Probabilistic Hierarchical Forecasting with Deep Poisson Mixtures

Anonymous Author(s)
Affiliation
Address
email

Abstract

Hierarchical forecasting problems arise when time series compose a group structure that naturally defines aggregation and disaggregation coherence constraints for the predictions. In this work, we explore a new forecast representation, the Poisson Mixture Mesh (PMM), that can produce *probabilistic, coherent* predictions; it is compatible with the neural forecasting innovations, and defines simple aggregation and disaggregation rules capable of accommodating hierarchical structures, unknown during its optimization. We perform an empirical evaluation to compare the PMM to other methods on Australian domestic tourism data.

1 Introduction and Motivation

There are several cases where forecast users need simultaneous forecasts throughout a hierarchy or group structure. The coherence between the predictions is critical for the user's trust. Some examples of hierarchical forecasting problems include retail demand where sales are tracked at the product and geographic levels (Seeger et al., 2017); electricity price forecasting where predictions are needed at different regional levels (Ben Taieb & Koo, 2019; Jeon et al., 2019).

The main goal of hierarchical forecasting is to leverage available information across all the levels of the structure, to produce coherent forecasts. Prior solutions to the challenge depend on two-stage approaches where *base forecasts* are created and later reconciled. More recent work extended the generation of probabilistic hierarchical forecasts into single and simplified models. Some limitations that we find in current solutions are (i) their efficiency, as any new hierarchical structure of interest would require to fit a whole new model, (ii) their accuracy, as the models tend to use restrictive probabilistic models, to enable reconciliation.

In this work we explore a Poisson Mixture Mesh Convolutional Neural Network (`PMMCNN`) solution for the hierarchical forecasting challenge, as we envision that its following characteristics can help to tackle the mentioned challenges:

1. **Probabilistic Coherence**: The Poisson Mixture Mesh (`PMM`) defines a joint distribution across all levels of the time series structures. And by construction, it guarantees probabilistic coherence of its predictions, through its simple aggregation and disaggregation rules.
2. **Enhanced Accuracy**: The `PMM` is a flexible and powerful improvement over available hierarchical methods capable of producing coherent, probabilistic forecasts that rely on strong assumptions on the distributions, like Gaussian noise, that enable its reconciliation strategies. Additionally, the `PMM` representation is independent of the model that supports it and can leverage the rich literature on neural forecasting models.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

3. **Efficiency**: Finally, the PMM can accommodate unrevealed hierarchical structures during its optimization. Its aggregation and disaggregation rules also allow users to easily generate any new marginal distribution across the hierarchy. In contrast, other more complex approaches require the hierarchical structure in advance, and changes to the structure need to fit a whole new specialized model.

The remainder of the paper is structured as follows. Section 2 introduces notation and reviews relevant literature on hierarchical forecasting. Section 3 describes the PMMCNN model. In Section 4 we provide an empirical evaluation on the Australian Tourism hierarchical data, where we showcase the advantages of our method. Finally, in Sections 5 and 6 we discuss future work and conclude.

2 Literature Review

2.1 Hierarchical Forecasting Notation

A hierarchical time series, is a multivariate time series that satisfies linear aggregation constraints. The aggregation is typically depicted with a tree structure, but with group structures beyond strict hierarchies, it may not necessarily be described by a tree (Hyndman et al., 2014; Athanasopoulos et al., 2017; Spiliotis et al., 2020). A hierarchical multivariate time series can be denoted by the vector $\mathbf{y}_{[a,b],t} = [\mathbf{y}_{[a],t}^{\top} \mid \mathbf{y}_{[b],t}^{\top}]^{\top} \in \mathbb{R}^{N}$, where a, b, t stand for the single aggregate, bottom and time indices for the time series $\mathbf{y}$, and $[a], [b], [t]$ listed indices respectively. With the total number of series in the hierarchy $N = N_a + N_b$, where N_a is the number of aggregated series and N_b the number of bottom series, that are at the most disaggregated level possible. The hierarchical aggregation constraints have the following convenient matrix representation:

$$\mathbf{y}_{[a,b],t} = \mathbf{S}\mathbf{y}_{[b],t} \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{y}_{[a],t} \\ \mathbf{y}_{[b],t} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{\text{sum}} \\ \mathbf{I}_{N_b} \end{bmatrix} \mathbf{y}_{[b],t}, \tag{1}$$

The matrix $\mathbf{S} \in \mathbb{R}^{(N_a+N_b)\times N_b}$ aggregates the bottom level series to the series above, it can be decomposed into a summing matrix $\mathbf{S}_{\text{sum}}$ and an identity matrix $\mathbf{I}_{N_b}$. For a simple example, hierarchy described by Figure 1, where each parent node is the sum of its children. In this example the dimensions are $N_a = 3$, $N_b = 4$, and the hierarchical, aggregated and base series are respectively

$$y_{\text{Total},t} = y_{b_1,t} + y_{b_2,t} + y_{b_3,t} + y_{b_4,t}$$
$$\mathbf{y}_{[a],t} = [y_{\text{Total},t},\ y_{b_1,t} + y_{b_2,t},\ y_{b_3,t} + y_{b_4,t}]^{\intercal} \qquad \mathbf{y}_{[b],t} = [y_{b_1,t},\ y_{b_2,t},\ y_{b_3,t},\ y_{b_4,t}]^{\intercal}$$

The summing matrix of the Figure 1 example can be written as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\text{sum}} \\ \hline \mathbf{I}_{N_b} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2 Reconciliation Strategies

