



Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Probabilistic hierarchical forecasting with deep Poisson mixtures

Kin G. Olivares^a, O. Nganba Meetei^{b,*}, Ruijun Ma^b, Rohan Reddy^b,
Mengfei Cao^b, Lee Dicker^b^a Auton Lab, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, United States of America^b Forecasting Science, Amazon, New York, NY, United States of America

ARTICLE INFO

Keywords:

Hierarchical forecasting
Probabilistic coherence
Neural networks
Poisson mixtures
Composite Likelihood

ABSTRACT

Hierarchical forecasting problems arise when time series have a natural group structure, and predictions at multiple levels of aggregation and disaggregation across the groups are needed. In such problems, it is often desired to satisfy the aggregation constraints in a given hierarchy, referred to as hierarchical coherence in the literature. Maintaining coherence while producing accurate forecasts can be a challenging problem, especially in the case of probabilistic forecasting. We present a novel method capable of accurate and coherent probabilistic forecasts for time series when reliable hierarchical information is present. We call it the deep Poisson mixture network (DPMN). It relies on the combination of neural networks and a statistical model for the joint distribution of the hierarchical multivariate time-series structure. By construction, the model guarantees hierarchical coherence and provides simple rules for aggregation and disaggregation of the predictive distributions. We perform an extensive empirical evaluation comparing the DPMN to other state-of-the-art methods which produce hierarchically coherent probabilistic forecasts on multiple public datasets. Compared to existing coherent probabilistic models, we obtain a relative improvement in the overall continuous ranked probability score (CRPS) of 11.8% on Australian domestic tourism data, and of 8.1% on the Favorita grocery sales dataset, where time series are grouped with geographical hierarchies or travel-intent hierarchies. For San Francisco Bay Area highway traffic, where the series' hierarchical structure is randomly assigned and their correlations are less informative, our method does not show significant performance differences over statistical baselines.

© 2023 The Authors. Published by Elsevier B.V. on behalf of International Institute of Forecasters. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction and motivation

We study the task of forecasting probabilistically coherent time series where users need predictive distributions for all related time series organized into a hierarchy or group structure (Athanasopoulos, Hyndman, Kourentzes, & Petropoulos, 2017; Ben Taieb, Taylor, & Hyndman, 2017; Hyndman, Lee, & Wang, 2014;

Panagiotelis, Gamakumara, Athanasopoulos, & Hyndman, 2023; Spiliotis, Petropoulos, Kourentzes, & Assimakopoulos, 2020). As forecasts for different aggregation levels drive different decisions, forecast coherence is desired to ensure aligned decision-making across the hierarchies (Fotios Petropoulos et al., 2021). Notable examples of hierarchical forecasting tasks include the need for energy planners to synchronize the electricity load at each level of the grid with total production (Jeon, Panagiotelis, & Petropoulos, 2019; Souhaib & Bonsoo, 2019), the short-term load category in the Global Energy Forecasting

* Corresponding author.

E-mail address: nganba@gmail.com (O.N. Meetei).

<https://doi.org/10.1016/j.ijforecast.2023.04.007>

0169-2070/© 2023 The Authors. Published by Elsevier B.V. on behalf of International Institute of Forecasters. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Competition 2012 (GEFCOM2012; Hong, Pinson, and Fan 2014), and the efforts from the forecasting community manifested at the fifth Makridakis Competition (M5; Makridakis, Spiliotis, and Assimakopoulos 2022).

Coherent forecasts are defined as those that satisfy the aggregation constraints of the hierarchy. That is, disaggregated forecasts "add up" to the forecasts of aggregate levels. This definition is accessible for mean forecasts, which are additive by linearity of the expectation. For probabilistic forecasts, coherence is achieved when the forecast distribution of the aggregate series is identical to the distribution of the sum of its children's forecast series under an implicit or explicit joint distribution (Ben Taieb et al., 2017; Ben Taieb, Taylor, & Hyndman, 2021; Panagiotelis, Gamakumara, Athanasopoulos, & Hyndman, 2020; Panagiotelis et al., 2023; Wickramasuriya, 2023). Hierarchical reconciliation strategies provide an interesting approach for bringing back mean and probabilistic hierarchical coherence into neural forecasting methods. Early work focused on reconciling independently generated mean base forecasts (Hyndman, Ahmed, Athanasopoulos, & Shang, 2011; Wickramasuriya, Athanasopoulos, & Hyndman, 2019). The reconciliation strategies improved accuracy, and recently, a better understanding of the reconciliation process was provided through the language of forecast combinations (Hollyman, Petropoulos, & Tipping, 2021). Similar two-step forecast reconciliation methods were later extended to probabilistic forecasts as well (Ben Taieb et al., 2017; Puwasala, Panagiotelis Anastasios, & Hyndman, 2018), first estimating the marginal distributions independently and then reconciling them. Finally (Han, Dasgupta, & Ghosh, 2021; Rangapuram et al., 2021) proposed combining these two steps into a single neural network. Efficiently leveraging the cross-learning approach (Makridakis, Spiliotis, & Assimakopoulos, 2018a; Semenovoglou, Spiliotis, Makridakis, & Assimakopoulos, 2021) to improve accuracy while maintaining probabilistic coherence remains a challenge.

In this work, we present a novel method for producing probabilistically coherent forecasts. It combines the strength of modern neural networks and an intuitive statistical model for the disaggregated-forecast joint distribution. In contrast to earlier efforts (Han et al., 2021; Rangapuram et al., 2021), our method is an extension to mixture density networks (Bishop, 1994). It is coherent by construction and does not require an explicit reconciliation step, either as part of a single end-to-end network or as a separate step. We call it the deep Poisson mixture network (DPMN). The DPMN models the joint probability mass function of the multivariate time series as a finite mixture of Poisson distributions and combines it with the well-established MQ-Forecaster neural architecture (Eisenach, Patel, & Madeka, 2021; Wen, Torkkola, Narayanaswamy, & Madeka, 2017). This is possible because we formulate the problem as a mixture density network, and we can choose a relevant class of probabilistic distributions for the statistical model and the neural architecture independently. The key advantages of our method are as follows:

1. **Flexible forecast distribution:** We model the forecast distribution as a finite mixture of Poisson random variables, which is analogous to a Poisson kernel density. The resulting distribution is flexible, capable of accurately modeling a wide range of joint probability distributions, and compatible as an output layer with state-of-the-art neural architectures. We demonstrate this empirically on three different forecasting tasks in Section 6.
2. **Computational efficiency:** Learning coherent forecast distributions in a high-dimensional hierarchical space can be computationally intractable. To alleviate this, we anchor the DPMN on a multivariate distribution of the bottom-level time series and employ composite likelihood optimization strategies, which enables it to extend to large-scale applications.

The rest of the work is structured as follows. In Section 2 we introduce mathematical notations and review the statistical and neural-network hierarchical forecast literature. We describe our method's probabilistic model in Section 3 and the learning and inference methods in Section 4. In Section 5 we discuss the neural network architecture and in Section 6 we perform an empirical evaluation. Finally, in Section 7 we discuss future work and conclude.

2. Literature review

2.1. Hierarchical forecasting notation

Mathematically a hierarchical multivariate time series can be denoted by the vector $\mathbf{y}_{[a,b],t} = [\mathbf{y}_{[a],t}^\top | \mathbf{y}_{[b],t}^\top]^\top \in \mathbb{R}^{(N_a+N_b)}$ for each time point t ; where $[a]$, $[b]$ stand for the set of all aggregate and bottom indices of the time series, respectively. The total number of series in the hierarchy is $|[a, b]| = (N_a + N_b)$, where $|[a]| = N_a$ is the number of aggregated series and $|[b]| = N_b$ the number of bottom series that are at the most disaggregated level possible. Time indices for past information until t are given by the set $[t]$ with length $|[t]| = N_t$. With this notation the hierarchical aggregation constraints at each time point t have the following matrix representation:

$$\mathbf{y}_{[a,b],t} = \mathbf{S}_{[a,b][b]}\mathbf{y}_{[b],t} \Leftrightarrow \begin{bmatrix} \mathbf{y}_{[a],t} \\ \mathbf{y}_{[b],t} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{[a][b]} \\ \mathbf{I}_{[b][b]} \end{bmatrix} \mathbf{y}_{[b],t} \quad (1)$$

The matrix $\mathbf{S}_{[a,b][b]} \in \mathbb{R}^{(N_a+N_b) \times N_b}$ aggregates the bottom level to the series above. It is composed by stacking the aggregation matrix $\mathbf{A}_{[a][b]} \in \mathbb{R}^{N_a \times N_b}$ and an $N_b \times N_b$ identity matrix $\mathbf{I}_{[b][b]}$.

For example, Fig. 1 represents a simple hierarchy where each parent node is the sum of its children. Here the dimensions are $N_a = 3$ and $N_b = 4$, and the hierarchical, aggregated, and base series are, respectively,

$$\begin{aligned} \mathbf{y}_{\text{Total},t} &= \mathbf{y}_{\beta_1,t} + \mathbf{y}_{\beta_2,t} + \mathbf{y}_{\beta_3,t} + \mathbf{y}_{\beta_4,t} \\ \mathbf{y}_{[a],t} &= [\mathbf{y}_{\text{Total},t}, \mathbf{y}_{\beta_1,t} + \mathbf{y}_{\beta_2,t}, \mathbf{y}_{\beta_3,t} + \mathbf{y}_{\beta_4,t}]^\top \\ \mathbf{y}_{[b],t} &= [\mathbf{y}_{\beta_1,t}, \mathbf{y}_{\beta_2,t}, \mathbf{y}_{\beta_3,t}, \mathbf{y}_{\beta_4,t}]^\top. \end{aligned} \quad (2)$$

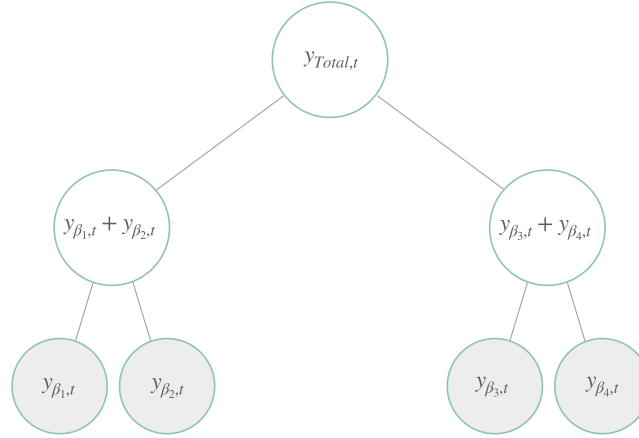


Fig. 1. A simple three-level time series hierarchical structure, with four bottom-level variables. The disaggregated bottom variables are marked with a gray background. In this description each node represents non-overlapping series for a single point in time.

The constraint matrix of Fig. 1 example and the corresponding aggregations from Eq. (2) is the following:

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

2.2. Mean forecast reconciliation strategies

Given a forecast creation date t and horizon h , the forecast indexes are denoted by $\tau \in [t + 1 : t + h]$. Most of the prior statistical solutions to the hierarchical forecasting challenge implement a two-stage process, first generating base forecasts $\hat{\mathbf{y}}_{[a,b],\tau} \in \mathbb{R}^{N_a + N_b}$, and then revising them into coherent forecasts $\tilde{\mathbf{y}}_{[a,b],\tau}$ through reconciliation. The reconciliation is compactly expressed by:

$$\tilde{\mathbf{y}}_{[a,b],\tau} = \mathbf{S}_{[a,b][b]} \mathbf{P}_{[b][a,b]} \hat{\mathbf{y}}_{[a,b],\tau} \quad (3)$$

where $\mathbf{S}_{[a,b]} \in \mathbb{R}^{(N_a + N_b) \times N_b}$ is the hierarchical aggregation matrix, and $\mathbf{P}_{[b][a,b]} \in \mathbb{R}^{N_b \times (N_a + N_b)}$ is a matrix determined by the reconciliation strategies. The most common reconciliation methods can be classified into top-down, bottom-up, and alternative approaches.

- Bottom-up: The simple bottom-up strategy, abbreviated as NaiveBU (Orcutt, Watts, & Edwards, 1968), first generates bottom-level forecasts and then aggregates them to produce forecasts for all the series in the multivariate structure.
- Top-down: The top-down strategy, abbreviated as TD (Fiedner, 1999; Gross & Sohl, 1990), distributes the total forecast, and then disaggregates it down the hierarchy using proportions that can be historical actuals or forecasted separately.
- Alternative: The more recent middle-out strategies, denoted as MO Hyndman et al., 2011; Hyndman &

Athanasopoulos, 2018), treat the second-stage reconciliation as an optimization problem for the matrix $\mathbf{P}_{[b][a,b]}$. These reconciliation techniques include, among others, game-theoretically optimal (GTOP; Van Erven & Cugliari, 2015), learning a projection for reducing quadratic loss; a generalized least squares model for minimizing the trace of the squared error matrix, namely the minimum trace reconciliation (MinT; Wickramasuriya et al., 2019); and the empirical risk minimization approach (ERM; Souhaib & Bonsoo, 2019).

Despite the advancements in alternative reconciliation strategies with statistical solutions, as mentioned in Section 1, there are still fundamental limitations. First, most post-process reconciliation methods produce mean forecasts but not probabilistic forecasts, with some exceptions that have relied on univariate statistical methods with strong probability assumptions for the base series that may be restrictive (Ben Taieb et al., 2017; Panagiotelis et al., 2020). Second, the mentioned methods independently learn the model parameters of the base-level forecasts, limiting the base model's optimization inputs to single series. This approach induces a setting that is prone to over-fitting for complex non-linear methods, which, as noted by the forecasting community, is one of their biggest challenges (Makridakis, Spiliotis, & Assimakopoulos, 2018b). The implied data scarcity translates into a missed opportunity to leverage the flexibility of non-linear methods.

2.3. Coherent probabilistic forecasting

There is a large body of related work on Bayesian hierarchical¹ modeling of spatiotemporal data with joint coherent predictive distributions (Diggle, 2013; Wikle,

¹ Note that the term "hierarchy" in Bayesian hierarchy is different from its use in hierarchical forecasting. The former refers to the conditional dependence of a posterior distribution's parameters, and the latter refers to the aggregation constraints across multiple time series.

