Learning Optimal Projection for Forecast Reconciliation of Hierarchical Time Series

Asterios Tsiourvas [1] **Wei Sun** [2] **Georgia Perakis** [1] **Pin-Yu Chen** [2] **Yada Zhu** [2]

Abstract

Hierarchical time series forecasting requires not only prediction accuracy but also coherency, i.e., forecasts add up appropriately across the hierarchy. Recent literature has shown that reconciliation via projection outperforms prior methods such as top-down or bottom-up approaches. Unlike existing work that pre-specifies a projection matrix (e.g., orthogonal), we study the problem of learning the optimal oblique projection from data for coherent forecasting of hierarchical time series. In addition to the unbiasedness-preserving property, oblique projection implicitly accounts for the hierarchy structure and assigns different weights to individual time series, providing significant adaptability over orthogonal projection which treats base forecast errors equally. We examine two broad classes of projections, namely Euclidean projection and general oblique projections. We propose to model the reconciliation step as a learnable, structured, projection layer in the neural forecaster architecture. The proposed approach allows for the efficient learning of the optimal projection in an end-to-end framework where both the neural forecaster and the projection layer are learned simultaneously. An empirical evaluation of real-world hierarchical time series datasets demonstrates the superior performance of the proposed method over existing state-of-the-art approaches.

1. Introduction

A hierarchical time series refers to a collection of time series that follows a hierarchical aggregation structure. Forecasting hierarchical time series has garnered increasing attention due to its crucial role in decision-making across various domains (Dai et al., 2017). For instance, in retail, demand forecasts across different levels of granularity (e.g., product, store, state, country) are essential for inventory control and revenue management (Seeger et al., 2016). In the energy and utility sector, accurate electricity consumption forecasts at individual, grid, and regional levels are vital for the efficient operation of power grids (Taieb et al., 2017; 2021).

To forecast hierarchical time series, besides accuracy, it is also critical to ensure *coherence*, i.e., the forecasts of each aggregation group are equal to those making up the group. When individual time series are learned independently, there is no guarantee that these forecasts would satisfy the aggregation constraints specified by the hierarchy. Previous methods address coherence by forecasting only a single level of the hierarchy and then reconciling either in a top-down (Athanasopoulos et al., 2009; Gross & Sohl, 1990; Das et al., 2023) or bottom-up approach (Kahn, 1998), or by utilizing a combination of both known as the middle-out method (Hollyman et al., 2021). There are two issues associated with this approach. Firstly, the model parameters for each time series are learned independently of the reconciliation method that follows. Secondly, since such approaches only utilize partial data in the hierarchy, valuable information present at other levels is neglected.

Several notable advances in forecast reconciliation literature that attempt to combine forecasts across all levels via solving a regression problem are shown to outperform prior methods (Hyndman et al., 2011; Wickramasuriya et al., 2019). Panagiotelis et al. (2021) provide a geometric interpretation of these reconciliation methods as specific instances of projections. In particular, Panagiotelis et al. (2021) show that the method proposed in Hyndman et al. (2011) is an orthogonal projection, while MinT method from Wickramasuriya et al. (2019) is a special case of generalized Euclidean projections. Reconciliation via projections enjoys desirable properties such as the unbiasedness-preserving property, i.e., the reconciled forecasts are unbiased if the initial forecasts are also unbiased.

Oblique projection (such as generalized Euclidean projection) offers significantly more flexibility in modeling as the standard orthogonal projection method implicitly treats

[1]Operations Research Center, Massachusetts Institute of Technology, USA [2]IBM Research, USA. Correspondence to: Asterios Tsiourvas <atsiour@mit.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

individual base forecast errors equally, disregarding the hierarchy structure. However, implementing oblique projection is remarkably difficult. Take MinT (Wickramasuriya et al., 2019) as an example, despite having a closed-form solution in theory, it requires estimating the covariances of forecast errors, rendering its implementation infeasible in practice, without resorting to approximations.

In this paper, we attempt to address the challenge of implementing oblique projection for forecast reconciliation of hierarchical time series by proposing a novel, flexible, and tractable framework. Our contributions are threefold.

1. Existing work on forecasting hierarchical time series via projection only considers pre-defined projection matrices. To the best of our knowledge, this is the first work that proposes a **learnable projection** method, where we integrate the oblique projection as a structured layer within a neural forecaster architecture. This allows for the efficient learning of the oblique projection on an end-to-end framework where both the neural forecaster and the projection layer are learned simultaneously while generating forecasts that are coherent by construction. The proposed approach of utilizing the structured layer for reconciliation is highly flexible and we show that it can be applied to both point and probabilistic forecasting of hierarchical time series.

2. We consider two types of oblique projections. The first is **generalized Euclidean projection**, where we learn a symmetric, positive-definite matrix from a hierarchical time series. To impose such a structure, we perform matrix decomposition and implement a symmetric, positive-definite, dense neural network layer. We also consider **general oblique projection**, where we only require the projection matrix to satisfy the idempotence property, which is achieved via regularization. The general oblique projection provides a higher degree of adaptability due to its minimal constraint on the neural network structure, while the generalized Euclidean projection offers an added benefit of interpretability as the learned matrix captures different weights applied to the forecasting errors of individual time series, accounting for the hierarchy. In comparison, instead of learning the projection from data, the state-of-the-art approach that utilizes orthogonal projection simply specifies this matrix as an identity matrix, i.e., effectively treating errors in individual time series as equal.

3. We validate the superior performance of our approach through **extensive empirical evaluation** on both point and probabilistic forecasting methods, using real-world datasets. Specifically, we compare against existing state-of-the-art reconciliation approaches, of both sequential and end-to-end frameworks. We also experiment with different neural forecasters. Our proposed methods consistently outperform benchmarks in all datasets and across different levels of the hierarchies, highlighting the advantages of using learnable, oblique projections for hierarchical time series forecasting.

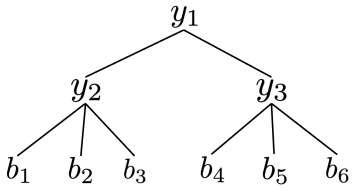


Figure 1. Example of a tree hierarchically structured time series for $n = 9$ time series with $m = 6$ bottom and $r = n - m = 3$ aggregated time series.

2. Reconciled Hierarchical Forecasting

2.1. Notations and preliminaries

Following the notation in Hyndman et al. (2011), we define a hierarchical time series as a collection of n variables indexed by time t, where $t = 1, \dots, T$. We denote the n-dimensional vector which includes observations of all variables in the hierarchy at time t as $\boldsymbol{y}_t \in \mathbb{R}^n$, with $y_{t,i} \in \mathbb{R}$ as the value of the i-th univariate time series at time t. We refer to the time series at the bottom of the hierarchy as bottom-level series of dimension m, and the rest of the series as aggregated-level series of dimension $n - m$. Based on this definition, $\boldsymbol{y}_t$ can be expressed as $[\boldsymbol{a}_t \ \boldsymbol{b}_t]^T$, where $\boldsymbol{b}_t \in \mathbb{R}^m$ and $\boldsymbol{a}_t \in \mathbb{R}^{n-m}$ represent the vectors of the bottom-level series and the aggregated-level series at time t respectively.

We assume that the indexing of each individual time series is given by the level-order traversal of the hierarchy going from left to right. Each hierarchical time series structure can be described by the aggregation matrix $S = \{0, 1\}^{n \times m}$ that is defined to satisfy that

$$\boldsymbol{y}_t = S\boldsymbol{b}_t \iff \begin{bmatrix} \boldsymbol{a}_t \\ \boldsymbol{b}_t \end{bmatrix} = \begin{bmatrix} S_{sum} \\ I_m \end{bmatrix} \boldsymbol{b}_t, \ \forall t \in [T] \quad (1)$$

where $S_{sum} \in \mathbb{R}^{r \times m}$ is the summation matrix and $I_m \in \mathbb{R}^{m \times m}$ is the identity matrix.

To illustrate the concepts, consider the following example with the hierarchy depicted in Figure 1. We have that $\boldsymbol{a}_t = [y_1, y_2, y_3]_t^T \in \mathbb{R}^3$ and $\boldsymbol{b}_t = [b_1, b_2, b_3, b_4, b_5, b_6]_t^T \in \mathbb{R}^6$. Furthermore, $S_{sum} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$. It is important to note that in addition to the tree structure, there are other examples of hierarchies including temporal hierarchies (Athanasopoulos et al., 2017), cross-temporal aggregation structures (Spiliotis et al., 2020), etc.