Prior solutions to the hierarchical forecasting challenge follow a two-stage process, where first a set of base forecasts $\hat{\mathbf{y}}_{[a,b],t} \in \mathbb{R}^{N_a+N_b}$ is created and then revised into coherent forecasts $\tilde{\mathbf{y}}_{[a,b],t}$ through a reconciliation method. The reconciliation can be compactly expressed by

$$\tilde{\mathbf{y}}_{[a,b],t} = \mathbf{S}\mathbf{P}\hat{\mathbf{y}}_{[a,b],t} \tag{2}$$

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

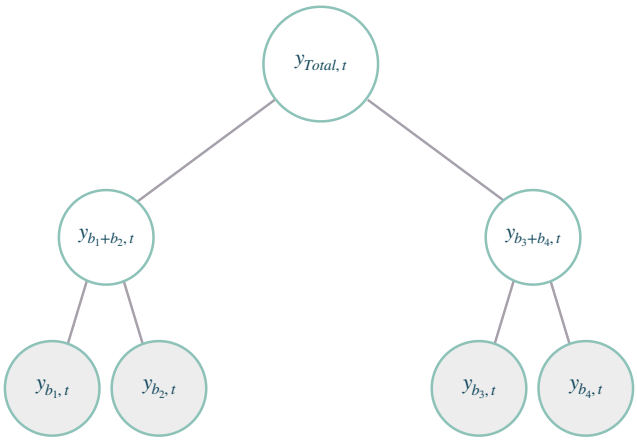

Figure 1: *A simple three level time series hierarchical structure, with four bottom level variables. The disaggregated bottom variables are marked with gray background. In this description each node represents non overlapping series for a single point in time, other indices can be used like locations.*

- Bottom-up: The simple *bottom-up* strategy (`NaiveBU`; Orcutt et al. 1968; Dunn et al. 1976), first generates bottom level forecasts and then aggregates them to produce predictions for all the series in the hierarchical structure. Here $\mathbf{P} = [\mathbf{0}_{N_b \times N_a} \mid \mathbf{I}_{N_b}]$.
- Top-down: The *top-down* strategy (`TD`; Gross & Sohl 1990; Fliedner 1999), distributes the total forecast, and then disaggregates it down the hierarchy using proportions that can be historical or forecasted. In this strategy $\mathbf{P} = [\mathbf{p}_{N_b \times 1} \mid \mathbf{0}_{N_b \times (N_a + N_b - 1)}]$. This method has several variants, depending on the way the proportions $\mathbf{p}$ are created, like the *average of historical proportions*, *proportions of historical averages*, or the *forecasted proportions*.
- Alternative: The more recent *middle-out* strategies (`MO`; Hyndman & Athanasopoulos 2017), treat the second stage reconciliation as an optimization problem for the projection matrix $\mathbf{P}$. These reconciliation techniques include among others the *least squares reconciliation* (Hyndman et al., 2011), the *minimum trace* reconciliation (`MinT`; Wickramasuriya et al. 2019) and the *empirical risk minimization* approach (`ERM`; Ben Taieb & Koo 2019), that relaxes unbiasedness assumptions in `MinT`.

Despite the advancements in post-processing methods, there are still fundamental limitations. First, most post-process reconciliation methods produce point rather than probabilistic forecasts, with some exceptions (Gneiting & Katzfuss, 2014; Taieb et al., 2017; Panagiotelis et al., 2020). Second, the mentioned methods learn the model parameters of the base level forecasts independently, missing the opportunity to share a common model that leverages information across all the time series in the hierarchical structure. Finally, most post-processing reconciliation strategies do not consider the model parameters obtained during the first stage but only their predictions.

2.3 Hierarchical Neural Forecasting

In the last decade, neural network-based forecasting methods have become ubiquitous in large-scale forecasting applications. Neural forecasting has transcended the boundaries of industry into academia, as it has redefined the state-of-the-art in many practical tasks, and forecasting competitions (Benidis et al., 2020; Makridakis et al., 2018, 2020). The latest contributions on neural hierarchical forecasting have incorporated the reconciliation strategies to the models. Methods like the *Simultaneous Hierarchically Reconciled Quantile Regression* (`SHARQ`; Han et al. 2021) and *Hierarchically Regularized Deep Forecasting* (`HIRED`; Paria et al. 2021) augment the training loss function with quadratic approximations to the hierarchical constraints, while the *Hierarchical End-to-End* learning (`HierE2E`; Rangapuram et al. 2021) method imposes exact hierarchical constraints to the

predictions through a projection in a single end-to-end model. None of the architectures and methods mentioned are readily usable for large-scale applications. For instance, the architectures do not use the *forking sequences* optimization technique (Wen et al., 2017), and the `HierE2E` reconciliation projection scales poorly as it needs to receive all the time series of the hierarchy simultaneously.

2.4 Multi-horizon Quantile Recurrent Forecaster

Many scalability and accuracy improvements of sequence modeling have been incorporated into the `MQ-Forecaster` family (Wen et al., 2017). The methods of the family produce probabilistic forecasts using the non parametric quantile regression (Koenker & Bassett, 1978). In spite of major advances in forecast accuracy for large-scale applications, the estimated quantiles are not additive and ensuring the hierarchical coherence of quantile regressors would potentially imply the need to fit a network for the marginal distributions associated to each hierarchical structure of interest, or coherency regularization, which encourages but does not guarantee coherence.

Our primary goal is then to create a probabilistic hierarchical forecasting model that is accurate, and efficient, by extending the proven `MQ-Forecaster` family.