Berliner, & Cressie, 1998). These models, however, come with strong assumptions, as they typically assume a stationary Gaussian process to induce correlations and rely on Markov chain Monte Carlo to estimate the posterior distribution (Diggle, 2013; Diggle & Brix, 2001). For modeling count data, variants of Bayesian hierarchical Poisson regression models were introduced (Christiansen & Morris, 1997; Park & Lord, 2009), but they usually operate with linear model assumptions. There are few specialized methods on coherent probabilistic forecasting, as most research on hierarchical forecasting has been limited to point predictions. Exceptions are the works by Jeon et al. (2019), Shang and Hyndman (2017), who provide an early exploration of forecast quantile reconciliation; and Ben Taieb et al. (2017, 2021), who propose the combination of bottom-level forecast marginal distributions with empirical copula functions describing their dependencies to create aggregate predictive distributions. To the best of our knowledge, a formal definition of probabilistic coherence has only been explored by (Ben Taieb et al., 2021; Panagiotelis et al., 2023; Puwasala et al., 2018; Wickramasuriya, 2023). Ben Taieb et al. (2021) provide a convolution-based definition, while Panagiotelis et al. (2023) provide a generalized and intuitive framework² that we follow in our work and introduce below:

Definition 2.1 (Probabilistic Coherence). Let $(\Omega_{[b]}, \mathcal{F}_{[b]}, \hat{\mathbb{P}}_{[b]})$ be a probabilistic forecast space, with sample space $\Omega_{[b]}$, $\mathcal{F}_{[b]}$ its σ -algebra, and $\hat{\mathbb{P}}_{[b]}$ a forecast probability. Let $\mathbf{S}_{[a,b][b]}(\cdot) : \Omega_{[b]} \mapsto \Omega_{[a,b]}$ be the linear transformation implied by the constraints matrix. A coherent probabilistic forecast space $(\Omega_{[a,b]}, \mathcal{F}_{[a,b]}, \hat{\mathbb{P}}_{[a,b]})$ satisfies:

$$\hat{\mathbb{P}}_{[a,b]}(\mathbf{S}_{[a,b][b]}(\mathcal{B})) = \hat{\mathbb{P}}_{[b]}(\mathcal{B})$$

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

(4)

That is, it assigns a zero probability to sets in $\mathbb{R}^{N_a+N_b}$ not containing any coherent forecasts.

For a simple definition-satisfying example, consider three random variables $(Y_\alpha, Y_{\beta_1}, Y_{\beta_2})$ with $Y_\alpha := Y_{\beta_1} + Y_{\beta_2}$. A coherent forecast assigns zero probability to the variable realizations $(y_\alpha, y_{\beta_1}, y_{\beta_2})$ if they do not satisfy the aggregation constraint $y_\alpha = y_{\beta_1} + y_{\beta_2}$. An equivalent definition of coherence is to require that the marginal distributions are derivable from the joint distribution of the bottom-level random variables. In this case, the probability function of interest is $\hat{\mathbb{P}}(Y_{\beta_1}, Y_{\beta_2})$, and the marginal probabilities can be derived from it using indicator functions as follows:

$$\hat{\mathbb{P}}(Y_{\beta_1} = y_{\beta_1}) = \sum_{y_{\beta_2}} \hat{\mathbb{P}}(Y_{\beta_1} = y_{\beta_1}, Y_{\beta_2} = y_{\beta_2})$$

and $\hat{\mathbb{P}}(Y_{\beta_2} = y_{\beta_2}) = \sum_{y_{\beta_1}} \hat{\mathbb{P}}(Y_{\beta_1} = y_{\beta_1}, Y_{\beta_2} = y_{\beta_2})$

² HierE2E by Rangapuram et al. (2021) informally introduced a similar notion of probabilistic coherence.

$$\hat{\mathbb{P}}(Y_\alpha = y_\alpha) = \sum_{y_{\beta_1}, y_{\beta_2}} \hat{\mathbb{P}}(Y_{\beta_1} = y_{\beta_1}, Y_{\beta_2} = y_{\beta_2}) \times \mathbb{1}(y_\alpha = y_{\beta_1} + y_{\beta_2}),$$

where $\mathbb{1}(y_\alpha = y_{\beta_1} + y_{\beta_2})$ equals 1 when $y_\alpha = y_{\beta_1} + y_{\beta_2}$, and 0 otherwise. The marginal probability for Y_α has the aggregation constraint built into its definition and is by construction a hierarchically coherent probability with respect to $\hat{\mathbb{P}}(Y_{\beta_1})$ and $\hat{\mathbb{P}}(Y_{\beta_2})$. Note that knowledge of the joint distribution is not required to generate hierarchically coherent forecasts. However, if we have access to the joint distribution, constructing hierarchically coherent marginal distributions is straightforward.

2.4. Hierarchical neural forecasting

In the last decade, neural network-based forecasting methods have become ubiquitous in large-scale forecasting applications (Böse et al., 2017; Eisenach et al., 2021; Madeka et al., 2018; Wen et al., 2017), transcending industry boundaries into academia, as they have redefined the state of the art in many practical tasks (Olivares, Challu, Marcjasz, Weron, & Dubrawski, 2023; Ravuri et al., 2021; Yu, Yin, & Zhu, 2018) and forecasting competitions (Makridakis et al., 2018a, 2022).

The latest neural network-based solutions to the hierarchical forecasting challenge include methods—such as simultaneous hierarchically reconciled quantile regression (SHARQ; Han et al., 2021), hierarchically regularized deep forecasting (HIRED; Paria, Sen, Ahmed, & Das, 2021), and the probabilistic robust hierarchical network (PROFHIT; Kamarthi, Kong, Rodriguez, Zhang, & Prakash, 2022)—that augment the training loss function with approximations to the hierarchical constraints. Hierarchical end-to-end learning (HierE2E; Rangapuram et al., 2021) integrates an alternative reconciliation strategy in its forecasts through linear projections. With the exception of HierE2E, these methods encourage probabilistic coherence through regularization but do not guarantee it. Additionally, if a user requires updating the hierarchical structure of interest, a whole new optimization of the networks is needed for the existing methods to forecast the structure correctly.

Our proposed method, DPMM, can address these deficiencies by specifying any network's output as our proposed Poisson mixture predictive distribution. With this predictive distribution, a model needs only to forecast the bottom-most level in the hierarchy, after which any desired hierarchical structure of interest can be predicted with guaranteed probabilistic hierarchical coherence.

3. Probabilistic model

3.1. Joint Poisson mixture distribution

In this work, we focus our attention on hierarchical forecasting of discrete events with non-negative variables. Many forecasting problems fall in this category, as shown by the datasets described in Section 6. However, the general framework presented here works for continuous distributions as well, with a suitably chosen base class for the

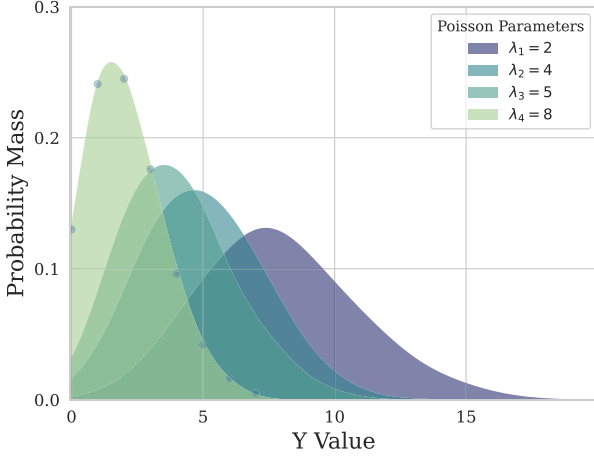


Fig. 2. The Poisson mixture distribution has desirable properties that make it well suited for probabilistic hierarchical forecasting for count data. Under minimal conditions, its aggregation rule implies probabilistic coherence of the random variables it models.

mixture distribution, e.g., a Gaussian distribution. For discrete events, we start by postulating that the forecasted joint probability of the bottom-level future multivariate time series' realization $\mathbf{y}_{[b][t+1:t+h]} \in \mathbb{N}^{N_b \times h}$ is:

$$\begin{aligned} \hat{\mathbb{P}}(\mathbf{y}_{[b][t+1:t+h]} | \boldsymbol{\lambda}_{[b][k][t+1:t+h]}) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} \prod_{(\beta, \tau) \in [b][t+1:t+h]} \text{Poisson}(y_{\beta, \tau} | \lambda_{\beta, \kappa, \tau}) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} \prod_{(\beta, \tau) \in [b][t+1:t+h]} \frac{\lambda_{\beta, \kappa, \tau}^{y_{\beta, \tau}}}{y_{\beta, \tau}!} e^{-\lambda_{\beta, \kappa, \tau}} \end{aligned} \quad (5)$$

The mixture model describes individual bottom-level time series and their correlations through the distribution of the mixture's latent variables defined by the Poisson rates $\boldsymbol{\lambda}_{[b][k][t+1:t+h]} \in \mathbb{R}^{N_b \times N_k \times h}$ and the associated weights $\mathbf{w}_{[k]} \in [0, 1]^{N_k}$, with $\mathbf{w}_{[k]} \geq 0$ and $\sum_{\kappa=1}^{N_k} w_{\kappa} = 1$. The number of Poisson components $|[k]| = N_k$ is a hyperparameter of the model that controls the flexibility of the mixture distribution. The mixture distribution can also be interpreted as a multivariate kernel density approximation, with Poisson kernels, to the actual joint probability distribution $\mathbb{P}(\mathbf{y}_{[b][t+1:t+h]})$. We show an example of the Poisson mixture distribution in Fig. 2.

The joint distribution in Eq. (5) assumes that the modeled observations $\mathbf{y}_{[b][t+1:t+h]}$ are conditionally independent, given the latent Poisson rates $\boldsymbol{\lambda}_{[b][k][t+1:t+h]}$. That is, for all bottom-level series and horizons, $(\beta, \tau) \neq (\beta', \tau')$, $(\beta, \tau), (\beta', \tau') \in [b][t+1:t+h]$ and $\kappa \in [k]$:

$$\begin{aligned} \hat{\mathbb{P}}(Y_{\beta, \tau}, Y_{\beta', \tau'} | \lambda_{\beta, \kappa, \tau}, \lambda_{\beta', \kappa, \tau'}) \\ &= \hat{\mathbb{P}}(Y_{\beta, \tau} | \lambda_{\beta, \kappa, \tau}) \hat{\mathbb{P}}(Y_{\beta', \tau'} | \lambda_{\beta', \kappa, \tau'}) \end{aligned} \quad (6)$$

3.2. Marginal distributions for bottom series

Eq. (5) describes the joint distribution of all bottom-level time series. We can derive the marginal distribution

for one of the bottom-level series $\beta \in [b]$ and for a single future time period $\tau \in [t+1:t+h]$ by integrating out the remaining series and time indices. The marginal distribution is:

$$\begin{aligned} \hat{\mathbb{P}}(Y_{\beta, \tau} = y_{\beta, \tau}) &= \sum_{\mathbf{y}_{[b][t+1:t+h] \setminus (\beta, \tau)}} \sum_{\kappa=1}^{N_k} w_{\kappa} \\ &\times \prod_{(\beta', \tau') \in [b][t+1:t+h]} \text{Poisson}(y_{\beta', \tau'} | \lambda_{\beta', \kappa, \tau'}) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} \text{Poisson}(y_{\beta, \tau} | \lambda_{\beta, \kappa, \tau}) \\ &\times \prod_{(\beta', \tau') \in [b][t+1:t+h] \setminus (\beta, \tau)} \sum_{y_{\beta', \tau'}} \text{Poisson} \\ &\times (y_{\beta', \tau'} | \lambda_{\beta', \kappa, \tau'}) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} \text{Poisson}(y_{\beta, \tau} | \lambda_{\beta, \kappa, \tau}). \end{aligned} \quad (7)$$

We get a clean final expression for the marginal distribution, which is equivalent to simply dropping all other time series and time periods from the product in Eq. (5).

3.3. Marginal distributions for aggregate series

An important consequence of the conditional independence in Eq. (6) is that the marginal distributions at aggregate levels $\mathbf{y}_{[a], \tau}$ can be computed via simple component-wise addition of the lower-level Poisson rates. For example consider an aggregate level variable $Y_{\alpha, \tau} := Y_{\beta_1, \tau} + Y_{\beta_2, \tau}$. The marginal distribution for $Y_{\alpha, \tau}$ can be derived from the joint distribution in Eq. (5) as follows. First we marginalize all other series and time indices (as done in Section 3.2, above), which gives us the joint distribution for $Y_{\beta_1, \tau}$ and $Y_{\beta_2, \tau}$:

$$\begin{aligned} \hat{\mathbb{P}}(Y_{\beta_1, \tau} = y_{\beta_1, \tau}, Y_{\beta_2, \tau} = y_{\beta_2, \tau}) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} \text{Poisson}(y_{\beta_1, \tau} | \lambda_{\beta_1, \kappa, \tau}) \times \text{Poisson}(y_{\beta_2, \tau} | \lambda_{\beta_2, \kappa, \tau}). \end{aligned}$$

Now, the aggregate marginal distribution is

$$\begin{aligned} \hat{\mathbb{P}}(Y_{\alpha, \tau} = y_{\alpha, \tau}) &= \sum_{y_{\beta_1, \tau}, y_{\beta_2, \tau}} \hat{\mathbb{P}}(Y_{\beta_1, \tau} = y_{\beta_1, \tau}, Y_{\beta_2, \tau} = y_{\beta_2, \tau}) \\ &\times \mathbb{1}(y_{\alpha, \tau} = y_{\beta_1, \tau} + y_{\beta_2, \tau}) \end{aligned}$$

For each mixture component with index κ , the distributions of $y_{\beta_1, \tau}$ and $y_{\beta_2, \tau}$ conditioned on the respective Poisson rates $\lambda_{\beta_1, \kappa, \tau}$ and $\lambda_{\beta_2, \kappa, \tau}$ are independent Poisson distributions, and therefore, the distribution of the aggregate is another Poisson mixture with parameters $\lambda_{\alpha, \kappa, \tau} = \lambda_{\beta_1, \kappa, \tau} + \lambda_{\beta_2, \kappa, \tau}$:

$$\begin{aligned} \hat{\mathbb{P}}(Y_{\alpha, \tau} = y_{\alpha, \tau}) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} \text{Poisson}(y_{\alpha, \tau} | \lambda_{\alpha, \kappa, \tau} = \lambda_{\beta_1, \kappa, \tau} + \lambda_{\beta_2, \kappa, \tau}). \end{aligned} \quad (8)$$

The aggregation rule can be concisely described as:

$$\mathbf{Y}_{[a][t+1:t+h]} = \mathbf{A}_{[a][b]} \mathbf{Y}_{[b][t+1:t+h]} \quad (9)$$

$$\lambda_{[a][k][t+1:t+h]} = \mathbf{A}_{[a][b]} \lambda_{[b][k][t+1:t+h]},$$

with $\mathbf{A}_{[a][b]}$ as the hierarchical aggregation matrix defined in Section 2.1. The joint predictive distribution is hierarchically coherent by construction. We offer a formal proof of DPMN's satisfaction of the probabilistic coherence property from Definition 2.1 in Appendix A.

