



Efficient probabilistic reconciliation of forecasts for real-valued and count time series

Lorenzo Zambon^{1,2} · Dario Azzimonti¹ · Giorgio Corani¹

Received: 8 November 2022 / Accepted: 12 October 2023
© The Author(s) 2023

Abstract

Hierarchical time series are common in several applied fields. The forecasts for these time series are required to be coherent, that is, to satisfy the constraints given by the hierarchy. The most popular technique to enforce coherence is called reconciliation, which adjusts the base forecasts computed for each time series. However, recent works on probabilistic reconciliation present several limitations. In this paper, we propose a new approach based on conditioning to reconcile any type of forecast distribution. We then introduce a new algorithm, called Bottom-Up Importance Sampling, to efficiently sample from the reconciled distribution. It can be used for any base forecast distribution: discrete, continuous, or in the form of samples, providing a major speedup compared to the current methods. Experiments on several temporal hierarchies show a significant improvement over base probabilistic forecasts.

Keywords Forecast reconciliation · Probabilistic reconciliation · Temporal hierarchies · Importance sampling

1 Introduction

Often time series are organized into a hierarchy. For example, the total visitors of a country can be divided into regions and the visitors of each region can be further divided into sub-regions. Such data structures are referred to as *hierarchical time series*; they are common in fields such as retail sales (Makridakis et al. 2021) and energy modelling (Taieb et al. 2021).

The forecasts for hierarchical time series should respect some summing constraints, in which case they are referred

to as *coherent*. For instance, the sum of the forecasts for the sub-regions should match the forecast for the entire region. However, the forecasts independently produced for each time series (*base forecasts*) are generally *incoherent*.

Reconciliation algorithms (Hyndman et al. 2011; Wickramasuriya et al. 2019) adjust the incoherent base forecasts, making them coherent. Reconciled forecasts are generally more accurate than base forecasts: indeed, forecast reconciliation is a special case of forecast combination (Hollyman et al. 2021). An important application of reconciliation algorithms is constituted by temporal hierarchies (Athanasopoulos et al. 2017; Kourentzes and Athanasopoulos 2021), which make coherent the forecasts produced for the same time series at different temporal scales.

Most reconciliation algorithms (Hyndman et al. 2011; Wickramasuriya et al. 2019, 2020; Di Fonzo and Girolimetto 2021, 2022) provide only *reconciled point forecasts*. It is however clear (Kolassa 2023) that *reconciled predictive distributions* are needed for decision making.

Probabilistic reconciliation has been addressed only recently; earlier attempts (Jeon et al. 2019; Taieb et al. 2021), though experimentally effective, lacked a strong formal justification. For the case of Gaussian base forecasts, Corani et al. (2020) obtains the reconciled distribution in analytical form introducing the approach of *reconciliation via conditioning*. Panagiotelis et al. (2023) provides a frame-

✉ Lorenzo Zambon
lorenzo.zambon@idsia.ch

Dario Azzimonti
dario.azzimonti@idsia.ch

Giorgio Corani
giorgio.corani@idsia.ch

¹ Istituto Dalle Molle di studi sull'intelligenza artificiale (IDSIA USI-SUPSI), Scuola universitaria professionale della Svizzera italiana, 6962 Lugano, Switzerland

² Department of Mathematics, University of Pavia, 27100 Pavia, Italy

work for probabilistic reconciliation via projection. However this approach cannot reconcile discrete distributions. Corani et al. (2023) performs probabilistic reconciliation via conditioning of count time series by adopting the concept of *virtual evidence* (Pearl 1988). However its implementation in probabilistic programming, based on Markov Chain Monte Carlo (MCMC), is too slow on large hierarchies; moreover it requires the base forecast distribution to be in parametric form.

The main contribution of this paper is the Bottom-Up Importance Sampling (BUIIS) algorithm, which samples from the reconciled distribution obtained via conditioning with a substantial speedup with respect to Corani et al. (2023). BUIIS can be used even when the base forecast distribution is only available through samples. This is the case of forecasts returned by models for time series of counts (Liboschik et al. 2017) or based on deep learning (Salinas et al. 2020). We prove the convergence of BUIIS to the actual reconciled distribution. An implementation of the algorithm in the R language is available in the R package *bayesRecon* (Azzimonti et al. 2023).

We provide two further formal contributions. The first is a definition of coherence for probabilistic forecasts that applies to both discrete and continuous distributions. The second is a novel interpretation of the reconciliation via conditioning, in which the base forecast distribution is conditioned on the hierarchy constraints. This allows for a unified treatment of the reconciliation of discrete and continuous forecast distributions. We test our method exhaustively on temporal hierarchies reporting positive results both for the accuracy and the efficiency of our method.

The paper is organized as follows. In Sect. 2, we introduce the notation and the reconciliation of point forecasts. In Sect. 3, we introduce our approach to reconciliation via conditioning and we compare it to the existing literature. In Sect. 4, we introduce the Bottom-Up Importance Sampling algorithm. We empirically verify its correctness in Sect. 5, while in Sect. 6 we test it on different data sets. We present the conclusions in Sect. 7.

2 Notation

Consider the hierarchy of Fig. 1. We denote by $\mathbf{b} = [b_1, \dots, b_{n_b}]^T$ the vector of bottom variables, and by $\mathbf{u} = [u_1, \dots, u_{n_u}]^T$ the vector of upper variables. We then denote by

$$\mathbf{y} = \begin{bmatrix} \mathbf{u} \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^n$$

the vector of all the variables. The hierarchy can be expressed as a set of linear constraints:

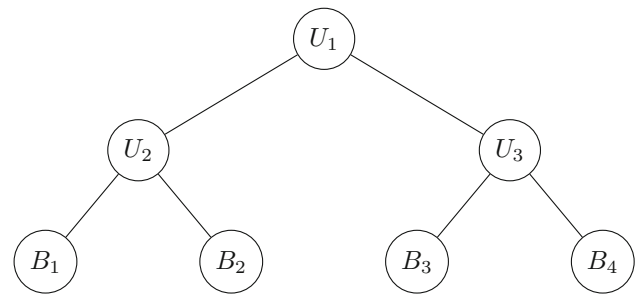


Fig. 1 A hierarchy with 4 bottom and 3 upper variables

$$\mathbf{y} = \mathbf{S}\mathbf{b}, \text{ where } \mathbf{S} = \begin{bmatrix} \mathbf{A} \\ \mathbf{I} \end{bmatrix}. \tag{1}$$

$\mathbf{I} \in \mathbb{R}^{n_b \times n_b}$ is the *identity matrix*; we refer to $\mathbf{S} \in \mathbb{R}^{n \times n_b}$ as the *summing matrix* and to $\mathbf{A} \in \mathbb{R}^{n_u \times n_b}$ as the *aggregating matrix*. We can thus write the constraints as $\mathbf{u} = \mathbf{A}\mathbf{b}$. For example, the aggregating matrix of the hierarchy in Fig. 1 is:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

A point $\mathbf{y} \in \mathbb{R}^n$ is *coherent* if it satisfies the constraints given by the hierarchy. We denote by \mathcal{S} the set of coherent points, which is a linear subspace of \mathbb{R}^n :

$$\mathcal{S} := \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} = \mathbf{S}\mathbf{b}\}. \tag{2}$$

2.1 Temporal hierarchies

In temporal hierarchies (Athanasopoulos et al. 2017; Kourentzes and Athanasopoulos 2021), forecasts are generated for the same time series at different temporal scales. For instance, a quarterly time series can be aggregated to the semi-annual and the annual scale. If we are interested in predictions up to one year ahead, we compute four quarterly forecasts $\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4$, two semi-annual forecasts \hat{s}_1, \hat{s}_2 , and an annual forecast \hat{a}_1 . We then obtain the hierarchy in Fig. 1. The base point forecasts, independently computed at each frequency, are $\hat{\mathbf{b}} = [\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4]^T$ and $\hat{\mathbf{u}} = [\hat{a}_1, \hat{s}_1, \hat{s}_2]^T$.

2.2 Point forecasts reconciliation

Let us denote by $\hat{\mathbf{y}} = [\hat{\mathbf{u}}^T | \hat{\mathbf{b}}^T]^T$ the vector of the base (incoherent) forecasts. Note that, for ease of notation, we drop the time subscript. Point reconciliation is generally performed in two steps (Hyndman et al. 2011; Wickramasuriya et al. 2019). First, the reconciled bottom forecasts are computed

by linearly combining the base forecasts of the entire hierarchy:

$$\tilde{\mathbf{b}} = \mathbf{G}\hat{\mathbf{y}},$$

for some matrix $\mathbf{G} \in \mathbb{R}^{m \times n}$. Then, the reconciled forecasts for the whole hierarchy are given by:

$$\tilde{\mathbf{y}} = \mathbf{S}\tilde{\mathbf{b}}.$$

The state-of-the-art reconciliation method is MinT (Wickramasuriya et al. 2019), which defines \mathbf{G} as:

$$\mathbf{G} = (\mathbf{S}^T \mathbf{W}^{-1} \mathbf{S})^{-1} \mathbf{S}^T \mathbf{W}^{-1},$$

where \mathbf{W} is the covariance matrix of the errors of the base forecasts. This method minimizes the expected sum of the squared errors of the reconciled forecasts, under the assumption of unbiased base forecasts.

2.3 Probabilistic framework

Probabilistic reconciliation requires a probabilistic framework, in which forecasts are in the form of probability distributions. We denote by $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$ the forecast distribution for \mathbf{y} , where $\mathcal{P}(\mathbb{R}^n)$ is the space of probability measures on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$, and $\mathcal{B}(\mathbb{R}^n)$ is the Borel σ -algebra on \mathbb{R}^n . Moreover, we denote by $\hat{\nu}_u$ and $\hat{\nu}_b$ the marginal distributions of, respectively, the forecasts for the upper and the bottom components of \mathbf{y} .