3 Poisson Mixture Mesh Model

3.1 Poisson Mixture Mesh

The foundation of `PMM` model is the assumption that the joint distribution of a multivariate time series $\mathbf{y}_{[b],[t]}$, is described by a Poisson Mixture distribution. The mixture first determines a latent categorical variable K that selects a state, and conditional on the selection an independent Poisson random variable that determines the observations. Using mixing weights w_k and rate $\lambda_{b,k,t}$ parameters the joint likelihood can be parametrized:

$$\begin{aligned} \mathrm{P}\left(\mathbf{y}_{[b],[t]}\right) &= \sum_{k=1}^{N_\lambda} \mathrm{P}(K=k)\mathrm{P}\left(\mathbf{y}_{[b],[t]} \,|\, K=k\right) \\ &= \sum_{k=1}^{N_\lambda} w_k \prod_{(b,t)\in\mathcal{B}} \left(\frac{(\lambda_{b,k,t})^{y_{b,t}} \exp^{-\lambda_{b,k,t}}}{(y_{b,t})!} \right) \end{aligned} \tag{3}$$

We denote such Poisson mixture distributed variable as $\mathbf{y}_{[b],[t]} \sim \mathrm{PM}(\mathbf{w}_{[k]},\, \lambda_{[b],[k],t})$. We describe properties of the Poisson Mixture distribution that make it exceptionally suited for the hierarchical forecasting task below.

3.2 Probabilistic Properties

Under conditional independence assumption and the unmixing property of the parameters $\mathbf{w}_{[k]}$ and $\boldsymbol{\lambda}_{[b],[k],t}$, we show in the Appendix A that the marginal distributions follow the simple aggregation and disaggregation rules:

- Aggregation Rule: Let a multivariate time series $\mathbf{y}_{[b],[t]}$ follow the `PMM` then aggregated levels satisfy
$$\mathbf{y}_{[a],\,t} = \mathbf{S}_{\mathrm{sum}}\mathbf{y}_{[b],t} \sim \mathrm{PM}(\mathbf{w}_{[k]}, \mathbf{S}_{\mathrm{sum}}\boldsymbol{\lambda}_{[b],[k],t}) \tag{4}$$
with $\mathbf{S}_{\mathrm{sum}} \in \mathbb{R}^{N_a \times N_b}$ the hierarchical aggregation matrix.

- Disaggregation Rule: Let the most aggregated level time series $y_{\mathrm{Total},t}$ follow the `PMM`, then l-th level series can be built with the proportions $\mathbf{p}^{(l)}_{[b],t} \in \mathbb{R}^{N_b}$ by
$$\mathbf{y}_{[b],\,t} = \mathbf{p}^{(l)}_{[b],t} y_{\mathrm{Total},t} \sim \mathrm{PM}(\mathbf{w}_{[k]}, \mathbf{p}^{(l)}_{[b],t} \odot \boldsymbol{\lambda}_{\mathrm{Total},[k],t}) \tag{5}$$
with proportions $0 \leq p^{(l)}_{b_i,t}$, $\sum_{b_i} p^{(l)}_{b_i,t} = 1$ and $\odot$ a multiplication over the b-th dimension.

3.3 Poisson Mixture Mesh Convolutional Neural Network

The PMM offers a solution to model the high dimensional regression problem

$$\mathrm{P}(\mathbf{y}_{[b],[t:t+h]} \mid \mathbf{y}_{[b],[:t]}, \mathbf{x}^{(h)}_{[:t]}, \mathbf{x}^{(f)}_{[:t]}, \mathbf{x}^{(s)}) \tag{6}$$

where $\mathbf{y}_{[b],[t:t+h]}, \mathbf{y}_{[b],[:t]}, \mathbf{x}^{(h)}_{[:t]}, \mathbf{x}^{(f)}_{[:t]}, \mathbf{x}^{(s)}$ represent future observations of the multivariate time series, observations of the target up until time t, the past covariates, known future information and static covariates, respectively. It is fully general and compatible with the MQ-Forecaster model family (Wen et al., 2017), for which we only need a few adjustments.

For this work, we combine the PMM with the convolutional encoder version into the PMMCNN. To alleviate the computational burden of optimizing for a full joint distribution from Equation (3), simpler variants need to be devised (Varin et al., 2011). The learning objective is implied by the different versions of the PMM that we design, we describe the variants in detail in Section 3.4.

3.4 Poisson Mixture Mesh Variants

The PMM aggregation and disaggregation rules, described in Section 3.2, allow for applying any of the post-processing reconciliation strategies from Section 2.2. Yet, we can further improve the PMMCNN, by augmenting the model's architecture or optimization with the said strategies.

- Naive Bottom-Up:
  The PMMCNN-NaiveBU model is estimated exclusively on the bottom-level series that are treated as conditionally independent from one another, however for a single time series its time dependencies are still estimated.

  Optimization:

$$\mathrm{P}\left(\mathbf{y}_{[b],[t]}\right) = \prod_{b\in[b]} \mathrm{P}\left(\mathbf{y}_{b,[t]}\right) = \prod_{b\in[b]} (\sum_{k=1}^{N_\lambda} w_k \prod_{t\in[t]} \mathrm{Poisson}\left(y_{b,t}|\lambda_{b,k,t}\right)) \tag{7}$$

- Group Bottom-Up:
  Given that the prediction intervals of the NaiveBU version are excessively wide, we created the GroupBU version of the model to help the PMM learn a learn a better correlation structure of the bottom-level series. The PMMCNN-GroupBU model learns a joint distribution for groups g of bottom-level series and uses a group sampling strategy for the network's SGD optimization. The independence treatment between the groups, maintains the tractability of the likelihood computation.

