End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series

Syama Sundar Rangapuram [1] **Lucien D. Werner** [2 3] **Konstantinos Benidis** [1] **Pedro Mercado** [1] **Jan Gasthaus** [1] **Tim Januschowski** [1]

Abstract

This paper presents a novel approach to forecasting of hierarchical time series that produces *coherent, probabilistic* forecasts without requiring any explicit post-processing step. Unlike the state-of-the-art, the proposed method simultaneously learns from all time series in the hierarchy and incorporates the reconciliation step as part of a single trainable model. This is achieved by applying the reparameterization trick and utilizing the observation that reconciliation can be cast as an optimization problem with a closed-form solution. These model features make end-to-end learning of hierarchical forecasts possible, while accomplishing the challenging task of generating forecasts that are both probabilistic and coherent. Importantly, our approach also accommodates general aggregation constraints including grouped, temporal, and cross-temporal hierarchies. An extensive empirical evaluation on real-world hierarchical datasets demonstrates the advantages of the proposed approach over the state-of-the-art.

1. Introduction

In many practically important applications, multivariate time series have a natural hierarchical structure, where time series at upper levels of hierarchy are aggregates of those at lower levels (Petropoulos et al., 2020). Prominent examples include retail sales, where sales are tracked at product, store, state and country levels (Seeger et al., 2016; 2017), and electricity forecasting, where consumption/production quantities are desired at the individual, grid, and regional levels (Taieb et al., 2020; Jeon et al., 2019). Although series at the bottom of the hierarchy are typically sparse, noisy and devoid of the high level patterns that are apparent in aggregate (Ben Taieb et al., 2017), forecasts have value at all levels: bottom-level forecasts may be of more interest for automated decision making on operational horizons whereas forecasts at the top level enable strategic decision making (Januschowski & Kolassa, 2019). However, generating forecasts independently for time series at each level does not guarantee that the forecasts are *coherent*, i.e., forecasts of aggregated time series are the sum of forecasts of the corresponding disaggregated time series.

Thus, the main challenge of forecasting hierarchical time series is to exploit the information available across all levels of a given hierarchy while producing coherent forecasts. Prior work in hierarchical forecasting follows a two-stage approach: *base forecasts* are first obtained independently for each time series in the hierarchy and are then combined and revised in a post-processing step to ensure coherence. Two main issues arise with such a two-stage procedure: (i) the model parameters for each time series are learned independently, thereby discarding information, and (ii) the base forecasts are revised without any regard to the learned model parameters. Another fundamental limitation of most existing methods is that they can only produce point (rather than probabilistic) forecasts. Probabilistic forecasts are required in practice for better decision making and risk management (Berrocal et al., 2010). The notable exception is (Ben Taieb et al., 2017), although it is still a two-step procedure.

In this work, we present a novel approach to probabilistic forecasting of hierarchical time series that incorporates both learning and reconciliation into a single end-to-end model. Model parameters are learned simultaneously from all time series in the hierarchy. The probabilistic forecasts from the model are guaranteed to be coherent without requiring any post-processing step. The key insights behind the proposed method are the differentiability of the sampling operation, thanks to the reparametrization trick (Kingma & Welling, 2013), and the implementation of the reconciliation step on samples as a convex optimization problem. This allows one to combine typically independent components (generation of base forecasts, sampling and reconciliation) into a single *trainable* model.

[1]AWS AI Labs, Germany. [2]Department of Computing & Mathematical Sciences, California Institute of Technology, Pasadena, California, USA. [3]Work done while at Amazon. Correspondence to: Syama S. Rangapuram <rangapur@amazon.de>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

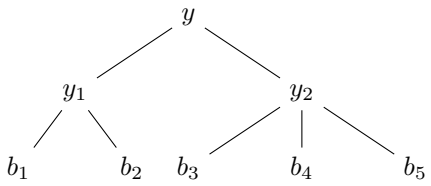

Figure 1. Example of hierarchical time series structure for $n = 8$ time series with $m = 5$ bottom and $r = 3$ aggregated time series.

While our approach is fully general and can be used with any multivariate model, we show its empirical effectiveness via a specific multivariate, nonlinear autoregressive model, `DeepVAR` (Salinas et al., 2019), which exploits information across all time series in the hierarchy to improve forecast accuracy.[1] Moreover, since our model generates coherent samples directly, it can be trained not only with the log-likelihood loss but with any loss function that is of practical interest. Our approach accommodates to incorporate complicated structural constraints on the forecasts via the use of differentiable convex optimization layers (Agrawal et al., 2019a). For hierarchical reconciliation as we consider it here, this reduces to a simple closed-form solution thus facilitating its inclusion as the last step of the trainable model.

In what follows, we provide the necessary background of the hierarchical forecasting problem and review the state-of-the art literature (Section 2). We then present our model in Section 3 and describe the training procedure. We provide a thorough empirical evaluation on several real-world datasets in Section 4 and conclude in Section 5.

2. Background and Related Work

A hierarchical time series is a multivariate time series that satisfies linear aggregation constraints. Such aggregation constraints typically encode a tree hierarchy (see Figure 1) but need not necessarily. For example, grouped (Hyndman et al., 2016), temporal (Athanasopoulos et al., 2017), and cross-temporal aggregations (Spiliotis et al., 2020) can also be expressed with linear constraints.

2.1. Preliminaries

Consider a time horizon $t = 1, \dots, T$. Let $\mathbf{y}_t \in \mathbb{R}^n$ denote the values of a hierarchical time series at time t, with $y_{t,i} \in \mathbb{R}$ the value of the i-th (out of n) univariate time series. Here we assume that the index i of the individual time series is given by the level-order traversal of the hierarchical tree going from left to right at each level. Further, let $\mathbf{x}_{t,i} \in \mathbb{R}^k$ be time varying covariate vectors associated to each univariate time series at time t, and $\mathbf{x}_t := [\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,n}] \in \mathbb{R}^{k \times n}$. We use the shorthand $\mathbf{y}_{1:T}$ to denote the sequence $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T\}$.

