Large Scale Hierarchical Industrial Demand Time-Series Forecasting incorporating Sparsity

Harshavardhan Kamarthi
Georgia Institute of Technology
Atlanta, USA
hkamarthi3@gatech.edu

Aditya B. Sasanur
Georgia Institute of Technology
Atlanta, USA
asasanur@gatech.edu

Xinjie Tong
The Dow Chemical Company
Houston, USA
xtong1@dow.com

Xingyu Zhou
The Dow Chemical Company
Houston, USA
xzhou14@dow.com

James Peters
The Dow Chemical Company
Midland, USA
japeters@dow.com

Joe Czyzyk
The Dow Chemical Company
Midland, USA
jczyczyk@dow.com

B. Aditya Prakash
Georgia Institute of Technology
Atlanta, USA
badityap@cc.gatech.edu

ABSTRACT

Hierarchical time-series forecasting (HTSF) is an important problem for many real-world business applications where the goal is to simultaneously forecast multiple time-series that are related to each other via a hierarchical relation. Recent works, however, do not address two important challenges that are typically observed in many demand forecasting applications at large companies. First, many time-series at lower levels of the hierarchy have high sparsity i.e., they have a significant number of zeros. Most HTSF methods do not address this varying sparsity across the hierarchy. Further, they do not scale well to the large size of the real-world hierarchy typically unseen in benchmarks used in literature. We resolve both these challenges by proposing HAILS, a novel probabilistic hierarchical model that enables accurate and calibrated probabilistic forecasts across the hierarchy by adaptively modeling sparse and dense time-series with different distributional assumptions and reconciling them to adhere to hierarchical constraints. We show the scalability and effectiveness of our methods by evaluating them against real-world demand forecasting datasets. We deploy HAILS at a large chemical manufacturing company for a product demand forecasting application with over ten thousand products and observe a significant 8.5% improvement in forecast accuracy and 23% better improvement for sparse time-series. The enhanced accuracy and scalability make HAILS a valuable tool for improved business planning and customer experience.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; Multi-task learning; **Neural networks**.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08.
https://doi.org/10.1145/3637528.3671632

KEYWORDS

Hierarchical Forecasting, Time-series Forecasting, Probabilistic Forecasting

ACM Reference Format:
Harshavardhan Kamarthi, Aditya B. Sasanur, Xinjie Tong, Xingyu Zhou, James Peters, Joe Czyzyk, and B. Aditya Prakash. 2024. Large Scale Hierarchical Industrial Demand Time-Series Forecasting incorporating Sparsity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3671632

1 INTRODUCTION

Hierarchical time-series forecasting is a problem that profoundly influences decision-making across various domains. These time-series data possess inherent hierarchical relationships and structures [1, 9]. Instances of such situations include predicting employment trends [23] across diverse geographical scales, forecasting the spread of epidemics [20], etc. When dealing with time-series datasets that exhibit underlying hierarchical dependencies, the objective of hierarchical time-series forecasting is to generate precise forecasts for all individual time-series while capitalizing on the hierarchical interconnections among them [8]. For instance, at a large manufacturing company, forecasting demand at various levels of aggregation is important [3]. Forecasts at a middle level of the business hierarchy are important for procuring raw materials and determining the amount of intermediate materials (or product families) to produce in the medium-term. Near-term forecasts at lower levels of the hierarchy relate more to specific products and even package sizes that are needed. Additionally, companies do not forecast based solely using historical data but include external variables (such as macroeconomic forecasts which incorporate reasonable assumptions about the future) to improve demand forecasts.

Previous forecasting methods have not typically placed an emphasis on providing well-calibrated probabilistic forecasts that model uncertainty. Instead, traditional methods have primarily concentrated on providing single-point predictions. In contrast,

recent post-processing techniques [2, 23, 26] refine forecast distributions generated by independent base models as a preprocessing step. These post-processing methods offer the advantage of being readily applicable to forecasts generated by various models and are usually simple to implement and tractable even for large-scale datasets with thousands of time-series in the hierarchy. However, they fall short in enabling the base forecasting models to grasp the intricate hierarchical relationships among time-series data within the hierarchy.

In contrast, end-to-end learning neural methods have taken a more direct approach by incorporating hierarchical relationships as an integral part of either the model architecture [19] or learning algorithm [6]. These comprehensive end-to-end approaches tend to surpass post-processing methods by imposing hierarchical constraints on forecast distribution parameters, such as the mean or fixed quantiles. Most end-to-end methods do not consistently enforce hierarchical coherence across the entirety of the distribution. Some recent methods [6] do impose some distributional constraints such as across specific quantiles. ProfHiT [13] is capable of generating well-calibrated forecasts by imposing hierarchical constraints on the forecast distributions. Further, large scale industrial demand time-series exhibit a range of distributional behavior across the hierarchy [24]. Importantly, many time-series corresponding to individual products for specific customers at lower levels have high sparsity due to infrequent demand [10, 18]. This can be attributed to various factors, such as the seasonality, novelty, or niche appeal of products at these levels. However, time-series at higher levels show much less sparsity being an aggregation of multiple time-series at lower levels. Post-processing methods, due to their inability to transfer information across base forecasts cannot capture this wide range of behavior and overcome this just by reconciliation at post-processing. These methods are not designed to model time-series of different sparsity simultaneously and also provide subpar performance.

Motivated by a real-world use-case at a large scale chemical company, we propose HAILS (Hierarchical forecasting with Adaptation for Industrial and Large Sparse time-series), a novel hierarchical forecasting framework that is both scalable and capable of generating well-calibrated forecasts with precise uncertainty measurements. We overcome this challenge by proposing to model the lower-level forecasts and higher-level forecasts using appropriate distributions. At levels of the hierarchy that have a mixture of both distributions, we use distributional approximations to have uniform distributions across subtree when reconciling the forecasts. To accomplish this, we propose a novel loss function that enables the model to adapt to sparse time-series data at lower levels of the hierarchy while being able to reconcile with denser time-series at higher levels. We summarize our contributions as follows:

- **Efficient Large Scale Probabilistic Hierarchical Forecasting**: We propose a novel hierarchical forecasting framework that is both scalable and capable of generating well-calibrated forecasts with reliable uncertainty measurements for large industrial time-series.
- **Adaptation to Time-series of different levels of sparsity**: We propose modelling the lower level forecasts and higher level forecasts using different distributions based on historical sparsity and propose a novel framework to reconcile them.
- **State-of-art performance on large datasets**: We demonstrate the effectiveness of our proposed method on large-scale demand datasets with thousands of time-series in the hierarchy. We evaluate on a public dataset as well as a proprietary dataset from a large chemical manufacturing company. We show that our method outperforms the state-of-art methods across most levels of the hierarchy both in terms of accuracy of point forecasts and probabilistic forecasts. We perform a detailed case study to demonstrate the impact of our proposed method on a real-world application at a large chemical company.