The quantity of our interest is to forecast each time series in the hierarchy for a time horizon h, i.e. for all times $t = T+1, \ldots, T+h$. The typical approaches of forecasting hierarchical time series follow a two-step procedure: (i) forecast each time series independently to obtain base forecasts of the multivariate time series τ time steps ahead, denoted by $\hat{\boldsymbol{y}}_{T+\tau} \in \mathbb{R}^n$, which are not necessarily reconciled, and (ii) produce adjusted forecasts $\tilde{\boldsymbol{y}}_{T+\tau}$ through reconciliation, which requires the forecasts to adhere to the aggregation constraint – a property referred to as *coherence*.

Definition 2.1. The m-dimensional linear subspace $\mathcal{S} \subseteq \mathbb{R}^n$ for which the linear aggregation constraints hold for all $\boldsymbol{y} \in \mathcal{S}$ is defined as the *coherent subspace*.

Definition 2.2. The forecast $\tilde{\boldsymbol{y}}_{T+\tau}$ of the multivariate time series τ time steps ahead is coherent if $\tilde{\boldsymbol{y}}_{T+\tau} \in \mathcal{S}$.

Definition 2.3. Let ξ be a mapping, $\xi : \mathbb{R}^n \to \mathcal{S}$. The forecast $\tilde{\boldsymbol{y}}_{T+\tau} = \xi(\hat{\boldsymbol{y}}_{T+\tau})$ reconciles the predictions with respect to the mapping.

All reconciliation methods that we are aware of consider linear mapping in the place of ξ, where the base forecasts are multiplied by an $n \times n$ matrix that has $\mathcal{S}$ as its image. Specifically, reconciled forecasts are always achieved by multiplying the base forecasts $\hat{\boldsymbol{y}}_{T+\tau}$ with the matrix SP, where $P \in \mathbb{R}^{m \times n}$, i.e., $\tilde{\boldsymbol{y}}_{T+\tau} = SP\hat{\boldsymbol{y}}_{T+\tau}$.

When the reconciliation matrix is defined as $P = [\mathbf{0}_{m \times r} | \mathbf{1}_{m \times m}]$, it represents the bottom-up approach. On the other hand, when the reconciliation matrix is defined as $P = [\boldsymbol{p}_{m \times 1} | \mathbf{0}_{m \times (n-1)}]$, where $\boldsymbol{p}$ is a vector that sums to 1 that disaggregates the top-level series proportionally to the bottom series, we obtain the top-down approach.

2.2. Coherent Reconciliation via Projection

In the literature on hierarchical forecasting, several works have considered a specific type of reconciliation where SP is a projection matrix onto $\mathcal{S}$ (Hyndman et al., 2011; Wickramasuriya et al., 2019; Panagiotelis et al., 2021).

Definition 2.4. Matrix SP is a projection matrix (onto $\mathcal{S}$) when the idempotence property holds, i.e. $(SP)^2 = SP$.

Hyndman et al. (2011) proposed to use $P = (S^T S)^{-1} S^T$, which is a solution to the so-called OLS reconciliation problem. Wickramasuriya et al. (2019) proposed $P = (S^T W_\tau^{-1} S)^{-1} (S^T W_\tau^{-1})$, also known as the MinT method, where W_τ is the covariance matrix of the τ time step ahead forecast errors $\hat{\boldsymbol{\epsilon}}_{T+\tau} = \boldsymbol{y}_{T+\tau} - \hat{\boldsymbol{y}}_{T+\tau}$. The authors show that when predictions are unbiased, this choice of P minimizes the sum of variances of the forecast errors and produces unbiased reconciled predictions. However, a disadvantage is that the error covariance matrix W_τ is hard to obtain for $\tau > 1$, and approximations are used instead.

Recently Panagiotelis et al. (2021) provided a geometric interpretation that encompasses these well-known reconciliation methods as specific instances of projections. In particular, the authors showed that the MinT projection matrix is a special case of generalized Euclidean projections $\min_{\boldsymbol{y} \in \mathcal{S}} ||\hat{\boldsymbol{y}}_{T+\tau} - \boldsymbol{y}||_W$ where the loss function is the generalized Euclidean norm with respect to matrix W, i.e. $||\boldsymbol{v}||_W^2 = \boldsymbol{v}^T W v$, under the assumption that W is an invertible and symmetric matrix. If W is known *a priori*, then the solution to the generalized Euclidean projection problem is $\tilde{\boldsymbol{y}}_{T+\tau} = SP\hat{\boldsymbol{y}}_{T+\tau}$, with $P = (S^T W S)^{-1}(S^T W)$. Meanwhile, the authors also show that the reconciliation matrix $P = (S^T S)^{-1} S^T$ proposed in Hyndman et al. (2011) is simply an orthogonal projection where $W = I$. To illustrate the benefit of oblique projection over orthogonal projection, consider the following example illustrated in Figure 2.

Example: We consider the multivariate time series $\boldsymbol{y} = (y_1, y_2) \in \mathbb{R}^2$. Let's assume that for $\tau = 1$, the actual values are $\boldsymbol{y}_1 = (1, 1)$ and the base predictions by a multivariate time series model are $\hat{\boldsymbol{y}}_1 = (2, 3)$. The reconciled predictions when the orthogonal projection reconciliation method is used, i.e. the reconciled predictions according to the Euclidean norm ($||\cdot||_W$, with $W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and, consequently $P = [0.5, 0.5]$), are $\tilde{\boldsymbol{y}}_{L_2} = (2.5, 2.5)$ and the corresponding Root Mean Squared Error (RMSE) is 0.5. On the other hand, if we could learn the matrix $W = \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix}$, or equivalently $P = [2, -1]$, then the learned general (oblique) projection would produce the reconciled prediction $\tilde{\boldsymbol{y}}_{obl} = (1, 1)$ that has a perfect RMSE of 0.

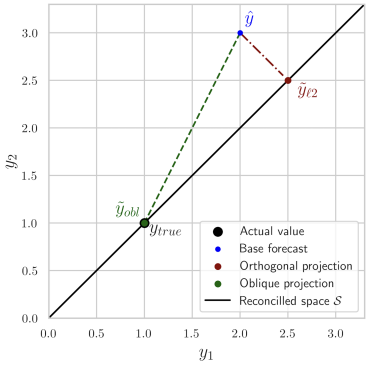


Figure 2. An example of forecast reconciliation via orthogonal (red) and oblique projection (green).

Existing approaches in the literature first pre-specify the projection matrix (i.e. matrix P) and then perform the reconciliation. In the next section, we propose a novel method that learns the optimal oblique projection matrix from data. Instead of performing the sequential two-stage procedure (as in Wickramasuriya et al. (2019) or Ben Taieb & Koo (2019)), we learn the optimal projection matrix and produce the reconciled forecasts in a single end-to-end model.

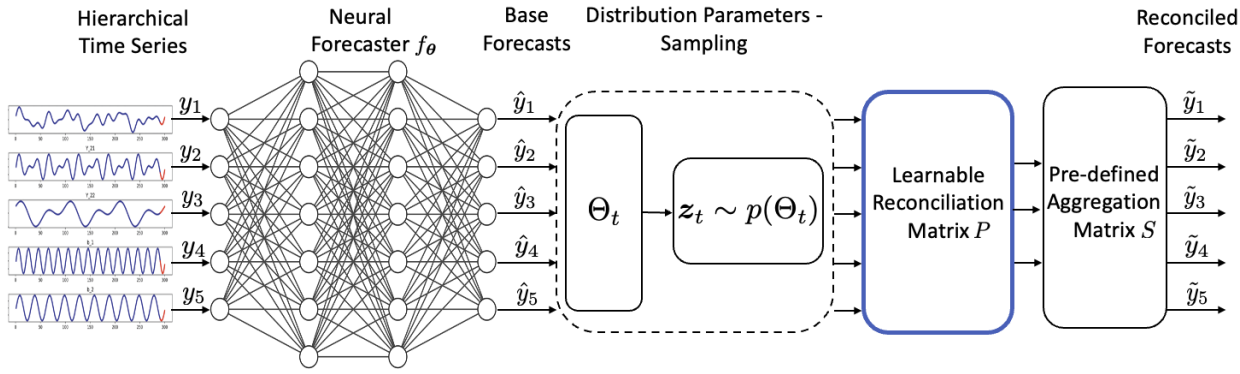


Figure 3. A representation of the proposed end-to-end architecture for both point and probabilistic forecasts. The components within the dashed box are used only during probabilistic forecasting. We model the reconciliation matrix P as a learnable, structured projection layer. P, $f_{\boldsymbol{\theta}}$ and Θ_t are learned simultaneously during training.