We refer to the time series at the leaf nodes of the hierarchy as *bottom-level* series and those of the remaining nodes as *aggregated* series. We also call a given set of forecasts for *all* time series in the hierarchy that are generated without heeding the aggregation constraint as *base* forecasts (not to be confused with bottom-level). For notational convenience we split the vector of all series $\mathbf{y}_t$ into m bottom entries and r aggregated entries such that $\mathbf{y}_t = [\mathbf{a}_t \ \mathbf{b}_t]^\top$ with $\mathbf{a}_t \in \mathbb{R}^r$ and $\mathbf{b}_t \in \mathbb{R}^m$. Clearly $n = r + m$. For an individual hierarchy or grouping, an aggregation matrix $S \in \{0, 1\}^{n \times m}$ is defined and the $\mathbf{y}_t, \mathbf{b}_t$, and S satisfy

$$\mathbf{y}_t = S\mathbf{b}_t \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{a}_t \\ \mathbf{b}_t \end{bmatrix} = \begin{bmatrix} S_{\text{sum}} \\ I_m \end{bmatrix} \mathbf{b}_t, \tag{1}$$

for every t. $S_{\text{sum}} \in \{0, 1\}^{r \times m}$ is a summation matrix and I_m is the $m \times m$ identity matrix. We also find it useful to equivalently represent (1) as

$$A\mathbf{y}_t = \mathbf{0}, \tag{2}$$

where $A := [I_r \mid -S_{\text{sum}}] \in \{0, 1\}^{r \times n}$, $\mathbf{0}$ is an r-vector of zeros, and I_r is the $r \times r$ identity. Formulation (2) allows for a natural definition of forecast error (see below).

We illustrate our notation with the example in Figure 1. For this hierarchy, $\mathbf{a}_t = [y, y_1, y_2]_t^\top \in \mathbb{R}^3$ and $\mathbf{b}_t = [b_1, b_2, b_3, b_4, b_5]_t^\top \in \mathbb{R}^5$. The aggregation matrix S is

$$S = \begin{bmatrix} S_{\text{sum}} \\ I_5 \end{bmatrix} = \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ \hline & & & & \\ & & I_5 & & \\ & & & & \end{array}\right].$$

In hierarchical time series forecasting, one is typically interested in producing forecasts for all the time series in the hierarchy for a given number τ of future time steps after the present time T. Here τ is the length of the prediction or forecast horizon. The forecasts are either point predictions or probabilistic in nature, in which case they can be represented as a set of Monte Carlo samples drawn from the forecast distribution. For $h \leq \tau$ we denote an h-period-ahead forecast sample by $\hat{\mathbf{y}}_{T+h}$, with the entire set of samples that comprises the probabilistic forecast written as$\{\hat{\mathbf{y}}_{T+h}\}$.

Clearly, an important aspect of hierarchical forecasting is the requirement that the forecasts generated respect the aggregation constraint, which can be formalized as follows.

Definition 2.1. Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a linear subspace defined as

$$\mathcal{S} := \{\mathbf{y} | \mathbf{y} \in \text{null}(A)\}.$$

[1] In fact, we show in experiments that even without reconciliation `DeepVAR` model outperforms classical hierarchical methods.

A point forecast $\hat{\mathbf{y}}_{T+h}$ is said to be *coherent* iff $\hat{\mathbf{y}}_{T+h} \in \mathcal{S}$. The coherency error of $\hat{\mathbf{y}}_{T+h}$ is defined as $\hat{\mathbf{r}}_{T+h} = A\hat{\mathbf{y}}_{T+h}$. Similarly, a probabilistic forecast represented as samples $\{\hat{\mathbf{y}}_{T+h}\}$ is coherent iff each of its samples is. Past observations $y_t, t = 1, 2, \ldots, T$ are coherent by construction.

Recent work Panagiotelis et al. (2020) proposes an alternate definition of coherence directly on probability densities.

2.2. Related Work

Existing approaches to hierarchical forecasting methods mainly consider point forecasts with the notable exception (Ben Taieb et al., 2017) tackling probabilistic forecasts.[2]. In this section, we review several of the state-of-the-art methods.

Mean Forecast Combination & Reconciliation. Approaches towards forecasting the means of hierarchical time series follow a two-step procedure: (i) forecast each time series independently to obtain base forecasts $\hat{\mathbf{y}}_{T+h}$ and (ii) produce revised forecasts $\tilde{\mathbf{y}}_{T+h}$ through reconciliation. Given base forecasts $\hat{\mathbf{y}}_{T+h}$, the methods in (Hyndman et al., 2011; Wickramasuriya et al., 2019) obtain reconciled forecasts

$$\tilde{\mathbf{y}}_{T+h} = SP\hat{\mathbf{y}}_{T+h}, \tag{3}$$

where S is the aggregation matrix and $P \in \mathbb{R}^{m\times n}$ is a matrix that depends on the choice of the hierarchical forecasting approach. When the method is bottom-up (BU), $P = [\mathbf{0}_{m\times r}|\mathbf{1}_{m\times m}]$. For top-down (TD), $P = [\mathbf{p}_{m\times 1}|\mathbf{0}_{m\times n-1}]$ where $\mathbf{p}$ is an m-vector summing to 1 that disaggregates the top-level series proportionally to the bottom level series. Middle-out (MO) can be analogously defined (Hyndman & Athanasopoulos, 2017). Connections to ensembling are presented in (Hollyman et al., 2021).

The MinT method (Wickramasuriya et al., 2019) proposes reconciled forecasts using $P = \left(S^\top W_h^{-1} S\right)^{-1}\left(S^\top W_h^{-1}\right)$, where W_h is the covariance matrix of the h-period-ahead forecast errors $\hat{\varepsilon}_{T+h} = \mathbf{y}_{T+h} - \hat{\mathbf{y}}_{T+h}$. It is shown that when the $\hat{\mathbf{y}}_{T+h}$ are unbiased, this choice of P minimizes the sum of variances of the forecast errors.

The advantages of the MinT approach are that its revised forecasts are coherent by construction and the reconciliation approach incorporates information from all levels of hierarchy simultaneously. Disadvantages are the strong assumption of base forecasts to be unbiased and that the error covariance W_h is hard to obtain for general h.

The unbiasedness assumption in MinT is relaxed in (Ben Taieb & Koo, 2019). Rather than computing the minimum-variance revised forecasts, the authors seek to find the optimal bias-variance trade-off by solving an empirical risk minimization (ERM) problem. This method also generates base forecasts, followed by reconciliation.

Van Erven & Cugliari (2015) follow the two-stage scheme too. Their reconciliation approach is a weighted projection of the base forecasts onto the coherent subspace $\mathcal{S}$:

$$\begin{aligned}\tilde{\mathbf{y}}_{t+h} = \arg\min_{\mathbf{x}} \quad & ||\mathbf{Q}(\hat{\mathbf{y}}_{t+h} - \mathbf{x})||_2^2 \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{S} \cap \mathcal{B}.\end{aligned}$$

$\mathbf{Q}$ is a diagonal weight matrix that encodes knowledge/belief about the relative magnitudes of the base forecast errors for each series in the hierarchy and $\mathcal{B}$ is a set of additional constraints to be imposed on the reconciled forecasts.