  Optimization:

$$\mathrm{P}\left(\mathbf{y}_{[b],[t]}\right) = \prod_{g\in[g]} \mathrm{P}\left(\mathbf{y}_{[g],[t]}\right) = \prod_{g\in[g]} (\sum_{k=1}^{N_\lambda} w_k \prod_{(b,t)\in g\times[t]} \mathrm{Poisson}\left(y_{b,t}|\lambda_{b,k,t}\right)) \tag{8}$$

  During the hierarchical reconciliation stage, the PMMCNN's predictions of the upper levels are created using bottom-up strategy and the aggregation rule:

$$\tilde{\mathbf{y}}_{[a,b],t} \sim \mathrm{PM}(\mathbf{w}_{[k]}, \mathbf{S}\hat{\boldsymbol{\lambda}}_{[b],[k],t}) \tag{9}$$

4 Empirical Evaluation

For the empirical evaluation of our proposed methods, we use a publicly available dataset from existing hierarchical forecasting literature. This allows to compare against well-performing benchmarks from the literature and test the `PMMCNN` model, on a real data with natural hierarchical structures.

The dataset we use is the `Tourism-L`, a detailed Australian Tourism dataset with origin in a National Visitor Survey, managed by Tourism Research Australia[1] (Tourism Australia, Canberra, 2019). As described in Table 1, the dataset contains 555 monthly series from 1998 to 2016, it is organized by geography and purpose of travel. The 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 (Wickramasuriya et al., 2019).

Table 1: Australian Tourism flows (`Tourism-L`).

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

4.1 Evaluation Metrics

The evaluation of the model's predictions is based on the *quantile loss* (QL); consider the estimated cumulative distribution function $\hat{\mathbf{F}}_{i,t}$ for an observation $\mathbf{y}_{i,t}$, then the quantile loss is defined by:

$$\mathrm{QL}(\hat{\mathbf{F}}_{i,t}, y_{i,t})_q = 2\left(\mathbb{1}\{y_{i,t} \leq \hat{\mathbf{F}}_{i,t}^{-1}(q)\} - q\right)\left(\hat{\mathbf{F}}_{i,t}^{-1}(q) - y_{i,t}\right) \tag{10}$$

We further 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* (CRPS; Matheson & Winkler 1976). The CRPS measures the predictive power of forecast distributions and has desirable theoretical guarantees (Gneiting & Ranjan, 2011). Following notation from Laio & Tamea 2007, the CRPS[2] is defined as

$$\mathrm{CRPS}(\hat{\mathbf{F}}_{i,t}, y_{i,t}) = \int_0^1 \mathrm{QL}(\hat{\mathbf{F}}_{i,t}, y_{i,t})_q dq \tag{11}$$

4.2 Training Methodology and Hyperparameter Optimization

We set the training window to 202 observations (16 years) preceding the generation of the 12 (1 year) steps ahead forecast. To train the neural network, we minimize the negative log likelihood of the `PMM` variant, using stochastic gradient descent with *Adaptive Moments* (`ADAM`; Kingma & Ba 2014). The network is trained for 3,000 epochs on a single machine with 2 NVIDIA K80 GPUs, and a grouped batch size of 4 regions and 4 purposes each. The `PMMCNN` is implemented using `MXNet` (Tianqi Chen et al., 2015) and the learning rate and random initialization hyperparameters are selected using `HYPEROPT` (Bergstra et al., 2011), a Bayesian optimization library that efficiently explores the hyperparameters using tree-structured Parzen estimators. The hyperparameter search process trains on data from 1998-2014 and evaluates performance on data from 2015. Once the optimal hyperparameter values are determined, we retrain shifting the training window forward one year and predict on data from 2016 to obtain the final CRPS scores.

[1] The `Tourism-L` dataset can be publicly accessed in the `MinT` reconciliation web page.

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

Table 2: Empirical evaluation of probabilistic, hierarchical forecasts on the Australian Tourism Flows (`Tourism-L`). Mean CRPS scores for predictions at each aggregation level, the best result is highlighted. CRPS scores averaged over 8 runs for GroupBU and NaiveBU; scores for Hier-E2E and reconciliations are quoted from `HierE2E` (Rangapuram et al., 2021). Only the best performing among the state-of-the-art reconciliation strategies is shown.

Level	`PMMCNN-GroupBU`	`PMMCNN-NaiveBU`	Hier-E2E	Best Reconciliation
Overall	**0.1260±0.0046**	0.1578±0.0084	0.1520	0.1609 (`ARIMA-MinT-shr`)
1 (geo.)	**0.0411±0.0110**	0.1130±0.0197	0.0810	0.0438 (`ARIMA-MinT-shr`)
2 (geo.)	**0.0624±0.0070**	0.1189±0.0161	0.1030	0.0816 (`ARIMA-MinT-shr`)
3 (geo.)	**0.1122±0.0049**	0.1466±0.0131	0.1361	0.1433 (`ARIMA-MinT-shr`)
4 (geo.)	**0.1571±0.0032**	0.1759±0.0125	0.1752	0.2036 (`ARIMA-MinT-shr`)
5 (prp.)	**0.0747±0.0056**	0.1315±0.0060	0.1027	0.0830 (`ARIMA-MinT-shr`)
6 (prp.)	**0.1100±0.0044**	0.1416±0.0058	0.1403	0.1479 (`ARIMA-MinT-shr`)
7 (prp.)	**0.1901±0.0046**	0.1908±0.0052	0.2050	0.2437 (`ARIMA-MinT-shr`)
8 (prp.)	0.2600±0.0039	**0.2428±0.0045**	0.2727	0.3406 (`ARIMA-MinT-shr`)

4.3 Forecasting Results

We compare against the next forecasting methods: `NaiveBU` that produces univariate bottom-level time series forecasts independently and then sums them according to the hierarchical constraints, a distribution is generated using Gaussian assumptions on the errors. `MinT` (Wickramasuriya et al., 2019) that reconciles unbiased independent forecasts and minimizes the variance of the forecast errors. `ERM` (Ben Taieb & Koo, 2019) that improves on the unbiasedness assumption of the base forecasts in `MinT` and aims to minimize the bias-variance trade-off of the forecast errors. The benchmark methods [3] are implemented or compiled in the `GluonTS` library (Alexandrov et al., 2020). The original implementation of `MinT` is available in the R package `hts` (Hyndman et al., 2020), it automatically selects `ARIMA` and `ETS` univariate base forecast methods (Hyndman & Khandakar, 2008).