3. Learning Optimal Oblique Projection

To learn optimal oblique projections during training as part of a single end-to-end framework, we model the matrix P as a learnable, structured, dense layer in the neural forecaster architecture that is used for the hierarchical time series prediction. The proposed end-to-end architecture can be seen in Figure 3. Note that while the learnable projection layer can be applied to both point and probabilistic forecasting methods, the step that learns the distribution parameters and performs sampling (shown in the dashed box in Figure 3) is only needed for the probabilistic setting. As the vast majority of the literature in the field of hierarchical time series is on point-wise predictions, we focus on this setting in this section. Later in Section 4 we will discuss how to adapt the framework for probabilistic forecasts, and present experimental results from both settings in Section 5.

3.1. Proposed Learnable Projections

In this work, we focus on learning two classes of projections, namely, the generalized Euclidean projection, and the broad class of general oblique projections.

For the generalized Euclidean oblique projection, we impose the following structure on $P = (S^T W S)^{-1}(S^T W)$, with $W \in \mathbb{R}^{n \times n}$ to be a symmetric, positive-definite, dense neural network layer. To model symmetry, we set $W = (Q + Q^T)/2$, where Q is a learnable, positive-definite dense neural network layer. By performing this decomposition, matrix W is always symmetric, while we only need to learn a single matrix, Q. In Figure 4, we demonstrate the proposed decomposition of W that preserves the symmetric property. To model the positive-definite requirement for Q, we perform the eigenvalue-like factorization proposed by Lezcano-Casado (2019).

For the general oblique projection, we model P as an arbitrary dense layer with input dimension n and output dimension m. We train the complete model (neural forecaster + projection) under the constraint $(SP)^2 = SP$. To impose the idempotence property, we introduce a Lagrange multiplier λ to penalize the Frobenius norm $||PS - I||_F$ of the constraint $PS = I$, where $I \in \mathbb{R}^{m \times m}$ is the identity matrix. The satisfaction of this constraint implies that SP is a general projection matrix onto $\mathcal{S}$ since if $PS = I$, then $(SP)^2 = SPSP = S(PS)P = SIP = SP$. Table 1 provides a summary that compares the two proposed reconciliation approaches.

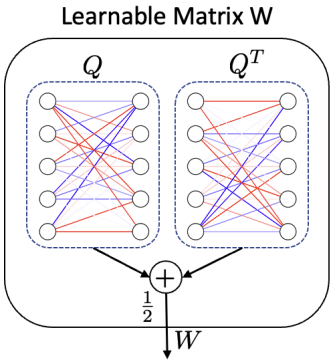


Figure 4. The proposed decomposition of matrix W into $\frac{Q+Q^T}{2}$, where Q is a positive-definite matrix.

3.2. End-to-End Learning

Formally, our proposed approach solves the following optimization problem:

$$\begin{aligned} \min_{P,\boldsymbol{\theta}} \quad & \frac{1}{T-1}\sum_{t=2}^{T} ||\boldsymbol{y}_t - SP f_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:t-1})|| \\ \text{subject to} \quad & \textit{projection constraints}, \end{aligned} \tag{2}$$

where $f_{\boldsymbol{\theta}}$ is the neural forecaster used for obtaining the base predictions, $\boldsymbol{\theta}$ is its set of trainable parameters, and the projection constraints depend on the type of the projection (see

Table 1. Summary of the proposed reconciliation approaches

Reconciliation	Structure of P	Projection constraints
Gen. Euclidean	$P = (S^T W S)^{-1}(S^T W)$	$W = \frac{Q+Q^T}{2}, Q \succ 0$
Gen. Projection	$P \in \mathbb{R}^{m \times n}$	$PS = I$

Table 1). Through the representation of the projection as a neural network layer, we can effectively leverage existing off-the-shelf frameworks to *efficiently* learn the neural forecaster and the projection layer at the same time using SGD. Furthermore, the proposed architecture enables the resulting projection to be informed *concurrently* by both the structure of the neural forecaster and the nature of the data.

3.3. Theoretical Guarantees of Reconciled Forecasts via Projection

Utilizing learnable, oblique projections for reconciling hierarchical time series forecasts not only enhances performance in terms of accuracy but also confers a set of significant properties to the reconciled forecasts. The first property is the unbiasedness preserving property that extends to both proposed oblique projections. We first state the following lemma.

Lemma 3.1 (Rao (1974))**.** *Any vector lying in the image of a projection is mapped to itself by that projection.*

This lemma implies that if SP is a projection matrix onto $\mathcal{S}$, then for every $\boldsymbol{v} \in \mathcal{S}$ we have that $SP\boldsymbol{v} = \boldsymbol{v}$. We now formally state the unbiasedness-preserving property for our proposed projections, while we present the proof in section A of the Appendix.

Proposition 3.2. *For an unbiased base forecast $\hat{\boldsymbol{y}}_\tau$, the reconciled point forecast produced by the proposed oblique projections is also an unbiased prediction.*

Intuitively, this property implies that if the predictions of the neural forecaster are unbiased, then the reconciled forecasts will remain unbiased. Furthermore, in addition to the unbiasedness property, the generalized Euclidean projection comes with some additional interesting properties, that enhance its interpretability and transparency. These properties extend the known properties of the original Euclidean projection to the setting of the generalized Euclidean projection. We state the first property in the following proposition.

Proposition 3.3. *The generalized Euclidean projection assigns different weights to different forecasts, i.e. transforms the space, and then applies an orthogonal projection to the weighted forecasts.*

Intuitively, the generalized Euclidean projection first transforms the space by multiplying all vectors and matrices involved with $W^{1/2}$ and then applies the orthogonal projection. Thus, by retrieving matrix W, we can identify the exact weights the projection gave to each time series in the hierarchy in order to reduce the overall error. Moreover, as shown in the following proposition, the generalized Euclidean projection never increases the error of the reconciled forecast, with respect to the norm induced by W.

Proposition 3.4. *The generalized Euclidean projection never increases the error of the reconciled forecast, with respect to the norm defined by W.*

Both proofs can be found in section A of the Appendix. It is worth noting, that despite these attractive properties associated with a generalized Euclidean projection, it is important to note that the loss function of interest is not always the norm induced by W. Furthermore, the generalized Euclidean projection imposes a specific structure on matrix P that might not always be essential or required. On the other hand, the general projection is the most flexible, and thus expressive projection, as it imposes the least structure on the learnable layer in order for SP to be a projection onto $\mathcal{S}$. In the experiments section, we also test the reconciliation scheme where P is unconstrained (and thus, not a projection) and we show empirically the advantages of using structured, projection layers for reconciliation.

4. Adaptation to Probabilistic Forecasting

Compared to point predictions, in probabilistic forecasting (Gneiting & Katzfuss, 2014; Salinas et al., 2020), the goal is to accurately estimate the conditional predictive CDF for each series i in the hierarchy, i.e. $F_{T+\tau,i}(y_i|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) = \mathbb{P}[y_i \leq y_{T+\tau,i}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$. The proposed method can be easily extended to produce coherent probabilistic forecasts.

Following the methodology proposed by Rangapuram et al. (2021) for both training and inference, we use the output of the multivariate forecaster to model the parameters Θ_t of the predictive distribution at time step t (typically the forecast distribution is assumed Gaussian, i.e. $\Theta_t = \{\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\}$, but it can be extended to other distributions) instead of the base forecasts. Given the estimated distribution parameters Θ_t, we generate probabilistic base forecasts by drawing a set of N Monte Carlo samples from the predicted distribution using the reparameterization trick. Then, we perform the projection step by performing the feed-forward pass through our proposed learnable projection layer and finally, we compute sufficient statistics from the samples and use them to

calculate the (log) likelihood loss function that is maximized during training (or any other relevant loss function). Similar to formulation (2) for point predictions, our approach solves the following optimization problem

$$\begin{aligned} \min_{P,\boldsymbol{\theta}} \quad & -\prod_{t=2}^{T} p(\boldsymbol{y}_t, \Theta_t^c) \\ \text{subject to} \quad & \textit{projection constraints}. \end{aligned} \tag{3}$$

This formulation maximizes the log-likelihood of the learned distribution. Θ_t^c are the sufficient statistics calculated on $SP\boldsymbol{z}_t$, with $\boldsymbol{z}_t \sim p(\Theta_t)$, where $\Theta_t = g(f_\theta(\boldsymbol{y}_{1:t-1}))$ and $g(\cdot)$ is the function for calculating the sufficient statistics of the assumed underlying distribution from the output of the neural forecaster. Depending on our assumption of the forecast distribution, $p(\cdot)$ and $g(\cdot)$ can be analytically expressed. Note that in our case, the projection matrix only affects the mean (and variance) of the distribution. In the following experiments section, to evaluate the performance of the proposed methods, we conduct extensive experiments on both point and probabilistic hierarchical time series forecasting.