Probabilistic Methods. In contrast to the methods in the previous section, Ben Taieb et al. (2017) consider forecasting probability distributions (Gneiting & Katzfuss, 2014) rather than just means (i.e., point forecasts). In particular they estimate the conditional predictive CDF for each series i in the hierarchy:

$$F_{i,T+h}(y_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \mathbb{P}(y_{i,T+h} \leq y_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_T).$$

Ben Taieb et al. (2017) start by generating independent forecasts of the conditional marginal distributions (e.g., mean and variance from MinT). They obtain probabilistic forecasts of the aggregate series by sampling from the bottom-level marginals of their children and re-ordering the samples to match an empirical copula generated from the forecast errors. Thus far, this method is a probabilistic "bottom-up" approach and emits coherent samples by construction. To share information between the levels, a combination step is performed on the means of the learned marginal distributions and the bottom-up samples are adjusted accordingly.

To the best of our knowledge, none of the existing approaches to probabilistic forecasting take an end-to-end view. This introduces an opportunity to handle the trade-off between forecast accuracy and coherence better through a single, joint model where reconciliation is performed alongside forecast learning. The flexibility of our framework means that we can take advantage of the increasingly rich literature on neural forecasting models, e.g., Benidis et al. (2020) provide an overview. Recent forecasting competitions have shown them to be highly effective (Makridakis et al., 2018; Bojer & Meldgaard, 2020; Makridakis et al., to appear). Of particular relevance are multivariate, probabilistic, forecasting models (Rasul et al., 2021; de Bézenac et al., 2020; Salinas et al., 2019). These estimate the dependency structure in the time series panels explicitly and can naturally be incorporated into our approach.

[2]This is despite the general recognition of the practical importance of probabilistic forecasting for downstream applications (e.g., (Böse et al., 2017; Faloutsos et al., 2019)).

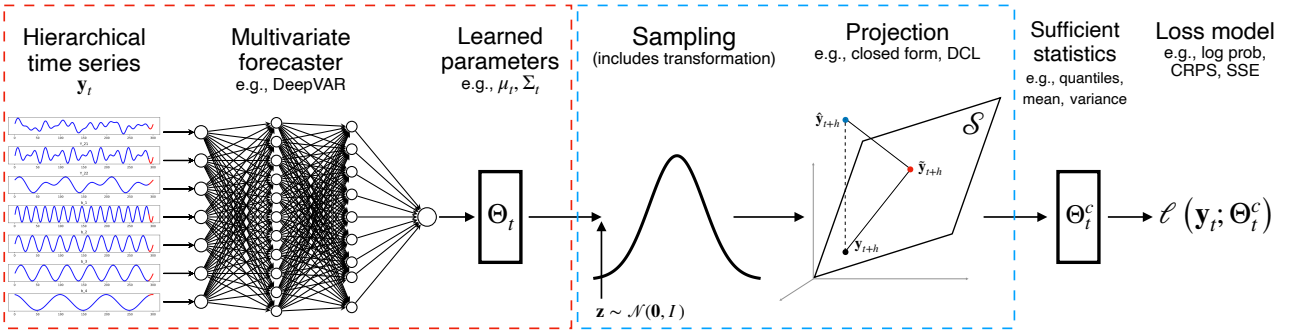

Figure 2. Model architecture. Hierarchical time series data is used to train a multivariate forecaster. Learned distribution parameters along with the reparameterization trick allow this distribution to be sampled during training. Optionally, a nonlinear transformation of the samples (e.g., normalizing flow) can account for data in a non-Gaussian domain. Samples are then projected to enforce coherency. From the empirical distribution represented by the samples, sufficient statistics Θ_t^c can be computed and used to define an appropriate loss.

3. End-to-End Hierarchical Forecasting

Two primary components comprise our approach: (i) a forecasting model that produces a multivariate forecast distribution over the prediction horizon; and (ii) a sampling & projection step where samples are drawn from the forecast distribution, and are then projected onto the coherent subspace. Figure 2 illustrates the architecture of the model.

It is important to note that when both of the above components are amenable to auto-differentiation, they constitute a single global model whose parameters are learned end-to-end by minimizing a loss on the coherent samples directly. In particular, the sampling step can be differentiated using the reparametrization trick (Kingma & Welling, 2013) and the projection step, which is an optimization problem, can be formed as a differentiable convex optimization layer (DCL) (Amos & Kolter, 2017; Agrawal et al., 2019a;b). In the setting of hierarchical and grouped time series, the optimization problem has a closed-form solution requiring only a matrix-vector multiplication (with a pre-computable matrix) and hence is trivially differentiable. However, the proposed approach can handle more sophisticated constraints than those imposed by hierarchical setting via DCL.

3.1. Model

We now describe the instantiation of our model that is explored in this work. We use DeepVAR (Salinas et al., 2019) as the base multivariate forecaster because of its simplicity and performance. A schematic of the full hierarchical model is shown in Figure 3. The red dashed line represents the DeepVAR model (described below) and the blue dashed line highlights the sampling and projection steps. Once trained, the model produces coherent forecasts by construction.

3.1.1. DeepVAR

DeepVAR is a multivariate, nonlinear generalization of classical autoregressive models (Salinas et al., 2019; 2020; Alexandrov et al., 2019).[3] It uses a recurrent neural network (RNN) to exploit relationships across the entire history of the multivariate time series and is trained to learn parameters of the forecast distribution. More precisely, given a feature vector $\mathbf{x}_t$ and the multivariate lags $\mathbf{y}_{t-1} \in \mathbb{R}^n$ as inputs, DeepVAR assumes the predictive distribution at time step t is parameterized by Θ_t, which are the outputs of the RNN:

$$\Theta_t = \Psi(\mathbf{x}_t, \mathbf{y}_{t-1}, \mathbf{h}_{t-1}; \Phi). \tag{4}$$

Here Ψ is a recurrent function of the RNN whose global shared parameters are given by Φ and hidden state by $\mathbf{h}_{t-1}$. Typically, DeepVAR assumes that the forecast distribution is Gaussian in which case $\Theta_t = \{\mu_t, \Sigma_t\}$, where $\mu_t \in \mathbb{R}^n$ and $\Sigma_t \in \mathbb{S}_+^n$, although it can be extended to handle other distributions. The unknown parameters Φ are then learned by the maximum likelihood principle given the training data. Note that for simplicity we specify only one lag $\mathbf{y}_{t-1}$ as the input to the recurrent function but in the implementation lags are chosen from a lag set determined by the frequency of the time series (Alexandrov et al., 2019).