2 RELATED WORKS

Classical methods in hierarchical time-series forecasting traditionally employed a two-phase method, concentrating on point predictions [8, 9]. These methods predicted time-series at a singular hierarchy level, then extrapolated forecasts to other levels using hierarchical relationships. Recent techniques, such as MinT and ERM, act as post-processing procedures, refining forecasts across all hierarchy levels. MinT [25, 26] operates under the assumption that baseline forecasts are independent and unbiased, aiming to minimize forecast error variance from historical data. ERM [2] modifies this by not assuming unbiased forecasts.

Recent neural network approaches offer more end-to-end learning of patterns of individual time-series as well as hierarchical relations across time-series. Rangapuram et al. [19] adopt a strategy of projecting the forecasts into a subspace of reconciled forecasts via a differentiable operation and optimize the loss on the projected forecasts. SHARQ [6] represents another novel deep-learning probabilistic method that employs quantile regression and regularizes consistency across various forecast distribution quantiles. ProfHiT [13] imposes hierarchical constraints on the forecast distributions by minimizing distributional distance between the parent forecast and the sum of the child forecasts. However, none of the methods are designed to adapt to sparse time-series and therefore perform sub-optimally on real-world industrial demand time-series.

3 PROBLEM STATEMENT

We denote the dataset $\mathcal{D}$ of N time-series over the time horizon $1, 2, \ldots, T$. Let $\mathbf{y}_i \in \mathrm{R}^T$ be time-series i and $y_i^{(t)}$ its value at time t. The hierarchical relations across time-series is denote as $\mathcal{T} = (G_{\mathcal{T}}, H_{\mathcal{T}})$ where $G_{\mathcal{T}}$ is a tree of N nodes rooted at time-series 1 (time-series 1 is the aggregate of all leaf time-series). Consider a non-leaf node (time-series) i with children C_i. The hierarchical relations are of the form $H_{\mathcal{T}} = \{\mathbf{y}_i = \sum_{j \in C_i} \phi_{ij} \mathbf{y}_j : \forall i \in \{1, 2, \ldots, N\}, |C_i| > 0\}$ where values of ϕ_{ij} are constant and known.

Our problem can be formulated as follows: Given a dataset $\mathcal{D}$ with underlying hierarchical relations $H_{\mathcal{T}}$, we learn a model M that provides *accurate* probabilistic forecast distributions $\{p_M(y_1^{(t+1)}|\mathcal{D}^t), \ldots p_M(y_N^{(t+\tau)}|\mathcal{D}^t)\}$ across all levels of the hierarchy where τ is the forecast horizon.

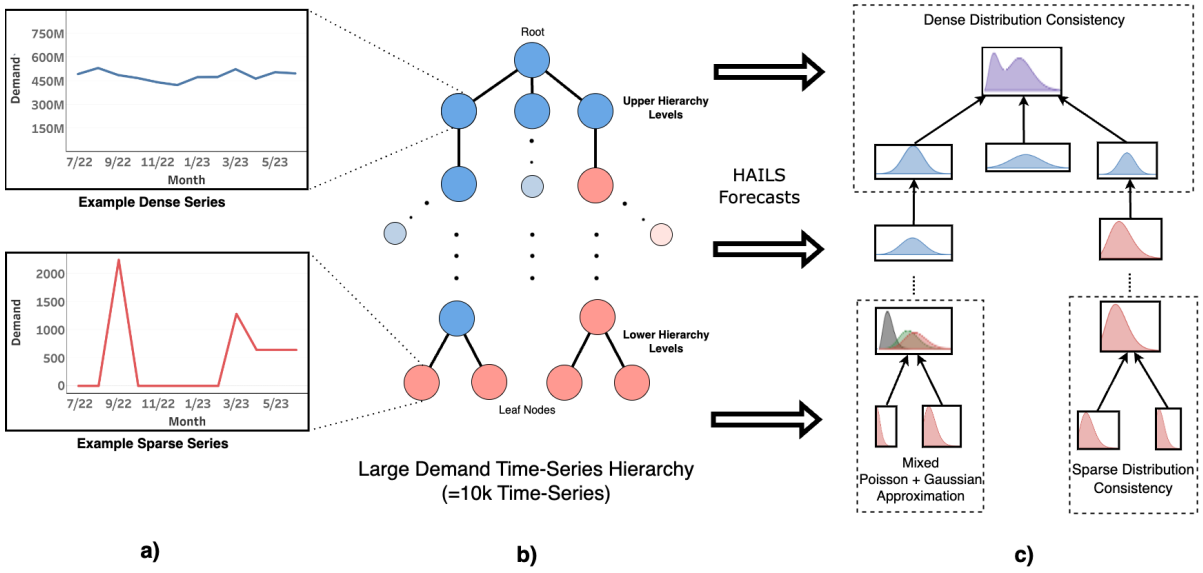

Figure 1: Overview of pipeline of HAILS. (a) The lower levels of the hierarchy tend to have sparse (red) time-series while the higher levels have denser (blue) time-series. (b) HAILS first generates forecasts for each of the time-series of hierarchy with their parametric form depending on the sparsity of the time-series. The denser time-series forecasts are modeled as Gaussians and sparser ones as Poisson. (c) The distributions are reconciled via a distribution consistency loss for each subtree. If the subtree has all distribution same the appropriate loss is applied. In case of mixed subtrees, Poisson distribution of children are first approximated as Gaussian.

4 METHODOLOGY

Most hierarchical forecasting models struggle to adapt to large hierarchies found in real-world industrial applications both in terms of effectiveness and efficiency in learning from tens of thousands of time-series as well as adapting to sparse time-series at the lower levels of the hierarchy. HAILS overcomes these challenges in two ways. First, we make architectural design choices to enable more efficient in learning from larger hierarchies as well as support sparse time-series. Second, we develop optimization methods to allow for learning accurate and consistent forecasts both for sparse and dense forecasts at different levels of the hierarchy. We first provide a brief overview of ProfHiT, state-of-art hierarchical forecasting model and then discuss in detail our innovations to enable dealing with these challenges.