5. Experiments

5.1. Datasets

We evaluate our proposed methodology on the publicly available hierarchical datasets used in Rangapuram et al. (2021). We consider the *Labour* dataset (Australian Bureau of Statistics, 2020) that contains monthly Australian employment data from Feb. 1978 to Dec. 2020, the *Traffic* dataset (Cuturi, 2011) that contains information about the occupancy rate of car lanes in San Fransisco and the *Wiki* dataset (Ben Taieb & Koo, 2019) that includes daily views for 145,000 Wikipedia articles starting from Jul. 2015 to Dec. 2016. We also test our methodology on the *Tourism* dataset (Tourism Research Australia, 2005) that presents a geographical hierarchy with quarterly observations of Australian tourism flows from 1998 to 2006, and on the *TourismLarge* dataset that is a larger, more detailed version of *Tourism* (Wickramasuriya et al., 2019) based on geography and purpose-of-travel. For each dataset, we use the prediction length presented in Table 2. For a given prediction horizon h, we assume that the total length of the multivariate time series at hand is $T + h$ with the total length of the time series used for training being T.

5.2. Metrics

To evaluate the accuracy of the point predictions, we use the root mean squared error (RMSE) that is defined as $\sqrt{\frac{1}{nh} \cdot \sum_{\tau=1}^{h} ||\boldsymbol{y}_{T+\tau} - \tilde{\boldsymbol{y}}_{T+\tau}||_2^2}$ and the weighted mean absolute percentage error (wMAPE) that is defined as $\frac{1}{nh} \cdot \frac{\sum_{\tau=1}^{h} ||\boldsymbol{y}_{T+\tau} - \tilde{\boldsymbol{y}}_{T+\tau}||_1}{\sum_{\tau=1}^{h} ||\boldsymbol{y}_{T+\tau}||_1}$. To evaluate the accuracy of the forecast distributions, we use the total continuous ranked probability score (CRPS; Gneiting & Ranjan (2011)). Given an estimated predictive CDF $\hat{F}_t$ for the multivariate time series $\boldsymbol{y}_t$, the CDF $\hat{F}_{t,i}$ for univariate time series i, and the ground-truth observation $y_{t,i}$, the total CRPS is be defined as $\text{CRPS}_{sum}(\hat{F}_t, \boldsymbol{y}_t) = \sum_i \int_0^1 \text{QS}_q\big(\hat{F}_{t,i}^{-1}, y_{t,i}\big)\, dq$, where QS_q is the quantile score for the q-th quantile, i.e. $\text{QS}_q\big(\hat{F}_{t,i}^{-1}, y_{t,i}\big) = 2\big(\mathbb{1}\{y_{t,i} \leq \hat{F}_{t,i}^{-1}(q)\} - q\big)\big(\hat{F}_{t,i}^{-1}(q) - y_{t,i}\big)$. For the experiments, we use the total CRPS implementation from GluonTS (Alexandrov et al., 2020), with all quantiles from 0.05 to 0.95 with a step of 0.05.

5.3. Models

We test our method using two backbone models as the neural forecaster $f_{\boldsymbol{\theta}}$, i.e., the TimesNet (Wu et al., 2022) and Autoformer (Wu et al. (2021)). For point predictions, we use the Mean Squared Error (MSE) as the loss function, while for probabilistic predictions we learn the set of parameters $\Theta_t = \{\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\}$ of a Gaussian distribution, where $\boldsymbol{\mu} \in \mathbb{R}^n$ corresponds to the vector of the mean and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{n \times n}$ corresponds to the diagonal covariance matrix, using Gaussian negative log-likelihood as the loss function. We selected the TimesNet and the Autoformer models as the neural forecaster backbones due to their high prediction accuracy in a variety of datasets.

5.4. Benchmarks

We compare our approach against several benchmarks, of both sequential and end-to-end nature. We use both TimesNet and Autoformer as the backbone neural forecasters and we perform cross-validation to find the best set of parameters for each dataset. For performing cross-validation on the neural forecaster, we train on the first $T - h$ time steps and validate on the following h time steps. For TimesNet, we use the hyperparameters presented in Wu et al. (2022) for short-term forecasting and we use the default hyperparameter selection method presented in Olivares et al. (2022a). For the Autoformer model, we use again the default hyperparameter selection method presented in Olivares et al. (2022a). For both models, for the family of general oblique projection, we use a Lagrange multiplier of $\lambda = 10^4$, since this value is large enough to guarantee reconciliation. In section C of the Appendix, we present an extensive experimental study investigating the impact of different Lagrange multipliers on the learned projection matrix and show why the selected value leads to certified projection matrices. As baseline models, we consider the Naive and SeasonalNaive benchmarks (Meyer, 2002; Garza et al., 2022).

For the sequential benchmarks, we first train a TimesNet and Autoformer (AF) to generate their respective base forecasts. Then, we apply a wide range of reconciliation methods to the base forecasts. Specifically, we use the Bottom-up (BU), the Top-Down (TD; forecast pro-

Table 2. Datasets Summary

Dataset	Total	Bottom	Levels	Observations	Horizon h	Seasonality
Labour	57	32	4	228	8	12
Traffic	207	200	4	366	1	7
Wiki	199	150	5	366	1	7
Tourism	89	56	4	36	8	4
TourismLarge	555	76, 304	4, 5	228	12	4

portions disaggregation strategy), the MinT-ols, MinT-shr, MinT-var, and the ERM reconciliation methods. MinT approaches impose that $P = (S^T W_\tau^{-1} S)^{-1}(S^T W_\tau^{-1})$. In MinT-ols, the OLS estimator is used, i.e. $W_\tau = I$. In MinT-shr, W_τ is the shrinkage estimator, with $W_\tau = (1-\alpha)W_s + \alpha W_d$, with $W_s = \frac{1}{T}\sum_{t=1}^{T} \hat{\boldsymbol{\epsilon}}_{t+1}(\hat{\boldsymbol{\epsilon}}_{t+1})^T$, $W_d = diag(W_s)$ and $\alpha \in (0, 1]$. Finally, in MinT-var W_τ is the diagonal matrix of the variances of the errors (shrinkage estimator for $\alpha = 0$). ERM corresponds to the emprirical risk minimization reconciliation method with $P = \arg\min_P \frac{1}{(T-T_1-h+1)n} \sum_{t=T_1}^{T-h} ||\boldsymbol{y}_{t+h} - SP\hat{\boldsymbol{y}}_{t+h}||_2^2$, where $T - T_1 - h + 1$ is the number of observations in the validation set. For the ERM method, we use the whole training sample to calculate P (i.e. $T_1 = 1$). To implement the aforementioned reconciliation approaches we used the libraries by Garza et al. (2022) and Olivares et al. (2022b).

For the end-to-end benchmarks, we implement our proposed methods, i.e., the generalized Euclidean projection and the general oblique projection that we deonte as **Eucl** and **GenProj** respectively. We also include two additional benchmarks where we relax the idempotence constraint on P, i.e. $\lambda = 0$, denoted as **Gen** in Table 3 and the end-to-end projection presented in (Rangapuram et al., 2021) denoted as `Hier-E2E` in Table 3. Note that since **Gen** is no longer a projection method, its forecasts are not guaranteed to have the unbiasedness preserving property.

5.5. Results

For each dataset, we run 10 independent simulations. We report the mean and standard deviation of the RMSE in Table 3, while due to space limitations, we report the mean and standard deviation of wMAPE in Table 4 and of CRPS for probabilistic forecasting in Table 5 in section B of the Appendix.

We observe that in all cases, our proposed end-to-end reconciliation methodologies produce the most accurate forecasts within the same backbone model. Among both backbone neural forecasters, TimesNet performs the best in two datasets (Labour and Wiki), AF performs the best in the Traffic dataset, while in Tourism and TourismLarge datasets both models perform roughly the same. It is worth noting that the Naive and SeasonalNaive baselines exhibit good enough accuracy on the *Traffic* and *Labour* datasets.

5.6. Insights on the Resulting Projection Matrices

Experiments reveal interesting insights into the structure of the learned projection matrices, especially regarding the type of projections learned. It is known that orthogonal projections have a spectral norm of 1 and oblique projections have a spectral norm greater than 1, implying that the greater the spectral norm, the farther the projection is from being orthogonal. By inspecting the distribution of the spectral norms in our experiments (see Tables 6 & 7 of the Appendix), we observe that higher spectral norms coincide with datasets that are harder to predict, i.e., exhibit the highest RMSEs. For such datasets, our method generates projections that are far from orthogonal. Essentially, this means that the learned projections assign higher weights to individual series in the hierarchy that are harder to predict, to reduce the overall error.

We have also included visualizations of the resulting orthogonal, generalized Euclidean, and general projection matrices in section E of the Appendix. For all the datasets in our experiments, we observe that the orthogonal projection matrix is symmetric and sparse, while for the generalized Euclidean and the general projection, the matrix is not symmetric and is much denser as it contains more information concerning the weights assigned at each forecast error. Furthermore, it can be seen that all three matrix categories preserve the hierarchical structure, as the constant patterns of S are observed across all matrices.