The forecast distribution $\hat{\nu}$ may be either discrete or absolutely continuous. In the following, if there is no ambiguity, we will use $\hat{\pi}$ to denote either its probability mass function, in the former case, or its density, in the latter. Therefore, if $\hat{\nu}$ is discrete, we have

$$\hat{\nu}(F) = \sum_{x \in F} \hat{\pi}(x),$$

for any $F \in \mathcal{B}(\mathbb{R}^n)$. Note that the sum is well-defined as $\hat{\pi}(x) > 0$ for at most countably many x 's. On the contrary, if $\hat{\nu}$ is absolutely continuous, for any $F \in \mathcal{B}(\mathbb{R}^n)$ we have

$$\hat{\nu}(F) = \int_F \hat{\pi}(x) dx.$$

3 Probabilistic Reconciliation

We now discuss coherence in the probabilistic framework and our approach to probabilistic reconciliation.

Recall that a point forecast is incoherent if it does not belong to the set \mathcal{S} , defined as in (2). Let $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$ be a forecast distribution. Intuitively, $\hat{\nu}$ is incoherent if there

exists a set T of incoherent points, i.e. $T \cap \mathcal{S} = \emptyset$, such that $\hat{\nu}(T) > 0$. Or, equivalently, if $\text{supp}(\hat{\nu}) \not\subseteq \mathcal{S}$. We now define the summing map $s : \mathbb{R}^{n_b} \rightarrow \mathbb{R}^n$ as

$$s(\mathbf{b}) = \mathbf{S}\mathbf{b}. \tag{3}$$

The image of s is given by \mathcal{S} . Moreover, from (3) and (1), s is injective. Hence, s is a bijective map between \mathbb{R}^{n_b} and \mathcal{S} , with inverse given by $s^{-1}(\mathbf{y}) = \mathbf{b}$, where $\mathbf{y} = (\mathbf{u}, \mathbf{b}) \in \mathcal{S}$. As explained in Panagiotelis et al. (2023), for any $\nu \in \mathcal{P}(\mathbb{R}^{n_b})$ we may obtain a distribution $\tilde{\nu} \in \mathcal{P}(\mathcal{S})$ as $\tilde{\nu} = s_{\#}\nu$, namely the pushforward of ν using s :

$$\tilde{\nu}(F) = \nu(s^{-1}(F)), \quad \forall F \in \mathcal{B}(\mathcal{S}),$$

where $s^{-1}(F) := \{\mathbf{b} \in \mathbb{R}^{n_b} : s(\mathbf{b}) \in F\}$ is the preimage of F . In other words, $s_{\#}$ builds a probability distribution for \mathbf{y} supported on the coherent subspace \mathcal{S} from a distribution on the bottom variables \mathbf{b} . Since s is a measurable bijective map, $s_{\#}$ is a bijection between $\mathcal{P}(\mathbb{R}^{n_b})$ and $\mathcal{P}(\mathcal{S})$, with inverse given by $(s^{-1})_{\#}$ (Appendix A). We thus propose the following definition.

Definition 1 We call coherent distribution any distribution $\nu \in \mathcal{P}(\mathbb{R}^{n_b})$.

This definition works with any type of distribution. Moreover, it can be used even if the constraints are not linear, as it does not require s to be a linear map.

3.1 Probabilistic reconciliation

The aim of probabilistic reconciliation is to obtain a coherent reconciled distribution $\tilde{\nu} \in \mathcal{P}(\mathbb{R}^{n_b})$ from the base forecast distribution $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$.

The probabilistic bottom-up approach, which simply ignores any probabilistic information about the upper series, is obtained by setting $\tilde{\nu} = \hat{\nu}_b$.

Panagiotelis et al. (2023) proposes a reconciliation method based on projection. Given a continuous map $\psi : \mathbb{R}^n \rightarrow \mathcal{S}$, the reconciled distribution $\tilde{\nu} \in \mathcal{P}(\mathcal{S})$ is defined as the pushforward of the base forecast distribution $\hat{\nu}$ using ψ :

$$\tilde{\nu} = \psi_{\#}\hat{\nu},$$

i.e. $\tilde{\nu}(F) = \hat{\nu}(\psi^{-1}(F))$, for any $F \in \mathcal{B}(\mathbb{R}^n)$. Hence, if $\mathbf{y}_1, \dots, \mathbf{y}_N$ are independent samples from $\hat{\nu}$, then $\psi(\mathbf{y}_1), \dots, \psi(\mathbf{y}_N)$ are independent samples from $\tilde{\nu}$. The map ψ is expressed as $\psi = s \circ g$, where $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n_b}$ combines information from all the levels by projecting on the bottom level. g is assumed to be in the form $g(\mathbf{y}) = \mathbf{d} + \mathbf{G}\mathbf{y}$, and the parameters $\gamma := (\mathbf{d}, \text{vec}(\mathbf{G})) \in \mathbb{R}^{n_b + n_b \times n}$ are optimized through stochastic gradient descent (SGD) to minimize a chosen scoring rule. This approach therefore can only be used with continuous distributions.

3.2 Probabilistic Reconciliation through conditioning

We now present our approach to probabilistic reconciliation, based on conditioning on the hierarchy constraints. Let $\hat{\mathbf{Y}} = (\hat{\mathbf{U}}, \hat{\mathbf{B}})$ be a random vector representing the probabilistic forecasts with distribution given by $\hat{\nu}$, so that $\hat{\nu}_u$ and $\hat{\nu}_b$ are the distributions of $\hat{\mathbf{U}}$ and $\hat{\mathbf{B}}$.

Let us first suppose that the base forecast distribution $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$ is discrete, and let $\hat{\pi}$ be its probability mass function. We define $\tilde{\nu}$ by conditioning on the coherent subspace \mathcal{S} :

$$\begin{aligned} \tilde{\nu}(F) &= \mathbb{P}(\hat{\mathbf{B}} \in F \mid \hat{\mathbf{Y}} \in \mathcal{S}) \\ &= \frac{\mathbb{P}(\hat{\mathbf{B}} \in F, \hat{\mathbf{Y}} \in \mathcal{S})}{\mathbb{P}(\hat{\mathbf{Y}} \in \mathcal{S})} \\ &= \frac{\mathbb{P}(\hat{\mathbf{B}} \in F, \hat{\mathbf{U}} = \mathbf{A}\hat{\mathbf{B}})}{\mathbb{P}(\hat{\mathbf{U}} = \mathbf{A}\hat{\mathbf{B}})} \\ &= \frac{\sum_{\mathbf{b} \in F} \hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b})}{\sum_{\mathbf{x} \in \mathbb{R}^{n_b}} \hat{\pi}(\mathbf{A}\mathbf{x}, \mathbf{x})}, \end{aligned} \tag{4}$$

for any $F \in \mathcal{B}(\mathbb{R}^{n_b})$, provided that $\mathbb{P}(\hat{\mathbf{Y}} \in \mathcal{S}) > 0$. The sums in (4) are well-defined, as $\hat{\pi}(\mathbf{u}, \mathbf{b}) = \hat{\pi}(\mathbf{y}) > 0$ for at most countably many \mathbf{y} 's. Hence, $\tilde{\nu}$ is a discrete probability distribution with pmf given by

$$\tilde{\pi}(\mathbf{b}) = \frac{\hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b})}{\sum_{\mathbf{x} \in \mathbb{R}^{n_b}} \hat{\pi}(\mathbf{A}\mathbf{x}, \mathbf{x})} \propto \hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b}). \tag{5}$$

Note that, if $\hat{\nu}$ is absolutely continuous, we have that $\hat{\nu}(\mathcal{S}) = 0$, since the Lebesgue measure of \mathcal{S} is zero. Hence, $\mathbb{P}(\hat{\mathbf{B}} \in F \mid \hat{\mathbf{Y}} \in \mathcal{S})$ is not well-defined. However, if we denote by $\hat{\pi}$ the density of $\hat{\nu}$, the last expression is still well-posed. We thus give the following definition.

Definition 2 Let $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$ be a base forecast distribution. The reconciled distribution through conditioning is defined as the probability distribution $\tilde{\nu} \in \mathcal{P}(\mathbb{R}^{n_b})$ such that

$$\tilde{\pi}(\mathbf{b}) \propto \hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b}), \tag{6}$$

where $\hat{\pi}$ and $\tilde{\pi}$ are the densities of (respectively) $\hat{\nu}$ and $\tilde{\nu}$, if $\hat{\nu}$ is absolutely continuous, or the probability mass functions otherwise.

To rigorously derive (6) in the continuous case, we proceed as follows. Let us define the random vector $\mathbf{Z} := \hat{\mathbf{U}} - \mathbf{A}\hat{\mathbf{B}}$. Note that the event $\{\hat{\mathbf{Y}} \in \mathcal{S}\}$ coincides with $\{\mathbf{Z} = \mathbf{0}\}$. The joint density of $(\mathbf{Z}, \hat{\mathbf{B}})$ can be easily computed (Appendix A):

$$\pi_{(\mathbf{Z}, \hat{\mathbf{B}})}(\mathbf{z}, \mathbf{b}) = \hat{\pi}(\mathbf{z} + \mathbf{A}\mathbf{b}, \mathbf{b}).$$

Then, the conditional density of $\hat{\mathbf{B}}$ given $\mathbf{Z} = \mathbf{0}$ is given by (Çınlar 2011, Chapter 4):

$$\begin{aligned} \tilde{\pi}(\mathbf{b}) &= \frac{\pi_{(\mathbf{Z}, \hat{\mathbf{B}})}(\mathbf{0}, \mathbf{b})}{\int_{\mathbb{R}^{n_b}} \pi_{(\mathbf{Z}, \hat{\mathbf{B}})}(\mathbf{0}, \mathbf{x}) d\mathbf{x}} \\ &= \frac{\hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b})}{\int_{\mathbb{R}^{n_b}} \hat{\pi}(\mathbf{A}\mathbf{x}, \mathbf{x}) d\mathbf{x}} \\ &\propto \hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b}), \end{aligned}$$

provided that $\int_{\mathbb{R}^{n_b}} \hat{\pi}(\mathbf{A}\mathbf{x}, \mathbf{x}) d\mathbf{x} > 0$. Finally, note that, if $\hat{\mathbf{U}}$ and $\hat{\mathbf{B}}$ are independent, (6) may be rewritten as

$$\tilde{\pi}(\mathbf{b}) \propto \hat{\pi}_u(\mathbf{A}\mathbf{b}) \cdot \hat{\pi}_b(\mathbf{b}), \tag{7}$$

where $\hat{\pi}_u$ and $\hat{\pi}_b$ are the densities of (respectively) $\hat{\nu}_u$ and $\hat{\nu}_b$. This approach can be applied to both continuous and discrete distributions, yielding the same expression (6) for the reconciled distribution.