4.1 Probabilistic Hierarchical Forecasting

ProfHiT [13] is a state of the art probabilistic forecasting model that. It optimizes the full distribution of forecasts of the hierarchy to be both accurate and consistent with the hierarchical constraints. It first produces base forecasts for each node of the hierarchy independently via a differentiable neural model. The authors of ProfHiT chose to use CaMuL [12], a state of the art neural probabilistic forecasting model to produce the base forecasts parameterized by normal distribution $\{(\mu_i, \sigma_i)\}_{i=1}^{N}$. The base forecast parameters are used as prior distribution parameters to generate refined distribution that leverage inter-series relations and hierarchical constraints to produce the refined parameters $\{(\mu_i, \sigma_i)\}_{i=1}^{N}$. This is achieved by the *Hierarchy-aware Refinement Module* and the whole model is trained on both the Log-likelihood loss for *accuracy* and Soft Distributional Consistency Regularization (SDCR) for *Distributional Consistency* by minimizing the Distributional Consistency Error defined as follows:

Definition 1. *(Distributional Consistency Error) [13] Given the forecasts at time $t+\tau$ as $\{p_M(y_1^{(t+\tau)}|\mathcal{D}^t), \ldots p_M(y_N^{(t+\tau)}|\mathcal{D}^t)\}$ distributional consistency error (DCE) is defined as*

$$\sum_{i\in\{1,\ldots,N\}, C_i\neq\emptyset} Dist\left(p_M(y_i^{(t+\tau)}|\mathcal{D}^t), p_M(\sum_{j\in C_i}\phi_{i,j}y_j^{(t+\tau)}|\mathcal{D}^t)\right) \quad (1)$$

where Dist is a distributional distance metric.

The final forecasts $\{P(y_i^{(t+\tau)}|\mathcal{D}^t)\}_{i=1}^{N}$ are thus optimized to be accurate and distributionally consistent across the hierarchy.

4.2 HAILS: Forecasting for large hierarchies with sparse time-series

HAILS models sparse and dense time-series using appropriate distributions when forecasting. A key challenge which we overcome in the process is to enable learning consistent forecasts in cases where parts of the hierarchy have both sparse and dense time-series. To enable this feature, HAILS proposes important architectural changes to ProfHiT and a novel loss: *Distributional consistency regularization with Sparse adaptation*. We describe the various modules of HAILS as follows.

4.2.1 Testing for Poisson Distribution. We first determine whether we should model a given node of the hierarchy as a sparse time-series. We use the Poisson distribution to model the time-series if it is deemed sparse since we can model the high probability of observing zeros. Therefore, to systematically classify the data from a given node of the hierarchy as sparse, we use the *Poisson dispersion test* on samples from training data of the time-series. Intuitively, the dispersion test tests if the mean and variance of the data samples are similar. We observe that using a p value threshold of 0.1 is a good measure to classify nodes as *sparse* or *dense*. We also make sure that the parents of a node classified as dense are automatically dense. We observe this to be always true for our benchmark datasets. But, in case it does not hold, we explicitly classify the parents as dense. Notationally, we denote all nodes in $\{1, \ldots, N\}$ that are classified as sparse as $\mathbb{S}$.

4.2.2 Base forecasting model. The requirements for choosing a base forecasting model are based on the application's specific needs. ProfHiT uses CaMuL [11, 12] due to its superior performance in terms of accuracy and uncertainty quantification. However, it makes deploying to large industrial hierarchies infeasible. First, CaMuL is a stochastic model [16] that leverages multiple sampling components that makes stable training a hard technical challenge when scaling it to train tens of thousands of time-series independently. Secondly, it requires significant amount of historical data that it uses as *reference points* to map similar patterns from historical data to current input time-series for uncertainty quantification. Along with the challenge of the high compute requirement of storing and embedding these historical time-series, in many real-world applications we do not have sufficient historical data to learn reliably. Finally, CaMuL is not designed to model sparse predictions and instead parameterized the output as a Gaussian. Instead, we chose a simpler model: a Gated Recurrent Unit (GRU) neural network, a widely adopted recurrent deep learning model. Depending on the nature of the node, the base forecasts output the forecast parameter. For a node $i \notin \mathbb{S}$ classified as dense, it outputs two parameters of the normal distribution: $(\mu_i, \exp(\sigma_i))$. For any node $j \in \mathbb{S}$ classified as sparse, it simply outputs only the Poisson mean parameter: $\lambda_j = \mu_j$.

4.2.3 Hierarchy-aware Refinement Module. This module uses the base forecasts from RNNs and refined them to 1) leverage information of the time-series across the hierarchy 2) enables them to be distributionally consistent by training on the SDCR. Let $\mu = [\mu_1 \ldots, \mu_N]$ be a vector of means of base distributions for all nodes. $\hat{\mu}_i$ is the weighted sum of μ_i and base mean of all time-series:

$$\gamma_i = \text{sigmoid}(\hat{w}_i), \quad \hat{\mu}_i = \gamma_i \mu_i + (1 - \gamma_i)\mathbf{w}_i^T \mu \tag{2}$$

where $\{\hat{w}_i\}_{i=1}^N$ and $\{\mathbf{w}_i\}_{i=1:N}$ are parameters of the model and sigmoid$(\cdot)$ denotes the sigmoid function. γ intuitively denotes the tradeoff between relying on the base mean and information from rest of the distribution. Let $\sigma = \{\sigma_i | i \notin \mathbb{S}\}$ be a vector of variances for dense nodes' base forecasts. The variance parameter $\hat{\sigma}_i$ of the refined distribution is derived from the base distribution parameters

$$\hat{\sigma}_i = c\sigma_i \text{sigmoid}(\mathbf{v}_{1i}^T \mu + \mathbf{v}_{2i}^T \sigma + b_i) \tag{3}$$

where $\{\mathbf{v}_{1i}\}_{i=1}^N$, $\{\mathbf{v}_{2i}\}_{i=1}^N$ and $\{b_i\}_{i=1}^N$ are parameters and c is a positive constant hyperparameter.