3.4. Covariance matrix

Using the law of total covariance and the conditional independence from Eq. (6), we show in B that the covariance of any two bottom-level series naturally follows:

$$\begin{aligned} \text{Cov}(Y_{\beta,\tau}, Y_{\beta',\tau'}) &= \bar{\lambda}_{\beta,\tau} \mathbb{1}(\beta = \beta') \mathbb{1}(\tau = \tau') \\ &+ \sum_{\kappa=1}^{N_k} w_{\kappa} (\lambda_{\beta,\kappa,\tau} - \bar{\lambda}_{\beta,\tau}) (\lambda_{\beta',\kappa,\tau'} - \bar{\lambda}_{\beta',\tau'}) \end{aligned} \quad (10)$$

where $\bar{\lambda}_{\beta,\tau} = \sum_{\kappa=1}^{N_k} w_{\kappa} \lambda_{\beta,\kappa,\tau}$. Appendix C shows that the non-diagonal covariance matrix expressivity, as determined by its rank, depends on the number of mixture components from Eq. (5).

4. Parameter estimation and inference

4.1. Maximum joint likelihood

To estimate model parameters, we can maximize the joint likelihood implied by the joint distribution from Eq. (5). Let θ represent the neural network parameters, as described in Section 5. We parameterize the probabilistic model with Poisson rates $\lambda_{[b][k][t+1:t+h]}$ as follows:

$$\begin{aligned} \lambda_{[b][k][t+1:t+h]} &:= \hat{\lambda}_{[b][k][t+1:t+h]}(\theta) \\ \mathbf{w}_{[k]} &:= \hat{\mathbf{w}}_{[k]}(\theta) \end{aligned} \quad (11)$$

The Poisson rates and the mixture weights are conditioned through the network's parameters on forecasting features discussed in Section 5.1. The negative log-likelihood can then be written as follows³:

$$\begin{aligned} \mathcal{L}(\theta) &= -\log \left[\sum_{\kappa=1}^{N_k} \hat{w}_{\kappa}(\theta) \right. \\ &\times \left. \prod_{(\beta,\tau) \in [b][t+1:t+h]} \left(\frac{(\hat{\lambda}_{\beta,\kappa,\tau}(\theta))^{y_{\beta,\tau}} \exp\{-\hat{\lambda}_{\beta,\kappa,\tau}(\theta)\}}{(y_{\beta,\tau})!} \right) \right] \end{aligned} \quad (12)$$

This is the same expression as the multivariate probability mass function in Eq. (5) but parametrized as a function of the neural network parameters θ . The maximum likelihood estimation method (MLE) has desirable properties like statistical efficiency and consistency. However, the mixture components cannot be estimated separately,

³ We kept notations simple and omitted the explicit conditioning on input features.

and for this reason, MLE is only feasible for hierarchical time series with a small number of series. Additional exploration is needed to make the estimation scalable.

4.2. Maximum composite likelihood

A computationally efficient alternative to MLE for estimating the model parameters is to maximize the composite likelihood. This method involves breaking up the high-dimensional space into smaller sub-spaces, and the composite likelihood consists of the weighted product of the marginal likelihoods of the subspaces (Lindsay, 1988). For simplicity, we used uniform weights.

In addition to the computational efficiency, maximizing the composite likelihood provides a robust and unbiased estimate of marginal model parameters with the drawback that the model inference may suffer from properties similar to a misspecified model (Varin, Reid, & Firth, 2011). We discuss variants of the composite likelihood below.

4.2.1. Naive bottom-up

A simple option of using composite likelihood is to define each bottom-level time series as its likelihood sub-space and treat it as independent during model training (Orcutt et al., 1968). We refer to this estimation method for the DPMN as naive bottom-up (DPMN-NaiveBU). The negative logarithm of the DPMN-NaiveBU composite likelihood is:

$$\begin{aligned} \mathcal{L}_{\text{NaiveBU}}(\theta) &= -\sum_{\beta \in [b]} \log \left[\sum_{\kappa=1}^{N_k} \hat{w}_{\kappa}(\theta) \right. \\ &\times \left. \prod_{\tau \in [t+1:t+h]} \left(\frac{(\hat{\lambda}_{\beta,\kappa,\tau}(\theta))^{y_{\beta,\tau}} \exp\{-\hat{\lambda}_{\beta,\kappa,\tau}(\theta)\}}{(y_{\beta,\tau})!} \right) \right] \end{aligned} \quad (13)$$

Even though the sub-space consists of single bottom-level time series, DPMN-NaiveBU is still a multivariate model with the composite likelihood defined over multiple time points $[t+1 : t+h]$. Maximizing the DPMN-NaiveBU composite likelihood will still learn correlations across the time points and will generate coherent forecast distributions for aggregations in the time dimension. It does not attempt, however, to discover correlations across different time series.

4.2.2. Group bottom-up

If prior information helps us identify groups of time series with interesting correlation structures, we may estimate them by including the groups in the composite likelihood. We refer to this estimation method for the DPMN as group bottom-up (DPMN-GroupBU). Let $\mathcal{G} = \{[g_i]\}$ be time-series groups. Then the negative log composite likelihood for the DPMN-GroupBU is:

$$\begin{aligned} \mathcal{L}_{\text{GroupBU}}(\theta) &= -\sum_{[g_i] \in \mathcal{G}} \log \left[\sum_{\kappa=1}^{N_k} \hat{w}_{\kappa}(\theta) \right. \\ &\times \left. \prod_{(\beta,\tau) \in [g_i][t+1:t+h]} \left(\frac{(\hat{\lambda}_{\beta,\kappa,\tau}(\theta))^{y_{\beta,\tau}} \exp\{-\hat{\lambda}_{\beta,\kappa,\tau}(\theta)\}}{(y_{\beta,\tau})!} \right) \right]. \end{aligned} \quad (14)$$

The main advantage over DPMN-NaiveBU is that the model now learns cross-series relationships. In this paper we only rely on intuitive grouping like geographic proximity, but one could in principle employ more sophisticated methods like clustering to define the groups. To optimize the learning objective we use stochastic gradient descent, and sample train series batches at the group level.

4.3. Forecast inference

As mentioned above, model inference from composite likelihood estimation suffers from problems similar to model misspecification. This is because of the independence assumed across sub-spaces. In our model the maximum composite likelihood estimates do not understand how mixture components learned for one sub-space relate to mixture components learned for a different sub-space. However, we need to identify mixture components across sub-spaces in order to define the joint distribution in Eq. (5) across all bottom-level time series. Knowing this joint distribution is at the crux of forecast inference from our model. Fortunately there is a natural way for us to resolve this issue. In both the DPMN-NaiveBU composite likelihood in Eq. (13) and the DPMN-GroupBU composite likelihood in Eq. (14), the weights $\hat{\mathbf{w}}_{[k]}(\theta) \in \mathbb{R}^{N_k}$ are shared across all sub-spaces. We identify components with the same weight as belonging to the same multivariate sample, and hence providing the full joint distribution. We call this method weight matching.

For the DPMN-NaiveBU estimation, the weight matching method is a strong statistical assumption, extrapolating from independent marginal distributions of each series to the full joint distribution. The DPMN-GroupBU approach significantly alleviates this problem because the model parameters are well defined within each group $[g_i] \in \mathcal{G}$, and if most of the interesting correlations are already captured within each group, then much less burden is placed on the weight matching method. We show in the evaluation of Section 6 that both DPMN-NaiveBU and DPMN-GroupBU perform favorably when compared to other mean and probabilistic hierarchical forecasting methods, and that between the two, DPMN-GroupBU is generally more accurate when the time series grouping given is informative.

5. Deep Poisson mixture network

Our primary goal is to create a probabilistically coherent forecasting model that is accurate and efficient. For this purpose, we opt to extend the MQ-Forecaster family (Eisenach et al., 2021; Wen et al., 2017), proven by its history of industry service, with the Poisson mixture distribution. We refer to this model as the deep Poisson mixture network (DPMN). Our MQ-Forecaster architecture selection is driven by its high computational efficiency, a consequence of the *forking-sequences* technique and multi-step forecasting strategy, in addition to its ability to incorporate static and known future temporal features (see Fig. 3).

5.1. Model features

As part of the innovations within our work, we propose to separate the bottom-level and aggregate-level features that we use in the forecasts. Sharing aggregate-level features across their respective bottom series simplifies the model's inputs and reduces redundant information, while greatly improving the model's memory usage efficiency.

We follow the hierarchical forecasting literature practice for the static features and use group identifiers implied by the hierarchy structure. We denote the static features, composed of features shared across the bottom series, by $\tilde{\mathbf{x}}^{(s)}$, and bottom level-specific features $\mathbf{x}_{[b]}^{(s)}$ by:

$$\mathbf{x}^{(s)} = \{\mathbf{x}_{[b]}^{(s)}, \tilde{\mathbf{x}}^{(s)}\}. \quad (15)$$

Regarding temporal features, for the bottom level, we use the bottom series' past; for the aggregate level, we use the parent node series' past. The future temporal information available can be task-specific inputs like prices or promotions in the context of product demand forecasting, or other simpler model forecasts as inputs, such as Naive1 or SeasonalNaive that help the model predict series levels and seasonalities. Similarly to the static features, we also distinguish the temporal shared features $\tilde{\mathbf{x}}_{[:t]}^{(h)}$, $\tilde{\mathbf{x}}_{[:t+h]}^{(f)}$ and the temporal bottom level-specific features $\mathbf{x}_{[b][:t]}^{(h)}$, $\mathbf{x}_{[b][:t+h]}^{(f)}$. With this consideration, we denote the historical and future temporal features by:

$$\mathbf{x}_{[:t]}^{(h)} = \{\mathbf{x}_{[b][:t]}^{(h)}, \tilde{\mathbf{x}}_{[:t]}^{(h)}\} \quad \text{and} \quad \mathbf{x}_{[:t+h]}^{(f)} = \{\mathbf{x}_{[b][:t+h]}^{(f)}, \tilde{\mathbf{x}}_{[:t+h]}^{(f)}\} \quad (16)$$

5.2. Model architecture

In summary, the DPMN builds on the MQ-Forecaster architecture that is based on a context-aware sequence-to-sequence network (Seq2SeqC; Cho et al., 2014). The DPMN uses dilated temporal convolutions (TempConv; van den Oord et al., 2016) to encode the available history into hidden states and uses forked decoders based on a multi-layer perceptron (MLP; Rosenblatt, 1961) in a direct multi-horizon forecast strategy (Amir & Souhaib, 2016). Below, we describe in further detail the components of the model. Other hyperparameter details are available in Table A.5.

5.2.1. Encoder

As explained above, the DPMN main encoder is a stack of dilated temporal convolutions. Additionally, we use a global dense layer to encode the static features and a local dense layer, shared across time, to encode the available future information. The encoder for each time t and its components are described in Eq. (17):

$$\begin{aligned} \mathbf{h}_t^{(h)} &= \{\mathbf{h}_{1,t}^{(h)}, \mathbf{h}_{2,t}^{(h)}\} = \{\text{TempConv}(\mathbf{x}_{[b][:t]}^{(h)}), \text{TempConv}(\tilde{\mathbf{x}}_{[:t]}^{(h)})\} \\ \mathbf{h}^{(s)} &= \{\mathbf{h}_1^{(s)}, \mathbf{h}_2^{(s)}\} = \{\text{MLP}(\mathbf{x}_{[b]}^{(s)}), \text{MLP}(\tilde{\mathbf{x}}^{(s)})\} \\ \mathbf{h}_t^{(f)} &= \text{MLP}_t(\mathbf{x}_{[b][:t+h]}^{(f)}) \end{aligned} \quad (17)$$

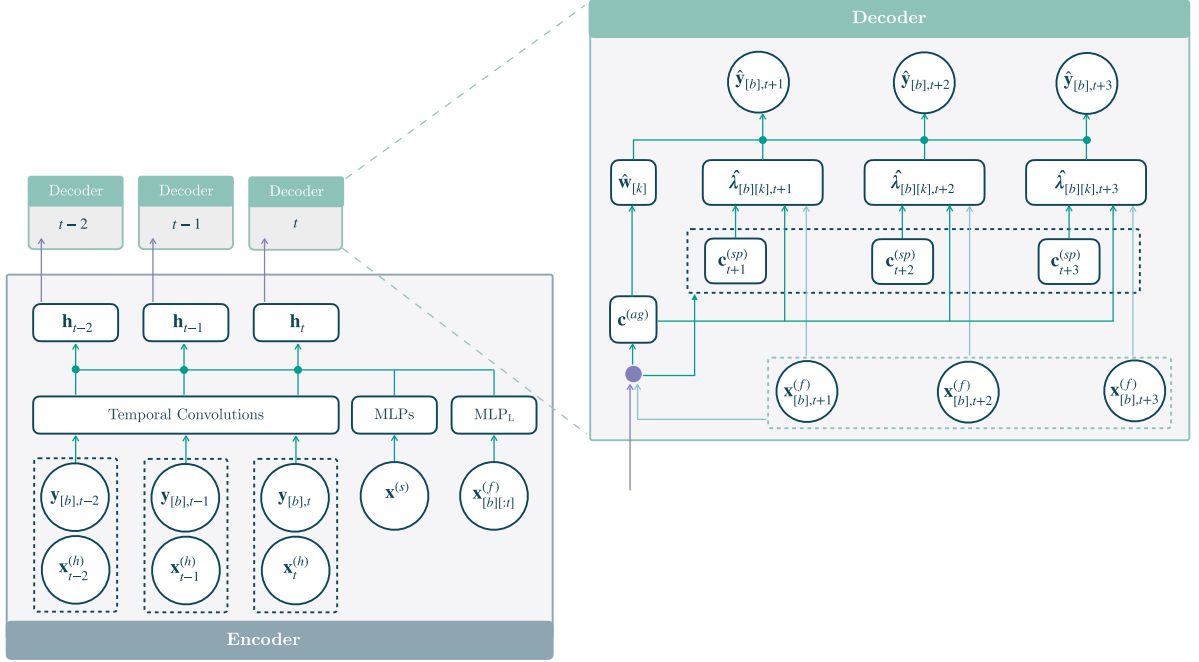


Fig. 3. The proposed deep Poisson mixture network (DPMN) is a context-aware sequence-to-sequence network that uses dilated temporal convolutions as the primary encoder and multi-layer perceptron-based decoders for a direct multi-step forecast. The forked decoders share their parameters and create the multivariate forecast distribution for each time point in the encoder, making the architecture efficient in its optimization and predictions.