Given two coherent points $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{S}$, the distribution reconciled through conditioning satisfies the following property:

$$\frac{\tilde{\pi}(\mathbf{y}_1)}{\tilde{\pi}(\mathbf{y}_2)} = \frac{\hat{\pi}(\mathbf{y}_1)}{\hat{\pi}(\mathbf{y}_2)} \tag{8}$$

if $\hat{\pi}(\mathbf{y}_2) \neq 0$, and $\tilde{\pi}(\mathbf{y}_2) = 0$ if $\hat{\pi}(\mathbf{y}_2) = 0$; i.e., the relative probabilities of the coherent points are preserved. Moreover, reconciliation via conditioning ignores the behaviour of the base distribution outside the coherent subspace. As shown by (6), $\tilde{\nu}$ only depends on the values of $\hat{\nu}$ on \mathcal{S} . Reconciliation via conditioning is therefore invariant under modifications of the base forecast probabilities outside the coherent subspace. This constitutes a major difference with respect to the method of Panagiotelis et al. (2023) that will be thoroughly studied in future work.

In Corani et al. (2023), the authors follow an approach based on *virtual evidence* (Pearl 1988) to reconcile discrete forecasts. They set the joint bottom-up distribution as a prior on the entire hierarchy, and the update is made by conditioning on the base upper forecasts, treated as uncertain observations. In contrast, we provide a unified treatment of reconciliation via conditioning for the discrete and the continuous case. Our approach has a clear interpretation, as the conditioning is done on the hierarchy constraints.

4 Sampling from the reconciled distribution

If the base forecasts are jointly Gaussian, then the reconciled distribution is also Gaussian. In this case, reconciliation via conditioning yields the same mean and variance (Corani et al. 2020) of MinT, which is optimal with respect to the log score (Wickramasuriya 2023).

In general, however, the reconciled distribution is not available in parametric form, hence we need to resort to sampling approaches. We propose a method based on Importance Sampling (IS, Kahn 1950; Elvira and Martino 2021).

4.1 Importance Sampling

Let X be an absolutely continuous random variable with density p . Suppose we want to compute the expectation $\mu = \mathbb{E}[f(X)]$, for some function f . Importance Sampling estimates the expectation μ by sampling from a different distribution q , and by weighting the samples to correct the mismatch between the target p and the proposal q .

In the following the term *density* denotes either the probability mass function (for discrete distributions) or the density with respect to the Lebesgue measure (for absolutely continuous distributions). Let q be a density such that $q(x) > 0$ if $f(x)p(x) \neq 0$, and let y_1, \dots, y_N be independent samples drawn from q . The self-normalized importance sampling estimate (Elvira and Martino 2021) is:

$$\mathbb{E}[f(X)] \approx \frac{\sum_{i=1}^N w(y_i) f(y_i)}{\sum_{i=1}^N w(y_i)}, \tag{9}$$

where w is defined as $w(y) = c \frac{p(y)}{q(y)}$, for some (typically unknown) constant c .

4.2 Probabilistic reconciliation via IS

Let $\tilde{\nu}$ (Definition 2) be the target distribution. We set $\hat{\nu}_b$ as proposal distribution. Given a sample $\mathbf{b}_1, \dots, \mathbf{b}_N$ drawn from $\hat{\nu}_b$, the weights are computed as

$$w_i := \frac{\hat{\pi}(\mathbf{A}\mathbf{b}_i, \mathbf{b}_i)}{\hat{\pi}_b(\mathbf{b}_i)}. \tag{10}$$

Then, $(\mathbf{b}_i, \tilde{w}_i)_{i=1, \dots, N}$ is a weighted sample from $\tilde{\nu}$, where $\tilde{w}_i := w_i / \sum_{j=1}^N w_j$ are the normalized weights. Note that (10) may be interpreted as the conditional density of $\hat{\mathbf{U}}$ at the point $\mathbf{A}\mathbf{b}_i$, given that $\hat{\mathbf{B}} = \mathbf{b}_i$. We thus draw samples $(\mathbf{b}_i)_i$ from the base bottom distributions, and then weight how likely they are using the base upper distributions. Under the assumption of independence between $\hat{\mathbf{B}}$ and $\hat{\mathbf{U}}$, the density of $\tilde{\nu}$ factorizes as in (7), hence:

$$w_i = \hat{\pi}_u(\mathbf{A}\mathbf{b}_i). \tag{11}$$

However, IS is affected by the curse of dimensionality (Agapiou et al. 2017). In Appendix D.2, we empirically show that IS has poor accuracy when reconciling large hierarchies. Another shortcoming of IS is that it is unreliable if the proposal distribution does not well approximate the target

distribution. Indeed, we prove in Appendix E that the performance of IS degrades as the Kullback–Leibler divergence between bottom-up and base forecast distributions (which is related to the incoherence of the base forecasts) increases. The Bottom-Up Importance Sampling (BUIS) algorithm addresses such problems.

4.3 Bottom-Up Importance Sampling algorithm

Algorithm 1 Bottom-Up Importance Sampling

```

1: Sample  $(\mathbf{b}^{(i)})_{i=1, \dots, N}$  from  $\hat{\pi}_b$ 
2: for  $l$  in levels do
3:   for  $j = 1, \dots, k_l$  do
4:      $\tilde{w}^{(i)} \leftarrow \hat{\pi}_{u_{j,l}} \left( \sum_{t=1}^{q_{j,l}} b_{t,(j,l)}^{(i)} \right)$  for  $i = 1, \dots, N$ 
5:      $w^{(i)} \leftarrow \frac{\tilde{w}^{(i)}}{\sum_h \tilde{w}^{(h)}}$  for  $i = 1, \dots, N$ 
6:      $(\bar{\mathbf{b}}_j^{(i)})_i \leftarrow \text{Resample} \left( \left( b_{1,(j,l)}^{(i)}, \dots, b_{q_{j,l},(j,l)}^{(i)} \right), w^{(i)} \right)_i$ 
7:   end for
8:    $\mathbf{b}^{(i)} \leftarrow [\bar{\mathbf{b}}_1^{(i)}, \dots, \bar{\mathbf{b}}_{k_l}^{(i)}]$  for  $i = 1, \dots, N$ 
9: end for
10: return  $(\mathbf{b}^{(i)})_i$ 

```

First, we state the main assumption of our algorithm:

Assumption 1 The base forecasts of each variable are conditionally independent, given the time series observations.

We leave for future work the extension of this algorithm to deal with correlations between the base forecasts. In this paper we perform experiments with temporal hierarchies, which commonly make this assumption.

In order to simplify the presentation, we also assume that the data structure is strictly *hierarchical*, i.e., that every node only has one parent and thus the hierarchy is represented by a tree. *Grouped* time series (Hyndman and Athanasopoulos 2021, Chapter 11), which do not satisfy this assumption, require a more complex treatment; we discuss it in Sect. 4.5.

The BUIS algorithm exploits the hierarchical structure to split a large n_u -dimensional importance sampling problem into n_u one-dimensional problems, thus deeply alleviating the curse of dimensionality. BUIS starts by drawing a sample from the base bottom distribution $\hat{\nu}_b$. Then, for each level of the hierarchy, from bottom to top, it updates the sample through an importance sampling step, using the “partially” reconciled distribution as proposal.

For each level $l = 1, \dots, L$ of the hierarchy, we denote the upper variables at level l by $u_{1,l}, \dots, u_{k_l,l}$. Moreover, for any upper variable $u_{j,l}$, we denote by $b_{1,(j,l)}, \dots, b_{q_{j,l},(j,l)}$ the bottom variables that sum up to $u_{j,l}$. In this way, we have that $\sum_{l=1}^L k_l = n_u$, the number of upper variables, while $\sum_{j=1}^{k_l} q_{j,l} = n_b$, the number of bottom variables, for each level l .

Let us consider, for example, the hierarchy in Fig. 1. For the first level $l = 1$, we have $k_1 = 2$, $u_{1,1} = U_2$, and $u_{2,1} =$

U_3 . Moreover, $q_{1,1} = q_{2,1} = 2$, and $b_{1,(1,1)} = B_1, b_{2,(1,1)} = B_2, b_{1,(2,1)} = B_3, b_{2,(2,1)} = B_4$. For the last level $l = 2$, we have $k_2 = 1, u_{1,2} = U_1, q_{1,2} = 4, b_{1,(1,2)} = B_1, b_{2,(1,2)} = B_2, b_{3,(1,2)} = B_3, b_{4,(1,2)} = B_4$.

Alg. 1 shows the BUIS algorithm. The ‘‘Resample’’ step samples with replacement from the discrete distribution given by

$$\mathbb{P}(\mathbf{b} = (b_{1,(j,l)}^{(i)}, \dots, b_{q_{j,l}(j,l)}^{(i)})) = w^{(i)}, \tag{12}$$

for all $i = 1, \dots, N$. Note that the algorithm can be easily parallelized by drawing batches of samples on different cores. This additional step would further reduce the computational times.