4.2.4 Soft Distributional Consistency Regularization. ProfHiT learns to generate forecasts that are distributionally consistent by introducing SDCR. It forces the model to minimize the Distributional Consistency Error across the forecasts of the hierarchy leading to the aggregated forecasts of the children being similar to the parent forecast. However, SDCR only deals with dense time-series since it models them as Gaussians. Moreover. it cannot deal with different types of distributions across the hierarchy. Therefore, SDCR cannot be directly applied to hierarchies that have sparse time-series ($\mathbb{S} \neq \Phi$). HAILS introduces *Distributional Consistency Regularization with Sparse adaptation* (DCRS) that allows for hierarchies with varying time-series sparsities to provide distributional consistency. DCRS applies different consistency losses across the subtrees of the hierarchies based on the the sparsity of the parents and children. We specifically look at three cases that are observed in the hierarchies:

Dense Parent-Dense Children: If the parent node as well as children nodes are dense, we use the same distributional consistency loss as ProfHiT: we model the parent and children forecast distributions as gaussians and compute the Jenson-Shannon divergence:

$$\mathcal{L}_{DCRS}^{(i)} = JSD(\mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i) | \mathcal{N}\left(\sum_{j \in C_i} \phi_{ij}\hat{\mu}_j, \sqrt{\sum_{j \in C_i} \phi_{ij}^2 \hat{\sigma}_j^2}\right)) =$$
$$\sum_{i=1}^{N} \frac{\hat{\sigma}_i^2 + \left(\hat{\mu}_i - \sum_{j \in C_i} \phi_{ij}\hat{\mu}_j\right)^2}{4\sum_{j \in C_i} \phi_{ij}^2 \hat{\sigma}_j^2} + \sum_{i=1}^{N} \frac{\sum_{j \in C_i} \phi_{ij}^2 \hat{\sigma}_j^2 + \left(\hat{\mu}_i - \sum_{j \in C_i} \phi_{ij}\hat{\mu}_j\right)^2}{4\hat{\sigma}_i^2} \tag{4}$$

Sparse Parent- Sparse Children: To calculate the distributional consistency error of sparse time-series at lower levels of the hierarchy we note that we assume the forecasts are Poisson distributions. The JSD between two Poissons has a closed form solution:

$$\mathcal{L}_{DCRS}^{(i)} = JSD\left(\lambda_i | \sum_{j \in C_i} \lambda_j\right) = \lambda_i \log\left(\frac{\lambda_1}{\sum_{j \in C_i} \lambda_j}\right) + \sum_{j \in C_i} \lambda_j \log\left(\frac{\sum_{j \in C_i} \lambda_j}{\lambda_i}\right). \tag{5}$$

Mixed Subtrees: Now we examine the case where the parent is a dense node but some or all of her children are sparse. We note that as we go further up the hierarchy, sparsity of time-series decreases. Therefore, the sparsity assumption on these nodes gets weaker. We therefore propose to approximate the sparse forecasts of these time-series as Gaussian distributions and apply Eq. 4 to optimize for distributional consistency. We perform the approximation leveraging the central limit theorem as follows:

Theorem 1. *Let $X_1, X_2, \ldots, X_N$ be N independent Poisson random variables with parameters $\lambda_1, \lambda_2, \ldots, \lambda_N$. Then denote Y as $Y = \sum_{i=1}^N X_i$. Then Y is a Poisson variable with parameter $\lambda_Y = \sum_{i=1} \lambda_i$. Then for sufficiently large λ_Y, Y can be approximated by a Gaussian distribution $\tilde{Y} = \mathcal{N}(\lambda, \sqrt{\lambda})$ [15].*

Therefore, all the children time-series forecasts of sparse node j of form λ_j are converted to normal distribution $\mathcal{N}(\lambda_j, \sqrt{\lambda})$. Then, once all the forecast distributions are modeled as Normal distributions, we apply Eq. 4.

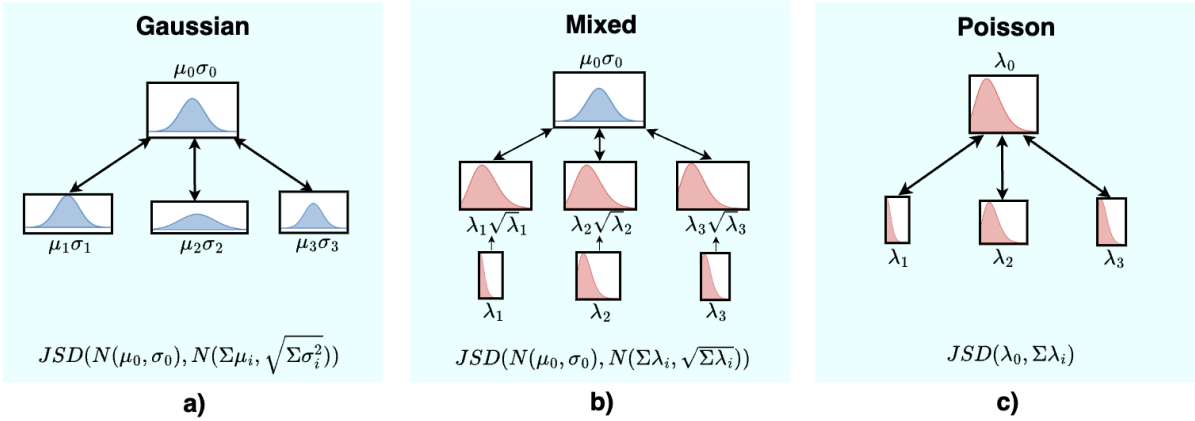

Figure 2: For homogeneous subtrees of Normal and Poisson distribution (a,c), JSD divergence loss is applied directly. In case of heterogeneous subtrees (b), Poisson distributions are first approximated as gaussians and then JSD is applied across resultant gaussians.

We summarize the three cases of performing distributional consistency in Figure 2. The total DCRS loss is denoted as:

$$\mathcal{L}_{DCRS} = \sum_{i:|C_i|>0} \mathcal{L}^{(i)}_{DCRS} \tag{6}$$

4.2.5 Other Training details. Likelihood loss. Similar to ProfHiT, we use a Log-likelihood loss along with the Distributional coherency loss over all the nodes of the hierarchy. The log-likelihood loss $\mathcal{L}_{LL}$ is trained on the refined parameters of the forecast distribution and summed across all nodes. The total loss is $\mathcal{L} = \mathcal{L}_{LL} + \gamma\mathcal{L}_{DCRS}$ where the hyperparameter γ dictates the relative importance of the importance of DCRS loss.

Pre-train base RNN models: Pre-training the weights of some of the layers of a neural model to improve the overall efficiency and convergence of a model is a well-known effective technique [7]. We therefore, first train the only RNN modules for each node independently for point forecasting for small (about 50) epochs before we train all the modules of HAILS for hierarchical probabilistic forecasting.

Hyperparameters We use a bidirectional GRU with 60 hidden units for all nodes of the time-series. For each node we use a 80-20 train validation split to tune the hyperparameters. For training we use batch size of 32 and learning rate of 0.001. We use early stopping to determine number of epochs to train. We observed that HAILS usually converges within 200 epochs for both datasets.

Data Preprocessing We first normalize the data as follows: For each non-leaf time-series we divide the time-series value by number of children. Then we use the weights $\phi_{ij} = \frac{1}{|C_i|}$ for hierarchical relations. This is so that the higher levels of the hierarchy do not hoave very large values as inputs ot the model to enable stable training.