We report two variants of the `PMMCNN` model as an ablation experiment to better analyze the source of the forecast accuracy improvements. The first version, the `PMMCNN-NaiveBU`, treats the bottom level series as independent and the `PMMCNN-GroupBU` version that considers groups of time series during its likelihood estimation, both methods obtain probabilistic coherent predictions through their aggregation rule using the bottom-up reconciliation.

Table 2 contains the CRPS scores for the predictions at each aggregation level through the `Tourism-L` hierarchy. The top row reports the overall CRPS score (averaged across all the hierarchy levels). We highlight the best result in **bolds**.

The `PMMCNN-GroupBU` improves the overall CRPS by **20** percent against the second-best alternative. Its predictions show uniform performance gains across all the levels of the hierarchy. Our results confirm previous observations from the community that a shared model, capable of learning from all the time series jointly improves the predictions over those from univariate time series methods. Additionally the comparison [4] between the `NaiveBU` and the `GroupBU` versions of the method show that an expressive joint distribution framework capable of leveraging the hierarchical structure of the data is also beneficial for the forecast accuracy, when compared to the more restrictive Gaussian distributions based methods from the hierarchical forecasting literature.

[3] The implementation of the models and experiments on `HierE2E` applications will be available in the future as an extension to the GluonTS library.

[4] Figure 2 and Figure 3 in the Appendix show a qualitative exploration of the two versions of the model.

5 Future Work

We constructed the PMMCNN extending the MQ-Forecaster architecture. Despite the significant improvements in computational efficiency and accuracy for large-scale applications, the MQ-Forecaster family may not be suited for small datasets. Given the time constraints, we restricted our current experiments to the Tourism-L dataset. We are looking forward to performing more experiments on smaller and larger datasets to continue strengthening this paper's findings.

The current version of the GroupBU model defines the groups by simple geographical distances. However, other exciting approaches could arise when using distances over different dimensions. New ways of describing the groups could create interesting connections with nearest neighbors and clustering literature.

We envision that we can include interactions of time series beyond the predefined groups implied by the GroupBU approach to enhance the ability of the PMM to model the joint likelihood of the hierarchically structured time series while maintaining its estimation tractable. The top-down disaggregation method to learn aggregate level Poisson rates and bottom level proportions could help the model capture higher hierarchical relationships inherited from the structure's upper levels.

6 Conclusion

We have introduced a novel method for probabilistic hierarchical forecasting, the *Poisson Mixture Mesh* (PMM). Our approach is a joint probability distribution that defines simple aggregation and disaggregation rules for its predictions. By construction PMM produces coherent probabilistic forecasts. It provides a flexible and expressive representation capable of highly accurate predictions and is highly efficient since it is scalable and can accommodate hierarchical structures unrevealed during its optimization. Additionally, the method is compatible with innovations in neural forecasting as a minimal adapting process is needed to augment existing deep forecasting architectures. We empirically showcase the advantages of PMM through its application with the PMMCNN model, which improves by **20** percent compared with previous state-of-the-art results on Australian domestic tourism data, a hierarchical dataset from the literature.

References

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Tarkmen, A. C., & Wang, Y. (2020). GluonTS: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, *21*, 1–6. URL: http://jmlr.org/papers/v21/19-820.html.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., & Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, *262*, 60–74.

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

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, B., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., Callot, L., & Januschowski, T. (2020). Neural forecasting: Introduction and literature overview. *Computing Research Repository*, . arXiv:2004.10240.

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* (pp. 2546–2554). Curran Associates, Inc. volume 24. URL: https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Dunn, D. M., Williams, W. H., & Dechaine, T. L. (1976). Aggregate versus subaggregate models in local area forecasting. *Journal of the American Statistical Association*, *71*, 68–71.

Fliedner, G. (1999). An investigation of aggregate variable time series forecast strategies with specific subaggregate time series statistical correlation. *Comput. Oper. Res.*, *26*, 1133–1149. URL: https://doi.org/10.1016/S0305-0548(99)00017-9. doi:10.1016/S0305-0548(99)00017-9.

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, *1*, 125–151. URL: https://ssrn.com/abstract=2405902.

Gneiting, T., & Ranjan, R. (2011). Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, *29*, 411–422.

Gross, C. W., & Sohl, J. E. (1990). Disaggregation methods to expedite product line forecasting. *Journal of Forecasting*, *9*, 233–254. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980090304. doi:10.1002/for.3980090304. arXiv:https://onlinelibrary.wiley.com/doi/pdf/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 The 24th International Conference on Artificial Intelligence and Statistics* (pp. 190–198). PMLR volume 130 of *Proceedings of Machine Learning Research*. URL: http://proceedings.mlr.press/v130/han21a.html.

Hyndman, R., Lee, A., Wang, E., & Wickramasuriya, S. (2020). *hts: Hierarchical and Grouped Time Series*. URL: https://CRAN.R-project.org/package=hts r package version 6.0.1.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., & Shang, H. L. (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, *55*, 2579 – 2589. URL: http://www.sciencedirect.com/science/article/pii/S0167947311000971. doi:https://doi.org/10.1016/j.csda.2011.03.006.