We explicit the BUIS algorithm on the simple hierarchy in Fig. 1:

1. Sample $(b_j^{(i)})_{i=1, \dots, N}$ from π_{B_j} , for $j = 1, 2, 3, 4$
2. Compute the weights $(w^{(i)})_{i=1, \dots, N}$ with respect to U_2 as

$$w^{(i)} = \pi_{U_2} (b_1^{(i)} + b_2^{(i)})$$

3. Sample $(\bar{b}_1^{(i)}, \bar{b}_2^{(i)})_i$ with replacement from $((b_1^{(i)}, b_2^{(i)}), w^{(i)})_{i=1, \dots, N}$
4. Repeat step 2 and 3 using B_3, B_4 and U_3 to get $(\bar{b}_3^{(i)}, \bar{b}_4^{(i)})_i$
5. Set $(b_1^{(i)}, b_2^{(i)}, b_3^{(i)}, b_4^{(i)})_i = (\bar{b}_1^{(i)}, \bar{b}_2^{(i)}, \bar{b}_3^{(i)}, \bar{b}_4^{(i)})_i$ and move to the next level
6. Compute the weights $(w^{(i)})_{i=1, \dots, N}$ with respect to U_1 as

$$w^{(i)} = \pi_{U_1} (b_1^{(i)} + b_2^{(i)} + b_3^{(i)} + b_4^{(i)})$$

7. Sample $(\bar{b}_1^{(i)}, \bar{b}_2^{(i)}, \bar{b}_3^{(i)}, \bar{b}_4^{(i)})_i$ with replacement from $((b_1^{(i)}, b_2^{(i)}, b_3^{(i)}, b_4^{(i)}), w^{(i)})_i$

In Appendix B we prove the following proposition:

Proposition 1 *The output of the BUIS algorithm is approximately a sample drawn from the reconciled distribution $\tilde{\nu}$.*

4.4 Sample-based BUIS

Sometimes the base forecasts are given as samples, without a parametric form; this is the case of models for time series of counts (Liboschik et al. 2017) or based on deep learning (Salinas et al. 2020). BUIS can reconcile also this type of base forecasts. Since we only deal with one-dimensional densities to compute the weights, we use approximations based on samples. For discrete distributions, we use the empirical distribution. For continuous distributions, we use kernel

density estimation (Chen 2017). Therefore, we only need to replace line 4 in Algorithm 1 with:

$$\begin{aligned} &\text{Sample } (u_{j,l}^{(i)})_{i=1, \dots, N} \text{ from } \hat{\pi}_{u_{j,l}} \\ &\tilde{\pi} \leftarrow \text{Density Estimation} \left((u_{j,l}^{(i)})_{i=1, \dots, N} \right) \\ &\check{w}^{(i)} \leftarrow \tilde{\pi} \left(\sum_{t=1}^{q_{j,l}} b_{t,(j,l)}^{(i)} \right) \text{ for } i = 1, \dots, N \end{aligned}$$

The sample-based algorithm becomes slightly slower due to the density estimation step.

4.5 More complex hierarchies: grouped time series

We refer to *grouped* time series when the data structure does not disaggregate in a unique hierarchical manner (Hyndman and Athanasopoulos 2021, Chapter 11). In this case, the aggregated series cannot be represented by a single tree, as a bottom node can have more than one parent. For instance, consider a weekly time series, for which we compute the following temporal aggregates: 2-weeks, 4-weeks, 13-weeks, 26-weeks, 1-year. A bottom node (weekly) is thus children of both the 2-weeks and of the 13-weeks aggregates. This structure cannot be represented as a tree.

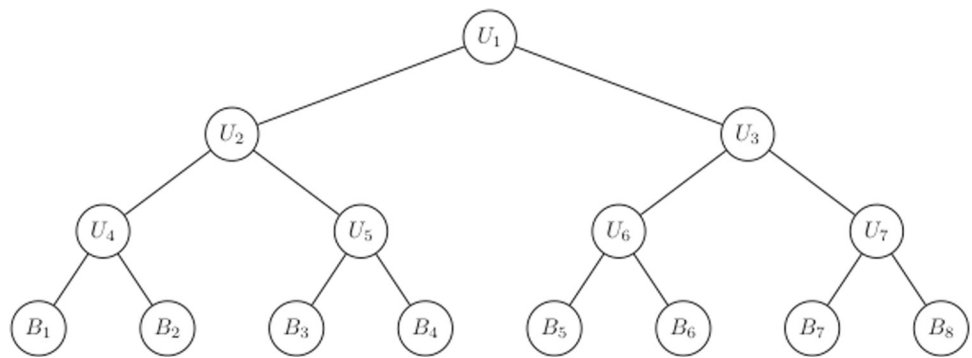
The BUIS algorithm, as described in Sect. 4.3, requires that the hierarchy is a tree, so it cannot be used in this case. Indeed, as highlighted in the proof, we need the independence of $\bar{\mathbf{b}}_1, \dots, \bar{\mathbf{b}}_{k_l}$ to multiply their densities. If the hierarchy is not a tree, correlations between bottom variables are created when conditioning on the upper levels.

To overcome this problem, we proceed as follows. First, we find the largest sub-hierarchy within the group structure. For instance, in the example above, we consider the sub-hierarchy given by the bottom variables and by the 2-weeks, 4-weeks and 1-year aggregates. All the other upper variables are then regarded as additional constraints. We use the BUIS algorithm on the sub-hierarchy, obtaining a sample \mathbf{b} . Then, we compute the weights on \mathbf{b} using the base distributions of the additional constraints. This is equivalent to performing a standard IS, where we use the output of BUIS on the hierarchical part as proposal distribution. In this way, we reduce the dimension of the IS task from n_u , the total number of upper constraints, to the number of constraints that are not included in the sub-hierarchy: in the above example, from 46 to 6. We highlight that the distribution we sample from would be the same even with different choices of sub-hierarchies. However, picking the largest one is the best choice from a computational perspective.

5 Experiments on synthetic data

We now empirically test the convergence of the BUIS algorithm to the true reconciled distribution. We compare

Fig. 2 A binary hierarchy



BUIS with IS and with the method by Corani et al. (2023), which we implement using the library *PyMC* (Salvatier et al. 2016). *PyMC* adopts an adaptive Metropolis-Hastings algorithm (Haario et al. 2001) for discrete distributions and the No-U-Turn Sampler (NUTS, Hoffman and Gelman 2014) for continuous distributions. We performed experiments on the hierarchy of Fig. 2, implementing the IS and BUIS algorithms in Python.

5.1 Reconciling Gaussian forecasts

Dealing with Gaussian base forecasts, the reconciled distribution can be obtained in closed form (Corani et al. 2020). We can thus check how the various algorithms approximates the exact solution. We set on each bottom node a Gaussian distribution with mean randomly chosen in the interval [5, 10], and standard deviation $\sigma_b = 2$. We denote by $\mu_b \in \mathbb{R}_+^8$ the vector of the base bottom means. We induce incoherence by setting the means of the base forecast of the upper variables as $\mu_u = (1 + \epsilon)\mathbf{A}\mu_b$, where \mathbf{A} is the aggregating matrix and ϵ is the incoherence level; we consider $\epsilon \in \{0.1, 0.3, 0.5\}$. Hence, if $\epsilon = 0.3$ the base upper means are 30% greater than the sum of the corresponding base bottom means. We set $\sigma_u = 3$ as standard deviation for the base forecast of each upper variable.

We run *PyMC* with 4 chains with 5,000 samples each. For IS and BUIS, we run multiple experiments, drawing each time a different number of samples, ranging from 10^4 to 10^6 . We repeat each experiment 30 times. We then compute the 2-Wasserstein distance (Panaretos and Zemel 2019) between the true reconciled distribution, obtained analytically, and the empirical distributions obtained via sampling. The results are reported in Fig. 3a, where we also show the 95% confidence interval over the 30 experiments. Note that the axes are in logarithmic scale.

As expected, the performance of IS and BUIS depends on the incoherence level ϵ . This behavior also affects BUIS, which is based on importance sampling. However, BUIS is significantly more robust than IS, and it works effectively even with extreme incoherence level such as $\epsilon = 0.5$. As

the number of samples grows, the performance of BUIS improves, eventually outperforming the reference method based on *PyMC*. We confirm the results by computing the percentage error on the reconciled mean (Appendix D.1). Even with an extreme incoherence level, $\epsilon = 0.5$, the percentage error on the mean obtained with 10^6 samples from BUIS is negligible ($< 0.1\%$) and comparable to *PyMC*. In the same setup IS achieves an error greater than 5%.

Both IS and BUIS are substantially faster than *PyMC* (Table 1). The computational time of BUIS with 10^5 samples is two orders of magnitude smaller than *PyMC*, while achieving comparable performances. Note that here BUIS is running on a single core. An insight about the reasons of such a speedup is given in Appendix C, where we provide a detailed comparison between IS and a bare-bones implementation of MCMC on a simple hierarchy.