5 EXPERIMENTS

5.1 Setup

We evaluate HAILS against top hierarchical forecasting baselines on two large hierarchical demand forecasting benchmarks. We evaluated all models on a system with Intel 64-core Xeon Processor with 128 GB memory and Nvidia Tesla V100 GPU with 32 GB VRAM. We provide our implementation of HAILS at https://github.com/AdityaLab/HAILS. We used PyTorch for training neural networks and Numpy for other data processing steps.

5.1.1 Datasets. While most hierarchical forecasting benchmarks consist of small hierarchies with all time-series being dense, we choose to evaluate on two benchmarks for our specific application: large hierarchies for demand forecasting. We choose one public dataset and also evaluate on a proprietary real-world use-case of product demand forecasting at a large chemical company.

M5 dataset: M5 forecasting competition featured a monthly retail sales forecasting dataset with hierarchically structured sales data with intermittent and erratic characteristics [18, 22]. The dataset had 12 levels of hierarchy and consisted of 3914 time-series in total. The forecast horizon was up to 28 months ahead.

Dow Demand forecasting: The dataset contains monthly historical sales (in volume) from January 2018 to June 2023 made by Dow in 10+ major industries across 160+ countries. The dataset has a hierarchical structure where the top levels represents the aggregated sales at the country and industry levels, and the lower levels contain the sales data in more granular product classes. The historical sales from January 2018 to June 2022 along with external business indicators were used to train the model. Product demand forecasts were generated for July 2022 to June 2023, and the actual sales during this period are used as ground truth. The forecast horizon was 12 months ahead with the following hierarchical structure:

5.1.2 Baselines. We compare HAILS's performance against state-of-the-art hierarchical forecasting methods as well as generic time-series forecasting methods. We first compare against a standard heuristic of averaging past 6 months' values (6-Average). ARIMA [17] is a commonly used statistical time-series models. We also use GRU [4] without any reconciliation as a common neural RNN-based forecasting baseline. For GRU, we used Monte-Carlo dropout [5] to generate multiple forecast samples for probabilistic forecasts. Finally we also considered DeepAR [21], popular deep probabilistic

Table 1: Weighted RMSSE for M5 dataset. HAILS achieves the best (bold) or second-best (underline) performance across most levels of hierarchy.

Model	**Total**	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
Sparsity		0.1	0.1	0.1	0.1	0.2	0.5	3.1	7.4	11.5	21.7	28.3	37.3
6-Average	0.851	0.472	0.534	0.554	0.614	0.693	0.711	0.775	0.913	0.871	0.985	0.986	0.984
ARIMA	0.681	0.271	0.392	0.455	0.491	0.577	0.631	0.695	0.744	0.871	0.989	0.994	0.993
RNN	0.653	0.231	0.337	0.274	0.375	0.413	0.533	0.581	0.572	0.766	0.968	0.984	0.997
DeepAR	0.612	0.216	0.342	0.316	0.322	0.384	0.481	0.529	0.618	0.669	0.873	0.996	0.965
DeepAR-MinT	0.592	0.201	0.317	0.301	0.328	0.356	0.432	0.504	0.628	0.674	0.819	0.959	0.997
DeepAR-ERM	0.585	0.221	0.283	0.275	0.316	0.384	0.442	0.481	0.611	0.629	0.779	0.986	0.969
HierE2E	0.614	0.215	0.291	0.318	0.337	0.397	0.405	0.477	0.656	0.748	0.886	0.924	0.966
SHARQ	0.565	0.24	0.391	0.352	0.425	0.491	0.552	0.591	0.582	0.6864	0.991	0.994	0.981
PEMBU-MINT	0.534	0.23	0.327	0.41	0.342	0.411	0.445	0.481	0.492	0.582	0.991	0.951	0.899
M5-Leader	0.512	**0.199**	0.31	0.422	**0.277**	**0.366**	**0.39**	**0.474**	0.48	0.573	0.966	0.929	0.884
ProfHiT	0.551	0.245	**0.216**	0.316	0.337	0.417	0.432	0.474	**0.439**	0.557	0.849	0.941	0.932
HAILS	**0.502**	0.211	0.233	**0.262**	0.311	0.382	0.416	0.462	0.443	**0.539**	**0.693**	**0.882**	**0.814**

Table 2: Normalized CRPS for M5 dataset. HAILS achieves the best performance across all levels of hierarchy. HAILS achieves the best (bold) or second-best (underline) performance across most levels of hierarchy.

Model	**Total**	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
Sparsity		0.1	0.1	0.1	0.1	0.2	0.5	3.1	7.4	11.5	21.7	28.3	37.3
ARIMA	5.233	2.563	2.742	3.643	4.145	5.672	7.264	7.984	9.335	10.445	13.244	16.264	17.894
RNN	0.442	0.285	0.277	0.293	0.322	0.527	0.766	0.912	0.935	0.982	0.993	0.994	1.144
DeepAR	0.423	0.271	0.269	0.274	0.361	0.481	0.718	0.897	0.917	0.995	0.994	0.986	1.211
DeepAR-MinT	0.492	0.224	0.253	0.248	0.339	0.441	0.683	0.863	0.884	0.956	0.948	0.981	0.974
DeepAR-ERM	0.229	0.214	0.231	0.283	0.318	0.429	0.668	0.822	0.826	0.926	0.956	0.977	0.993
HierE2E	0.126	0.113	0.111	0.117	0.309	0.392	0.592	0.731	0.718	0.885	0.933	0.942	0.942
SHARQ	0.139	0.082	0.283	0.294	0.323	0.388	0.732	0.782	0.811	0.895	0.926	0.993	0.942
PEMBU-MINT	0.126	0.064	0.067	**0.074**	0.298	0.493	0.693	0.750	0.841	0.943	0.972	0.991	0.973
M5-Leader	0.119	**0.029**	0.087	0.188	**0.173**	0.283	0.429	0.572	**0.715**	0.774	0.881	0.893	0.942
ProfHiT	0.132	0.032	0.088	0.163	0.294	0.481	0.622	0.637	0.824	0.872	0.937	0.924	0.982
HAILS	**0.081**	0.030	**0.060**	0.152	0.246	**0.257**	**0.380**	**0.521**	0.763	**0.717**	**0.570**	**0.826**	**0.728**

Level	Time-series	Sparsity
L1 (Area)	4	0%
L2 (Country)	25	0%
L3 (Industry)	418	3.69%
L4 (Business Group)	1111	15.61%
L5	1956	21.05%
L6	3462	32.76%
L7	5459	36.49%
L8	7587	40.93%

Table 3: Number of time-series and sparsity (% of zeros) by Level for Dow time-series.

forecasting models which do not exploit hierarchy relations. Note that 6-Average cannot produce probabilistic forecasts due to its deterministic mechanics.