The encoder's output in Eq. (18) is a set of shared and bottom-level encoded features $\mathbf{h}_{1,t}$ and $\mathbf{h}_{2,t}$. The first concatenates all the encoded past $\mathbf{h}_{1,t}^{(h)}, \mathbf{h}_{2,t}^{(h)} \in \mathbb{R}^{N_{cf}}$, static $\mathbf{h}_1^{(s)}, \mathbf{h}_2^{(s)} \in \mathbb{R}^{N_s}$ and available future $\mathbf{h}_t^{(f)} \in \mathbb{R}^{N_f}$ information.⁴ The second one concatenates the encoded past $\mathbf{h}_{2,t}^{(h)} \in \mathbb{R}^{N_{cf}}$ and static $\mathbf{h}_2^{(s)} \in \mathbb{R}^{N_s}$ shared features.

$$\mathbf{h}_t = \{\mathbf{h}_{1,t}, \mathbf{h}_{2,t}\} = \{[\mathbf{h}_{1,t}^{(h)}|\mathbf{h}_{2,t}^{(h)}|\mathbf{h}_1^{(s)}|\mathbf{h}_2^{(s)}|\mathbf{h}_t^{(f)}], [\mathbf{h}_{2,t}^{(h)}|\mathbf{h}_2^{(s)}]\}. \quad (18)$$

5.2.2. Forked decoders

The DPMN uses a two-branch MLP decoder. The first decoder branch summarizes the encoder output and future available information into two contexts: the horizon-agnostic context set $\mathbf{c}^{(ag)} \in \mathbb{R}^{N_{ag}}$, and the horizon-specific context $\mathbf{c}_{[t+1:t+h]}^{(sp)} \in \mathbb{R}^{N_{sp} \times h}$ that provides structural awareness of the forecast horizon and plays a crucial role in expressing any recurring patterns in the time series. Eq. (19) describes the first decoder branch:

$$\begin{aligned} \mathbf{c}^{(ag)} &= \{\mathbf{c}_1^{(ag)}, \mathbf{c}_2^{(ag)}\} = \{\text{MLP}(h_{1,t}), \text{MLP}(h_{2,t})\} \\ \mathbf{c}_{[t+1:t+h]}^{(sp)} &= \text{MLP}_L(\mathbf{h}_{1,t}) \end{aligned} \quad (19)$$

The second decoder branch adapts the horizon-specific and horizon-agnostic contexts into the parameters of the Poisson mixture distribution. For the horizon-specific Poisson rates, we use the forking-sequence technique with a series of decoders with shared parameters for each

time point in $[t+1:t+h]$, and for the mixture weights, we apply an MLP followed by a softmax on the aggregate horizon-agnostic context. Eq. (20) describes the second decoder branch:

$$\begin{aligned} \hat{\lambda}_{[b][k][t+1:t+h]} &= \text{MLP}_L(\mathbf{c}_1^{(ag)}, \mathbf{c}_{[t+1:t+h]}^{(sp)}, \mathbf{x}_{[b][t+1:t+h]}^{(f)}) \\ \hat{\mathbf{w}}_{[k]} &= \text{SoftMax}(\text{MLP}(\mathbf{c}_2^{(ag)})) \end{aligned} \quad (20)$$

6. Empirical evaluation

6.1. Hierarchical forecasting datasets

To evaluate our method, we consider three forecasting tasks where the objective is to provide quantile forecasts for each time series in the group or hierarchy structure. The three datasets that we use in the empirical evaluation⁵ are all publicly available and have been used in the hierarchical forecasting literature (Paria et al., 2021; Rangapuram et al., 2021; Souhaib & Bonsoo, 2019; Wickramasuriya et al., 2019). Table 1 summarizes the datasets' characteristics.⁶

Tourism-L (Tourism Australia, Canberra, 2019) is an Australian tourism dataset that contains 555 monthly visit series from 1998 to 2016, grouped by geographical region and travel purpose. Favorita (Corporación Favorita, 2018) is a Kaggle competition dataset of the

⁴ The local horizon-specific MLP_L aligns future seasonalities and events and improves the forecast's sharpness.

⁵ Traffic is available in the UCI ML repository. Tourism-L is available on the MinT reconciliation webpage. Favorita is available at its Kaggle Competition url.

⁶ We include more details for the Traffic, Tourism-L, and Favorita datasets in Appendix D.

Table 1
Summary, hierarchical structure, and forecast horizon of datasets used in our empirical study.

Dataset	Total	Aggregated	Bottom	Levels	Observations	Horizon (h)
Traffic	207	7	200	4	366	1
Tourism-L	555	175	76 / 304	4 / 5	228	12
Favorita	371,312	153,386	217,944	4	1688	34

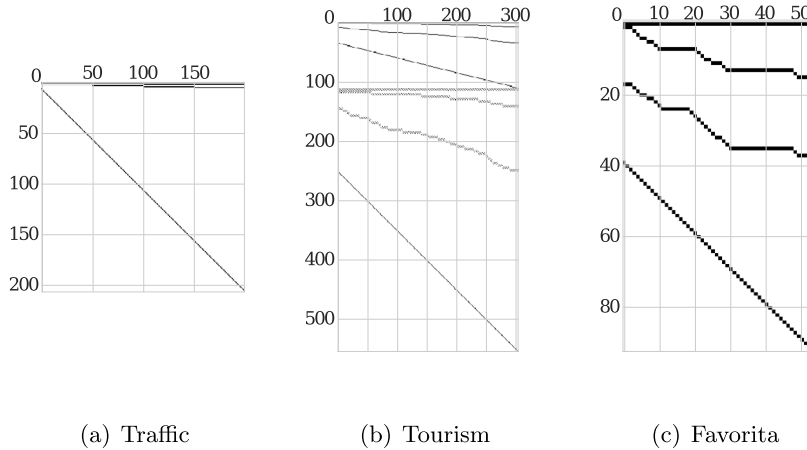


Fig. 4. Visualization of the hierarchical constraints of the empirical evaluation datasets. (a) *Traffic* groups 200 highways' occupancy series into quarters, halves, and total. (b) *Tourism-L* groups its 555 regional visit series into a combination of travel purpose, zones, states, and country geographical aggregations. (c) *Favorita* groups its grocery sales geographically, by store, city, state, and, country levels.

daily history of grocery item sales, with additional information on promotions, items, stores, and holidays, containing 371,312 series from January 2013 to August 2017, with a geographic hierarchy of states, cities, and stores. We show their hierarchical constraints matrix in Fig. 4. *Traffic* (Dua & Graff, 2017; Souhaib & Bonsoo, 2019) measures the occupancy of 200 car lanes in the San Francisco Bay Area, randomly grouped into a year of daily observations with a 207-series hierarchical structure.

The datasets provide an opportunity to showcase the broad applicability of the DPMN, as each has unique characteristics. *Tourism-L* allows us to test the DPMN to model group structures with multiple hierarchies. *Favorita* allows us to test the DPMN on a large-scale dataset. *Traffic* is composed of randomly assigned hierarchical groupings that may not have any informative structures for the DPMN to learn with GroupBU. Finally, *Favorita* contains some non-count demand values (grocery produce sold by weight) and *Traffic* aggregated occupancy rates are non-count data, so modeling these datasets with a Poisson mixture limits the maximum accuracy we can achieve.

6.2. Time series covariance modeling

We present in this subsection an illustrative example demonstrating how the DPMN leverages the flexibility and expressiveness of the multivariate Poisson mixture distribution to capture interesting correlations present in hierarchical time series datasets to improve forecast sharpness. Fig. 5 shows a comparison of forecasts generated by the DPMN-NaiveBU and the DPMN-GroupBU methods at various aggregation levels of the *Tourism-L* dataset. The

DPMN-GroupBU method accurately estimates correlations in bottom-level series, improving the forecast distribution concentration of the upper-level series. In contrast, the DPMN-NaiveBU method performs well on disaggregated series and mean forecasts, but suffers from significant model misspecification, which reduces the sharpness of forecasts at the aggregated levels. We find that the DPMN-GroupBU generally does better when an informative group series structure is present in the data, like in the *Tourism-L* and *Favorita* datasets. DPMN-NaiveBU produces comparable results at disaggregated levels of all datasets and outperforms hierarchical forecasting baselines when the hierarchical structure is noisy or uninformative, as in the *Traffic* dataset. Our intuitions are validated by the empirical results presented in Section 6.6.

6.3. Dataset partitioning and preprocessing

For the main experiments, we separate the training, validation and test datasets' partitions as follows: We hold out the final horizon-length observations as the test set. In a sliding-window fashion, we use the horizon-length that precedes the test set as the validation set and treat the rest of the past information as the training set. A partition example is depicted in Fig. 6.

For comparability purposes with the most recent hierarchical forecasting literature, we keep as close as possible to the preprocessing and wrangling of the datasets in Rangapuram et al. (2021).⁷ In general, the static variables that we consider on all the datasets correspond to the

⁷ The preprocessed datasets are available in the hierarchical forecasting extension to the [GluonTS library](#).

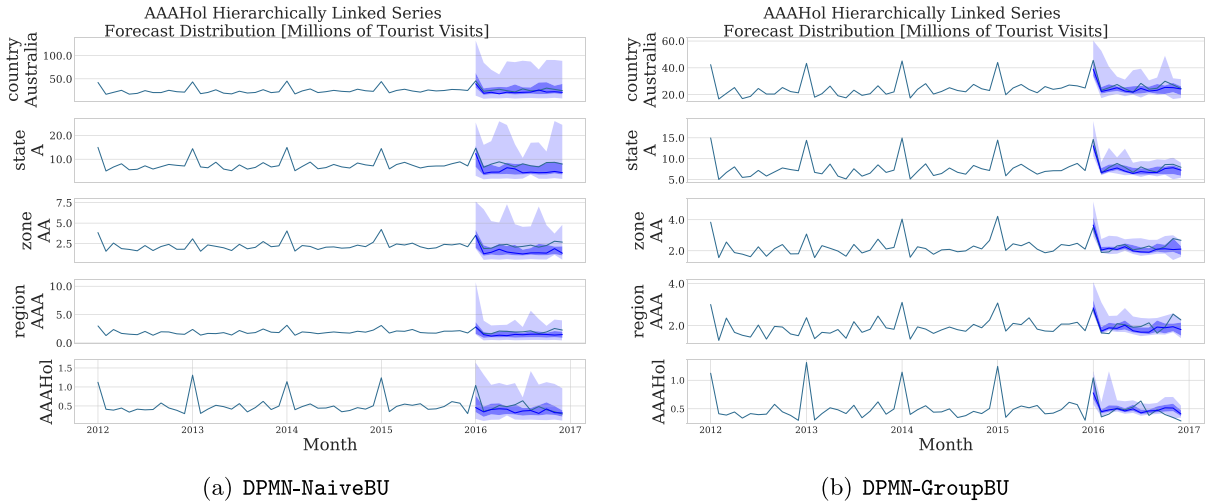


Fig. 5. DPMN-NaiveBU and DPMN-GroupBU forecast distributions on a Tourism-L hierarchically linked time series. The top row shows total tourist visits in Australia, the second row shows the visits to Australia for the North South Wales state (A), the third row shows the holiday visits in the metropolitan area of New South Wales (AA), the fourth row shows the total visits to Sydney (AAA), and the final row shows the holiday visits to Sydney. Forecast distributions, 99% and 75% prediction intervals in light and dark blue.

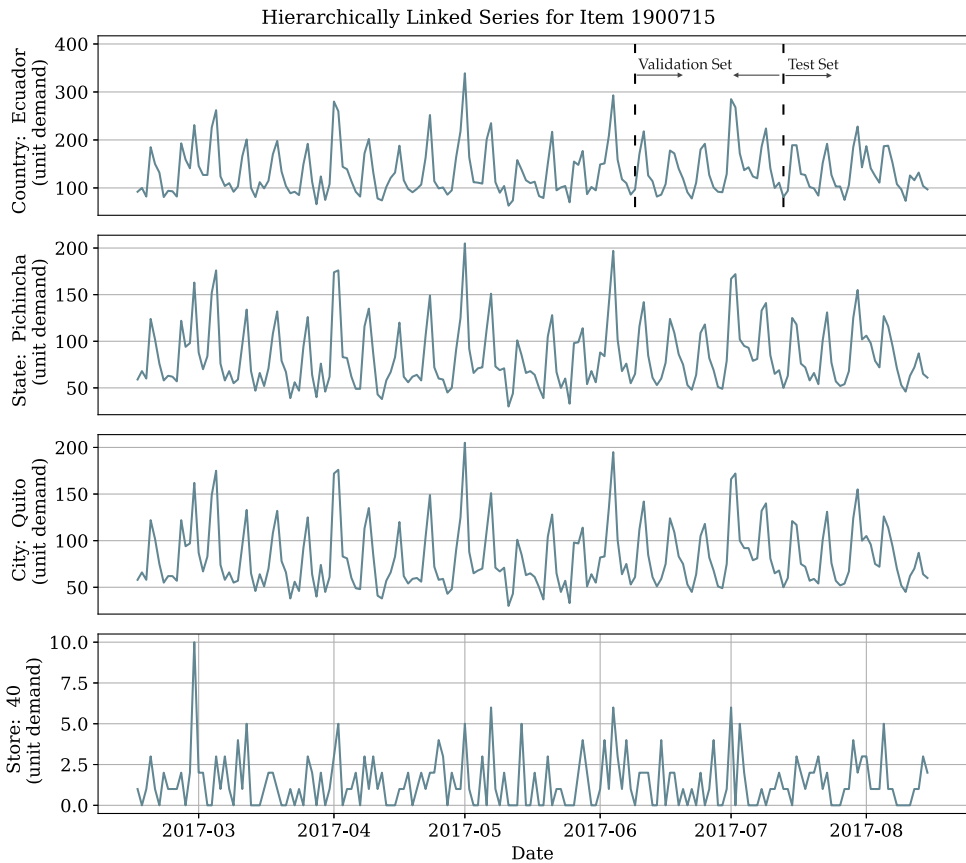


Fig. 6. Example of a Favorita geographically linked time series. The top level shows the sales for a grocery item in the country of Ecuador. The second level shows the sold units within Pichincha state, the third level shows the sales for Quito city, and the final level shows Store 40 item sales. For this dataset, the training set comprises all the observations preceding the validation and test sets. The validation set (between the first and second dotted lines) is the 34 days before the test set. The held-out test set (marked by the last dotted line) is the last 34 observations.

hierarchical and group designators as categorical variables implied by the hierarchical constraint matrix. The temporal covariates that we consider are the time series for the upper levels of the hierarchy, as well as calendar covariates associated with the time-series frequency of each dataset. As future data, we include calendar covariates to help the DPMN capture seasonalities.

6.4. Evaluation metrics

The primary evaluation metric of the model's forecasts is based on the quantile loss/pinball loss (QL) (Matheson & Winkler, 1976). For a given a forecast creation date t and horizon indexes $\tau \in [t + 1 : t + h]$, consider the estimated cumulative distribution function $\hat{F}_{i,\tau}$ of the variable $Y_{i,\tau}$ and its observation $y_{i,\tau}$. The loss is defined as:

$$QL(\hat{F}_{i,\tau}, y_{i,\tau})_q = \left(\mathbb{1}[y_{i,\tau} \leq \hat{F}_{i,\tau}^{-1}(q)] - q \right) \left(\hat{F}_{i,\tau}^{-1}(q) - y_{i,\tau} \right). \quad (21)$$

We summarize the evaluation, for convenience of exposition and to ensure the comparability of our results with the existing literature, using the continuous ranked probability score, abbreviated as CRPS (Matheson & Winkler, 1976).⁸ We use the following mean scaled CRPS Bolin and Wallin (2019), Makridakis, Spiliotis, Assimakopoulos, Chen et al. (2022) version:

$$CRPS(\hat{F}_{[i],\tau}, \mathbf{y}_{[i],\tau}) = \frac{2}{|[i]|} \sum_i \int_0^1 QL(\hat{F}_{i,\tau}, y_{i,\tau})_q dq \quad (22)$$

$$sCRPS(\hat{F}_{[i],\tau}, \mathbf{y}_{[i],\tau}) = \frac{CRPS(\hat{F}_{[i],\tau}, \mathbf{y}_{[i],\tau})}{\sum_i |y_{i,\tau}|}. \quad (23)$$

The CRPS measures the forecast distributions' accuracy and has desirable properties (Bolin & Wallin, 2019; Gneiting & Ranjan, 2011). For instance it is a *proper* scoring rule, since for any forecast distribution $\hat{F}_{i,\tau}$ and true distribution $F_{i,\tau}$, the expected score satisfies:

$$\mathbb{E}_{Y_{i,\tau}} [CRPS(F_{i,\tau}, Y_{i,\tau})] \leq \mathbb{E}_{Y_{i,\tau}} [CRPS(\hat{F}_{i,\tau}, Y_{i,\tau})], \quad (24)$$

which implies that it will prefer an ideal probabilistic forecasting system over any other.