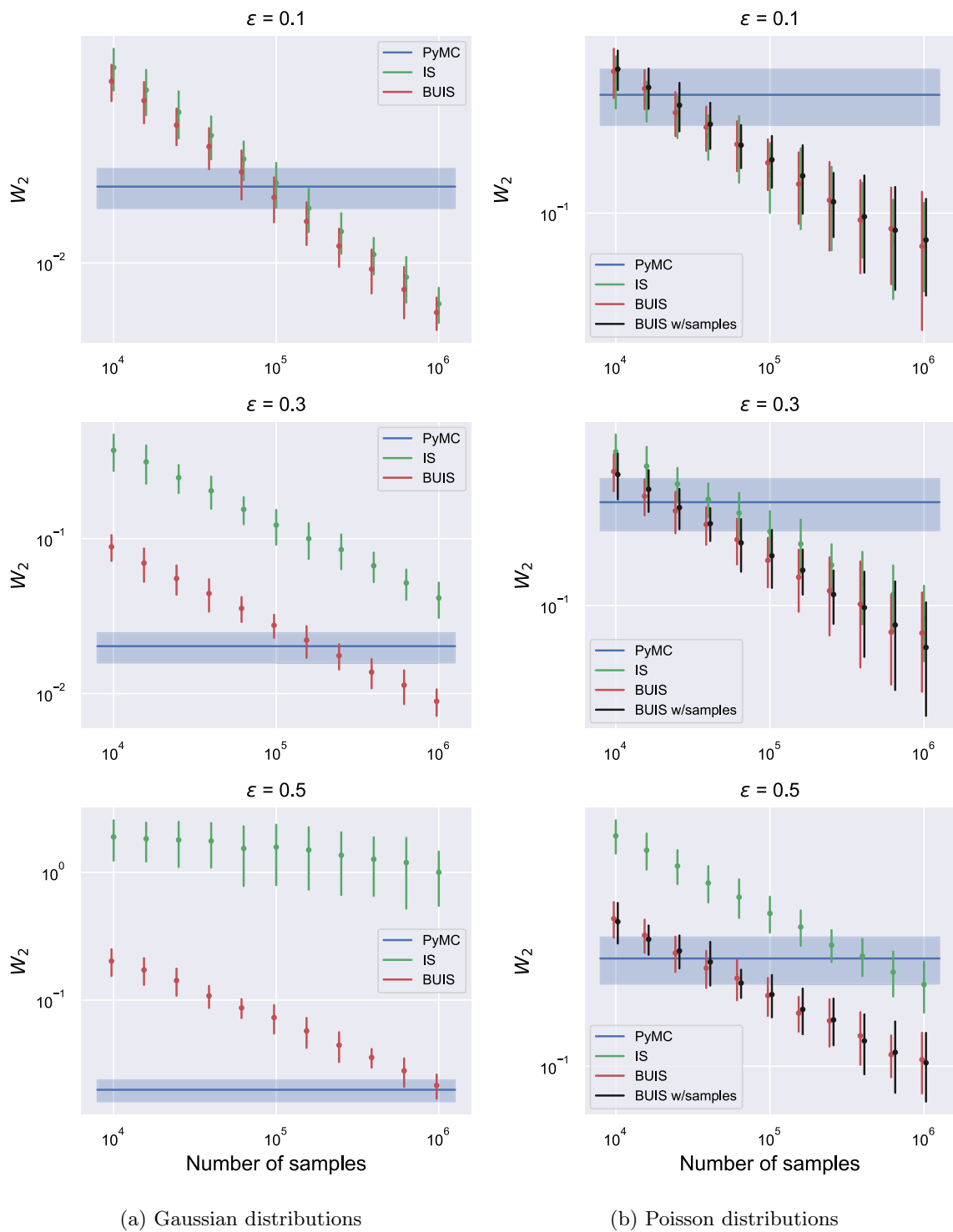
We also conduct similar experiments using a larger hierarchy; the results, reported in Appendix D.2, confirm that the BUIS is robust and computationally efficient.

5.2 Reconciling Poisson forecasts

We now consider discrete base forecasts. We set a Poisson distribution on each bottom variable, with mean randomly chosen in the interval [5, 10]. We denote by $\lambda_b \in \mathbb{R}_+^8$ the vector of the base bottom means. As before, for each incoherence level $\epsilon \in \{0.1, 0.3, 0.5\}$, we set the mean of the upper variables as $\lambda_u = (1 + \epsilon)\mathbf{A}\lambda_b$. In the Poisson case, the reconciled distribution cannot be analytically computed. We thus run an extensive experiment using *PyMC*, with 20 chains with 50,000 samples each. We consider these samples as the true reconciled distribution.

We run the same experiments described in Sect. 5.1. Since probabilistic forecasts of count time series are typically given as samples (Liboschik et al. 2017), we also run sample-based BUIS (Sect. 4.4): we assume that the parametric form of the base distribution is unknown, and that only samples are available.

The 2-Wasserstein distances are reported in Fig. 3b. As the number of samples grows, BUIS and sample-based BUIS



(a) Gaussian distributions

(b) Poisson distributions

Fig. 3 Wasserstein distance between true and empirical distributions. The axes are logarithmic

Table 1 Average computational times with the standard deviations (in seconds). The average times for *PyMC* (4 chains with 5,000 samples each) are: 26.81 ± 2.38 (Gaussian), 26.26 ± 4.14 (Poisson)

		Number of samples		
		10 ⁴	10 ⁵	10 ⁶
Gaussian	IS	0.01 ± 0.00	0.06 ± 0.02	0.66 ± 0.17
	BUIS	0.02 ± 0.00	0.15 ± 0.00	2.20 ± 0.02
Poisson	IS	0.01 ± 0.01	0.09 ± 0.01	1.01 ± 0.03
	BUIS	0.02 ± 0.01	0.19 ± 0.03	2.68 ± 0.41
	sample-based BUIS	0.03 ± 0.00	0.24 ± 0.01	3.49 ± 0.09

eventually outperform *PyMC*, for all levels of incoherence. As for the Gaussian case, the performance of IS deteriorates for larger values of the incoherence level. The results are confirmed by the percentage error on the reconciled mean (Appendix D.1), which is lower than 0.2% for BUIS with 10⁵ samples and about 0.4% for *PyMC*.

In Table 1 we show the average computational times. Sample-based BUIS is slightly slower than BUIS because of the density estimation step. Note that, using 10⁵ samples, BUIS and sample-based BUIS are 2 orders of magnitude faster than *PyMC*, while achieving a better performance for all incoherence levels.

6 Experiments on real data

We now perform probabilistic reconciliation on temporal hierarchies, using time series extracted from two different data sets: *carparts*, available from the R package *expsmooth* (Hyndman 2018), and *syph*, available from the R package *ZIM* (Yang et al. 2018).

The *carparts* data set is about monthly sales of car parts. As in (Hyndman et al. 2008, Chapter 16), we remove time series with missing values, with less than 10 positive monthly demands and with no positive demand in the first 15 and final 15 months. After this selection, there are 1046 time series left. Note that we use less restrictive criteria in the selection of the time series than Corani et al. (2023), where only 219 time series from *carparts* were considered. Monthly data are aggregated into 2-months, 3-months, 4-months, 6-months and 1-year levels.

The *syph* data set is about the weekly number of syphilis cases in the United States. We remove the time series with ADI greater than 20. The ADI is computed as $ADI = \frac{\sum_{i=1}^P p_i}{P}$, where p_i is the time period between two non-zeros values and P is the total number of periods (Syntetos and Boylan 2005). We also remove the time series corresponding to the total number of cases in the US. After this selection, there are 50 time series left. Weekly data are aggregated into 2-weeks, 4-weeks, 13-weeks, 26-weeks and 1-year levels.

For both data sets, we fit a generalized linear model with the *tscount* package (Liboschik et al. 2017). We use a negative

Table 2 Skill scores on the time series extracted from *carparts*, detailed by each level of the hierarchy

Metric	hier-level	<i>N</i> vs base	<i>NB</i> vs base	samples vs base
ES		0.07	0.52	0.53
MASE	Monthly	-1.02	0.14	0.13
	2-Monthly	-0.53	0.25	0.27
	Quarterly	-0.42	0.21	0.26
	4-Monthly	-0.40	0.16	0.21
	Biannual	-0.33	0.14	0.16
	Annual	-0.26	0.18	0.17
	average	-0.49	0.18	0.20
MIS	Monthly	-0.08	0.45	0.63
	2-Monthly	0.28	0.45	0.56
	Quarterly	0.22	0.43	0.46
	4-Monthly	0.03	0.35	0.36
	Biannual	-0.07	0.37	0.26
	Annual	-0.17	0.40	0.22
	average	0.03	0.41	0.42

The best result for each row is in bold

binomial predictive distribution, with a first-order regression on past observations. The test set has length 1 year for both data sets. We thus compute up to 12 steps ahead at monthly level, and up to 52 steps ahead at weekly level. Probabilistic forecasts are returned in the form of samples.

Reconciliation is performed in three different ways. In the first case, we fit a Gaussian distribution on the returned samples. Then, we follow (Corani et al. 2020) to analytically compute the Gaussian reconciled distribution. In the second case, we fit a negative binomial distribution on the samples, and we reconcile using the BUIS algorithm. Since these are grouped time series rather than hierarchical time series, we use the method of Sect. 4.5 for grouped time series. Finally, we use the sample-based BUIS (Sect. 4.4), without fitting a parametric distribution. Although the sample-based algorithm is slightly slower, this method yields a computational gain over BUIS, as fitting a negative binomial distribution on the samples requires about 1.2 s for the monthly hierarchy and 3.9 s for the weekly hierarchy. We refer to these methods,

Table 3 Skill scores on the time series extracted from *syph*, detailed by each level of the hierarchy

Metric	Hier-level	N vs base	NB vs base	samples vs base
ES		0.08	0.11	0.15
MASE	Weekly	-0.63	0.14	0.14
	2-Weekly	-0.40	0.16	0.14
	4-Weekly	-0.22	0.13	0.12
	Quarterly	-0.10	0.01	0.04
	Biannual	0.01	0.07	0.15
	Annual	-0.05	-0.00	0.04
	average	-0.23	0.08	0.10
	MIS	Weekly	-0.06	0.46
2-Weekly		0.08	0.33	0.34
4-Weekly		0.03	0.19	0.25
Quarterly		-0.15	-0.11	-0.08
Biannual		-0.34	-0.27	-0.21
Annual		-0.33	-0.23	-0.22
average		-0.13	0.06	0.09

The best result for each row is in bold

respectively, as *N*, *NB*, and *samples*. Furthermore, we denote by *base* the unreconciled forecasts.

We use different indicators to assess the performance of each method. The mean scaled absolute error (MASE) (Hyndman 2006) is defined as

$$MASE = \frac{MAE}{Q},$$

where $MAE = \frac{1}{h} \sum_{j=1}^h |y_{t+j} - \hat{y}_{t+j|t}|$ and $Q = \frac{1}{T-1} \sum_{t=2}^T |y_t - y_{t-1}|$. Here, y_t denotes the value of the time series at time t , while $\hat{y}_{t+j|t}$ denotes the point forecast computed at time t for time $t + j$. The median of the distribution is used as point forecast, since it minimizes MASE (Kolassa 2016).

The mean interval score (MIS) (Gneiting 2011) is defined, for any $\alpha \in (0, 1)$, as

$$MIS = (u - l) + \frac{2}{\alpha}(l - y)\mathbb{1}(y < l) + \frac{2}{\alpha}(y - u)\mathbb{1}(y > u),$$

where l and u are the lower and upper bounds of the $(1 - \alpha)$ forecast coverage interval and y is the actual value of the time series. In the following, we use $\alpha = 0.1$. MIS penalizes wide prediction intervals, as well as intervals that do not contain the true value.