In the hierarchical setting, the covariance matrix Σ_t captures the correlations imposed by the hierarchy as well as the relationships among the bottom-level time series. In our experience with industrial applications, we often find that the bottom-level time series are too sparse to learn any covariance structure let alone more complicated nonlinear relationships between them. Given this, we propose to learn a diagonal covariance matrix Σ_t when producing the initial base forecasts; if more flexibility is needed to capture the nonlinear relationships one could transform base forecasts using normalizing flows (see Section 3.1.2). Note that the linear relationships between the aggregated and bottom-

[3] Here we use the version without the Copula transformation.

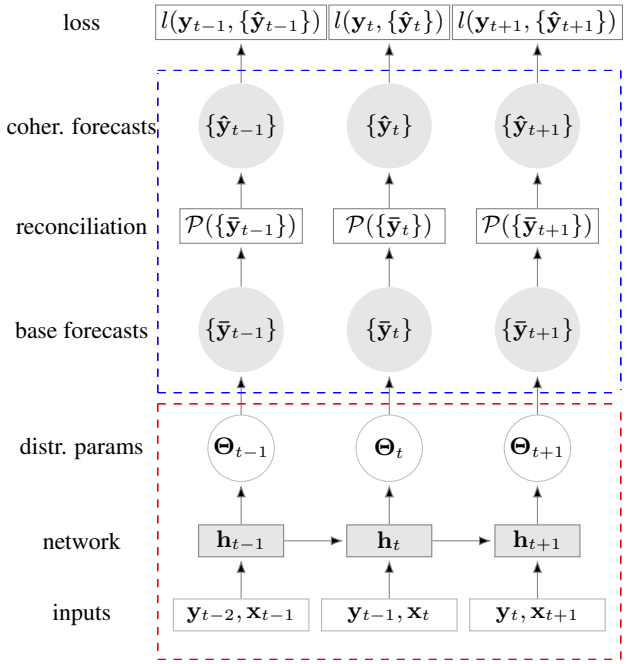

Figure 3. Specific instantiation of our approach with `DeepVAR` (Salinas et al., 2019) multivariate forecasting model (red boundary). Sampling and projection steps are highlighted by the blue boundary.

level time series are enforced via projection.

Although we assume Σ_t is diagonal, this is not equivalent to learning independent models for each of the n time series in the hierarchy. In fact, the mean $\mu_{t,i}$ and the variance $\Sigma_{t,(i,i)}$ of the forecast distribution for each time series are predicted by combining the lags of all time series $\mathbf{y}_{t-1}$ and features $\mathbf{x}_t$ in a nonlinear way using shared parameters Φ. In our experiments, we notice that this global learning already produces much better results than the hierarchical forecasting methods that do explicit reconciliation of the forecasts produced independently by univariate models.

3.1.2. Sampling and Projection

Next we describe how to generate coherent forecasts given distribution parameters $\Theta_t = \{\mu_t, \Sigma_t\}$ from the RNN. To this end, we first generate a set of N Monte Carlo samples from the predicted distribution, $\{\bar{\mathbf{y}}_t \in \mathbb{R}^n\} \sim \mathcal{N}(\mu_t, \Sigma_t)$. Note that this sampling step is differentiable with a simple reparameterization of $\mathcal{N}(\mu_t, \Sigma_t)$:

$$\bar{\mathbf{y}}_t = \mu_t + \Sigma_t^{1/2}\mathbf{z},$$

with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$. That is, given the samples from the standard multivariate normal distribution, which are independent of the network parameters, the actual forecast samples are deterministic functions of μ_t and Σ_t.

Optionally, in order to capture the nonlinear relationships among the bottom-level time series, the generated samples can be transformed using a learnable nonlinear transformation. In this case, "forecasts" from the base model do not directly correspond to un-reconciled forecasts for the base series, but rather represent predictions of an unobserved latent state. This is similar to the standard technique of handling non-Gaussian, nonlinear data by transforming it via normalizing flows into a Gaussian space where tractable methods can be applied. The main difference in our case is that the nonlinear transformation need not be invertible since our loss is computed on the samples directly (see Section 3.2).

Finally, we enforce coherence on the (transformed) samples $\{\bar{\mathbf{y}}_t\}$ obtained from the forecast distribution by solving the following optimization problem:

$$\begin{aligned} \hat{\mathbf{y}}_t = \underset{\mathbf{y} \in \mathbb{R}^n}{\arg\min} \quad & ||\mathbf{y} - \bar{\mathbf{y}}_t||_2 \\ \text{s.t.} \quad & A\mathbf{y} = 0. \end{aligned} \tag{5}$$

Note that this is essentially projection onto the null space of A which can be computed with a closed-form projection operator:

$$M := I - A^\top (AA^\top)^{-1} A. \tag{6}$$

In other words, $\hat{\mathbf{y}}_t = M\bar{\mathbf{y}} \in \mathcal{S}$. Note that $AA^\top$ is invertible for the hierarchical setting. M, which is time-invariant, can be computed offline once, prior to the start of training. In principle, the projection problem (5) can accommodate additional convex constraints. Although this precludes the possibility of a closed-form solution, the projection can be implemented with a differentiable layer with the `DCL` framework (Agrawal et al., 2019a).

Accuracy-coherency trade-off. In practice, coherency seems to improve accuracy (Wickramasuriya et al., 2019); our experiments confirm that as well. However, if there is a trade-off, we can convert the constrained optimization problem (5) into unconstrained problem with a penalty parameter (which would then be a hyperparameter giving the trade-off). Values close to zero for this penalty parameter enforce no coherency and larger values make the predictions more coherent. Depending on the application, one could select either the soft penalized version or the hard constrained version.

3.2. Training

The training of our hierarchical forecasting model is similar to `DeepVAR` except that the loss is directly computed on the coherent predicted samples. Given a batch of training series $Y := \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$, where $\mathbf{y}_t \in \mathbb{R}^n$, and associated time series features $X := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, the likelihood

of the shared parameters Φ is given by

$$\ell(\Phi) = p(Y; X, \Phi) = \prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{y}_{1:t-1}; \mathbf{x}_{1:t}, \Phi)$$
$$= \prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{y}_{t-1}; \mathbf{x}_t, \mathbf{h}_{t-1}, \Phi) = \prod_{t=1}^{T} p(\mathbf{y}_t; \Theta_t),$$

where Θ_t are the distribution parameters (Eq. (4)) predicted by the `DeepVAR` model.