The main focus of the paper is probabilistically coherent forecasting, and the main results comparing the sCRPS of DPMN to other hierarchical forecasting methods are presented in Section 6.6. We complement the main results with a comparison of the mean hierarchical forecasts accuracy in Section 6.7. It demonstrates the robustness of our method in point forecasting tasks as well.

6.5. Training methodology and hyperparameter optimization

For the overall hyperparameter selection, we used a standard two-stage approach where we first fixed the

architecture and the estimated probability distribution, and a second stage where we optimized the architecture's training procedure. Keeping the explored hyperparameter space small in the second stage serves two purposes: it keeps space exploration computationally tractable, and it showcases the DPMN's robustness, broad applicability, and accuracy with minor modifications. We defer some hyperparameter selection details to Appendix F.

In the first stage we select the number of DPMN's mixture components that are responsible for single-series forecasting and modeling bottom-level correlations, as stated in Section 4 and shown in Appendix C. For each dataset, we selected the components optimally using temporal cross-validation in the Appendix E ablation study, where we found that complex correlation structures favored a higher number of components. To observe the effects of modeling the series covariance, we compared the DPMN-GroupBU and DPMN-NaiveBU variants.

In the second stage, as shown in Table 2, the hyperparameter space that we consider for optimization is minimal. We only tune the learning rate, random seed to escape underperforming local minima, and the number of SGD epochs as a form of regularization (Yao, Rosasco, & Andrea, 2007). During the hyperparameter optimization phase, we measure the model sCRPS performance on the validation set described in Section 6.3, and use HYPEROPT (Bergstra, Bardenet, Bengio, & Kégl, 2011), a Bayesian optimization library, to efficiently explore the hyperparameters based on the validation measurements.

After the optimal hyperparameters are determined, we estimate the model parameters again by shifting the training window forward, called the re-training phase, and predict for the final test set. We refer to the combination of the hyperparameter optimization and re-training phases as a *run*. The DPMN is implemented using MXNet (Chen et al., 2015). To train the network, we minimize the negative log-composite likelihood variant from Section 4, using stochastic gradient descent with adaptive moments (ADAM; Kingma and Ba 2014).

6.6. Probabilistic forecasting results

We compare against the forecasts of the following probabilistic methods across the hierarchical levels: (1) HierE2E (Rangapuram et al., 2021), which combines DeepVAR with hierarchical constraints on a multivariate normal distribution⁹; (2) PERMBU-MinT (Ben Taieb et al., 2017), which synthesizes hierarchy levels' information with a probabilistic hierarchical aggregation of ARIMA forecasts¹⁰; (3) automatic ARIMA (Hyndman & Khandakar, 2008), which performs a step-wise exploration of ARIMA models using the AIC; and (4) GLM-Poisson a special case of generalized linear model regression suited for count data (Nelder & Wedderburn, 1972).

⁸ In practice the evaluation of the CRPS uses a numerical integration technique that discretizes the quantiles and treats the integral with a left Riemann approximation, averaging over uniformly distanced quantiles.

⁹ The HierE2E and PERMBU-MinT baseline models are available in a [GluonTS library](#) extension.

¹⁰ The original PERMBU-MinT is implemented in the [supplementary material](#) of the work of Ben Taieb et al. (2017).

Table 2

Hyperparameters considered for the deep Poisson mixture network (DPMN). The learning rate, random seed, and SGD epochs that performed best on the validation set were selected automatically in each HYPEROPT run. The remaining model parameters were configured once per dataset, as explained in Appendix F.

HYPERPARAMETER	CONSIDERED VALUES
Initial learning rate for SGD optimization.	$lr \in \{0.00001, \dots, 0.01\}$
SGD full passes to dataset (epochs).	$n_epochs \in \{10, \dots, 3000\}$
Random seed that controls initialization of weights.	$seed_train \in \{1, \dots, 10\}$
SGD batch size.	$batch_size \in \{4, \dots, 100\}$
Activation function.	PeLU ^a
Temporal convolution kernel size.	$N_{ck} \in \{2, 7\}$
Temporal convolution layers.	$N_{cl} \in \{3, 5\}$
Temporal convolution filters.	$N_{cf} \in \{10, 30\}$
Future encoder dimension.	$N_f \in \{50\}$
Static encoder dimension.	$N_s \in \{100\}$
Horizon agnostic decoder dimensions.	$N_{ag} \in \{50\}$
Horizon specific decoder dimensions.	$N_{sp} \in \{20\}$
Poisson mixture weights decoder layers.	$N_{wdl} \in \{3, 4\}$
Poisson mixture rate decoder layers.	$N_{rdl} \in \{2, 3, 4\}$
Local decoder dimensions.	$N_k \in \{25, 50, 100\}$

^aThe parametrized exponential linear unit (PeLU) modifies the ReLU activation, improving the network's training speed (Clevert, Unterthiner, & Hochreiter, 2015).

Table 3

Empirical evaluation of probabilistically coherent forecasts. Mean scaled continuous ranked probability score (sCRPS), averaged over eight runs, at each aggregation level. The best result is highlighted (lower measurements are preferred). Methods without standard deviation have deterministic solutions.

DATASET	LEVEL	DPMN-GroupBU (coherent)	DPMN-NaiveBU (coherent)	HierE2E ^a (coherent)	PERMBU-MinT ^b (coherent)	ARIMA (not coherent)	GLM-Poisson (not coherent)
Traffic	Overall	0.0907 ± 0.0024	0.0704 ± 0.0014	0.0375 ± 0.0058	0.0677 ± 0.0061	0.0751	0.0771
	1 (geo.)	0.0397 ± 0.0044	0.0134 ± 0.0022	0.0183 ± 0.0091	0.0331 ± 0.0085	0.0376	0.0063
	2 (geo.)	0.0537 ± 0.0024	0.0289 ± 0.0017	0.0183 ± 0.0081	0.0341 ± 0.0081	0.0412	0.0194
	3 (geo.)	0.0538 ± 0.0022	0.0290 ± 0.0011	0.0209 ± 0.0071	0.0417 ± 0.0061	0.0549	0.0406
	4 (geo.)	0.2155 ± 0.0022	0.2101 ± 0.0008	0.0974 ± 0.0021	0.1621 ± 0.0027	0.1665	0.2420
Tourism-L	Overall	0.1249 ± 0.0020	0.1274 ± 0.0028	0.1472 ± 0.0029	–	0.1416	0.1762
	1 (geo.)	0.0431 ± 0.0042	0.0514 ± 0.0030	0.0842 ± 0.0051	–	0.0263	0.0854
	2 (geo.)	0.0637 ± 0.0032	0.0705 ± 0.0026	0.1012 ± 0.0029	–	0.0904	0.1153
	3 (geo.)	0.1084 ± 0.0033	0.1068 ± 0.0019	0.1317 ± 0.0022	–	0.1389	0.1691
	4 (geo.)	0.1554 ± 0.0025	0.1507 ± 0.0014	0.1705 ± 0.0023	–	0.1878	0.2165
	5 (prp.)	0.0700 ± 0.0038	0.0907 ± 0.0061	0.0995 ± 0.0061	–	0.0770	0.0954
	6 (prp.)	0.1070 ± 0.0023	0.1175 ± 0.0047	0.1336 ± 0.0042	–	0.1270	0.1682
	7 (prp.)	0.1887 ± 0.0032	0.1836 ± 0.0038	0.1955 ± 0.0025	–	0.2022	0.2458
8 (prp.)	0.2629 ± 0.0034	0.2481 ± 0.0026	0.2615 ± 0.0016	–	0.2834	0.3134	
Favorita	Overall	0.4020 ± 0.0182	0.5301 ± 0.0120	0.5298 ± 0.0091	0.4670 ± 0.0096	0.4373	0.4524
	1 (geo.)	0.2760 ± 0.0149	0.4166 ± 0.0195	0.4714 ± 0.0103	0.2692 ± 0.0076	0.3112	0.3611
	2 (geo.)	0.3865 ± 0.0207	0.5128 ± 0.0108	0.5182 ± 0.0107	0.3824 ± 0.0092	0.4183	0.4398
	3 (geo.)	0.4068 ± 0.0206	0.5317 ± 0.0115	0.5291 ± 0.0129	0.6838 ± 0.0108	0.4446	0.4598
	4 (geo.)	0.5387 ± 0.0253	0.6594 ± 0.0150	0.6012 ± 0.0131	0.5532 ± 0.0116	0.5749	0.5490

^aThe HierE2E results differ from Rangapuram et al. (2018), as the sCRPS quantile interval space has granularity of 1% over its original 5%.

^bPERMBU-MinT on Tourism-L is unavailable because the original implementation cannot currently be applied to structures beyond single hierarchies.

For our proposed methods, we report the DPMN-NaiveBU and the DPMN-GroupBU. As described in Section 4, the DPMN-NaiveBU treats the bottom-level series as independent observations, and the DPMN-GroupBU considers groups of time series during its composite likelihood estimation. Both methods obtain probabilistically coherent forecasts using the bottom-up reconciliation. The comparison of the DPMN variants serves as an ablation experiment to better analyze the source of the accuracy improvements. It also showcases the ability of the Poisson mixture model to give good results for unseen hierarchical structures—and, in the case of the Traffic dataset, of uninformative or noisy time-series group structures—to explore the limits of the GroupBU estimation method.

Table 3 contains the sCRPS measurements for the predictive distributions at each aggregate level through the whole dataset hierarchy. The top row reports the overall sCRPS score (averaged across all hierarchy levels). We highlight the best result in bold. The DPMN significantly and consistently improves the overall sCRPS for Tourism-L and Favorita. In particular, the DPMN-GroupBU variant shows improvements of 11.8% against the second-best alternative in the Tourism-L dataset, and of 8.1% against the second-best choice in the Favorita dataset. In the Traffic dataset, the DPMN-GroupBU variant does not benefit from modeling the uninformative correlations between highways, and subsequently does not improve upon the other compared

Table 4

Empirical evaluation of mean hierarchical forecasts. Relative mean squared error (relMSE), averaged over eight runs, at each aggregation level. The best result is highlighted (lower measurements are preferred). Methods without standard deviation have deterministic solutions.

DATASET	LEVEL	DPMN-GroupBU (hier.)	DPMN-NaiveBU (hier.)	ARIMA-ERM ^a (hier.)	ARIMA- MinT-ols (hier.)	ARIMA- NaiveBU (hier.)	ARIMA (not hier.)	GLM- Poisson (not hier.)	Seasonal- Naive (not hier.)
Traffic	Overall	0.1750 ± 0.0099	0.0168 ± 0.0026	0.0199	0.0425	0.0217	0.0433	0.0175	0.0709
	1 (geo.)	0.1619 ± 0.0099	0.0033 ± 0.0026	0.0133	0.0344	0.0168	0.0302	0.0001	0.0547
	2 (geo.)	0.1835 ± 0.0101	0.0240 ± 0.0027	0.0135	0.0380	0.0180	0.0392	0.0109	0.0676
	3 (geo.)	0.1819 ± 0.0100	0.0239 ± 0.0027	0.0373	0.0647	0.0295	0.0850	0.0462	0.0989
	4 (geo.)	0.9964 ± 0.043	0.9561 ± 0.0022	0.6355	0.5876	0.5669	0.5669	1.2119	1.3118
Tourism-L	Overall	0.1113 ± 0.0158	0.2680 ± 0.0748	0.1178	0.1251	0.2979	0.1414	0.1944	0.1306
	1 (geo.)	0.0597 ± 0.0212	0.3371 ± 0.1506	0.0596	0.0472	0.4002	0.0343	0.2015	0.0582
	2 (geo.)	0.1121 ± 0.0152	0.3186 ± 0.1130	0.1293	0.1476	0.3340	0.2530	0.2274	0.1628
	3 (geo.)	0.2250 ± 0.0196	0.3909 ± 0.0822	0.2529	0.3556	0.4238	0.4429	0.3913	0.3695
	4 (geo.)	0.2980 ± 0.0197	0.4198 ± 0.0668	0.3236	0.4288	0.4012	0.4835	0.4238	0.4766
	5 (prp.)	0.0798 ± 0.0195	0.1459 ± 0.0177	0.0895	0.0856	0.1703	0.0973	0.0961	0.0615
	6 (prp.)	0.1403 ± 0.0150	0.1576 ± 0.0113	0.1466	0.1537	0.1986	0.1663	0.1840	0.1577
	7 (prp.)	0.2654 ± 0.0212	0.2537 ± 0.0100	0.2705	0.3017	0.3151	0.2914	0.3293	0.3699
	8 (prp.)	0.3302 ± 0.0235	0.3030 ± 0.0083	0.3543	0.3970	0.3769	0.3769	0.3908	0.4969
Favorita	Overall	0.7563 ± 0.0713	0.9533 ± 0.0201	0.8163	0.9465	0.8276	0.9665	0.8346	1.1420
	1 (geo.)	0.7944 ± 0.0568	0.9188 ± 0.0187	0.8362	0.8999	0.8415	0.9217	0.9054	1.1269
	2 (geo.)	0.7355 ± 0.1057	1.0451 ± 0.0310	0.7830	1.0057	0.8050	1.0451	0.8037	1.1078
	3 (geo.)	0.7303 ± 0.1035	1.0317 ± 0.0333	0.7986	1.0418	0.8192	1.0881	0.8003	1.1315
	4 (geo.)	0.6770 ± 0.0351	0.8090 ± 0.0180	0.8199	0.8808	0.8228	0.8228	0.6499	1.2815

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

methods. We hypothesize that holiday features explain the Traffic New Year's Day performance gap between HierE2E's and alternative approaches. As neither DPMN nor other baselines use these features. The DPMN-NaiveBU variant performs well on Traffic relative to statistical baselines, and gives acceptable performance on Tourism-L and Favorita compared to all alternatives.

Our results confirm observations from the community that a shared model, capable of learning from all the time series jointly, improves the forecasts over those from univariate time series methods. Additionally, the qualitative comparison¹¹ between the NaiveBU and GroupBU methods shows that an expressive joint distribution framework capable of leveraging the hierarchical structure of the data, when informative, benefits the forecasts' accuracy.