In the case of hierarchical forecasting, we considered PEMBU [23], the state-of-art post-processing method applied on DeepAR forecasts reconciled by MinT. With respect to the state-of-art neural hierarchical forecasting methods, we compare against SHARQ [6] a deep learning-based approach that reconciles forecast distributions by using quantile regressions and making the quantile values consistent. We also compare against HierE2E [19], a deep learning-based approach that projects the base predictions onto a space of consistent forecasts and trains the model in an end-to-end manner. For the M5 benchmark, we also include the scores from the top submission of the M5 competition, denoted as M5-Leader.

5.1.3 Evaluation Metrics. We evaluate our models and baselines using carefully chosen metrics to measure both point accuracy and probabilistic distribution calibration of the forecasts. For a ground truth $y^{(t)}$, let the predicted probability distribution be $\hat{p}_{y^{(t)}}$ with mean $\hat{y}^{(t)}$. Also let $\hat{F}_{y^{(t)}}$ be the CDF.

•**Weighted Root Mean Squared Scaled Error (WRMSSE)** is a scale-invariant metric for point-predictions that can be used to compare across different time-series of varying scales. RMSSE for a time-series is defined as:

$$RMSSE = \sqrt{\frac{1/N \sum_{t=n}^{n+N} (y^{(t)} - y^{\hat{(t)}})^2}{1/(n-1) \sum_{t=2}^{n} (y^{(t)} - y^{(t-1)})^2}}$$

where N is the forecast horizon and n is the length of the training data. We then weight each of the time-series's RMSSE with the average value of ground truth in training dataset to get the weighted RMSSE. This metric was used in the M5 competition to evaluate the accuracy of point predictions [18].

•**Cumulative Ranked Probability Score (CRPS)** is a widely used standard metric for the evaluation of probabilistic forecasts that measures *both accuracy and calibration*. Given ground truth y and the predicted probability distribution $\hat{p}_y$, let $\hat{F}_y$ be the CDF. Then, CRPS is defined as:

$$CRPS(\hat{F}_y, y) = \int_{-\infty}^{\infty} (\hat{F}_y(\hat{y}) - \mathbf{1}\{\hat{y} > y\})^2 d\hat{y}.$$

We approximate $\hat{F}_y$ as a Gaussian distribution formed from samples of the model to derive CRPS. We normalize the value of CRPS for each time-series by the average value of ground-truth in training data to get *normalized CRPS*.

•**Root Mean Squared Error (RMSE)** is used to calculate the forecast performance at specific time-step since the opther metrics are usually used to calculate over the full forecast horizon.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

where N is the number of observations.

5.2 Results

We evaluate the performance of HAILS on the M5 dataset and then perform a detailed case study showcasing the impact of HAILS on demand forecasting at Dow.

5.2.1 Forecasting performance on M5. We evaluate the forecasting performance at each of the individual levels and across the entire hierarchy in Table 1. We observe that the average performance of HAILS is significantly better than all the other baselines as well as ProfHiT and M5-Leader across the hierarchy as well as in most of the hierarchy levels. Specifically, we observe significant increase of about 12% in performance at lower levels (L10- L12) with sparse time-series over ProfHiT, showcasing the importance of leveraging poisson distributions at the lower level and using the novel DCRS loss. Overall HAILS achieves best or close to best performance at all levels of the hierarchy.

In terms of the performance of probabilistic forecasts (Table 2), we also observe over 40% better CRPS scores of HAILS over ProfHiT and 32% over best baselines with consistently better performance across all levels of the hierarchy. Similarly, we observe a significant 20% better performance at the lower levels of the hierarchy.

5.2.2 Case Study: Demand Forecasting at Dow. Background: At Dow, hierarchical time series models are developed to forecast product demand and raw material price to facilitate business planning. These models offer substantial value to the businesses by minimizing the cost to serve through improved planning and forecasting, thereby enhancing customer experience and relationships. Currently, forecasts are performed by applying Microsoft Azure Auto Machine Learning (AutoML), a cloud-based service that automates the selection and tuning of machine learning models. One of the main drawbacks of this approach is the restriction on the number of predictor variables that can be included in the model, resulting from poor model scalability. In addition, the relationships among different layers in the hierarchy are expected to provide useful insights on the product demand but are not accounted for in the model training (i.e., aggregated data at a higher level of granularity were provided for model training and inference and are disaggregated to lower levels based on proportions derived from historical data). Last but not the least, the lack of transparency, and uncertainty-driven risk assessment associated with the model performance and forecasts impose significant challenges to business decision-making processes.

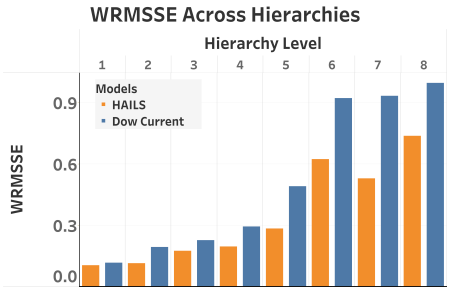

Figure 3: HAILS has significantly lower WRMSSE than Dow baseline across all levels of the hierarchy.

Impact: We developed HAILS to overcome these challenges that are commonly existed in large-scale business planning. HAILS alleviates these crucial challenges: First, it efficiently scales to predict the time-series across all levels of the hierarchy. It additionally leverages DCRS to optimize for distributional consistency according to underlying hierarchical relationships. Finally, being a state-of-art probabilistic model it provides reliable forecast distributions that are both accurate and have dependable uncertainty measures. These benefits allow HAILS to have a vastly lower RMSE across the entire forecast horizon.

Historical demand is used as the criterion to identify the top countries and industries. This is based on the assumption that higher demand corresponds to higher value, and thus more potential for profit. By improving forecast accuracy for these segments, we can optimize our business planning and reduce costs. We summarize the forecasting performance of HAILS, Dow's AutoML baseline and other baselines in Tables 4, 5. HAILS outperforms the previous baseline used by Dow by over 8.5% overall in RMSSE with an average improvement of 26% for the last three layers which have over 10% of the values zeroes (Fig. 3). Similarly, HAILS's CRPS score is 30% better than the best baseline models with over 23% better in the last 3 sparser levels of the hierarchy. The improvement in forecast performance is seen consistently during testing across the year (Fig. 4). We also observe that HAILS is 70% faster to train than the next best model. We also observe 44.14% average improvement in performance for forecasts in the top seven countries and industries identified by magnitude of past demand (Fig. 5). We also visualize few examples forecasts. We also observe that the confidence intervals of the forecasts closely follow the ground truth compared the the Dow baseline (Fig. 6).