In our hierarchical setting, the learnable parameters are still given by Φ but the model outputs coherent Monte Carlo samples $\{\hat{\mathbf{y}}_t\}$ at each time step t. We are then able to compute sufficient statistics Θ_t^c on $\{\hat{\mathbf{y}}_t\}$ and define the following likelihood model:

$$\ell(\Phi) = \prod_{t=1}^{T} p(\mathbf{y}_t; \Theta_t^c).$$

The exact distribution $p(\mathbf{y}_t; \Theta_t^c)$ can be chosen according to the data. One can then maximize the likelihood to estimate the parameters Φ. More importantly, we have the flexibility to estimate the parameters Φ by optimizing any other loss function such as quantile loss, continuous ranked probability score (Section 4 contains more details) or any of the metrics typically preferred in the forecasting community. This is possible because any quantile of interest can be computed given sufficiently many samples (N large enough.)

In our model, the sampling step is differentiable as long as the distribution chosen allows for a suitable reparameterization where the random "noise" component of the distribution can be separated from the deterministic values of the parameters. This is the case for several distributions including Gaussian (Kingma & Welling, 2013), Gamma, log-Normal, Beta (Ruiz et al., 2016) and Student-t (Abiri & Ohlsson, 2019). Figurnov et al. (2018) present an alternative approach to compute reparameterization gradients showing broader applicability to Student-t, Dirichlet and mixture distributions.

The projection step in our setting is just a matrix-vector multiplication and poses no problem in automatically determining the gradients. When reconciliation is a more involved optimization problem, as long as the problem can be written as a conic program, techniques such the ones presented in (Agrawal et al., 2019b) enable automatic differentiation.

3.3. Prediction

Prediction is performed by unrolling the RNN step-by-step over the prediction horizon as shown in Figure 4 (Salinas et al., 2019). Given an observed hierarchical time series $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$, we wish to predict its values for τ subsequent periods. Starting with $t = T + 1$, we obtain forecast

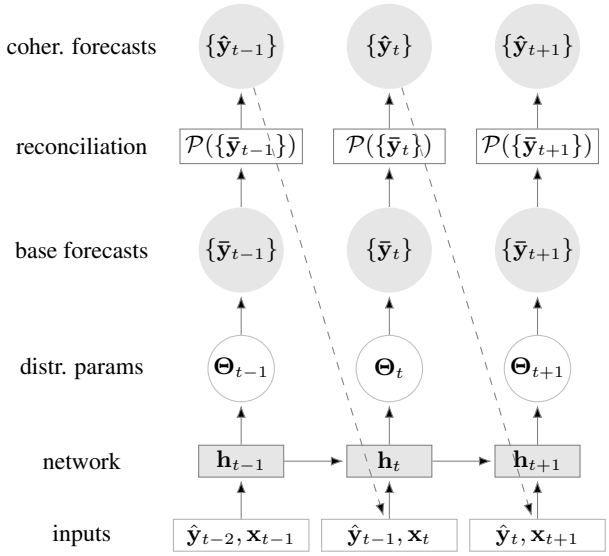

Figure 4. Multistep prediction using our model. The dashed line indicates that the model prediction from the previous time step $t-1$ is used as the lag input for time step $t, t > T+1$.

distribution parameters Θ_{T+1} by unrolling the RNN for one time step using the last hidden state from training $\mathbf{h}_t$, time series features $\mathbf{x}_{1:T+1}$ and the observed lag values $\mathbf{y}_{t-1}, t = 2, 3, \ldots, T+1$. We then generate a set of sample predictions $\{\hat{\mathbf{y}}_{T+1}\}$ by first taking Monte Carlo samples from parameters Θ_{T+1} and then projecting them with the same matrix M used in training. For each $t > T+1$, a sample predicted in the previous step $\hat{\mathbf{y}}_{t-1}$ is used as the lag input, shown as the dotted line in Figure 4, to generate prediction $\hat{\mathbf{y}}_t$. We repeat this procedure for each of the N samples generated at the beginning of the prediction horizon $T+1$. This way, we have obtained a set of sample paths $\{\hat{\mathbf{y}}_T + 1, \ldots, \hat{\mathbf{y}}_T + \tau\}$ that is coherent when the end of the prediction horizon is reached.

These samples may then be used to generate point (mean) or probabilistic forecasts by computing appropriate sample statistics (e.g., quantiles). Evaluation of the sample forecasts is discussed further in Section 4.

Here we would like to highlight the key benefits of the overall approach in comparison to the state-of-the art. By design our method is guaranteed to produce *coherent* and *probabilistic* forecasts in a straightforward way without performing reconciliation independently. By training using all time series simultaneously in a single nonlinear model we improve the fit for each individual time series thereby achieving better accuracy. One could easily replace `DeepVAR` with any recently proposed multivariate forecasting model without requiring major changes. Moreover, the overall model can be trained using application-dependent loss functions. Our method can naturally handle structural constraints

APPROACH	NAIVEBU	MINT	ERM	PERMBU-GTOP	HIER-E2E (OURS)
PROBABILISTIC	×	×	×	✓	✓
MULTIVARIATE	×	×	×	✓	✓
IMPLICIT RECONCILIATION	×	×	×	×	✓
STRUCTURAL CONSTRAINTS (≥ 0)	✓	✓	✓	×	✓

Table 1. Comparative summary of competing approaches on various dimensions.

DATASET	TOTAL	BOTTOM	AGGREGATED	LEVELS	OBSERVATIONS	τ
TOURISM	89	56	33	4	36	8
TOURISM-L (GROUPED)	555	76, 304	175	4, 5	228	12
LABOUR	57	32	25	4	514	8
TRAFFIC	207	200	7	4	366	1
WIKI	199	150	49	5	366	1

Table 2. Datasets summary. Here, τ refers to the prediction horizon length.

such as non-negativity etc. Table 1 summarizes the benefits along these key dimensions.