6. Related Literature

Most recent works on hierarchical forecasting have focused on implementing end-to-end frameworks. Rangapuram et al. (2021) was the first to propose an end-to-end model that consists of a neural forecaster followed by an orthogonal projection. While Rangapuram et al. (2021) only considers probabilistic forecasts, our paper also considers point forecasts. More importantly, we propose how an oblique projection can be learned instead of a pre-specified orthogonal projection. Theodosiou & Kourentzes (2021) introduced a deep learning method to augment temporal hierarchy learning by combining the generation of the base forecasts and the

Table 3. Test RMSE for the hierarchical dataset across all models. The best RMSE achieved per dataset and per model is highlighted in **bold**, while the second-best is highlighted in *italics*. Naive and SeasonalNaive's predictions are always reconciled and thus, produce the same forecasts when traditional reconciliation methods (BU, TD, MinT) are used. As a result, we omit them due to space limitations.

Method	Labour	Traffic	Wiki	Tourism	TourismLarge
Naive	22.73	8.31	890.41	729.21	164.80
Naive-ERM	27.94	9.36	796.72	871.11	273.06
SeasonalNaive	24.52	3.41	752.01	454.61	167.69
SeasonalNaive-ERM	23.86	1.29	897.15	567.55	172.64
TimesNet-Unreconciled	$11.32_{\pm 1.53}$	$2.21_{\pm 0.30}$	$619.12_{\pm 53.52}$	$419.15_{\pm 17.59}$	$113.75_{\pm 4.33}$
TimesNet-Bottom-Up	$10.58_{\pm 1.17}$	$2.38_{\pm 0.45}$	$620.58_{\pm 62.26}$	$419.91_{\pm 19.62}$	$128.13_{\pm 6.16}$
TimesNet-Top-Down	$11.64_{\pm 1.68}$	$2.21_{\pm 0.30}$	$730.95_{\pm 52.60}$	$424.65_{\pm 17.44}$	$134.07_{\pm 1.29}$
TimesNet-MinT-ols	$11.52_{\pm 1.64}$	$2.22_{\pm 0.29}$	$645.16_{\pm 49.38}$	$421.42_{\pm 18.00}$	$112.31_{\pm 3.99}$
TimesNet-MinT-shr	$10.99_{\pm 1.44}$	$2.35_{\pm 0.44}$	$623.04_{\pm 59.30}$	$416.49_{\pm 18.99}$	$117.36_{\pm 4.87}$
TimesNet-MinT-var	$10.95_{\pm 1.42}$	$2.37_{\pm 0.40}$	$618.85_{\pm 56.63}$	$421.84_{\pm 17.84}$	$112.36_{\pm 3.73}$
TimesNet-ERM	$22.73_{\pm 3.76}$	$2.47_{\pm 0.33}$	$646.05_{\pm 37.39}$	$586.16_{\pm 24.25}$	$188.78_{\pm 5.21}$
TimesNet-Hier-E2E	$10.51_{\pm 1.42}$	$2.20_{\pm 0.37}$	$617.94_{\pm 51.42}$	$422.78_{\pm 19.81}$	$112.26_{\pm 3.76}$
TimesNet-Gen	$10.39_{\pm 1.37}$	$2.37_{\pm 0.41}$	$622.95_{\pm 46.87}$	$\mathit{415.27}_{\pm 22.56}$	$111.84_{\pm 4.95}$
TimesNet-Eucl	$\mathit{10.22}_{\pm 1.67}$	$\mathit{2.18}_{\pm 0.60}$	$\mathit{615.41}_{\pm 48.95}$	$419.56_{\pm 18.48}$	$\mathit{111.51}_{\pm 3.60}$
TimesNet-GenProj	$\mathbf{10.12}_{\pm 1.35}$	$\mathbf{2.14}_{\pm 0.39}$	$\mathbf{613.95}_{\pm 49.37}$	$\mathbf{414.33}_{\pm 19.94}$	$\mathbf{110.82}_{\pm 3.71}$
AF-Unreconciled	$27.43_{\pm 5.67}$	$1.11_{\pm 0.07}$	$643.02_{\pm 6.19}$	$425.05_{\pm 18.93}$	$129.28_{\pm 13.47}$
AF-Bottom-Up	$24.52_{\pm 0.15}$	$1.06_{\pm 0.28}$	$658.39_{\pm 4.74}$	$424.11_{\pm 19.68}$	$126.78_{\pm 12.49}$
AF-Top-Down	$30.91_{\pm 6.17}$	$1.15_{\pm 0.15}$	$642.80_{\pm 9.49}$	$432.17_{\pm 14.38}$	$130.58_{\pm 7.56}$
AF-MinT-ols	$29.74_{\pm 7.91}$	$0.82_{\pm 0.13}$	$642.38_{\pm 4.76}$	$477.12_{\pm 14.77}$	$125.17_{\pm 14.82}$
AF-MinT-shr	$25.49_{\pm 3.78}$	$0.82_{\pm 0.15}$	$645.87_{\pm 9.47}$	$433.39_{\pm 19.62}$	$127.28_{\pm 18.28}$
AF-MinT-var	$24.61_{\pm 3.82}$	$0.79_{\pm 0.25}$	$643.25_{\pm 4.77}$	$421.59_{\pm 17.77}$	$122.33_{\pm 12.98}$
AF-ERM	$29.14_{\pm 0.28}$	$1.07_{\pm 0.14}$	$686.63_{\pm 0.13}$	$420.49_{\pm 12.48}$	$165.52_{\pm 12.07}$
AF-Hier-E2E	$24.94_{\pm 4.78}$	$0.75_{\pm 0.19}$	$645.63_{\pm 11.24}$	$422.02_{\pm 20.48}$	$121.59_{\pm 17.77}$
AF-Gen	$25.63_{\pm 5.52}$	$\mathit{0.73}_{\pm 0.18}$	$642.68_{\pm 13.94}$	$419.36_{\pm 15.65}$	$119.17_{\pm 19.82}$
AF-Eucl	$\mathit{22.81}_{\pm 3.78}$	$\mathbf{0.71}_{\pm 0.11}$	$\mathbf{640.63}_{\pm 15.52}$	$\mathit{415.37}_{\pm 14.78}$	$\mathit{117.12}_{\pm 14.77}$
AF-GenProj	$\mathbf{21.96}_{\pm 2.38}$	$\mathbf{0.71}_{\pm 0.24}$	$\mathit{641.63}_{\pm 16.37}$	$\mathbf{413.18}_{\pm 17.04}$	$\mathbf{115.17}_{\pm 14.38}$

reconciliation in an end-to-end method. The paper does not focus on the projection but instead proposes two approaches to perform the reconciliation step. Das et al. (2023) proposed an end-to-end neural forecaster model, that follows the principle of the classical top-down reconciliations strategy, and learns the distribution of the root time series, and the proportions according to which each parent time series are split. Olivares et al. (2023) proposed a model that combines neural networks and a statistical model for learning the joint distribution of the hierarchical multivariate time-series structure. Even though in this setting, model parameters are learned with respect to the reconciliation method that follows, the reconciliation is either not guaranteed (Theodosiou & Kourentzes, 2021), pre-defined (Rangapuram et al., 2021; Olivares et al., 2023) or does not come with important theoretical guarantees (e.g. preserving unbiasedness) (Das et al., 2023).

7. Conclusion

In this work, we propose a novel method for learning the optimal reconciliation step from data. In contrast to existing state-of-the-art methods, which employ a pre-defined reconciliation step to the coherent subspace, our proposed approach learns the optimal projection during training. This is achieved by modeling the projections as a learnable, structured, projection layer in the neural forecaster architecture used for the hierarchical time series prediction. In this framework, we utilize two broad classes of oblique projections; the generalized Euclidean and the general projection. Our proposed approach effectively addresses the challenge of weighing forecast errors of individual time series differently according to the hierarchy. We evaluate our proposed approaches by conducting extensive experiments on real-world hierarchical datasets, where we demonstrate the superior performance of our approach compared to state-of-the-art reconciliation approaches both in point and probabilistic forecasting.

Impact Statement

This paper presents work whose goal is to advance the field of hierarchical time series forecasting. In this work, we propose a novel method for learning the optimal reconciliation step as an oblique projection from data for coherent forecasting of hierarchical time series. Our proposed approach is efficient, and scalable and can be applied to any neural forecaster. Since what we propose is a fundamental machine learning and optimization methodology, we do not anticipate any direct negative impact on society resulting from the proposed methods.