Table 4: Weighted RMSSE for Dow dataset. HAILS achieves the best (bold) or second-best (underline) performance across all levels of hierarchy.

Model	**Total**	L1	L2	L3	L4	L5	L6	L7	L8
Sparsity		0	0	3.69	15.61	21.05	32.72	36.49	40.93
6-Average	0.814	0.477	0.492	0.612	0.731	0.855	0.923	0.987	0.985
ARIMA	0.582	0.215	0.225	0.304	0.381	0.592	0.778	0.924	0.973
RNN	0.527	0.187	0.176	0.244	0.287	0.698	0.729	0.988	0.997
DeepAR	0.496	0.143	0.168	0.236	0.305	0.494	0.891	0.973	0.996
DeepAR-MinT	0.483	0.137	0.166	0.226	0.284	0.428	0.842	0.946	0.975
DeepAR-ERM	0.487	0.133	0.144	0.229	0.286	0.441	0.885	0.941	0.936
HierE2E	0.438	0.119	0.142	0.205	0.244	0.428	0.914	0.952	0.983
SHARQ	0.427	0.106	0.153	0.196	0.249	0.448	0.934	0.944	0.955
PEMBU-MINT	0.431	0.126	0.173	0.217	0.257	0.427	0.847	0.933	0.981
Dow Current	0.443	0.117	0.194	0.227	0.294	0.491	0.921	0.932	0.995
ProfHiT	0.421	**0.101**	0.143	0.194	0.218	0.399	0.834	0.873	0.926
HAILS	**0.405**	0.105	**0.115**	**0.175**	**0.196**	**0.284**	**0.623**	**0.529**	**0.737**

Table 5: Normalized CRPS for Dow dataset. HAILS achieves the best (bold) or second-best (underline) performance across all levels of hierarchy.

Model	**Total**	L1	L2	L3	L4	L5	L6	L7	L8
Sparsity		0	0	3.69	15.61	21.05	32.72	36.49	40.93
ARIMA	5.265	3.027	3.335	4.326	4.502	6.120	8.927	9.044	11.952
RNN	0.783	0.308	0.313	0.358	0.393	0.683	0.858	1.116	0.957
DeepAR	0.715	0.302	0.299	0.286	0.393	0.525	0.768	0.968	1.008
DeepAR-MinT	0.218	0.285	0.276	0.264	0.342	0.473	0.705	0.899	0.912
DeepAR-ERM	0.117	0.240	0.254	0.320	0.377	0.431	0.816	0.870	0.830
HierE2E	0.161	0.088	0.284	0.298	0.416	0.513	0.871	0.865	0.881
SHARQ	0.154	0.066	0.073	**0.074**	0.326	0.494	0.873	0.939	0.849
PEMBU-MinT	0.132	**0.032**	0.096	0.228	**0.193**	**0.329**	0.473	0.686	0.937
ProfHiT	0.146	0.033	0.099	0.203	0.342	0.548	0.663	0.754	0.837
HAILS	**0.090**	**0.032**	**0.065**	0.154	0.297	0.480	**0.453**	**0.658**	**0.746**

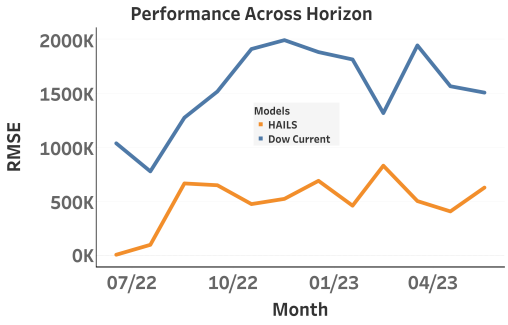

Figure 4: RMSE of HAILS is consistently lower than Dow baseline across the forecast horizon.

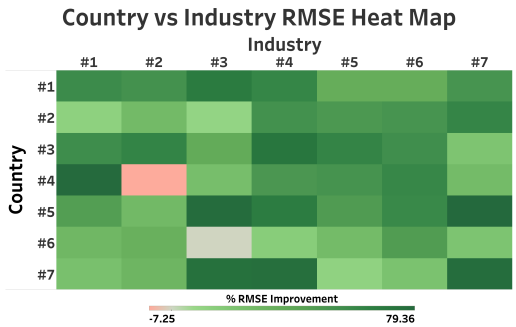

Figure 5: HAILS provides an average of 44.14% improvement over Dow Model over top 7 industries and countries.

5.2.3 Efficiency. HAILS leverages DCSR and Poisson selection to model sparse time-series as Poisson distribution. Moreover, we also leverage pre-training to improve the convergence and training efficiency of the model since HAILS typically takes lesser epochs to achieve state-of-the-art performance.

We measure the total training time in hours for the model and baselines in Table 7. We run the code on workstation with Intel Xeon CPU with 64 cores, 128 GB RAM and a Nvidia V100 GPU with 32GB VRAM. HAILS is more efficient than end-to-end neural models like SHARQ, HierE2E and ProfHiT, finishing training in less than 42% of the total time of the second best baseline for most

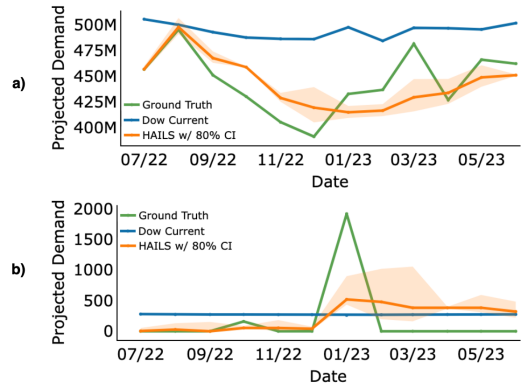

Figure 6: Examples of forecasts of HAILS and Dow baseline for (a) dense and (b) sparse time-series. HAILS's forecasts are much more accurate with uncertainty bands close to the ground truth.

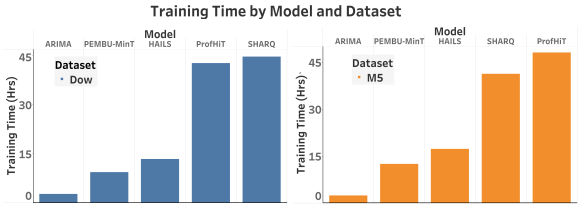

Figure 7: HAILS takes significantly less training time than state-of-art neural baselines like ProfHiT and SHARQ.

of the datasets. This is due to effective modeling of sparse time series as well as asynchronous updates of model weights.