4. Experiments

We present empirical evaluation of the proposed method on publicly available hierarchical datasets. Table 2 lists key dataset features and we give a brief summary here. `Labour` (Australian Bureau of Statistics, 2020) contains monthly Australian employment data from Feb. 1978 to Dec. 2020. Using included category labels, we construct a 57-series hierarchy. `Traffic` (Cuturi, 2011; Dua & Graff, 2017) records the occupancy rate of car lanes on Bay Area freeways. We aggregate sub-hourly data to obtain daily observations for one year and generate a 207-series hierarchy using the same aggregation strategy as in (Ben Taieb & Koo, 2019). `Tourism` (Tourism Australia, Canberra, 2005) consists of an 89-series geographical hierarchy with quarterly observations of Australian tourism flows from 1998 to 2006. This dataset is frequently referenced in hierarchical forecasting studies (Athanasopoulos et al., 2009; Hyndman et al., 2011). `Tourism-L` (Wickramasuriya et al., 2019) is a larger, more detailed version of `Tourism` with 555 total series in a grouped structure and 228 observations; it has two hierarchies, based on geography and purpose-of-travel, sharing a common root. `Wiki` includes daily views for 145,000 Wikipedia articles starting from Jul. 2015 to Dec. 2016.[4] We follow the procedure described by Ben Taieb & Koo (2019) to filter the dataset to 150 bottom series (199 total). Table 2 also gives the prediction length used for the evaluation. We use the same prediction lengths as those in the publications where the datasets were first referenced. For τ, the prediction length, let $T + \tau$ be the length of the time series available for a dataset. Then each method initially receives time series for the first T time steps which are used to tune hyperparameters in a back-test fashion, e.g., training on the first $T - \tau$ steps and validating on the last τ time steps. Once the best hyperparameters are found, each model is once again trained on T time steps and is evaluated on the time steps from $T + 1$ to $T + \tau$.

We use the continuous ranked probability score (CRPS) to evaluate the accuracy of our forecast distributions.[5] CRPS is a strictly proper scoring rule (Gneiting & Ranjan, 2011), meaning a sample scores better (lower) when it is drawn from the true distribution. Following Laio & Tamea (2007), given a univariate predictive CDF $\hat{F}_{t,i}$ for time series i, and a ground-truth observation $y_{t,i}$, CRPS can be defined as

$$\mathrm{CRPS}(\hat{\mathbf{F}}_t, \mathbf{y}_t) := \sum_i \int_0^1 \mathrm{QS}_q \left(\hat{F}^{-1}_{t,i}, y_{t,i} \right) \, dq, \qquad (7)$$

where QS_q is the quantile score (or pin-ball loss) for the q-th quantile:

$$\mathrm{QS}_q = 2 \left(\mathbb{1}\{ y_{t,i} \leq \hat{F}^{-1}_{t,i}(q) \} - q \right) \left(\hat{F}^{-1}_{t,i}(q) - y \right).$$

A discrete version of this is implemented in GluonTS, with the integral in (7) replaced by the weighted sum over the quantile set. We use the quantiles ranging from 0.05 to 0.95 in steps of 0.05.

We compare against the following hierarchical forecasting methods. `NaiveBU` generates univariate point forecasts for the bottom-level time series independently and then sums them according to the hierarchical constraint to get point forecasts for the aggregate series. `MinT` (Wickramasuriya et al., 2019) is the general reconciliation procedure that

[4] `https://www.kaggle.com/c/web-traffic-time-series-forecasting/data`

[5] Because our model produces forecasts end-to-end, we do not gauge the performance of our forecasts as an increase/decrease in accuracy versus an incoherent base forecasts, as in (Hyndman et al., 2011; Wickramasuriya et al., 2019; Ben Taieb et al., 2017).

	method	Labour	Traffic	Tourism	Tourism-L	Wiki
	ARIMA-NaiveBU	0.0453	0.0808	0.1138	0.1741	0.3772
	ETS-NaiveBU	0.0432	0.0665	0.1008	0.1690	0.4673
	ARIMA-MinT-shr	0.0467	0.0770	0.1171	0.1609	0.2467
	ARIMA-MinT-ols	0.0463	0.1116	0.1195	0.1729	0.2782
	ETS-MinT-shr	0.0455	0.0963	0.1013	0.1627	0.3622
	ETS-MinT-ols	0.0459	0.1110	0.1002	0.1668	0.2702
	ARIMA-ERM	0.0399	0.0466	0.5887	0.5635	0.2206
	ETS-ERM	0.0456	0.1027	2.3755	0.5080	0.2217
	PERMBU-MINT	0.0393±0.0002	0.0677±0.0061	**0.0771±0.0001**	—	0.2812±0.0240
	Hier-E2E (Ours)	**0.0340±0.0088**	**0.0376±0.0060**	0.0834±0.0052	**0.1520±0.0032**	**0.2038±0.0110**
ablation study	DeepVAR	0.0382±0.0045	0.0400±0.0026	0.0925±0.0022	0.1581±0.0102	0.2294±0.0158
ablation study	DeepVAR+	0.0433±0.0079	0.0434±0.0049	0.0958±0.0062	0.1882±0.0242	0.2439±0.0224

Table 3. CRPS numbers (lower is better) averaged over 5 runs. State-of-the-art methods except for PERMBU-MINT produced same results over multiple runs. PERMBU-MINT didn't work for the grouped-dataset Tourism-L.

revises unbiased independent univariate base forecasts in such a way that variances of forecast errors are minimized. ERM (Ben Taieb & Koo, 2019) relaxes the unbiasedness assumption of the base forecasts in MinT and instead optimizes the bias-variance tradeoff of the forecast errors.

We report results for different combinations of base forecasting methods and reconciliation strategies. For NaiveBU, MinT and ERM we compute base forecasts with both ARIMA and ETS with auto-tuning enabled using the R package hts (Hyndman et al., 2020). For MinT we consider the covariance matrix with shrinkage operator (MinT-shr) and the diagonal covariance matrix corresponding to ordinary least squares weights (MinT-ols) (Wickramasuriya et al., 2019). We reimplemented ERM and set the context length T_1 to be $T - (\tau + 1)$ in order to maximize the number of observations available to build the projection matrix used in the method. Since the underlying base forecast methods are set to auto-tuning mode, the best hyperparameters are determined on the same training-validation split as our method uses. There are no other hyperparameters to tune for the reconciliation strategies apart from the covariance settings for MinT, for which we explicitly report results. These methods returned identical results over multiple runs for a given set of hyperparameters and hence standard deviations of the errors are not mentioned. PERMBU (Ben Taieb et al., 2017) is the only existing method that produces probabilistic forecasts for the hierarchical setting. After correspondence with the authors, we decided together to focus our experiments on the replication of the results with MinT reconciliation.