Finally, the Energy score (Székely and Rizzo 2013) is defined as

$$ES(P, \mathbf{y}) = \mathbb{E}_P [\|\mathbf{y} - \mathbf{s}\|^\alpha] - \frac{1}{2} \mathbb{E}_P [\|\mathbf{s} - \mathbf{s}'\|^\alpha],$$

where P is the forecast distribution on the whole hierarchy, $\mathbf{s}, \mathbf{s}' \sim P$ are a pair of independent random variables and

\mathbf{y} is the vector of the actual values of all the time series. The energy score is a proper scoring rule for distributions defined on the entire hierarchy (Panagiotelis et al. 2023). We compute ES , with $\alpha = 2$, using *samples*, as explained in Wickramasuriya (2023).

We use the skill score to compare the performance of a method with respect to a baseline method, in terms of percentage improvement. We use *base* as baseline method. For example, the skill score of *NB* on MASE is given by

$$Skill(NB, base) = \frac{MASE(base) - MASE(NB)}{(MASE(base) + MASE(NB)) / 2}.$$

Note that the skill score is symmetric and scale-independent. For each level, we compute the skill score for each forecasting horizon, and take the average.

The skill scores for *carparts* are reported in Table 2. Both *NB* and *samples* methods yield a significant improvement for all the indicators, and for all the hierarchy levels. For both methods, the average improvement is about 20% for MASE, 40% for MIS and 50% for ES. The skill scores for *syph* are reported in Table 3. As before, the average improvement of *NB* and *samples* is significant for all indicators. For both datasets, the *N* method performs poorly, in many cases yielding negative skill scores. As observed in Corani et al. (2023), this method does not capture the asymmetry of the base forecasts. Finally, *samples* appears to perform better than *NB*. Indeed, the step of fitting a Negative Binomial distribution on the forecast samples may yield an additional source of error.

7 Conclusions

Our approach to probabilistic reconciliation based on conditioning allows to treat continuous and discrete forecast distributions in a unified framework. Moreover, the proposed BUIS is able to efficiently sample from continuous and discrete predictive distributions, provided in parametric form or as samples. We make available the BUIS algorithm within the *R* package *bayesRecon* (Azzimonti et al. 2023).

A future research direction is how to relax the assumption of conditional independence of the base forecasts. A second one is to study the implications of ignoring the behavior of the base forecast distribution outside the coherent subspace, which is a feature of reconciliation via conditioning and constitutes a major difference from reconciliation via projection.

Acknowledgements Work partially funded by the Swiss National Science Foundation (grant 200021_212164/1) and by the Hasler foundation (project 23057).

Author Contributions L. Zambon did the implementation and wrote the main manuscript text. D. Azzimonti supervised the formal part of

the paper. G. Corani supervised the experimental part of the paper. All authors reviewed the manuscript and participated in critical discussions.

Funding Open access funding provided by SUPSI - University of Applied Sciences and Arts of Southern Switzerland

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A Proofs

Proposition 2 Let $s : X \rightarrow Y$ be a measurable bijection between two measure spaces (X, \mathcal{X}) and (Y, \mathcal{Y}) . Then, the pushforward $s_{\#} : \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ is a bijection, with inverse given by $(s^{-1})_{\#}$.

Proof First, we recall that the pushforward $s_{\#}$ is defined, for any $\nu \in \mathcal{P}(X)$ and $F \in \mathcal{Y}$, as

$$s_{\#}\nu(F) = \nu(s^{-1}(F)).$$

Hence, for any $\nu \in \mathcal{P}(X)$ and $G \in \mathcal{X}$, we have

$$\begin{aligned} ((s^{-1})_{\#} \circ s_{\#})\nu(G) &= (s^{-1})_{\#}(s_{\#}\nu)(G) \\ &= s_{\#}(\nu)((s^{-1})^{-1}(G)) \\ &= s_{\#}(\nu)(s(G)) \\ &= \nu(s^{-1}(s(G))) \\ &= \nu(G), \end{aligned}$$

and therefore $(s^{-1})_{\#} \circ s_{\#}$ is the identity map. Analogously, for any $\mu \in \mathcal{P}(Y)$ and $F \in \mathcal{X}$, we have

$$\begin{aligned} (s_{\#} \circ (s^{-1})_{\#})\mu(F) &= s_{\#}((s^{-1})_{\#}\mu)(F) \\ &= (s^{-1})_{\#}(\mu)(s^{-1}(F)) \\ &= \mu((s^{-1})^{-1}(s^{-1}(F))) \\ &= \mu(s(s^{-1}(F))) \\ &= \mu(F). \end{aligned}$$

Proposition 3 Let $\hat{\pi}$ be the joint density of the random vector (\hat{U}, \hat{B}) . Then, the density of (Z, \hat{B}) , where $Z := \hat{U} - \mathbf{A}\hat{B}$, is given by

$$\pi_{(Z, \hat{B})}(z, \mathbf{b}) = \hat{\pi}(z + \mathbf{A}\mathbf{b}, \mathbf{b}).$$

Proof The joint density of (Z, \hat{B}) can be computed using the rule of change of variables (Billingsley 2008, Chapter 17). Let $\mathbf{H} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be defined as

$$\mathbf{H} : \begin{bmatrix} \mathbf{u} \\ \mathbf{b} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{u} - \mathbf{A}\mathbf{b} \\ \mathbf{b} \end{bmatrix}.$$

\mathbf{H} is invertible, with inverse given by

$$\mathbf{H}^{-1} : \begin{bmatrix} \mathbf{z} \\ \mathbf{b} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{z} + \mathbf{A}\mathbf{b} \\ \mathbf{b} \end{bmatrix},$$

and we have that

$$|J\mathbf{H}^{-1}(\mathbf{b}, \mathbf{z})| = \begin{vmatrix} \mathbf{I} & \mathbf{A}^T \\ \mathbf{0} & \mathbf{I} \end{vmatrix} = 1.$$

Then, the joint density of (Z, \hat{B}) is given by

$$\begin{aligned} \pi_{(Z, \hat{B})}(z, \mathbf{b}) &= \hat{\pi}(\mathbf{H}^{-1}(z, \mathbf{b})) \cdot |J\mathbf{H}^{-1}(z, \mathbf{b})| \\ &= \hat{\pi}(z + \mathbf{A}\mathbf{b}, \mathbf{b}). \end{aligned}$$

□

Appendix B Proof of BUIS algorithm

We show that the output $(\mathbf{b}^{(i)})_i$ of the BUIS algorithm is approximately a sample drawn from the target distribution $\tilde{\nu}$.

From (7), and from Assumption 1, we have that

$$\begin{aligned} \tilde{\pi}(\mathbf{b}) &\propto \hat{\pi}_b(\mathbf{b}) \cdot \hat{\pi}_u(\mathbf{A}\mathbf{b}) \\ &= \prod_{t=1}^{n_b} \pi_{b_t}(b_t) \cdot \prod_{l=1}^L \prod_{j=1}^{k_l} \pi_{u_{j,l}} \left(\sum_{k=1}^{q_{j,l}} b_{k,(j,l)} \right), \end{aligned}$$

where we are using the notation of Sect. 4.3. The initial distribution of the sample $(\mathbf{b}^{(i)})_{i=1, \dots, N}$ is given by $\hat{\pi}_b = \prod_{t=1}^{n_b} \pi_{b_t}(b_t)$. We show that each iteration of the algorithm corresponds to multiplying by a $\pi_{u_{j,l}} \left(\sum_{k=1}^{q_{j,l}} b_{k,(j,l)} \right)$ term.

Let π_X be a density over \mathbb{R}^d , and $w : \mathbb{R}^d \rightarrow \mathbb{R}$ a continuous function. Let X_1, \dots, X_N be independent samples from π_X , and compute the unnormalized weights $(\hat{w}^{(i)})_{i=1, \dots, N}$ as $\hat{w}^{(i)} = w(X_i)$. Then, if we draw Y_1, \dots, Y_m from the discrete distribution given by

$$\mathbb{P}(Y = X_i) = w^{(i)}, \quad i = 1, \dots, N,$$

□

where $w^{(i)} = \frac{\hat{w}^{(i)}}{\sum_{j=1}^N \hat{w}^{(j)}}$, then $(Y_i)_{i=1, \dots, m}$ is approximately an IID sample from the density $\pi_Y(x) \propto \pi_X(x) \cdot w(x)$. This technique is known as importance resampling or weighted bootstrap (Smith and Gelfand 1992). The same holds also for discrete distributions, using the pmf instead of the density.

Hence, if we compute the weights $w^{(i)}$'s as in the algorithm and sample $(\tilde{\mathbf{b}}_j^{(i)})_i$ from (12), it is approximately equivalent to sampling from $\hat{\pi}_b(\mathbf{b}) \cdot \pi_{u_{j,l}} \left(\sum_{t=1}^{q_{j,l}} b_t \right)$, where $\hat{\pi}_b$ is the original density of $(b_{1,(j,l)}, \dots, b_{q_{j,l},(j,l)})$. In other words, the weighting-resampling step corresponds to multiplying the density of the sample by a $\pi_{u_{j,l}} \left(\sum_{t=1}^{q_{j,l}} b_t \right)$ term.

Finally, note that in this way we are conditioning with respect to $u_{j,l}$. After the weighting-resampling step, $(b_{1,(j,l)}, \dots, b_{q_{j,l},(j,l)})$ are correlated. Since the hierarchy is given by a tree, we are guaranteed that for any level l and for all $j = 1, \dots, k_l$, $\tilde{\mathbf{b}}_j$ only depends on $b_{1,(j,l)}, \dots, b_{q_{j,l},(j,l)}, u_{j,l}$ and each upper variable that is under $u_{j,l}$. From Assumption 1, we have that $\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_{k_l}$ are independent. Hence, the density of $[\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_{k_l}]$ is given by the product of the densities of all $\tilde{\mathbf{b}}_j$'s, and the proof is concluded.

Appendix C MCMC-IS comparison

In order to fully understand the reasons for the significant difference in computational time between the MCMC and the IS approach, we compare the two methods on a minimal example. Let us consider a hierarchy given by two bottom variables, b_1 and b_2 , and just one upper variable u , which is the sum of b_1 and b_2 . We set a Gaussian distribution for each variables.