References

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 and neural time series modeling in python. *The Journal of Machine Learning Research*, 21 (1):4629–4634, 2020.

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, 2020. URL https://www.abs.gov.au/statistics/labour/employment-and-unemployment/labour-force-australia/latest-release.

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.

Cuturi, M. Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 929–936, 2011.

Dai, X., Fu, R., Lin, Y., Li, L., and Wang, F.-Y. Deeptrend: A deep hierarchical neural network for traffic flow prediction. *arXiv preprint arXiv:1707.03213*, 2017.

Das, A., Kong, W., Paria, B., and Sen, R. Dirichlet proportions model for hierarchically coherent probabilistic forecasting. In *Uncertainty in Artificial Intelligence*, pp. 518–528. PMLR, 2023.

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

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.

Gross, C. W. and Sohl, J. E. Disaggregation methods to expedite product line forecasting. *Journal of forecasting*, 9(3):233–254, 1990.

Hollyman, R., Petropoulos, F., and Tipping, M. E. Understanding forecast reconciliation. *European Journal of Operational Research*, 294(1):149–160, 2021.

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.

Kahn, K. B. Revisiting top-down versus bottom-up forecasting. *The Journal of Business Forecasting*, 17(2):14, 1998.

Lezcano-Casado, M. Trivializations for gradient-based optimization on manifolds. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 9154–9164, 2019.

Meyer, D. Naive time series forecasting methods. *R news*, 2(2):7–10, 2002.

Olivares, K. G., Challú, C., Garza, F., Canseco, M. M., and Dubrawski, A. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022a. URL https://github.com/Nixtla/neuralforecast.

Olivares, K. G., Garza, F., Luo, D., Challú, C., Mergenthaler, M., Taieb, S. B., Wickramasuriya, S. L., and Dubrawski, A. HierarchicalForecast: A reference framework for hierarchical forecasting in python. *Work in progress paper, submitted to Journal of Machine Learning Research.*, abs/2207.03517, 2022b. URL https://arxiv.org/abs/2207.03517.

Olivares, K. G., Meetei, O. N., Ma, R., Reddy, R., Cao, M., and Dicker, L. Probabilistic hierarchical forecasting with deep poisson mixtures. *International Journal of Forecasting*, 2023.

Panagiotelis, A., Athanasopoulos, G., Gamakumara, P., and Hyndman, R. J. Forecast reconciliation: A geometric view with new insights on bias correction. *International Journal of Forecasting*, 37(1):343–359, 2021.

Rangapuram, S. S., Werner, L. D., Benidis, K., Mercado, P., Gasthaus, J., and Januschowski, T. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pp. 8832–8843. PMLR, 2021.

Rao, C. R. Projectors, generalized inverses and the blue's. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(3):442–448, 1974.

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. W., Salinas, D., and Flunkert, V. Bayesian intermittent demand forecasting for large inventories. *Advances in Neural Information Processing Systems*, 29, 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. Coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pp. 3348–3357. PMLR, 2017.

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*, 116(533):27–43, 2021.

Theodosiou, F. and Kourentzes, N. Forecasting with deep temporal hierarchies. *Available at SSRN 3918315*, 2021.

Tourism Research Australia. Tourism Australia, 2005. URL https://robjhyndman.com/publications/hierarchicaltourism/.

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.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.

A. Proofs of Propositions

Proposition 3.2 *For an unbiased base forecast $\hat{\boldsymbol{y}}_\tau$, the reconciled point forecast produced by the proposed oblique projections is also an unbiased prediction.*

Proof. **Step 1**: We first show that SP is a projection matrix onto $\mathcal{S}$ for both our methods. Showing that $SPS = S$ implies that SP is a projection matrix onto $\mathcal{S}$. We study both cases.

1. For the generalized Euclidean projection, we have that:

$$SPS = S(S^TWS)^{-1}(S^TW)S = S(S^TWS)^{-1}(S^TWS) = SI = S. \tag{4}$$

2. For the general projection, given that $PS = I$, we have that:

$$SPS = S(PS) = SI = S. \tag{5}$$

Therefore, both of our methods model a projection matrix onto $\mathcal{S}$.

Step 2: We show the main result.

$$\mathbb{E}[\tilde{\boldsymbol{y}}_\tau] = \mathbb{E}[SP\hat{\boldsymbol{y}}_\tau] = SP\mathbb{E}[\hat{\boldsymbol{y}}_\tau] = SP\boldsymbol{\mu}_\tau, \tag{6}$$

where $\boldsymbol{\mu}_\tau := \mathbb{E}[\boldsymbol{y}_\tau] = E[\boldsymbol{y}_{T+\tau}|\boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$, where the expectation is taken over the predictive density (i.e., the distribution that y_t follows), and $E[\hat{\boldsymbol{y}}_\tau] = E[\hat{\boldsymbol{y}}_{T+\tau}]$ since the base forecasts are unbiased. Given that $\boldsymbol{\mu}_\tau$ is the expectation taken over the predictive density, we have that $\boldsymbol{\mu}_\tau \in \mathcal{S}$. Since SP is an oblique projection onto $\mathcal{S}$, then it maps $\boldsymbol{\mu}_\tau$ onto $\mathcal{S}$, as shown in step 1. Therefore, $SP\boldsymbol{\mu}_\tau = \boldsymbol{\mu}_\tau$, which concludes the proof. □

Proposition 3.3 *The generalized Euclidean projection assigns different weights to different forecasts, i.e. transforms the space, and then applies an orthogonal projection to the weighted forecasts.*

Proof. This can be seen by pre-multiplying all vectors and matrices involved with $W^{1/2}$, i.e. $\hat{\boldsymbol{y}}_W := W^{1/2}\hat{\boldsymbol{y}}$, $\tilde{\boldsymbol{y}}_W := W^{1/2}\tilde{\boldsymbol{y}}$ and $S_W = W^{1/2}S$, since

$$\begin{aligned}\tilde{\boldsymbol{y}}_W &= W^{1/2}\tilde{\boldsymbol{y}} = W^{1/2}S(S^TWS)^{-1}S^TW\hat{\boldsymbol{y}} = \\ &= W^{1/2}S((W^{1/2}S)^TW^{1/2}S)^{-1}(W^{1/2}S)^TW^{1/2}\hat{\boldsymbol{y}} = \\ &= S_W(S_W^TS_W)^{-1}S_W^T\hat{\boldsymbol{y}}_W.\end{aligned} \tag{7}$$

□

Proposition 3.4 *The generalized Euclidean projection never increases the error of the reconciled forecast, with respect to the norm defined by W.*

Proof. Let P be such that SP is a generalized Euclidean projection onto $\mathcal{S}$ (i.e. $P = (S^TWS)^{-1}(S^TW)$, with W symmetric and positive definite). Then, given that that $\tilde{\boldsymbol{y}}_\tau = SP\hat{\boldsymbol{y}}_\tau \in \mathcal{S}$, we have that $||\boldsymbol{y}_\tau - \hat{\boldsymbol{y}}_\tau||_W^2 = ||\boldsymbol{y}_\tau - \tilde{\boldsymbol{y}}_\tau||_W^2 + ||\tilde{\boldsymbol{y}}_\tau - \hat{\boldsymbol{y}}_\tau||_W^2$. Therefore, since $||\tilde{\boldsymbol{y}}_\tau - \hat{\boldsymbol{y}}_\tau||_W^2 \geq 0$, we obtain that $||\boldsymbol{y}_\tau - \hat{\boldsymbol{y}}_\tau||_W \leq ||\boldsymbol{y}_\tau - \tilde{\boldsymbol{y}}_\tau||_W$. □

B. Additional Results

We report the mean and standard deviation of wMAPE and CRPS in Tables 4 and 5. The results presented in the experiments section remain consistent across different metrics.

Table 4. Test wMAPE for the hierarchical dataset across all models. The best wMAPE achieved per dataset and per model is highlighted in **bold**, while the second-best is highlighted in *italics*. Naive and SeasonalNaive's predictions are always reconciled and thus, produce the same forecasts when traditional reconciliation methods (BU, TD, MinT) are used. As a result, we omit them due to space limitations.