Hyndman, R. J., & Athanasopoulos, G. (2017). Forecasting: Principles and practice. *https://otexts.com/fpp2/*, *987507109*.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, *27*, 1–22. URL: `https://www.jstatsoft.org/v027/i03`. doi:`10.18637/jss.v027.i03`.

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*, 364–379.

Kingma, D. P., & Ba, J. (2014). ADAM: A method for stochastic optimization. URL: `http://arxiv.org/abs/1412.6980` cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations (ICLR), San Diego, 2015.

Koenker, R., & Bassett, G. (1978). Regression quantiles. *Econometrica*, *46*, 33–50. URL: `http://www.jstor.org/stable/1913643`.

Laio, F., & Tamea, S. (2007). Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, *11*, 1267–1277.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*, 802 – 808. URL: `http://www.sciencedirect.com/science/article/pii/S0169207018300785`. doi:`https://doi.org/10.1016/j.ijforecast.2018.06.001`.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M5 accuracy competition: Results, findings and conclusions. *International Journal of Forecasting*, . URL: `https://www.researchgate.net/publication/344487258_The_M5_Accuracy_competition_Results_findings_and_conclusions`.

Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, *22*, 1087–1096. URL: `http://www.jstor.org/stable/2629907`.

Orcutt, G. H., Watts, H. W., & Edwards, J. B. (1968). Data aggregation and information loss. *The American Economic Review*, *58*, 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`.

Paria, B., Sen, R., Ahmed, A., & Das, A. (2021). Hierarchically regularized deep forecasting. `arXiv:2106.07630`.

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 the 38th International Conference on Machine Learning* Proceedings of Machine Learning Research. PMLR.

Seeger, M., Rangapuram, S., Wang, Y., Salinas, D., Gasthaus, J., Januschowski, T., & Flunkert, V. (2017). Approximate bayesian inference in linear state space models for intermittent demand forecasting at scale.

Spiliotis, E., Petropoulos, F., Kourentzes, N., & Assimakopoulos, V. (2020). Cross-temporal aggregation: Improving the forecast accuracy of hierarchical electricity consumption. *Applied Energy*, *261*, 114339.

Taieb, S. B., Taylor, J. W., & Hyndman, R. J. (2017). Coherent probabilistic forecasts for hierarchical time series. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (pp. 3348–3357). PMLR volume 70 of *Proceedings of Machine Learning Research*. URL: `http://proceedings.mlr.press/v70/taieb17a.html`.

Tianqi Chen et al. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, *abs/1512.01274*. URL: `http://arxiv.org/abs/1512.01274`.

Tourism Australia, Canberra (2019). Detailed tourism Research Australia (2005), Travel by Australians. Accessed at https://robjhyndman.com/publications/hierarchical-tourism/.

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, *21*, 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*. URL: `https://arxiv.org/abs/1711.11053`. `arXiv:1711.11053`.

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*, 804–819. URL: `https://robjhyndman.com/publications/mint/`. doi:`10.1080/01621459.2018.1448825`.

A Appendix

Poisson Mixture Mesh Properties

To describe the joint distribution across all levels of the hierarchical time series structure we assume that for a given time series $\mathbf{y}_{b,[t]}$, its non overlapping observations are distributed Poisson mixture, denoted $y_{b,\,t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t})$, the observations are conditionally independent given the Poisson rates, the mixing weights $\mathbf{w}_{[k]}$ are shared between the observations, and their respective Poisson components do not mix[5], under these assumptions we show in A.1, A.2 and A.4 the properties are summarized in Table A1:

A.1 Unmixing Property

To describe the joint distribution across all levels of the time series structure we assume that for a given time series $\mathbf{y}_{b,[t]}$ with non overlapping observations $y_{b,\,t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t})$, the observations are conditionally independent once the Poisson rates are known, the mixing weights $\mathbf{w}$ are shared between the observations, and their respective Poisson components do not mix. We prove the properties from Table A1.

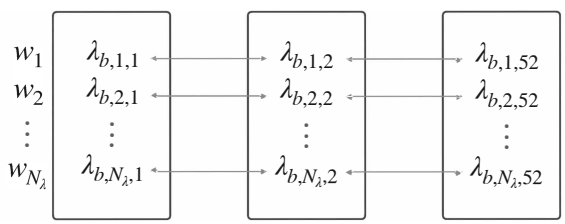

Figure 1: Representation of the un-mixing property of the Poisson components of the PMM.

A.2 Joint Distribution

Let two bottom random variables have a PM distribution with mixing weights $\mathbf{w}_{[k]}$ and rates $\boldsymbol{\lambda}_{b,[k],t}, \boldsymbol{\lambda}_{b,[k],t'}$, that is $y_{b,t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t})$ and $y_{b,t'} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t'})$, then under the conditional independence assumption, the independence between the weights and the bottom series and the deterministic evolution property the joint distribution of the variables is:

$$P(y_{b,t}, y_{b,t'}) = \sum_k w_k \text{Poisson}(y_{b,t} |\ \lambda_{b,k,t}) \text{Poisson}(y_{b,t'} |\ \lambda_{b,k,t'})$$

Proof

$$\begin{aligned}
&P(y_{b,t}, y_{b,t'}) \\
&= \sum_k \sum_l P(y_{b,t},\ y_{b,t'} |\ \lambda_{b,k,t}, \lambda_{b,l,t'}) P(\lambda_{b,k,t} | \lambda_{b,l,t'}) P(\lambda_{b,l,t'}) \\
&= \sum_k P(y_{b,t} |\ \lambda_{b,k,t}) P(y_{b,t'} |\ \lambda_{b,k,t'}) P(\lambda_{b,k,t'}) \\
&= \sum_k w_k \text{Poisson}(y_{b,t} |\ \lambda_{b,k,t}) \text{Poisson}(y_{b,t'} |\ \lambda_{b,k,t'})
\end{aligned}$$

[5]The un-mixing property of the Poisson components refers to the assumption that the count populations of a given Poisson component are deterministic.