We implemented our method with the public GluonTS forecasting library (Alexandrov et al., 2019). We will make our code available in GluonTS. Experiments are run on Amazon SageMaker (Liberty et al., 2020). The details of hyperparameters are given in the supplementary material. For prediction, we sampled 200 times from the learned parameters to generate an empirical predictive distribution. We ran our method 5 times and report the mean and standard deviation of the CRPS scores. The results are shown in Table 3. Our method achieves the best results across all datasets (except for Tourism where it is second-best), showing the advantages of a single end-to-end model. We performed ablation studies on two variants of our model to further analyze the source of improvement. The first was vanilla DeepVAR with no reconciliation. This is not guaranteed to produce coherent forecasts. The second version was DeepVAR+, which refers to applying reconciliation during prediction only as a post-processing step. This produced coherent forecasts but did not exploit hierarchical relationships during training. As the scores in Table 3 show, reconciliation done independently of training as a post-processing step can worsen the forecasts. Note that DeepVAR and DeepVAR+ are treated as different models and are tuned accordingly; hence they potentially have different optimal hyperparameter settings. Interestingly, DeepVAR being a global model was able to achieve better results than the state-of-the-art hierarchical methods without explicit reconciliation. This further supports the claim that learning from all the time series jointly improves forecast quality especially when there are few available observations (e.g., Tourism of length 36). The univariate method ETS-ERM returns a sizable error on this dataset, although its advantages become evident on a larger, more involved dataset like Wiki. Overall, by enforcing the aggregation constraint, our method improves over unreconciled DeepVAR.

In order to assess if the gains in the performance are uniform across aggregation levels, we present CRPS scores by level of aggregation for all datasets in the supplementary material. Our method achieves performance gains consistently across all aggregation levels unlike some of the state-of-the-art, which trade off favorable accuracy at the aggregated levels with less favorable accuracy at the disaggregated levels; see supplementary for details. To highlight this, Table 4 provides a summary by showing only the results of the best

Dataset	Level	Hier-E2E(Ours)	DeepVAR	DeepVAR+	Best of Competing Methods
Labour	1	**0.0311±0.0120**	*0.0352±0.0079*	0.0416±0.0094	0.0406±0.0002 (PERMBU-MINT)
	2	**0.0336±0.0089**	*0.0374±0.0051*	0.0437±0.0078	0.0389±0.0002(PERMBU-MINT)
	3	**0.0336±0.0082**	0.0383±0.0038	0.0432±0.0076	*0.0382±0.0002*(PERMBU-MINT)
	4	**0.0378±0.0060**	0.0417±0.0038	0.0448±0.0066	*0.0397±0.0003*(PERMBU-MINT)
Traffic	1	*0.0184±0.0091*	0.0225±0.0109	0.0250±0.0082	**0.0087**(ARIMA-ERM)
	2	*0.0181±0.0086*	0.0204±0.0044	0.0244±0.0063	**0.0112**(ARIMA-ERM)
	3	*0.0223*±0.0072	**0.0190±0.0031**	0.0259±0.0054	0.0255(ARIMA-ERM)
	4	**0.0914±0.0024**	*0.0982±0.0012*	0.0982±0.0017	0.1410(ARIMA-ERM)
Tourism	1	**0.0402±0.0040**	0.0519±0.0057	0.0508±0.0085	*0.0472±0.0012* (PERMBU-MINT)
	2	*0.0658±0.0084*	0.0755±0.0011	0.0750±0.0066	**0.0605±0.0006** (PERMBU-MINT)
	3	*0.1053±0.0053*	0.1134±0.0049	0.1180±0.0053	**0.0903±0.0006** (PERMBU-MINT)
	4	*0.1223±0.0039*	0.1294±0.0060	0.1393±0.0048	**0.1106±0.0005** (PERMBU-MINT)
Tourism-L	1	*0.0810±0.0053*	0.1029±0.0188	0.1214±0.0360	**0.0438** (ARIMA-MinT-shr)
	2 (geo.)	*0.1030±0.0030*	0.1076±0.0119	0.1364±0.0299	**0.0816** (ARIMA-MinT-shr)
	3 (geo.)	**0.1361±0.0024**	*0.1407±0.0081*	0.1713±0.0243	0.1433(ARIMA-MinT-shr)
	4 (geo.)	*0.1752±0.0026*	**0.1741±0.0066**	0.2079±0.0215	0.2036(ARIMA-MinT-shr)
	2 (trav.)	*0.1027±0.0062*	0.1100±0.0139	0.1370±0.0289	**0.0830** (ARIMA-MinT-shr)
	3 (trav.)	**0.1403±0.0047**	0.1485±0.0099	0.1776±0.0221	*0.1479* (ARIMA-MinT-shr)
	4 (trav.)	**0.2050±0.0028**	*0.2078±0.0076*	0.2435±0.0170	0.2437(ARIMA-MinT-shr)
	5 (trav.)	**0.2727±0.0017**	*0.2731±0.0066*	0.3108±0.0164	0.3406(ARIMA-MinT-shr)
Wiki	1	**0.0419±0.0285**	0.0905±0.0323	*0.0755±0.0165*	0.1558 (ETS-ERM)
	2	**0.1045±0.0151**	0.1418±0.0249	*0.1289±0.0171*	0.1614 (ETS-ERM)
	3	*0.2292±0.0108*	0.2597±0.0150	0.2583±0.0281	**0.2010**(ETS-ERM)
	4	*0.2716±0.0091*	0.2886±0.0112	0.3108±0.0298	**0.2399**(ETS-ERM)
	5	0.3720±0.0150	*0.3664±0.0068*	0.4460±0.0271	**0.3507**(ETS-ERM)

Table 4. Mean CRPS scores (lower is better) computed for time series at each aggregation level, averaged over 5 runs. Level 1 corresponds to the root of the hierarchy. To show the high-level summary, here we include only the result of the best performing competing method along with our method and its variants. For each dataset, we choose the competing method (among the state-of-the-art without the proposed method and its variants) that achieves the best result in as many aggregation levels as possible. In case of ties, we choose the method that achieved the best overall CRPS score (averaged across all the levels in the hierarchy). Among the methods shown here, the best result is shown in **boldface** and the second best result is *italicized*. The detailed CRPS scores for all methods and all levels are also given in the following tables.

performing competing method along with our method and its variants. For each dataset, we choose the competing method (among the state-of-the-art without the proposed method and its variants) that achieves the best result in as many aggregation levels as possible. In case of ties, we choose the method that achieved the best overall CRPS score (averaged across all the levels in the hierarchy). Our method consistently performs better at all levels achieving the **best** result (**boldface**) for 12 out of 25 total levels. For the remaining 13 levels it is the *second-best* (*italicized*) except for one level (Wiki Level-5). The next-best competing method, PERMBU-MINT, achieved the **best** and the *second-best* result only for 3 levels each. ETS-ERM and ARIMA-MinT-shr also achieved the **best** result in 3 levels each; however unlike PERMBU-MINT, they achieved this result only in a single dataset.