Method	Labour	Traffic	Wiki	Tourism	TourismLarge
`Naive`	2.34	31.76	24.14	18.65	29.78
`Naive-ERM`	2.86	35.80	28.89	24.77	64.79
`SeasonalNaive`	2.52	13.04	26.29	15.23	28.63
`SeasonalNaive-ERM`	2.61	11.94	30.88	18.93	32.41
`TimesNet-Unreconciled`	$1.07_{\pm 0.17}$	$8.46_{\pm 1.14}$	$22.82_{\pm 4.01}$	*10.90*$_{\pm 0.48}$	$22.71_{\pm 0.81}$
`TimesNet-Bottom-Up`	$0.99_{\pm 0.12}$	$9.10_{\pm 1.72}$	$23.25_{\pm 4.66}$	$11.10_{\pm 0.56}$	$26.64_{\pm 1.35}$
`TimesNet-Top-Down`	$1.11_{\pm 0.19}$	$8.47_{\pm 1.16}$	$27.38_{\pm 7.69}$	$11.03_{\pm 0.48}$	$28.15_{\pm 0.20}$
`TimesNet-MinT-ols`	$1.09_{\pm 0.18}$	$8.48_{\pm 1.12}$	$24.17_{\pm 3.70}$	$11.00_{\pm 0.48}$	$22.28_{\pm 0.70}$
`TimesNet-MinT-shr`	$1.04_{\pm 0.16}$	$8.98_{\pm 1.68}$	$22.59_{\pm 4.44}$	$10.96_{\pm 0.56}$	$23.87_{\pm 1.00}$
`TimesNet-MinT-var`	$1.12_{\pm 0.24}$	$8.67_{\pm 1.41}$	$23.49_{\pm 4.05}$	*10.90*$_{\pm 0.52}$	$26.83_{\pm 1.23}$
`TimesNet-ERM`	$1.42_{\pm 0.18}$	$8.60_{\pm 1.27}$	$26.13_{\pm 4.93}$	$11.40_{\pm 0.59}$	$24.56_{\pm 1.19}$
`TimesNet-Hier-E2E`	$1.01_{\pm 0.17}$	$8.48_{\pm 1.74}$	$22.64_{\pm 4.86}$	$10.94_{\pm 0.56}$	$22.20_{\pm 1.24}$
TimesNet-Gen	$0.98_{\pm 0.15}$	$8.56_{\pm 1.82}$	$23.16_{\pm 4.97}$	*10.90*$_{\pm 0.52}$	*22.04*$_{\pm 1.34}$
TimesNet-Eucl	*0.97*$_{\pm 0.16}$	*8.42*$_{\pm 1.54}$	*22.46*$_{\pm 4.62}$	*10.90*$_{\pm 0.50}$	*22.04*$_{\pm 1.38}$
TimesNet-GenProj	**0.95**$_{\pm 0.18}$	**8.40**$_{\pm 1.48}$	**22.44**$_{\pm 4.54}$	**10.88**$_{\pm 0.54}$	**22.02**$_{\pm 1.26}$
`AF-Unreconciled`	$2.50_{\pm 0.63}$	$6.06_{\pm 0.26}$	$22.56_{\pm 0.22}$	$12.21_{\pm 1.09}$	$24.45_{\pm 4.05}$
`AF-Bottom-Up`	$2.36_{\pm 0.03}$	$6.01_{\pm 1.08}$	$23.02_{\pm 0.17}$	$12.26_{\pm 0.73}$	$23.92_{\pm 4.05}$
`AF-Top-Down`	$3.06_{\pm 0.04}$	$6.64_{\pm 0.53}$	$22.40_{\pm 0.33}$	$12.73_{\pm 1.61}$	$23.82_{\pm 4.05}$
`AF-MinT-ols`	$2.42_{\pm 0.50}$	$6.11_{\pm 0.48}$	$22.42_{\pm 0.19}$	$12.29_{\pm 0.76}$	$24.54_{\pm 4.05}$
`AF-MinT-shr`	$2.40_{\pm 0.43}$	$6.22_{\pm 0.57}$	$22.58_{\pm 0.33}$	$12.29_{\pm 1.37}$	$23.26_{\pm 4.05}$
`AF-MinT-var`	$2.35_{\pm 0.56}$	$6.92_{\pm 0.98}$	$22.49_{\pm 0.26}$	$12.86_{\pm 0.92}$	$24.08_{\pm 4.05}$
`AF-ERM`	$2.77_{\pm 0.54}$	$6.94_{\pm 0.13}$	$23.51_{\pm 0.23}$	$12.23_{\pm 1.43}$	$25.06_{\pm 1.78}$
`AF-Hier-E2E`	$2.18_{\pm 0.32}$	$6.17_{\pm 0.45}$	$22.38_{\pm 0.64}$	$12.27_{\pm 1.52}$	$23.78_{\pm 2.34}$
AF-Gen	$2.11_{\pm 0.95}$	$6.09_{\pm 0.38}$	$22.48_{\pm 0.32}$	$12.22_{\pm 1.08}$	$23.22_{\pm 2.67}$
AF-Eucl	*2.05*$_{\pm 0.77}$	*5.99*$_{\pm 0.54}$	*22.35*$_{\pm 0.81}$	*12.16*$_{\pm 1.42}$	*23.18*$_{\pm 3.41}$
AF-GenProj	**1.97**$_{\pm 0.12}$	**5.96**$_{\pm 0.34}$	**22.26**$_{\pm 0.76}$	**12.13**$_{\pm 1.29}$	**23.04**$_{\pm 2.14}$

Table 5. Test CRPS for the hierarchical dataset across all models. The best CRPS achieved per dataset and per model is highlighted in **bold**, while the second-best is highlighted in *italics*. Naive and SeasonalNaive's predictions are always reconciled and thus, produce the same forecasts when traditional reconciliation methods (BU, TD, MinT) are used. As a result, we omit them due to space limitations.

Method	Labour	Traffic	Wiki	Tourism	TourismLarge
Naive	0.01	0.32	0.24	0.19	0.30
Naive-ERM	0.02	0.36	0.29	0.25	0.65
SeasonalNaive	0.02	0.13	0.26	0.15	0.29
SeasonalNaive-ERM	0.03	0.12	0.31	0.19	0.32
TimesNet-Unreconciled	$0.01_{\pm 0.00}$	$0.10_{\pm 0.03}$	$0.25_{\pm 0.10}$	$0.12_{\pm 0.00}$	$0.24_{\pm 0.01}$
TimesNet-Bottom-Up	$0.01_{\pm 0.00}$	$0.10_{\pm 0.02}$	$0.31_{\pm 0.14}$	$0.13_{\pm 0.01}$	$0.23_{\pm 0.02}$
TimesNet-Top-Down	$0.01_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.29_{\pm 0.18}$	$0.12_{\pm 0.00}$	$0.24_{\pm 0.01}$
TimesNet-MinT-ols	$0.01_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.25_{\pm 0.09}$	$0.12_{\pm 0.00}$	$0.23_{\pm 0.01}$
TimesNet-MinT-shr	$0.01_{\pm 0.00}$	$0.10_{\pm 0.02}$	$0.26_{\pm 0.11}$	$0.13_{\pm 0.00}$	$0.24_{\pm 0.01}$
TimesNet-MinT-var	$0.01_{\pm 0.00}$	$0.10_{\pm 0.02}$	$0.25_{\pm 0.10}$	$0.13_{\pm 0.00}$	$0.22_{\pm 0.01}$
TimesNet-ERM	$0.01_{\pm 0.00}$	$\mathit{0.08}_{\pm 0.04}$	$\mathit{0.20}_{\pm 0.02}$	$0.21_{\pm 0.01}$	$0.24_{\pm 0.02}$
TimesNet-Hier-E2E	$0.01_{\pm 0.00}$	$\mathit{0.08}_{\pm 0.02}$	$\mathit{0.20}_{\pm 0.02}$	$\mathit{0.11}_{\pm 0.00}$	$0.22_{\pm 0.01}$
TimesNet-Gen	$0.11_{\pm 0.00}$	$\mathit{0.08}_{\pm 0.02}$	$\mathit{0.20}_{\pm 0.02}$	$\mathbf{0.10}_{\pm 0.02}$	$\mathit{0.20}_{\pm 0.01}$
TimesNet-Eucl	$0.01_{\pm 0.00}$	$\mathbf{0.07}_{\pm 0.03}$	$\mathbf{0.18}_{\pm 0.03}$	$\mathit{0.11}_{\pm 0.01}$	$\mathit{0.20}_{\pm 0.01}$
TimesNet-GenProj	$0.01_{\pm 0.00}$	$\mathbf{0.07}_{\pm 0.03}$	$\mathbf{0.18}_{\pm 0.02}$	$\mathbf{0.10}_{\pm 0.01}$	$\mathbf{0.18}_{\pm 0.00}$
AF-Unreconciled	$0.04_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.20_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-Bottom-Up	$0.05_{\pm 0.01}$	$0.21_{\pm 0.02}$	$0.26_{\pm 0.02}$	$0.25_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-Top-Down	$0.04_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.17_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-MinT-ols	$0.03_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-MinT-shr	$0.04_{\pm 0.01}$	$0.22_{\pm 0.02}$	$0.37_{\pm 0.03}$	$0.22_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-MinT-var	$0.04_{\pm 0.01}$	$0.23_{\pm 0.02}$	$0.37_{\pm 0.03}$	$0.22_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-ERM	$\mathit{0.02}_{\pm 0.00}$	$0.16_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.23_{\pm 0.01}$
AF-Hier-E2E	$\mathit{0.02}_{\pm 0.00}$	$\mathit{0.10}_{\pm 0.01}$	$0.25_{\pm 0.01}$	$0.18_{\pm 0.01}$	$\mathit{0.21}_{\pm 0.01}$
AF-Gen	$\mathit{0.02}_{\pm 0.00}$	$0.12_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.18_{\pm 0.01}$	$\mathbf{0.19}_{\pm 0.01}$
AF-Eucl	$\mathbf{0.01}_{\pm 0.00}$	$0.11_{\pm 0.03}$	$\mathit{0.19}_{\pm 0.01}$	$\mathit{0.15}_{\pm 0.01}$	$\mathbf{0.19}_{\pm 0.01}$
AF-GenProj	$\mathbf{0.01}_{\pm 0.00}$	$\mathbf{0.08}_{\pm 0.01}$	$\mathbf{0.18}_{\pm 0.01}$	$\mathbf{0.14}_{\pm 0.01}$	$\mathbf{0.19}_{\pm 0.01}$