The first equality is the chain rule of probability, the second equality comes from the deterministic evolution property, the final part comes from the conditional independence property. By induction,

$$P(\mathbf{y}_{b,[t]}) = \sum_{k=1}^{N_\lambda} w_k \prod_{t=1}^{N_T} \text{Poisson}(y_{b,\,t}|\ \lambda_{b,k,t})$$

A.3 Covariance Structure

From the joint distribution for $y_{b,t},\ y_{b,t'}$ it can be shown using linearity of expectation and the Poisson distribution properties that the covariance structure of the bottom series can be described by the covariance structure of the Poisson rates:

$$Cov(y_{b,t},\ y_{b,t'}) = \sum_{k=1}^{N_\lambda} w_k(\lambda_{b,k,t} - \bar{\boldsymbol{\lambda}}_{b,[k],t})(\lambda_{b,k,t'} - \bar{\boldsymbol{\lambda}}_{b,[k],t'})$$

where $\bar{\boldsymbol{\lambda}}_{b,[k],t} = \sum_k w_k \lambda_{b,k,t}$ and $\bar{\boldsymbol{\lambda}}_{b,[k],t'} = \sum_k w_k \lambda_{b,k,t'}$. This shows that in spite of the strong assumptions made for the PMM the model is capable of describing a rich correlation structure from the dispersion inherited of the mixture.

A.4 Aggregation Rule

Let two bottom level random variables have a PM distribution with mixing weights $\mathbf{w}_{[k]}$ and rates $\boldsymbol{\lambda}_{b,[k],t},\ \boldsymbol{\lambda}_{b',[k],t}$, that is $y_{b,t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t})$ and $y_{b',t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b',[k],t})$, then under the conditional independence assumption, the independence between the weights and the bottom series and the deterministic evolution property we have the following aggregation rule:

$$\mathbf{y}_{b,t} + \mathbf{y}_{b',t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t} + \boldsymbol{\lambda}_{b',[k],t})$$

Proof

$$\begin{aligned}
&P(y_{b,t} + y_{b',t}) \\
&\quad = \sum_k \sum_l P(y_{b,t} + y_{b',t}|\ \lambda_{b,k,t}, \lambda_{b',l,t}) P(\lambda_{b,k,t}|\lambda_{b',l,t}) P(\lambda_{b',l,t}) \\
&\quad = \sum_k P(y_{b,t} + y_{b',t}|\ \lambda_{b,k,t},\ \lambda_{b',k,t}) P(\lambda_{b',k,t}) \\
&\quad = \sum_k w_k \text{Poisson}(y_{b,t} + y_{b',t}|\ \lambda_{b,k,t} + \lambda_{b',k,t})
\end{aligned}$$

Table A1: Summary properties of the Mixed Poisson State-Space Model

1. Marginal distribution	$y_{b,t} \sim PM(\mathbf{w}_{[k]}, \boldsymbol{\lambda}_{b,[k],t})$
2. Assumptions	
2.1 Conditional Independence	$y_{b_1,t}\vert\lambda_{b_1,k,t} \perp\!\!\!\perp y_{b_2,t}\vert\lambda_{b_2,h,t}$ if b_1, b_2 do not overlap
2.2 Unmixing property	$P(\lambda_{b,k,t}\vert\lambda_{b,h,t}) = \delta_{k,h}$ Poisson components do not mix where $\delta_{k,h}$ representing the Kronecker delta
3. Joint Distribution	$P(\mathbf{y}_{b,[t]}) = \sum_{k=1}^{N_\lambda} w_k \prod_{t=1}^{N_T} \text{Poisson}(y_{b,t}\vert\ \lambda_{b,k,t})$
4. Hierarchical Coherence	
4.1 Aggregation Rule	$\mathbf{y}_{[a],t} = \mathbf{S}_{\text{sum}}\mathbf{y}_{[b],t} \sim PM(\mathbf{w}_{[k]}, \mathbf{S}_{\text{sum}}\boldsymbol{\lambda}_{[b],[k],t})$
4.2 Disaggregation Rule	$\mathbf{y}_{[b],t} = \mathbf{p}_{[b],t}y_{\text{Total},t} \sim PM(\mathbf{w}_{[k]}, \mathbf{p}_{[b],t} \odot \boldsymbol{\lambda}_{\text{Total},[k],t})$) with shares $0 \leq p_{b_i}$, $\sum_{b_i} p_{b_i,t} = \mathbb{1}$ and $\odot$ multiplication in the b-th dimension
5. Covariance Structure	$\text{Cov}(y_{b,t},\ y_{b,t'}) = \sum_{k=1}^{N_\lambda} w_k(\lambda_{b,k,t} - \bar{\boldsymbol{\lambda}}_{b,[k],t})(\lambda_{b,k,t'} - \bar{\boldsymbol{\lambda}}_{b,[k],t'})$