We implement a simple Metropolis-Hastings algorithm with a Gaussian proposal distribution with fixed variance τI to sample from the reconciled distribution $\tilde{\pi}(\mathbf{b}) = \pi_{b_1}(b_1) \cdot \pi_{b_2}(b_2) \cdot \pi_u(b_1 + b_2)$. The algorithm reads as follows:

```

Initialize  $\mathbf{b}^{(0)}$ 
for  $j = 1, \dots, N$  do
  Sample  $\mathbf{y}^{(j)} \sim \mathcal{N}(\mathbf{b}^{(j-1)}, \tau I)$ 
   $\alpha \leftarrow \min \left( 1, \frac{\tilde{\pi}(\mathbf{y}^{(j)})}{\tilde{\pi}(\mathbf{b}^{(j-1)})} \right)$ 
   $u \leftarrow \text{Unif}(0, 1)$ 
  if  $u < \alpha$  then
     $\mathbf{b}^{(j)} \leftarrow \mathbf{y}^{(j)}$ 
  else
     $\mathbf{b}^{(j)} \leftarrow \mathbf{b}^{(j-1)}$ 
  end if
end for
return  $(\mathbf{b}^{(i)})_i$ 

```

On a standard laptop, it takes about 4 s to get 10, 000 samples from $\tilde{\pi}$. In particular, most of the time is employed by the computation of the acceptance probability α , which requires

about $3.7 \cdot 10^{-4}$ seconds per loop. Sampling from the proposal distribution only requires about $3 \cdot 10^{-5}$ seconds.

We then implement an IS algorithm on the same hierarchy, using Python:

```

Sample  $\mathbf{b}^{(1)}, \dots, \mathbf{b}^{(N)} \stackrel{\text{IID}}{\sim} \hat{\pi}_b$ 
 $w_i \leftarrow \hat{\pi}_u \left( b_1^{(i)} + b_2^{(i)} \right)$ 
return  $(\mathbf{b}^{(i)}, w_i)_i$ 

```

It takes about $7 \cdot 10^{-3}$ seconds to draw 100, 000 IID samples from $\hat{\pi}_b$, and about the same time to compute all the weights. The significant improvement in computational time is due to the fact that both sampling and computation of the weights are done simultaneously for all the samples, rather than sequentially as in MCMC.

Appendix D Additional results on synthetic data

D.1 Percentage error on the mean

Besides computing the 2-Wasserstein distance between the true reconciled distribution and the empirical reconciled distribution obtained via sampling (Sect. 5), we also compute the error on the reconciled mean. More precisely, if we denote by m_i the true mean and by \bar{m}_i the sample mean, we compute the average percentage error as:

$$\frac{1}{n} \sum_{i=1}^n \frac{|m_i - \bar{m}_i|}{m_i} \cdot 100,$$

where n is the number of nodes of the hierarchy. The average percentage errors are reported in Table 4.

D.2 Large hierarchy

We test the IS, BUIS, and *PyMC* algorithms on a larger hierarchy. We set a binary hierarchy, similar to that of Fig. 2, but with 5 levels: hence, there are 32 bottom and 31 upper nodes. We use the same procedure described in Sect. 5.1 to set the Gaussian base forecasts. Using BUIS with 10^5 samples we achieve a small average percentage error ($< 0.5\%$) on the reconciled means (Table 5), even with a large incoherence ($\epsilon = 0.5$). On the other hand, the error using IS is over 20%, even with 10^6 samples, proving that IS is not able to scale to large hierarchies. The results are confirmed by the plot of the 2-Wasserstein distance (Fig. 4). In conclusion, BUIS is able to correctly sample from the reconciled distribution, even in case of rather big hierarchies (~ 60 nodes) and large incoherence level ($\epsilon = 0.5$), while providing an impressive gain in terms of computational time with respect to *PyMC* (Table 6).

Table 4 Average percentage errors on the reconciled mean

			ϵ			
			0.1	0.3	0.5	
Gaussian	IS	10 ⁴ samples	0.39%	2.02%	11.3%	
		10 ⁵ samples	0.13%	0.6%	8.96%	
		10 ⁶ samples	0.04%	0.22%	5.69%	
	BUIS	10 ⁴ samples	0.34%	0.45%	0.92%	
		10 ⁵ samples	0.12%	0.14%	0.34%	
		10 ⁶ samples	0.04%	0.05%	0.09%	
	PyMC	4 × 5000 samples	0.11%	0.09%	0.07%	
	Poisson	IS	10 ⁴ samples	0.44%	0.75%	2.24%
			10 ⁵ samples	0.13%	0.25%	0.65%
10 ⁶ samples			0.06%	0.09%	0.23%	
BUIS		10 ⁴ samples	0.5%	0.58%	0.67%	
		10 ⁵ samples	0.16%	0.16%	0.21%	
		10 ⁶ samples	0.06%	0.07%	0.09%	
sample-based BUIS		10 ⁴ samples	0.52%	0.55%	0.59%	
		10 ⁵ samples	0.17%	0.17%	0.21%	
		10 ⁶ samples	0.07%	0.07%	0.08%	
PyMC	4 × 5000 samples	0.09%	0.07%	0.07%		

Appendix E Efficiency of IS

It is well-known that vanilla importance sampling is not effective to sample from high dimensional distributions; this prevents using it to reconcile large hierarchies. We also obtain low performances when the proposal distribution $\hat{\nu}_b$ is not a good approximation of the target distribution $\tilde{\nu}$. The following result relates the Kullback–Leibler divergence (Kullback and Leibler 1951) between the base and reconciled distribution to the efficiency of IS.

Proposition 4 *Let $\hat{\mathbf{B}}$ be a random vector distributed as $\hat{\nu}_b$, and let $W := \hat{\pi}(\mathbf{A}\hat{\mathbf{B}}, \hat{\mathbf{B}})/\hat{\pi}_b(\hat{\mathbf{B}})$. Then, the Kullback–Leibler divergence of the base bottom distribution from the reconciled bottom distribution is given by*

$$KL(\hat{\nu}_b \parallel \tilde{\nu}) = \log(\mathbb{E}[W]) - \mathbb{E}[\log(W)]. \tag{E1}$$

Proof First, we recall that, given a pair of absolutely continuous probability distributions μ and ν , the Kullback–Leibler (KL) divergence is defined as

$$KL(\mu \parallel \nu) = \int \log\left(\frac{p(x)}{q(x)}\right) p(x) dx,$$

where p and q are the densities of, respectively, μ and ν . The discrete case is completely analogous.

Now, let $\hat{\nu}_b$ be the base bottom forecast distribution, and $\tilde{\nu}$ the reconciled distribution. We recall that the density of $\tilde{\nu}$

is given by

$$\tilde{\pi}(\mathbf{b}) = \frac{1}{c} \hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b}),$$

where

$$\begin{aligned} c &:= \int \hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b}) d\mathbf{b} \\ &= \int \frac{\hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b})}{\hat{\pi}_b(\mathbf{b})} \hat{\pi}_b(\mathbf{b}) d\mathbf{b} \\ &= \mathbb{E}\left[\frac{\hat{\pi}(\mathbf{A}\hat{\mathbf{B}}, \hat{\mathbf{B}})}{\hat{\pi}_b(\hat{\mathbf{B}})}\right] \end{aligned}$$

is the normalizing constant, and $\hat{\mathbf{B}} \sim \hat{\nu}_b$. Then, we have

$$\begin{aligned} KL(\hat{\nu}_b \parallel \tilde{\nu}) &= \int \log\left(c \frac{\hat{\pi}_b(\mathbf{b})}{\hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b})}\right) \hat{\pi}_b(\mathbf{b}) d\mathbf{b} \\ &= \log(c) - \int \log\left(\frac{\hat{\pi}(\mathbf{A}\mathbf{b}, \mathbf{b})}{\hat{\pi}_b(\mathbf{b})}\right) \hat{\pi}_b(\mathbf{b}) d\mathbf{b} \\ &= \log\left(\mathbb{E}\left[\frac{\hat{\pi}(\mathbf{A}\hat{\mathbf{B}}, \hat{\mathbf{B}})}{\hat{\pi}_b(\hat{\mathbf{B}})}\right]\right) - \mathbb{E}\left[\log\left(\frac{\hat{\pi}(\mathbf{A}\hat{\mathbf{B}}, \hat{\mathbf{B}})}{\hat{\pi}_b(\hat{\mathbf{B}})}\right)\right] \\ &= \log(\mathbb{E}[W]) - \mathbb{E}[\log(W)]. \end{aligned} \tag{E2}$$

Note that the right-hand side of (E1) is a measure of the dispersion of the random variable W . Indeed, by the Jensen’s inequality, it is always non-negative, and it is zero when W is constant a.s.; it gets larger as W becomes more dispersed. In the context of the measures of inequality, it usually referred

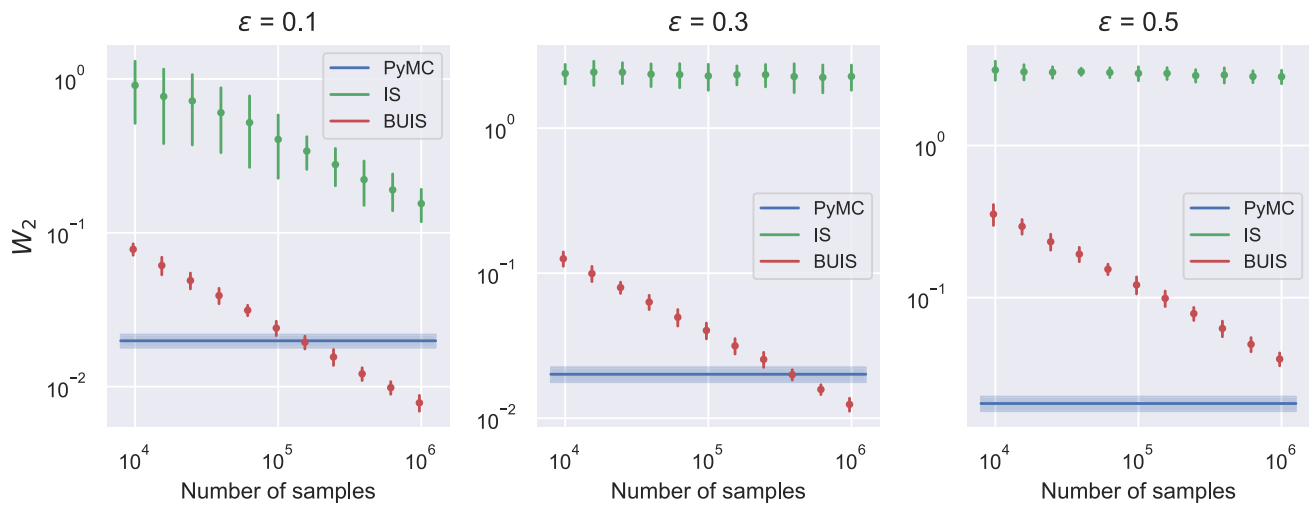


Fig. 4 Wasserstein distance between true and empirical distributions (Gaussian case, large hierarchy). The axes are logarithmic