C. Selection of the Lagrangian multiplier

In the experiments, we use $\lambda = 10^4$, a value large enough to guarantee reconciliation. In this section, we repeat the experiments for different values of λ ($10^2, 10^3, 10^4$) and we provide the spectral norm and the reconciliation error $||PS-I||_F$ to verify that the reconciliation indeed holds. We report the results in Table 6.

Table 6. We report the (spectral norm, reconciliation error) for various values of λ for the general projection case using both the AF model. All experiments were executed 10 times and we report the average value and the standard deviation.

Method	Labour	Traffic	Wiki	Tourism	TourismLarge
AF-GenProj $\lambda = 10^2$	$(3.02 \pm 0.52, 0.08 \pm 0.01)$	$(9.86 \pm 0.42, 0.22 \pm 0.02)$	$(10.68 \pm 1.39, 0.18 \pm 0.02)$	$(6.02 \pm 1.46, 0.10 \pm 0.10)$	$(10.54 \pm 1.98, 0.27 \pm 0.01)$
AF-GenProj $\lambda = 10^3$	$(3.24 \pm 0.32, 0)$	$(10.82 \pm 0.32, 0)$	$(9.02 \pm 0.48, 0)$	$(5.38 \pm 0.76, 0)$	$(11.04 \pm 0.57, 0)$
AF-GenProj $\lambda = 10^4$	$(3.46 \pm 0.24, 0)$	$(10.61 \pm 0.27, 0)$	$(8.08 \pm 0.37, 0)$	$(4.97 \pm 0.38, 0)$	$(12.56 \pm 0.25, 0)$

We observe that for $\lambda = 10^3$ and $\lambda = 10^4$ the reconciliation error is 0 (in practice less than 10^{-10}) and therefore, the resulting matrices SP are indeed projection matrices onto S. On the other hand, for $\lambda = 10^2$ we observe a small reconciliation error (magnitude of 10^{-1} and 10^{-2}). In this case, the reconciliation is not verified (even though it is close to 0).

D. On the Spectral Norm of the Generalized Euclidean Projection

To understand the kind of projection matrices obtained we report the spectral norm of each general projection matrix SP in Table 6. We observe that the resulting projection matrices are far from orthogonal as their spectral norm is always higher than 1. Furthermore, we also report in Table 7 the spectral norm for the generalized Euclidean projection using both the DeepVar model and the transformer-based model as the backbone in order to understand how far the resulting projections differ from the typical Euclidean which has spectral norm equal to 1.

Table 7. We report the spectral norm for the generalized Euclidean projection case using the AF backbone model. All experiments were executed 10 times and we report the average value and the standard deviation.

Method	Labour	Traffic	Wiki	Tourism	TourismLarge
AF-Eucl	21.44 ± 6.76	16.21 ± 7.28	13.85 ± 4.78	19.85 ± 6.72	22.85 ± 9.32

We observe that in all cases the spectral norms are greater than 1 (especially for the transformer-based model) and therefore, we can infer that the resulting projection is far from orthogonal.

E. Visualization of the Resulting Projection Matrices

In Tables 8, we present visualizations of the resulting orthogonal, generalized Euclidean, and general projection. For all the datasets in our experiments, we observe that the orthogonal projection matrix is symmetric and very sparse, while for the generalized Euclidean and the general projection, the matrix is not symmetric and less sparse as it contains more information concerning the weights assigned at each forecast error. Furthermore, it can be seen that all three matrix categories preserve the hierarchical structure, as constant patterns are observed across all matrices.

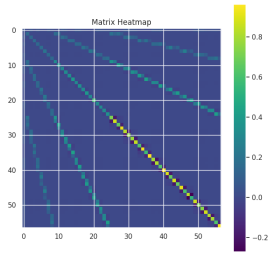
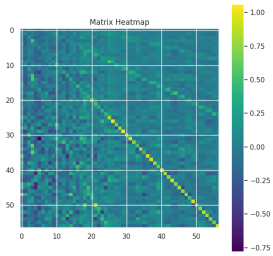
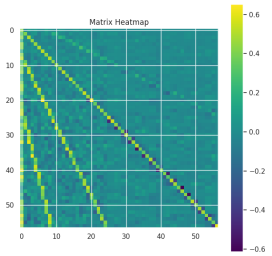
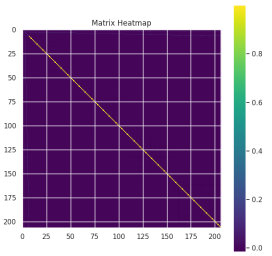
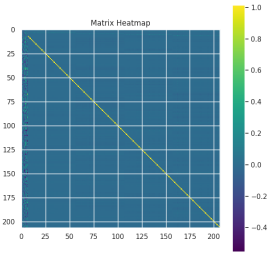
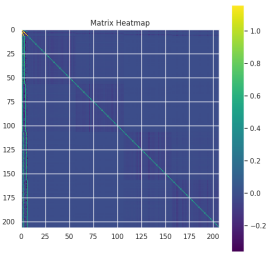
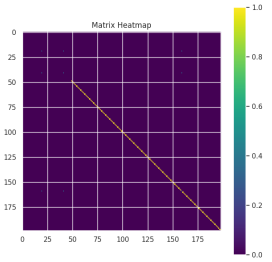
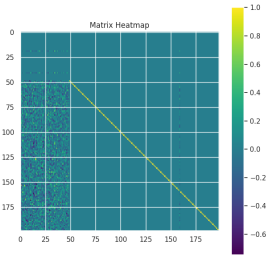
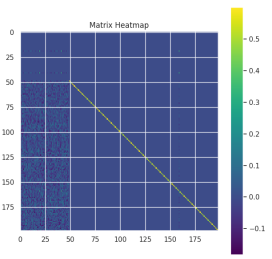
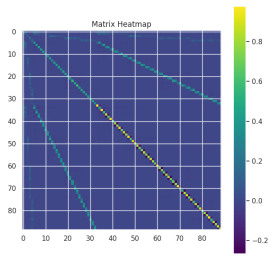
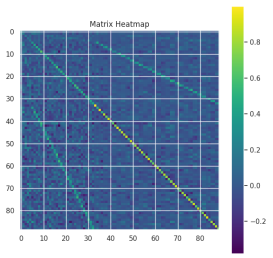
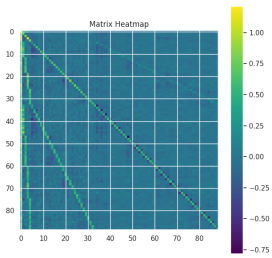
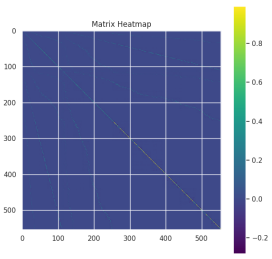
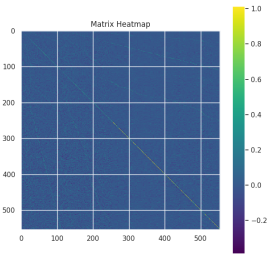
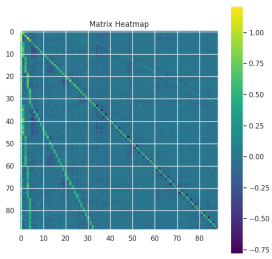
	Orthogonal	Generalized Euclidean	General Projection
Labour			
Traffic			
Wiki			
Tourism			
TourismL			

Table 8. Visualizations of the resulting orthogonal, generalized Euclidean, and general projection matrices