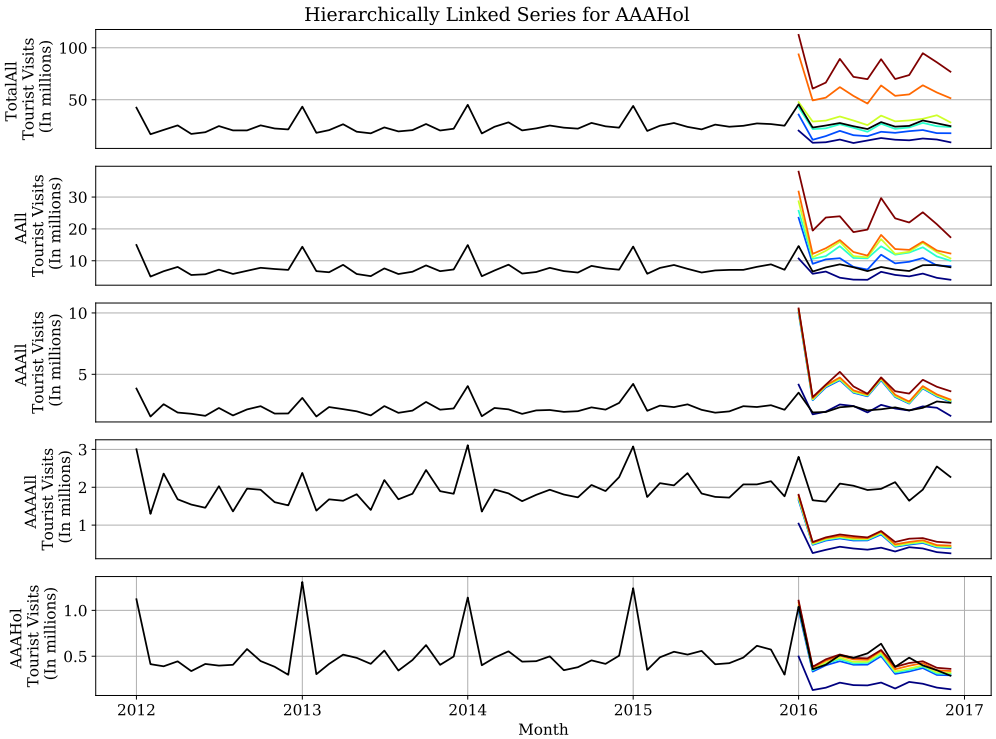

Figure 2: Example of a hierarchically linked time series from the `Tourism-L` dataset. The top row shows the total number of tourist visits in Australia (TotalAll), the second row shows the visits to Australia for the North South Wales state (AAll), the third row shows the holiday visits in the metropolitan Area of New South Wales (AAAll), the fourth row shows the holiday visits in SIDNEY (AAAAll), the final row shows the holiday visits to Sidney. Quantile predictions are shown in colored lines. The `PMMCNN-NaiveBU` performs well in disaggregated series and means, but makes the predictions of the aggregated series unnecessarily wide.

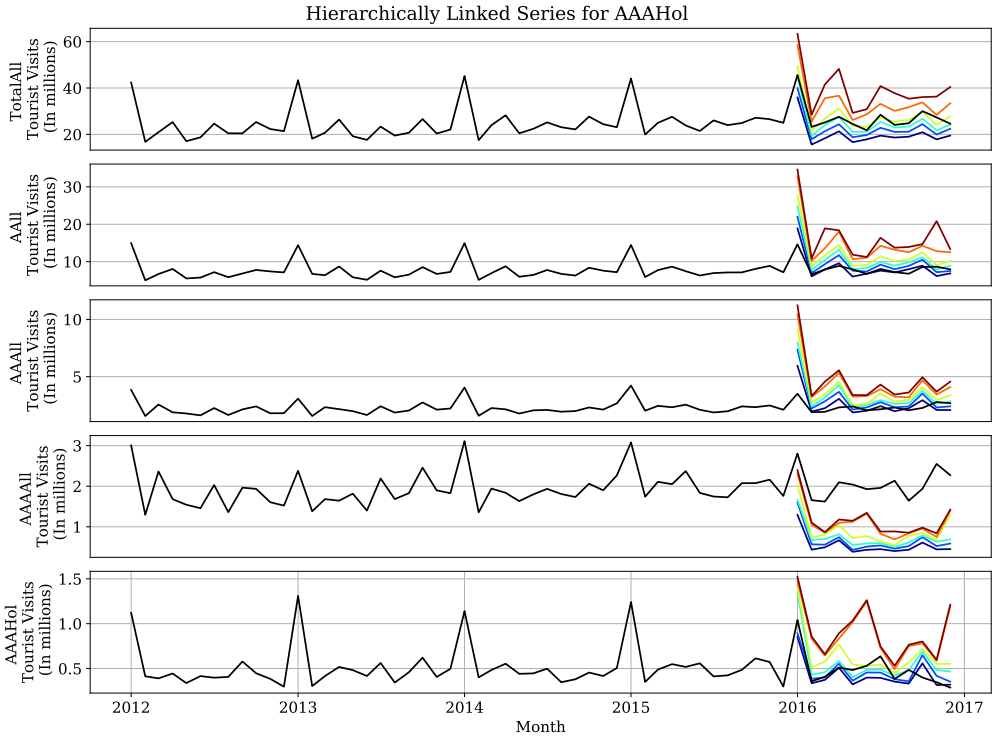

Figure 3: Example of a hierarchically linked time series from the `Tourism-L` dataset. The top row shows the total number of tourist visits in Australia (TotalAll), the second row shows the visits to Australia for the North South Wales state (AAll), the third row shows the holiday visits in the metropolitan Area of New South Wales (AAAll), the fourth row shows the holiday visits in SIDNEY (AAAAll), the final row shows the holiday visits to Sidney. Quantile predictions are shown in colored lines. The `PMMCNN-GroupBU` performs well in disaggregated series and aggregate series.

B Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract, and model characteristics listed in Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Future Work.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Per company policy, we are not able to share the code used to obtain the main experimental results; however, all datasets used are publicly available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.3.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] The data come from the National Visitor Survey managed by Tourism Research Australia, made publicly available by the Australian government. It is publicly accessible at https://robjhyndman.com/publications/mint/
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See (b).
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data we are using does not contain personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]