6.7. Complementary mean forecasting results

As shown in Section 3, the DPMN's multivariate Poisson mixture defines a probabilistically coherent system for its forecast distributions; the mean hierarchical coherence is naturally implied. In this experiment, we compare DPMN mean hierarchical forecasts (weighted average of Poisson rates) with the following point forecasting methods' forecasts: (1) ARIMA-ERM (Souhaib & Bonsoo, 2019), which performs an optimization-based reconciliation free of the unbiasedness assumption of the base forecasts; (2) ARIMA-MinT (Wickramasuriya et al., 2019), meant to reconcile unbiased independent forecasts and minimize the variance of the forecast errors; (3) ARIMA-NaiveBU (Orcutt et al., 1968), which produces univariate bottom-level time series forecasts independently and then sums them according to the hierarchical constraints;

(4) automatic ARIMA (Hyndman & Khandakar, 2008); (5) GLM-Poisson (Nelder & Wedderburn, 1972); and (6) the SeasonalNaive model.

To evaluate these, we take recommendations from (Hyndman & Koehler, 2006) and define the relative mean square error (relMSE) based on the following equation:

$$\text{relMSE}(\mathbf{y}_{[i],\tau}, \hat{\mathbf{y}}_{[i],\tau}, \tilde{\mathbf{y}}_{[i],\tau}) = \frac{\text{MSE}(\mathbf{y}_{[i],\tau}, \hat{\mathbf{y}}_{[i],\tau})}{\text{MSE}(\mathbf{y}_{[i],\tau}, \tilde{\mathbf{y}}_{[i],\tau})}, \quad (25)$$

where $\mathbf{y}_{[i],\tau}$, $\hat{\mathbf{y}}_{[i],\tau}$, and $\tilde{\mathbf{y}}_{[i],\tau} \in \mathbb{R}^{N \times h}$ represent the time series observations, the mean forecasts, and the Naive1 baseline forecasts, respectively.

Table 4 contains the relMSE measurements for the predicted means at each aggregation level. The top row reports the overall relMSE (averaged across all hierarchy levels). We highlight the best result in bold. DPMN shows overall improvements or comparable results with the baselines'. With respect to mean hierarchical baselines, DPMN shows 4% Traffic improvements, 5% Tourism-L improvements, and Favorita improvements of 7%.

7. Conclusions and future work

In this work, we introduced a novel method for coherent probabilistic forecasting, the deep Poisson mixture network (DPMN), which focuses on learning the joint distribution of bottom-level time series and naturally guarantees hierarchical probabilistic coherence. We also showed through empirical evaluations that our model is accurate for count data. We observed significant overall improvements in the sCRPS when compared with previous state-of-the-art probabilistically coherent models on two hierarchical datasets: Australian domestic tourism (11.8%) and Ecuadorian grocery sales (8.1%). However, the model did not show improvement in the sCRPS over alternative approaches when evaluated on San Francisco Bay Area traffic data.

¹¹ Figs. 5(a) and 5(b) show a qualitative exploration of the DPMN-NaiveBU and DPMN-GroupBU versions.

The framework presented here is also extensible. We chose to focus on forecasting count data and used Poisson kernels, but one could also use Gaussian kernels to model the joint distributions of real-valued hierarchical data. In fact, any kernel which admits closed-form expression for aggregated distributions under conditional independence akin to Eq. (6) will work well, and this includes kernels like the gamma and negative binomial distributions, in addition to the Poisson and Gaussian distributions already mentioned.

With respect to the definition of the groups considered in DPMN-GroupBU, we followed the natural structure of the data and defined them based on geographic proximity in this work. A promising line of research is an informed creation of such groups based on the series characteristics, for example via clustering.

By formulating the model as a mixture density network, we separated the probabilistic model of the predictive distribution from the underlying network, making it compatible with any other architecture. In the current paper, we relied on the convolutional encoder version of the MQ-Forecaster architecture, but significant progress has been made in the last few years on neural network-based forecasting models, for example, transformer-based deep learning architectures (Eisenach et al., 2021), that can improve performance. We plan to explore both directions, new kernels, and new neural network architectures in future work.

DPMN has its drawbacks as well. As is the case with any finite mixture model, the fidelity of the estimated distribution depends on the number of mixture components. A few hundred samples may be sufficient to describe a single marginal distribution, but can be too sparse to describe the joint distribution in a high-dimensional space. The sparsity will be particularly obvious if customers of hierarchical forecasting are interested in forecast distributions conditioned on partially observed data. The small number of samples will lead to overly confident posterior distributions. Another issue is the model misspecification during inference. The weight matching method performs quite well in empirical evaluations but is somewhat unsatisfactory as a statistical model. To mitigate both issues, we are exploring generative factor models where the mixture components are truly samples from an underlying distribution and correlations between marginal distributions will be captured by common factors. It will bring DPMN closer to a standard hierarchical Bayesian formulation but with fewer and less strict assumptions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. DPMN's probabilistic coherence

In this appendix, we prove DPMN's probabilistic coherence. Given access to the joint bottom-level forecast

probability $\hat{\mathbb{P}}_{[b]}$ defined in Eq. (5), and given the aggregation rule for $\hat{\mathbb{P}}_{[a]}$ defined in Eq. (9), the implied forecast probability for the hierarchical series $\hat{\mathbb{P}}_{[a,b]}$ is coherent and satisfies Definition 2.1. The proof first shows that $\hat{\mathbb{P}}_{[a]}$ is well defined, and then shows that DPMN's aggregate marginal probability assigns a zero probability to any set that does not contain any coherent forecasts, which implies probabilistic coherence.

Lemma A.1. *Let $(\Omega_{[b]}, \mathcal{F}_{[b]}, \hat{\mathbb{P}}_{[b]})$ be a probabilistic forecast space, where $\mathcal{F}_{[b]}$ is a σ -algebra on $\Omega_{[b]}$. The aggregation rule defines a probability measure over $\Omega_{[a]} = \mathbf{A}_{[a][b]}(\Omega_{[b]})$:*

$$\hat{\mathbb{P}}_{[a]}(\mathbf{y}_{[a]}) = \int_{\Omega_{[b]}} \hat{\mathbb{P}}_{[b]}(\mathbf{y}_{[b]}) \mathbb{1}\{\mathbf{y}_{[a]} = \mathbf{A}_{[a][b]}\mathbf{y}_{[b]}\} d\mathbf{y}_{[b]} \quad (\text{A.1})$$

Proof. We prove that $\hat{\mathbb{P}}_{[a]}$ satisfies the Kolmogorov axioms on $(\Omega_{[a]}, \mathcal{F}_{[a]}, \hat{\mathbb{P}}_{[a]})$ with $\Omega_{[a]} = \mathbf{A}_{[a][b]}(\Omega_{[b]})$.

1. $\hat{\mathbb{P}}_{[a]}(\mathcal{A}) \geq 0 \quad \forall \mathcal{A} \in \mathcal{F}_{[a]}$: This follows from the positivity of $\hat{\mathbb{P}}_{[b]}(\mathcal{B})$ and the indicator function.
2. $\hat{\mathbb{P}}_{[a]}(\Omega_{[a]}) = 1$: The unit measure assumption holds because

$$\begin{aligned} \hat{\mathbb{P}}_{[a]}(\mathbf{A}_{[a][b]}(\Omega_{[b]})) &= \int_{\Omega_{[a]}} \int_{\Omega_{[b]}} \hat{\mathbb{P}}_{[b]}(\mathbf{y}_{[b]}) \mathbb{1}\{\mathbf{y}_{[a]} \\ &= \mathbf{A}_{[a][b]}\mathbf{y}_{[b]}\} d\mathbf{y}_{[b]} d\mathbf{y}_{[a]} \\ &= \int_{\Omega_{[b]}} \hat{\mathbb{P}}_{[b]}(\mathbf{y}_{[b]}) d\mathbf{y}_{[b]} = 1. \end{aligned}$$

3. $\hat{\mathbb{P}}_{[a]}(\bigcup_{i=1}^{\infty} \mathcal{A}_i) = \sum_{i=1}^{\infty} \hat{\mathbb{P}}_{[a]}(\mathcal{A}_i)$ for disjoint sets \mathcal{A}_i 's: The σ -additivity assumption holds that

$$\begin{aligned} \hat{\mathbb{P}}_{[a]} \left(\bigcup_{i=1}^{\infty} \mathcal{A}_i \right) &= \hat{\mathbb{P}}_{[a]} \left(\bigcup_{i=1}^{\infty} \mathbf{A}_{[a][b]}(\mathcal{B}_i) \right) \\ &= \int \hat{\mathbb{P}}_{[b]} \left(\bigcup_{i=1}^{\infty} \mathcal{B}_i \right) \mathbb{1}\{\mathbf{y}_{[a]} \\ &= \mathbf{A}_{[a][b]}\mathbf{y}_{[b]}\} d\mathbf{y}_{[b]} \\ &= \int \hat{\mathbb{P}}_{[b]} \left(\bigcup_{i=1}^{\infty} \mathcal{B}_i \right) d\mathbf{y}_{[b]} = \sum_{i=1}^{\infty} \hat{\mathbb{P}}_{[b]}(\mathcal{B}_i) \\ &= \sum_{i=1}^{\infty} \hat{\mathbb{P}}_{[a]}(\mathcal{A}_i). \quad \square \end{aligned}$$

Lemma A.2. *Let $(\Omega_{[b]}, \mathcal{F}_{[b]}, \hat{\mathbb{P}}_{[b]})$ be a probabilistic forecast space, where $\mathcal{F}_{[b]}$ is a σ -algebra on $\Omega_{[b]}$. If a forecast distribution assigns a zero probability to sets that do not contain coherent forecasts, it defines a coherent probabilistic forecast space $(\Omega_{[a,b]}, \mathcal{F}_{[a,b]}, \hat{\mathbb{P}}_{[a,b]})$ with $\Omega_{[a,b]} = \mathbf{S}_{[a,b][b]}(\Omega_{[b]})$.*

$$\begin{aligned} \hat{\mathbb{P}}_{[a]}(\mathbf{y}_{[a]} \notin \mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B}) &= 0 \\ \implies \hat{\mathbb{P}}_{[a,b]}(\mathbf{S}_{[a,b][b]}(\mathcal{B})) &= \hat{\mathbb{P}}_{[b]}(\mathcal{B}) \quad \forall \mathcal{B} \in \mathcal{F}_{[b]} \quad (\text{A.2}) \end{aligned}$$

Proof.

$$\begin{aligned} \hat{\mathbb{P}}_{[a,b]}(\mathbf{S}_{[a,b][b]}(\mathcal{B})) &= \hat{\mathbb{P}}_{[a,b]} \left(\left[\begin{array}{c} \mathbf{A}_{[a][b]} \\ \mathbf{I}_{[b][b]} \end{array} \right] (\mathcal{B}) \right) \\ &= \hat{\mathbb{P}}_{[a,b]} \left(\left\{ \left[\begin{array}{c} \mathbf{A}_{[a][b]} \\ \mathbb{R}^{N_b} \end{array} \right] (\mathcal{B}) \right\} \cap \left\{ \left[\begin{array}{c} \mathbb{R}^{N_a} \\ \mathcal{B} \end{array} \right] \right\} \right) \end{aligned}$$

$$\begin{aligned} &= \hat{\mathbb{P}}_{[a]}(\mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B}) \hat{\mathbb{P}}_{[b]}(\mathcal{B}) \\ &= (1 - \hat{\mathbb{P}}_{[a]}(\mathbf{y}_{[a]} \notin \mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B})) \\ &\quad \times \hat{\mathbb{P}}_{[b]}(\mathcal{B}) = \hat{\mathbb{P}}_{[b]}(\mathcal{B}). \end{aligned}$$

The first equality is the image of a set $\mathcal{B} \in \Omega_{[b]}$ corresponding 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. And the final equality uses the zero probability assumption. \square

Theorem A.3. *With $(\Omega_{[b]}, \mathcal{F}_{[b]}, \hat{\mathbb{P}}_{[b]})$ probabilistic forecast space, we can construct a coherent probabilistic forecast space $(\Omega_{[a,b]}, \mathcal{F}_{[a,b]}, \hat{\mathbb{P}}_{[a,b]})$ with Lemma A.1's aggregation.*

Proof. It follows from the aggregation rule in Section 3.3 that $\hat{\mathbb{P}}_{[a]}(\mathbf{y}_{[a]} \notin \mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B}) = 0$. Using Lemma A.2, we obtain a probabilistic coherent space $(\Omega_{[a,b]}, \mathcal{F}_{[a,b]}, \hat{\mathbb{P}}_{[a,b]})$. \square

Appendix B. Covariance formula

Here we present the derivation of the covariance formula in Eq. (10).

Proof. Using the law of total covariance, we get

$$\begin{aligned} \text{Cov}(Y_{\beta,\tau}, Y_{\beta',\tau'}) &= \mathbb{E}[\text{Cov}(Y_{\beta,\tau}, Y_{\beta',\tau'} \mid \lambda_{\beta,\kappa,\tau}, \lambda_{\beta',\kappa,\tau'})] \\ &\quad + \text{Cov}(\mathbb{E}[Y_{\beta,\tau} \mid \lambda_{\beta,\kappa,\tau}], \mathbb{E}[Y_{\beta',\tau'} \mid \lambda_{\beta',\kappa,\tau'}]) \end{aligned} \quad (\text{B.1})$$

Using the conditional independence from Eq. (6), we can rewrite the expectation of the conditional covariance:

$$\begin{aligned} &\mathbb{E}[\text{Cov}(Y_{\beta,\tau}, Y_{\beta',\tau'} \mid \lambda_{\beta,\kappa,\tau}, \lambda_{\beta',\kappa,\tau'})] \\ &= \mathbb{E}[\text{Var}(Y_{\beta,\tau} \mid \lambda_{\beta,\kappa,\tau}) \mathbb{1}(\beta = \beta') \mathbb{1}(\tau = \tau')] \\ &= \mathbb{E}[\lambda_{\beta,\kappa,\tau} \mathbb{1}(\beta = \beta') \mathbb{1}(\tau = \tau')] \\ &= \bar{\lambda}_{\beta,\tau} \mathbb{1}(\beta = \beta') \mathbb{1}(\tau = \tau') \end{aligned} \quad (\text{B.2})$$

where $\bar{\lambda}_{\beta,\tau} = \sum_{\kappa=1}^{N_k} w_{\kappa} \lambda_{\beta,\kappa,\tau}$.