Table 5 Average percentage errors on the reconciled mean (Gaussian, large hierarchy)

		ϵ		
		0.1	0.3	0.5
IS	10^4 samples	6.72%	17.13%	23.96%
	10^5 samples	2.78%	16.05%	21.61%
	10^6 samples	1.0%	16.25%	20.11%
BUIS	10^4 samples	0.48%	0.65%	1.7%
	10^5 samples	0.15%	0.21%	0.52%
	10^6 samples	0.05%	0.07%	0.18%
<i>PyMC</i>	4×5000 samples	0.09%	0.07%	0.07%

Table 6 Average computational times (Gaussian, large hierarchy). The average time for *PyMC* (4 chains with 5,000 samples each) is 102.56 ± 29.93

		Number of samples		
		10^4	10^5	10^6
IS		0.02 ± 0.00	0.17 ± 0.03	1.74 ± 0.37
BUIS		0.10 ± 0.01	0.90 ± 0.15	13.1 ± 2.19

to as Mean Logarithm Deviation (Haughton and Khandker 2009). Moreover, from (10), we have that the importance sampling weights are IID copies of W . Hence, the more distant are the base and the reconciled distribution, in terms of Kullback-Leibler divergence, the more dispersed are the IS weights. A large dispersion of the weights leads to a poor performance of importance sampling (Martino et al. 2017). As the incoherence level ϵ grows, the distance between the distributions of $\mathbf{A}\hat{\mathbf{B}}$ and $\hat{\mathbf{U}}$ grows, and therefore also the distance between \hat{v}_b and \tilde{v} , as the reconciled distribution merges the information coming from the bottom and the upper variables.

References

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., Stuart, A.M.: Importance sampling: intrinsic dimension and computational cost. *Stat. Sci.* **32**, 405–431 (2017)

Athanasopoulos, G., Hyndman, R.J., Kourntzes, N., Petropoulos, F.: Forecasting with temporal hierarchies. *Eur. J. Oper. Res.* **262**(1), 60–74 (2017)

Azzimonti, D., Rubattu, N., Zambon, L., Corani, G.: bayesRecon: Probabilistic Reconciliation via Conditioning, (2023). R package version 0.1.2

Billingsley, P.: Probability and measure. Wiley, New York (2008)

Chen, Y.-C.: A tutorial on kernel density estimation and recent advances. *Biostat. Epidemiol.* **1**(1), 161–187 (2017)

Çinlar, E.: Probability and stochastics, vol. 261. Springer, Berlin (2011)

Corani, G., Azzimonti, D., Augusto, J.P., Zaffalon, M.: Probabilistic reconciliation of hierarchical forecast via Bayes’ rule. In Proc. European Conf. On Machine Learning and Knowledge Discovery in Database ECML/PKDD, vol. 3, pp. 211–226, (2020)

Corani, G., Azzimonti, D., Rubattu, N.: Probabilistic reconciliation of count time series. *Int. J. Forecast.* (2023). <https://doi.org/10.1016/j.ijforecast.2023.04.003>

Di Fonzo, T., Girolimetto, D.: Cross-temporal forecast reconciliation: optimal combination method and heuristic alternatives. *Int. J. Forecast.* **39**, 39–57 (2021)

Di Fonzo, T., Girolimetto, D.: Forecast combination-based forecast reconciliation: insights and extensions. *Int. J. Forecast.* (2022). <https://doi.org/10.1016/j.ijforecast.2022.07.001>

Elvira, V., Martino, L.: Advances in importance sampling. Wiley, New York (2021)

Gneiting, T.: Quantiles as optimal point forecasts. *Int. J. Forecast.* **27**(2), 197–207 (2011)

Haario, H., Saksman, E., Tamminen, J.: An adaptive metropolis algorithm. *Bernoulli*, pp. 223–242, (2001)

Haughton, J., Khandker, S.R.: Handbook on poverty+ inequality. World Bank Publications, Washington, D.C. (2009)

Hoffman, M.D., Gelman, A., et al.: The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **15**(1), 1593–1623 (2014)

Hollyman, R., Petropoulos, F., Tipping, M.E.: Understanding forecast reconciliation. *Eur. J. Oper. Res.* **294**(1), 149–160 (2021)

Hyndman, R.: Another look at forecast-accuracy metrics for intermittent demand. *Foresight Int. J. Appl. Forecast.* **4**(4), 43–46 (2006)

- Hyndman, R., Athanasopoulos, G.: Forecasting: principles and practice, 3rd edition. OTexts: Melbourne, Australia, (2021). [OTexts.com/fpp3](https://otexts.com/fpp3)
- Hyndman, R., Koehler, A.B., Ord, J.K., Snyder, R.D.: Forecasting with exponential smoothing: the state space approach. Springer, Berlin (2008)
- Hyndman, R.J.: expsmooth: Data sets from “Exponential smoothing: a state space approach” by Hyndman, Koehler, Ord and Snyder (Springer, 2008), (2018). URL <http://pkg.robjhyndman.com/expsmooth>. R package version 2.4
- Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L.: Optimal combination forecasts for hierarchical time series. *Comput. Stat. Data Anal.* **55**(9), 2579–2589 (2011). (ISSN 0167-9473)
- Jeon, J., Panagiotelis, A., Petropoulos, F.: Probabilistic forecast reconciliation with applications to wind power and electric load. *Eur. J. Oper. Res.* **279**(2), 364–379 (2019)
- Kahn, H.: Random sampling (Monte Carlo) techniques in neutron attenuation problems. *I. Nucleonics (US) Ceased publication*, 6, (1950)
- Kolassa, S.: Evaluating predictive count data distributions in retail sales forecasting. *Int. J. Forecast.* **32**(3), 788–803 (2016)
- Kolassa, S.: Do we want coherent hierarchical forecasts, or minimal MAPEs or MAEs? (We won’t get both!). *Int. J. Forecast.* **39**(4), 1512–1517 (2023)
- Kourentzes, N., Athanasopoulos, G.: Elucidate structure in intermittent demand series. *Eur. J. Oper. Res.* **288**(1), 141–152 (2021)
- Kullback, S., Leibler, R.A.: On information and sufficiency. *Annals Math. Stat.* **22**(1), 79–86 (1951)
- Liboschik, T., Fokianos, K., Fried, R.: tscount: an R package for analysis of count time series following generalized linear models. *J. Stat. Softw.* **82**(5), 1–51 (2017)
- Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M5 competition: background, organization, and implementation. *Int. J. Forecast.* **38**(4), 1325–36 (2021)
- Martino, L., Elvira, V., Louzada, F.: Effective sample size for importance sampling based on discrepancy measures. *Signal Process.* **131**, 386–401 (2017). <https://doi.org/10.1016/j.sigpro.2016.08.025>
- Panagiotelis, A., Gamakumara, P., Athanasopoulos, G., Hyndman, R.J.: Probabilistic forecast reconciliation: properties, evaluation and score optimisation. *Eur. J. Oper. Res.* **306**(2), 693–706 (2023). <https://doi.org/10.1016/j.ejor.2022.07.040>
- Panaretos, V.M., Zemel, Y.: Statistical aspects of Wasserstein distances. *Annual Rev. Stat. Appl.* **6**, 405–431 (2019)
- Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan kaufmann, Massachusetts (1988)
- Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: DeepAR: probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **36**(3), 1181–1191 (2020)
- Salvatier, J., Wiecki, T.V., Fonnesbeck, C.: Probabilistic programming in Python using PyMC3. *PeerJ Comput. Sci.* **2**, e55 (2016)
- Smith, A.F., Gelfand, A.E.: Bayesian statistics without tears: a sampling-resampling perspective. *Am. Stat.* **46**(2), 84–88 (1992)
- Syntetos, A.A., Boylan, J.E.: The accuracy of intermittent demand estimates. *Int. J. Forecast.* **21**(2), 303–314 (2005). (ISSN 0169-2070)
- Székely, G.J., Rizzo, M.L.: Energy statistics: a class of statistics based on distances. *J. Stat. Plan. Inference* **143**(8), 1249–1272 (2013)
- Taieb, S.B., Taylor, J.W., Hyndman, R.J.: Hierarchical probabilistic forecasting of electricity demand with smart meter data. *J. Am. Stat. Assoc.* **116**(533), 27–43 (2021)
- Wickramasuriya, S.L.: Probabilistic forecast reconciliation under the Gaussian framework. *J. Bus. Econ. Stat.* (2023). <https://doi.org/10.1080/07350015.2023.2181176>
- Wickramasuriya, S.L., Athanasopoulos, G., Hyndman, R.J.: Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *J. Am. Stat. Assoc.* **114**(526), 804–819 (2019)
- Wickramasuriya, S.L., Turlach, B.A., Hyndman, R.J.: Optimal non-negative forecast reconciliation. *Stat. Comput.* **30**(5), 1167–1182 (2020)
- Yang, M., Zamba, G., Cavanaugh, J.: ZIM: Zero-Inflated Models (ZIM) for Count Time Series with Excess Zeros, (2018). URL <https://CRAN.R-project.org/package=ZIM>. R package version 1.1.0

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.