6 CONCLUSION

HAILS is designed to solve challenges motivated by our experience dealing with real-world large scale demand forecasting problem: scalability and modeling sparse time-series across the hierarchy. HAILS improves on ProfHiT to support sparse time-series at lower levels of the hierarchy, an important property of real-world demand forecasting scenario that enables it to perform 8-30% better than previous best baselines with consistent performance across all levels of the hierarchy. HAILS also outperformed the baselines by over 20% in the sparse layers of the hierarchy. Our model design and training enables HAILS to train up to three times more efficiently than similarly sized state-of-art models enabling effective and accurate real-time forecasting. Our model was successfully applied to a real-world application of demand forecasting in one of the world's largest chemical companies and yielded significantly superior performance across the hierarchy. This enables significant reductions in cost due to manufacturing planning, inventory management and fulfillment scheduling.

There are other deployment challenges for HAILS that include data collection, data cleaning, choosing the right hierarchy, explainability and deployment. Collecting reliable data across the hierarchy in a large corporation is complicated by the number of systems, businesses and geographical areas and various product units of measure that need to be standardized. Therefore, building systems that can understand and leverage data quality information to improve the robustness of the forecasts is an important problem [14]. Another important challenge is providing interpretability as block-box neural models are not readily accepted in the business process. Developing reliable interpretability methods for hierarchical forecasting is essential for successful deployment. Additionally, the hierarchy structures may change due to reasons such as reclassifications from one business grouping to another, addition or deletion of products, etc. While we can recalculate the time-series values of the past for new hierarchy, deriving information from a dynamic hierarchy structure is a novel research direction.

ACKNOWLEDGEMENTS

This paper was supported in part by The Dow Chemical Company, the NSF (Expeditions CCF-1918770, CAREER IIS-2028586, Medium IIS-1955883, Medium IIS-2106961, PIPP CCF-2200269), CDC MInD program, Meta faculty gifts, and funds/computing resources from Georgia Tech.

REFERENCES

[1] George Athanasopoulos, Puwasala Gamakumara, Anastasios Panagiotelis, Rob J Hyndman, and Mohamed Affan. 2020. Hierarchical forecasting. *Macroeconomic forecasting in the era of big data: Theory and practice* (2020), 689–719.

[2] Souhaib Ben Taieb and Bonsoo Koo. 2019. Regularized regression for hierarchical forecasting without unbiasedness conditions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1337–1347.

[3] Joos-Hendrik Böse, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Dustin Lange, David Salinas, Sebastian Schelter, Matthias Seeger, and Yuyang Wang. 2017. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1694–1705.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[5] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[6] Xing Han, Sambarta Dasgupta, and Joydeep Ghosh. 2021. Simultaneously Reconciled Quantile Forecasting of Hierarchically Related Time Series. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 190–198.

[7] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[8] Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang. 2011. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis* 55, 9 (2011), 2579–2589.

[9] Rob J Hyndman and George Athanasopoulos. 2018. *Forecasting: principles and practice*. OTexts.

[10] Arindam Jati, Vijay Ekambaram, Shaonli Pal, Brian Quanz, Wesley M Gifford, Pavithra Harsha, Stuart Siegel, Sumanta Mukherjee, and Chandra Narayanaswami. 2023. Hierarchical Proxy Modeling for Improved HPO in Time Series Forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 891–900.

[11] Harshavardhan Kamarthi, Lingkai Kong, Alexander Rodriguez, Chao Zhang, and B Aditya Prakash. 2021. When in Doubt: Neural Non-Parametric Uncertainty Quantification for Epidemic Forecasting. *Thirty-fifth Conference on Neural Information Processing Systems* (2021).

[12] Harshavardhan Kamarthi, Lingkai Kong, Alexander Rodriguez, Chao Zhang, and B Aditya Prakash. 2022. CAMul: Calibrated and Accurate Multi-view Time-Series Forecasting. *ACM The Web Conference (WWW)* (2022).

[13] Harshavardhan Kamarthi, Lingkai Kong, Alexander Rodríguez, Chao Zhang, and B Aditya Prakash. 2023. When Rigidity Hurts: Soft Consistency Regularization for Probabilistic Hierarchical Time Series Forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1057–1072.

[14] Harshavardhan Kamarthi, Alexander Rodríguez, and B Aditya Prakash. 2022. Back2future: Leveraging backfill dynamics for improving real-time predictions in future. *ICLR* (2022).

[15] Scott M Lesch and Daniel R Jeske. 2009. Some suggestions for teaching about normal approximations to poisson and binomial distribution functions. *The American Statistician* 63, 3 (2009), 274–277.
[16] Christos Louizos, Xiahan Shi, Klamer Schutte, and Max Welling. 2019. The functional neural process. *arXiv preprint arXiv:1906.08324* (2019).
[17] Spyros Makridakis and Michele Hibon. 1997. ARMA models and the Box–Jenkins methodology. *Journal of forecasting* 16, 3 (1997), 147–163.
[18] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2022. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting* 38, 4 (2022), 1346–1364.
[19] Syama Sundar Rangapuram, Lucien D Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. 2021. End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series. In *International Conference on Machine Learning*. PMLR, 8832–8843.
[20] Nicholas G Reich, Logan C Brooks, Spencer J Fox, Sasikiran Kandula, Craig J McGowan, Evan Moore, Dave Osthus, Evan L Ray, Abhinav Tushar, Teresa K Yamana, et al. 2019. A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the United States. *Proceedings of the National Academy of Sciences* 116, 8 (2019), 3146–3154.
[21] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.
[22] Aris A Syntetos and John E Boylan. 2005. The accuracy of intermittent demand estimates. *International Journal of forecasting* 21, 2 (2005), 303–314.
[23] Souhaib Ben Taieb, James W Taylor, and Rob J Hyndman. 2017. Coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*. PMLR, 3348–3357.
[24] Ali Caner Türkmen, Tim Januschowski, Yuyang Wang, and Ali Taylan Cemgil. 2021. Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. *Plos one* 16, 11 (2021), e0259764.
[25] Shanika L Wickramasuriya. 2021. Probabilistic forecast reconciliation under the Gaussian framework. *arXiv preprint arXiv:2103.11128* (2021).
[26] Shanika L Wickramasuriya, George Athanasopoulos, and Rob J Hyndman. 2019. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *J. Amer. Statist. Assoc.* 114, 526 (2019), 804–819.