In the second term, because the conditional distributions are Poisson, we have

$$\mathbb{E}[Y_{\beta,\tau} \mid \lambda_{\beta,\kappa,\tau}] = \lambda_{\beta,\kappa,\tau} \quad \text{and} \quad \mathbb{E}[Y_{\beta',\tau'} \mid \lambda_{\beta',\kappa,\tau'}] = \lambda_{\beta',\kappa,\tau'}$$

which implies

$$\begin{aligned} &\text{Cov}(\mathbb{E}[Y_{\beta,\tau} \mid \lambda_{\beta,\kappa,\tau}], \mathbb{E}[Y_{\beta',\tau'} \mid \lambda_{\beta',\kappa,\tau'}]) \\ &= \sum_{\kappa=1}^{N_k} w_{\kappa} (\lambda_{\beta,\kappa,\tau} - \bar{\lambda}_{\beta,\tau}) (\lambda_{\beta',\kappa,\tau'} - \bar{\lambda}_{\beta',\tau'}) \end{aligned} \quad (\text{B.3})$$

Therefore, the covariance formula is:

$$\begin{aligned} \text{Cov}(Y_{\beta,\tau}, Y_{\beta',\tau'}) &= \bar{\lambda}_{\beta,\tau} \mathbb{1}(\beta = \beta') \mathbb{1}(\tau = \tau') \\ &\quad + \sum_{\kappa=1}^{N_k} w_{\kappa} (\lambda_{\beta,\kappa,\tau} - \bar{\lambda}_{\beta,\tau}) (\lambda_{\beta',\kappa,\tau'} - \bar{\lambda}_{\beta',\tau'}). \quad \square \end{aligned}$$

Table A.1

San Francisco Bay Area highway traffic.	
Geographical level ^a	Series per level
Bay Area	1
Halves	2
Quarters	4
Bottom	200
Total	207

^aThe hierarchical structure is randomly defined.

Appendix C. Mixture components and covariance matrix rank

As mentioned in Section 6.5, complex correlations across series benefit from a higher number of mixture components. We ground this intuition on the expressiveness of DPMN's Poisson mixture covariance matrix controlled by its rank. We can show that for DPMN's finite Poisson mixture distribution, the bottom level's estimator of the *non-diagonal* covariance matrix series is a matrix of rank at most $K - 1$, given by:

$$\text{Cov}(\mathbf{y}_{[b],\tau}) = \sum_{\kappa=1}^{N_k} \mathbf{w}_{\kappa} (\lambda_{[b],\kappa,\tau} - \bar{\lambda}_{[b],\tau}) (\lambda_{[b],\kappa,\tau} - \bar{\lambda}_{[b],\tau})^T \in \mathbb{R}^{N_b \times N_b} \quad (\text{C.1})$$

Proof. One can easily extend the pair-wise covariance from Eq. (B.3) to multivariate covariance Eq. (C.1).

Let $\mathbf{z}_{\kappa} = \mathbf{w}_{\kappa} (\lambda_{[b],\kappa,\tau} - \bar{\lambda}_{[b],\tau})$. By construction we can show that $\sum_{\kappa=1}^{N_k} \mathbf{z}_{\kappa} = \mathbf{0}$.

Rewriting the last vector $\mathbf{z}_{\kappa} = -\sum_{\kappa=1}^{N_k-1} \mathbf{z}_{\kappa}$, we obtain a sum of $K - 1$ rank-1 matrices

$$\begin{aligned} \sum_{\kappa=1}^{N_k} \frac{1}{\mathbf{w}_{\kappa}} \mathbf{z}_{\kappa} \mathbf{z}_{\kappa}^T &= \sum_{\kappa=1}^{N_k-1} \frac{1}{\mathbf{w}_{\kappa}} \mathbf{z}_{\kappa} \mathbf{z}_{\kappa}^T + \left(-\sum_{\kappa=1}^{N_k-1} \mathbf{z}_{\kappa} \right) \frac{1}{\mathbf{w}_K} \mathbf{z}_K^T \\ &= \sum_{\kappa=1}^{N_k-1} \mathbf{z}_{\kappa} \left(\frac{\mathbf{z}_{\kappa}}{\mathbf{w}_{\kappa}} - \frac{\mathbf{z}_K}{\mathbf{w}_K} \right)^T \end{aligned}$$

which implies that DPMN's modeled covariance matrix rank is upper bounded by $N_k - 1$. \square

Appendix D. Dataset details

The Traffic dataset, as mentioned, measures the occupancy rates of 963 freeway lanes from the Bay Area. The original data are at a 10-minute frequency from January 1st, 2008 to March 30th, 2009. The dataset is further aggregated from the 10-minute frequency into daily frequency with 366 observations. We match the sample procedure from previous hierarchical forecasting literature (Rangapuram et al., 2021; Souhaib & Bonsou, 2019), and use the same 200 bottom-level series from the 963 available. From these 200 bottom-level series, a hierarchy is randomly defined by aggregating them into quarters and halves of 50 and 100 series each. Finally, we consider the total aggregation. Table A.1 describes the hierarchical structure.

Table A.2
Australian tourism flows.

Geographical level	Series per level	Series per level & purpose	Total
Australia	1	4	5
States	7	28	35
Zones	27	108	135
Regions	76	304	380
Total	111	444	555

Table A.3
Favorita grocery sales.

Geographical level	Nodes per level	Series per level	Total
Ecuador	1	4036	4036
States	16	64,576	64,576
Cities	22	88,792	88,792
Stores	54	217,944	217,944
Total	93	371,312	371,312

The *Tourism-L* dataset contains 555 monthly series from 1998 to 2016. It is organized by geography and purpose of travel. The four-level geographical hierarchy comprises seven states, divided further into 27 zones and 76 regions. The categories for purpose of travel are holiday, visiting friends and relatives, business, and other. This dataset has been referenced by important hierarchical forecasting studies like the one of the *MinT* reconciliation strategy and the more recent *HierE2E* (Rangapuram et al., 2021; Wickramasuriya et al., 2019). *Tourism-L* is a grouped dataset. It has two dimensions in which it is aggregated: the total level aggregation, and its four associated purposes. Table A.2 describes the group and hierarchical structures.

The *Favorita* dataset, once balanced for items and stores, contains 217,944 bottom-level series. In contrast, the original competition considered 210,645 series. We resort for this balance because, for the moment, the *GroupBU* version of the *PMM* requires balanced hierarchies for its estimation. In the case of the *Favorita* experiment, we consider a geographical hierarchy (93 nodes) conditional of each of grocery item (4036). The hierarchy defines 153,368 new aggregate series at the item–country, item–state, and item–city levels. Table A.3 describes the structure.

Regarding the dataset preprocessing, we confirmed observations from the best submissions to the *Kaggle competition*. Most holiday distances included in the dataset, and covariates like oil production, lack value for the forecasts. The models did not benefit from a long history. Filtering the training window to the 2017 year produced consistently better results.

Appendix E. Poisson mixture size ablation study

As shown in Appendix B, the job of *DPMN*’ mixture distribution goes beyond forecasting single time series by modeling correlations across them too. Based on the theoretical properties of covariance matrix expressiveness, it

is reasonable to expect that the number of optimal components will grow with the complexity of the modeled time series structure.

In this ablation study, we empirically test these intuitions, showing how the number of optimally selected *DPMN*’s mixture components grows with larger datasets. For the experiment, we measured the cross-validation performance of *DPMN*’ configurations as defined in Table 2, explored automatically with *HYPEROPT* (Bergstra et al., 2011), where we fix the number of mixture components.

Table A.4 reports the validation probabilistic forecast accuracy measured with *sCRPS*, across *Traffic*, *Tourism-L*, and *Favorita*, for *DPMN-GroupBU* with different numbers of Poisson mixture components. For this experiment, we report the overall validation *sCRPS* averaged over four independent *HYPEROPT* runs with 12 optimization steps and eight steps in the *Traffic* dataset.

Table A.4’s *sCRPS* measurements suggest that there is a bias–variance tradeoff controlled by the Poisson mixture size. When $K = 1$, the *DPMN-GroupBU* model corresponds to Poisson regression and treats each series as probabilistically independent, such that the high-bias simple model produced forecasts with the worst *sCRPS*. The forecast accuracy improves as the number of Poisson components increases from $K = 1$, but the accuracy begins to deteriorate beyond a certain threshold. We hypothesize that a small number of mixture components does not have enough degrees of freedom to describe the data, and too many mixture components lead to over-fitting the training data, resulting in large variance in the validation data.

We observed that the optimal number of Poisson mixture components varies across the datasets. Larger datasets, or datasets with a complex time series correlation structure, appear to benefit from more flexible probability mixtures (see Fig. E.1). *Traffic*, our smallest dataset, produced optimal results with $K = 25$ components, *Tourism-L*, a medium-sized dataset, produced optimal results with $K = 25$ components. Finally, *Favorita* our largest dataset, did not saturate even with the largest number of components we experimented with; we capped the choice of the number of mixture components at $K = 100$, due to GPU memory constraints.

Appendix F. Model parameter details

As mentioned in Section 6.5, for the overall hyperparameter selection, we used a standard two-stage approach where we fixed the architecture and the probability distribution to estimate (and implicit training loss), and a second stage where we optimized the architecture’s training procedure.

In the first stage, we carefully fixed the architecture and the probability distribution to estimate. The most important heuristic guiding this selection was to increase the architecture’s and probability’s capacity for larger or more complex datasets. To increase the network’s capacity, we increased the number of convolution layers N_{cl} and convolution filters N_p , as well as the mixture weight and rate decoder layers N_{wdl} , N_{rdl} . In particular, since *Traffic* is the smallest dataset, we opted for a reduced model

Table A.4

Empirical evaluation of probabilistically coherent forecasts for different DPMN-GroupBU, varying the Poisson mixture size. Mean overall scaled continuous ranked probability score (sCRPS).

DATASET	LEVEL	$K = 1$	$K = 10$	$K = 25$	$K = 50$	$K = 100$
Traffic	Overall	0.1647 \pm 0.0009	0.1435 \pm 0.0947	0.0958 \pm 0.0005	0.1337 \pm 0.0004	0.1261 \pm 0.0037
Tourism-L	Overall	0.1673 \pm 0.0052	0.1380 \pm 0.0017	0.1247 \pm 0.0025	0.1284 \pm 0.0031	0.1251 \pm 0.0034
Favorita	Overall	0.8390 \pm 0.0124	0.4416 \pm 0.0152	-	0.4204 \pm 0.0108	0.3758 \pm 0.0040

The best result is highlighted (lower measurements are preferred).

Table A.5

Deep Poisson mixture network (DPMN) architecture parameters configured once per dataset. These hyperparameters correspond to the first selection phase preceding the automatic optimization.

PARAMETER	Notation	Considered values		
		Traffic	Tourism-L	Favorita
SGD Batch Size. ^a	-	4	4	100
Activation Function.	-	ReLU	ReLU	ReLU
Temporal Convolution Kernel Size.	N_{ck}	{7}	{2}	{2}
Temporal Convolution Layers.	N_{cl}	{3}	{5}	{5}
Temporal Convolution Filters.	N_p	{10}	{30}	{30}
Future Encoder Dimension.	N_f	{50}	{50}	{50}
Static Encoder Dimension.	N_s	{100}	{100}	{100}
Horizon Agnostic Decoder Dimensions.	N_{ag}	{50}	{50}	{50}
Horizon Specific Decoder Dimensions.	N_{sp}	{20}	{20}	{20}
Poisson Mixture Weight Decoder Layers.	N_{wdl}	{3}	{4}	{4}
Poisson Mixture Rate Decoder Layers.	N_{rdl}	{2}	{3}	{3}
Poisson Mixture Components.	N_k	{25}	{25}	{100}
GPU Training configuration.	-	2 \times NVIDIA V100	2 \times NVIDIA V100	4 \times NVIDIA V100

^aSGD batch selection follows an ablation study considering values between {2, 4, 8, 16, 32, 64, 100}. We report the best validation batch size.

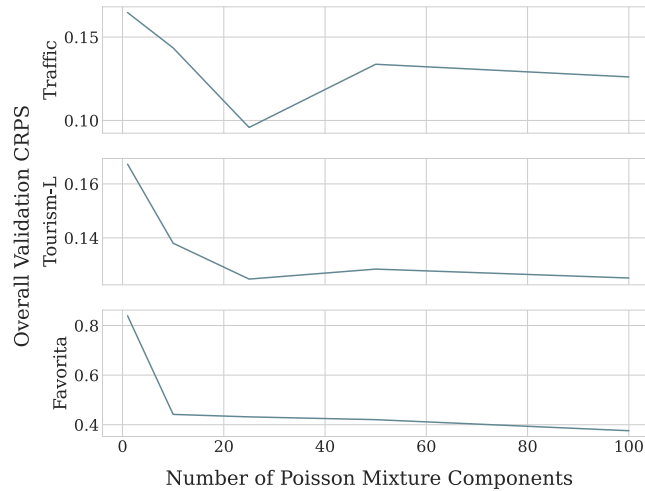


Fig. E.1. Poisson mixture size ablation study. We found interesting bias-variance tradeoffs controlled by the number of mixture components. Both Traffic and Tourism-L have an optimal value of 25 components, beyond which the sCRPS validation performance worsens. We observed a classic “U”-shaped pattern. In the case of Favorita, the largest dataset, the validation sCRPS continued to improve through $K = 100$.

size and the number of Poisson mixture components to control the model’s variability. Additionally, due to the dataset’s strong weekly seasonality pattern, we adjusted the convolution kernel size to encompass seven days. We controlled the probability’s capacity with the mixture size and SGD batchsize. In an ablation study similar to Appendix E, we found that for datasets with strong correlations like Favorita and Tourism-L maximizing the batch size with respect to GPU memory limitations resulted in better validation performance; for Traffic, though the entire dataset could fit in memory at once, it

was preferable to feed in subsets to allow the model to learn from different randomly sampled highway groups in each epoch.

For the second hyperparameter selection stage, as reported in Section 6.5, for each fixed architecture and probability, we optimally explored its training procedure hyperparameters defined in Table 2 using the HYPEROPT algorithm’s Bayesian optimization (Bergstra et al., 2011). The second phase only considered the optimal exploration of the learning rate, random initialization, and the number of SGD epochs. The selection was guided by a temporal

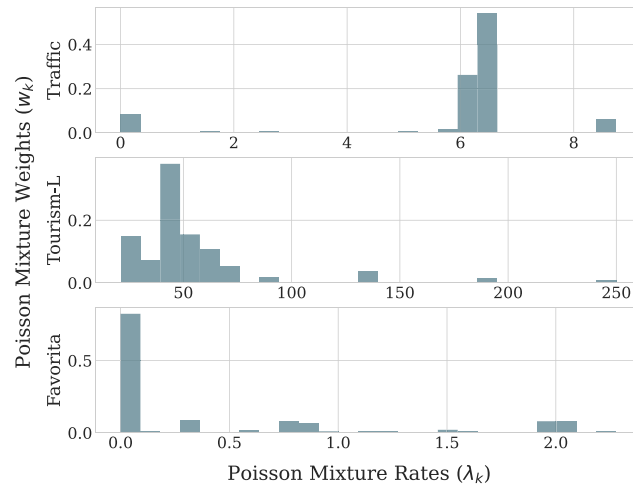


Fig. G.2. Poisson mixture rates distribution for the DPMN-GroupBU model reported in Table 3. A single bottom-level series and a time step ahead in the forecast horizon are considered. The mixture distribution is capable of flexibly modeling multimodal processes, overcoming Poisson regression's limitations for aggregated data, and modeling zero-inflated processes for disaggregated data. These qualities make it exceptionally useful for hierarchical forecasting tasks.

cross-validation signal obtained from the dataset partition introduced in Section 6.3.