5. Conclusion

We have presented a new approach for probabilistic forecasting of hierarchical time series. The main novelty is the proposal of a single, global model that does not require any adjustments to produce coherent, probabilistic forecasts, a first of its kind. Moreover, the proposed approach can readily handle more general structural constraints beyond the hierarchical set up via a differentiable convex optimization layer. Our approach is generic in the sense that we can add it to most existing deep forecasting models. We empirically showed that training a single, global model together with the enforcement of coherency achieves better results than the prior state-of-the-art which uses a two-step procedure instead of end-to-end-learning.[6] Although we found empirically that a multivariate Gaussian distribution performed well on the datasets considered, as future work we would like to explore the use of nonlinear transformations like normalizing flows to better model non-Gaussian data.

Acknowledgements

The authors thank Souhaib Ben Taieb for insightful discussions on this topic and his timely help in reproducing results with PERMBU-MINT. The authors are also grateful for being able to build on the work of Valentin Flunkert and David Salinas in both concepts and code.

[6] In concurrent and independent work, Han et al. (2021) propose a two-step procedure for making forecasts, for pre-specified quantiles, as coherent as possible: it first promotes the coherence of median forecasts via regularized quantile loss and then reconciles the other (unconstrained) quantile forecasts in a bottom-up fashion. Note that their forecasts are not coherent in general (see Table 7 of supplementary, (Han et al., 2021)). Future work will provide a detailed comparison with this method.

References

Abiri, N. and Ohlsson, M. Variational auto-encoders with student's t-prior. In *Proceedings, 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2019)*, pp. 415–420, Bruges, 2019.

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, Z. Differentiable convex optimization layers. *arXiv preprint arXiv:1910.12430*, 2019a.

Agrawal, A., Barratt, S., Boyd, S., Busseti, E., and Moursi, W. M. Differentiating through a cone program. *arXiv preprint arXiv:1904.09043*, 2019b.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. GluonTS: Probabilistic Time Series Models in Python. *JMLR*, 2019.

Amos, B. and Kolter, J. Z. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145. PMLR, 2017.

Athanasopoulos, G., Ahmed, R. A., and Hyndman, R. J. Hierarchical forecasts for australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166, 2009.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., and Petropoulos, F. Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1):60–74, 2017.

Australian Bureau of Statistics. Labour Force, Australia, Dec 2020. URL https://www.abs.gov.au/statistics/labour/employment-and-unemployment/labour-force-australia/latest-release. Accessed on 01.12.2021.

Ben Taieb, S. and Koo, B. 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, 2019.

Ben Taieb, S., Taylor, J. W., and Hyndman, R. J. Coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pp. 3348–3357, 2017.

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

Berrocal, V. J., Raftery, A. E., Gneiting, T., and Steed, R. C. Probabilistic weather forecasting for winter road maintenance. *Journal of the American Statistical Association*, 105(490):522–537, 2010.

Bojer, C. S. and Meldgaard, J. P. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, Sep 2020. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2020.07.007. URL http://dx.doi.org/10.1016/j.ijforecast.2020.07.007.

Böse, J.-H., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., Schelter, S., Seeger, M., and Wang, Y. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12):1694–1705, 2017.

Cuturi, M. Fast global alignment kernels. In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, 2011.

de Bézenac, E., Rangapuram, S. S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P., and Januschowski, T. Normalizing kalman filters for multivariate time series analysis. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Faloutsos, C., Gasthaus, J., Januschowski, T., and Wang, Y. Classical and contemporary approaches to big time series forecasting. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, New York, NY, USA, 2019. ACM.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 441–452. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/92c8c96e4c37100777c7190b76d28233-Paper.pdf.

Gneiting, T. and Katzfuss, M. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

Gneiting, T. and Ranjan, R. Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, 29(3):411–422, 2011.

Han, X., Dasgupta, S., and Ghosh, J. Simultaneously reconciled quantile forecasting of hierarchically related time series. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 190–198. PMLR, 13–15 Apr 2021.

Hollyman, R., Petropoulos, F., and Tipping, M. E. Understanding forecast reconciliation. *European Journal of Operational Research*, 2021.

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

Hyndman, R. J. and Athanasopoulos, G. Forecasting: Principles and practice. *www. otexts. org/fpp.*, 987507109, 2017.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., and Shang, H. L. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9):2579–2589, 2011.

Hyndman, R. J., Lee, A. J., and Wang, E. Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational statistics & data analysis*, 97:16–32, 2016.

Januschowski, T. and Kolassa, S. A classification of business forecasting problems. *Foresight: The International Journal of Applied Forecasting*, 52:36–43, 2019.

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

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Liberty, E., Karnin, Z., Xiang, B., Rouesnel, L., Coskun, B., Nallapati, R., Delgado, J., Sadoughi, A., Astashonok, Y., Das, P., Balioglu, C., Chakravarty, S., Jha, M., Gautier, P., Arpin, D., Januschowski, T., Flunkert, V., Wang, Y., Gasthaus, J., Stella, L., Rangapuram, S., Salinas, D., Schelter, S., and Smola, A. Elastic machine learning algorithms in amazon sagemaker. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pp. 731–737, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367356.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 2018. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2018.06.001. URL `http://www.sciencedirect.com/science/article/pii/S0169207018300785`.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The m5 accuracy competition: Results, findings and conclusions. *International Journal of Forecasting*, to appear.

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G., and Hyndman, R. J. Probabilistic Forecast Reconciliation: Properties, Evaluation and Score Optimisation. Technical report, 2020.

Petropoulos, F. et al. Forecasting: theory and practice. *arXiv preprint arXiv:2012.03854*, 2020.

Rasul, K., Sheikh, A.-S., Schuster, I., Bergmann, U., and Vollgraf, R. Multivariate probabilistic time series forecasting via conditioned normalizing flows, 2021.

Ruiz, F. R., Titsias RC AUEB, M., and Blei, D. The generalized reparameterization gradient. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29, pp. 460–468. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/f718499c1c8cef6730f9fd03c8125cab-Paper.pdf`.

Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R., and Gasthaus, J. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32:6827–6837, 2019.

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

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

Seeger, M. W., Salinas, D., and Flunkert, V. Bayesian intermittent demand forecasting for large inventories. In *Advances in Neural Information Processing Systems*, pp. 4646–4654, 2016.

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

Taieb, S. B., Taylor, J. W., and Hyndman, R. J. Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, 0(0):1–17, 2020.

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

Van Erven, T. and Cugliari, J. Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts. In *Modeling and stochastic learning for forecasting in high dimensions*, pp. 297–317. Springer, 2015.

Wickramasuriya, S. L., Athanasopoulos, G., and Hyndman, R. J. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019. doi: 10.1080/01621459.2018.1448825. URL https://doi.org/10.1080/01621459.2018.1448825.