Appendix G. Qualitative analysis of Poisson mixture rates

We use a histogram in Fig. G.2 to visualize the distribution of Poisson mixture rates learned by the DPMN-GroupBU method on the Traffic, Tourism-L, and Favorita datasets. These rates correspond to those of the models from Table 3, for a base time series and a single time step ahead in the forecast horizon. In the histogram, we use the learned mixture weights w_k in the vertical axis to account for the probability of each Poisson rate λ_k represented in the horizontal axis.

Exciting patterns arise from visualizing the learned Poisson mixture rates and their weights. First, the DPMN likelihood is capable of modeling zero-inflated data in the case of the Favorita bottom-level series; as we can see from the probability accumulation around zero, this may explain its superior performance on Favorita bottom-level series, which tend to be sparse, as shown in Fig. 6. Second, all the Poisson rate distributions show multiple modes, a quality that cannot be replicated by a Gaussian, unimodal, or symmetric distribution. This observation further motivates using a flexible distribution capable of better modeling the underlying data generation processes. Third, the distribution of Poisson rates in the Tourism-L dataset can overcome the limitations of simple Poisson distributions that tend to have collapsed variances for aggregated series due to their scale, since the variance of the series can still be correctly modeled by the variance of the Poisson rates.

References

Amir, A., & Souhaib, B. (2016). A bias and variance analysis for multistep-ahead time series forecasting. *IEEE Transactions on Neural*

- Networks and Learning Systems*, 27(1), 2162–2388, URL <https://pubmed.ncbi.nlm.nih.gov/25807572/>.
- Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., & Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1), 60–74.
- Ben Taieb, S., Taylor, J. W., & Hyndman, R. J. (2017). Coherent probabilistic forecasts for hierarchical time series. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of machine learning research: vol. 70, Proceedings of the 34th international conference on machine learning* (pp. 3348–3357). PMLR, URL <http://proceedings.mlr.press/v70/taieb17a.html>.
- Ben Taieb, S., Taylor, J. W., & Hyndman, R. J. (2021). Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, 116(533), 27–43. <http://dx.doi.org/10.1080/01621459.2020.1736081>, arXiv:DOI: 10.1080/01621459.2020.1736081.
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, Vol. 24 (pp. 2546–2554). Curran Associates, Inc., URL <https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf>.
- Bishop, C. M. (1994). *Mixture density networks: Technical Report*, Birmingham: Aston University, URL <https://publications.aston.ac.uk/id/eprint/373/>.
- Bolin, D., & Wallin, J. (2019). Local scale invariance and robustness of proper scoring rules. <http://dx.doi.org/10.48550/ARXIV.1912.05642>, URL <https://arxiv.org/abs/1912.05642>.
- Böse, J.-H., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., et al. (2017). Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12), 1694–1705. <http://dx.doi.org/10.14778/3137765.3137775>.
- Chen, T., et al. (2015). MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *Computing Research Repository*, 1512.01274, URL <http://arxiv.org/abs/1512.01274>.
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1724–1734). arXiv:1406.1078, arXiv:1406.1078, URL <http://arxiv.org/abs/1406.1078>.
- Christiansen, C. L., & Morris, C. N. (1997). Hierarchical Poisson regression modeling. *Journal of the American Statistical Association*, 92(438), 618–632.
- Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289.

- Corporación Favorita (2018). Corporación favorita grocery sales forecasting. URL <https://www.kaggle.com/c/favorita-grocery-sales-forecasting/>, Kaggle Competition.
- Diggle, P. J. (2013). *Statistical analysis of spatial and spatio-temporal point patterns* (3rd). Routledge.
- Diggle, P., & Brix, A. (2001). Spatio-temporal prediction for log-Gaussian Cox processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4), 823–841. <http://dx.doi.org/10.1111/1467-9868.00315>.
- Dua, D., & Graff, C. (2017). *UCI machine learning repository*. University of California, Irvine, School of Information and Computer Sciences, URL <http://archive.ics.uci.edu/ml>.
- Eisenach, C., Patel, Y., & Madeka, D. (2021). MQTransformer: Multi-horizon forecasts with context dependent and feedback-aware attention. In M. F. Balcan, & M. Meila (Eds.), *Proceedings of the 38th international conference on machine learning*. PMLR, submitted for publication, Working Paper version available at [arXiv:2009.14799](https://arxiv.org/abs/2009.14799).
- Fliedner, G. (1999). An investigation of aggregate variable time series forecast strategies with specific subaggregate time series statistical correlation. *Computers & Operations Research*, 26(10–11), 1133–1149. [http://dx.doi.org/10.1016/S0305-0548\(99\)00017-9](http://dx.doi.org/10.1016/S0305-0548(99)00017-9).
- Fotios Petropoulos, et al. (2021). Forecasting: Theory and practice. [arXiv:2012.03854](https://arxiv.org/abs/2012.03854).
- Gneiting, T., & Ranjan, R. (2011). Comparing density forecasts using threshold- and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, 29(3), 411–422.
- Gross, C. W., & Sohl, J. E. (1990). Disaggregation methods to expedite product line forecasting. *Journal of Forecasting*, 9(3), 233–254. <http://dx.doi.org/10.1002/for.3980090304>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980090304>.
- Han, X., Dasgupta, S., & Ghosh, J. (2021). Simultaneously reconciled quantile forecasting of hierarchically related time series. In A. Banerjee, & K. Fukumizu (Eds.), *Proceedings of machine learning research: vol. 130, Proceedings of the 24th international conference on artificial intelligence and statistics* (pp. 190–198). PMLR, URL <http://proceedings.mlr.press/v130/han21a.html>.
- Hollyman, R., Petropoulos, F., & Tipping, M. E. (2021). Understanding forecast reconciliation. *European Journal of Operational Research*, 294(1), 149–160. <http://dx.doi.org/10.1016/j.ejor.2021.01.017>, URL <https://www.sciencedirect.com/science/article/pii/S0377221721000199>.
- Hong, T., Pinson, P., & Fan, S. (2014). Global energy forecasting competition 2012. *International Journal of Forecasting*, 30(2), 357–363.
- Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., & Shang, H. L. (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9), 2579–2589. <http://dx.doi.org/10.1016/j.csda.2011.03.006>, URL <http://www.sciencedirect.com/science/article/pii/S0167947311000971>.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. Melbourne, Australia: OTexts, available at <https://otexts.com/fpp2/>.
- Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software, Articles*, 27(3), 1–22. <http://dx.doi.org/10.18637/jss.v027.i03>, URL <https://www.jstatsoft.org/v027/i03>.
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. <http://dx.doi.org/10.1016/j.ijforecast.2006.03.001>, URL <http://www.sciencedirect.com/science/article/pii/S0169207006000239>.
- Hyndman, R. J., Lee, A., & Wang, E. (2014). *Fast computation of reconciled forecasts for hierarchical and grouped time series: Monash Econometrics and Business Statistics Working Papers 17/14*, Monash University, Department of Econometrics and Business Statistics, URL <https://ideas.repec.org/p/msh/ebswps/2014-17.html>.
- Jeon, J., Panagiotelis, A., & Petropoulos, F. (2019). Probabilistic forecast reconciliation with applications to wind power and electric load. *European Journal of Operational Research*, 279(2), 364–379. <http://dx.doi.org/10.1016/j.ejor.2019.05.020>, URL <https://www.sciencedirect.com/science/article/pii/S0377221719304242>.
- Kamarthi, H., Kong, L., Rodriguez, A., Zhang, C., & Prakash, B. (2022). PROFHIT: Probabilistic robust forecasting for hierarchical time-series. *Computing Research Repository*, URL <https://arxiv.org/abs/2206.07940>.
- Kingma, D. P., & Ba, J. (2014). ADAM: A method for stochastic optimization. URL <http://arxiv.org/abs/1412.6980>, cite arxiv:1412.6980 Comment: Published as a conference paper at the 3rd International Conference for Learning Representations (ICLR), San Diego, 2015.
- Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, 80, 221–239.
- Madeka, D., Swiniarski, L., Foster, D., Razoumov, L., Torkkola, K., & Wen, R. (2018). Sample path generation for probabilistic demand forecasting. In *ICML workshop on theoretical foundations and applications of deep generative models*.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018a). The M4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4), 802–808. <http://dx.doi.org/10.1016/j.ijforecast.2018.06.001>, URL <http://www.sciencedirect.com/science/article/pii/S0169207018300785>.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018b). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3), 1–26. <http://dx.doi.org/10.1371/journal.pone.0194889>.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4), 1346–1364, URL https://www.researchgate.net/publication/344487258_The_M5_Accuracy_competition_Results_findings_and_conclusions.
- Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Tsetlin, I., et al. (2022). The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting*, 38(4), 1365–1385. <http://dx.doi.org/10.1016/j.ijforecast.2021.10.009>, URL <https://www.sciencedirect.com/science/article/pii/S0169207021001722>, Special Issue: M5 competition.
- Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, 22(10), 1087–1096, URL <http://www.jstor.org/stable/2629907>.
- Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3), 370–384, URL <http://www.jstor.org/stable/2344614>.
- Olivares, K. G., Challu, C., Marczasz, G., Weron, R., & Dubrawski, A. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2), 884–900. <http://dx.doi.org/10.1016/j.ijforecast.2022.03.001>, URL <https://www.sciencedirect.com/science/article/pii/S0169207022000413>.
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., et al. (2016). WaveNet: A generative model for raw audio. *Computer Research Repository*, [arXiv:1609.03499](https://arxiv.org/abs/1609.03499), [arXiv:1609.03499](https://arxiv.org/abs/1609.03499), URL <http://arxiv.org/abs/1609.03499>.
- Orcutt, G. H., Watts, H. W., & Edwards, J. B. (1968). Data aggregation and information loss. *The American Economic Review*, 58(4), 773–787, URL <http://www.jstor.org/stable/1815532>.
- Panagiotelis, A., Gamakumara, P., Athanasopoulos, G., & Hyndman, R. J. (2020). *Probabilistic forecast reconciliation: Properties, evaluation and score optimisation: Monash Econometrics and Business Statistics Working Papers 26/20*, Monash University, Department of Econometrics and Business Statistics, URL <https://ideas.repec.org/p/msh/ebswps/2020-26.html>.
- Panagiotelis, A., Gamakumara, P., Athanasopoulos, G., & Hyndman, R. J. (2023). Probabilistic forecast reconciliation: Properties, evaluation and score optimisation. *European Journal of Operational Research*, 306(2), 693–706. <http://dx.doi.org/10.1016/j.ejor.2022.07.040>, URL <https://www.sciencedirect.com/science/article/pii/S0377221722006087>.
- Paria, B., Sen, R., Ahmed, A., & Das, A. (2021). Hierarchically regularized deep forecasting. In *Proceedings of the 39th international conference on machine learning*. PMLR, submitted for publication, Working Paper version available at [arXiv:2106.07630](https://arxiv.org/abs/2106.07630).
- Park, B.-J., & Lord, D. (2009). Application of finite mixture models for vehicle crash data analysis. *Accident Analysis and Prevention*, 41(4), 683–691.
- Puwasa, G., Panagiotelis Anastasios, G., & Hyndman, R. J. (2018). Probabilistic forecasts in hierarchical time series. *Department of Econometrics and Business Statistics Working Paper Series 11/18*.
- Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., & Januschowski, T. (2018). Deep state space models for time series forecasting. In S. Bengio, H. Wallach, H. Larochelle,

- K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*, Vol. 31. Curran Associates, Inc., URL <https://proceedings.neurips.cc/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf>.
- Rangapuram, S. S., Werner, L. D., Benidis, K., Mercado, P., Gasthaus, J., & Januschowski, T. (2021). End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In M. F. Balcan, & M. Meila (Eds.), *Proceedings of machine learning research, Proceedings of the 38th international conference on machine learning*. PMLR.
- Ravuri, S. V., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., et al. (2021). Skillful precipitation nowcasting using deep generative models of radar. *Nature*, 597, 672–691, URL <https://www.nature.com/articles/s41586-021-03854-z.pdf>.
- Rosenblatt, F. (1961). *Principles of neurodynamics. Perceptrons and the theory of brain mechanisms: Technical Report*, Cornell Aeronautical Lab Inc Buffalo NY.
- Semenoglou, A.-A., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2021). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, 37(3), 1072–1084. <http://dx.doi.org/10.1016/j.ijforecast.2020.11.009>, URL <https://www.sciencedirect.com/science/article/pii/S0169207020301850>.
- Shang, H. L., & Hyndman, R. J. (2017). Grouped functional time series forecasting: An application to age-specific mortality rates. *Journal of Computational and Graphical Statistics*, 26(2), 330–343. <http://dx.doi.org/10.1080/10618600.2016.1237877>, arXiv:10.1080/10618600.2016.1237877.
- Souhaib, B., & Bonsoo, K. (2019). Regularized regression for hierarchical forecasting without unbiasedness conditions. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1337–1347). New York, NY, USA: Association for Computing Machinery, <http://dx.doi.org/10.1145/3292500.3330976>.
- Spiliotis, E., Petropoulos, F., Kourentzes, N., & Assimakopoulos, V. (2020). Cross-temporal aggregation: Improving the forecast accuracy of hierarchical electricity consumption. *Applied Energy*, 261, Article 114339.
- Tourism Australia, Canberra (2019). Detailed tourism research Australia (2005) travel by Australians. Accessed at <https://robjhyndman.com/publications/hierarchical-tourism/>.
- Van Erven, T., & Cugliari, J. (2015). Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts. In *Modeling and stochastic learning for forecasting in high dimensions* (pp. 297–317). Springer.
- Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 21(1), 5–42, URL <http://www.jstor.org/stable/24309261>.
- Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. In *31st Conference on neural information processing systems NIPS 2017, Time series workshop*. arXiv:1711.11053, URL <https://arxiv.org/abs/1711.11053>.
- Wickramasuriya, S. L. (2023). Probabilistic forecast reconciliation under the Gaussian framework. *Journal of Business & Economic Statistics*, in press.
- Wickramasuriya, S. L., Athanasopoulos, G., & Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526), 804–819. <http://dx.doi.org/10.1080/01621459.2018.1448825>, URL <https://robjhyndman.com/publications/mint/>.
- Wikle, C. K., Berliner, L. M., & Cressie, N. (1998). Hierarchical Bayesian space-time models. *Environmental and Ecological Statistics*, 5(2), 117–154.
- Yao, Y., Rosasco, L., & Andrea, C. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, 26(2), 289–315. <http://dx.doi.org/10.1007/s00365-006-0663-2>.
- Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting. In *Proceedings of the 27th international joint conference on artificial intelligence*. URL <http://arxiv.org/abs/1709.04875>.