When Rigidity Hurts: Soft Consistency Regularization for Probabilistic Hierarchical Time Series Forecasting

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

Lingkai Kong
College of Computing, Georgia Institute of Technology, USA
lkkong@gatech.edu

Alexander Rodríguez
College of Computing, Georgia Institute of Technology, USA
arodriguezc@gatech.edu

Chao Zhang
College of Computing, Georgia Institute of Technology, USA
chaozhang@gatech.edu

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

ABSTRACT

Probabilistic hierarchical time-series forecasting is an important variant of time-series forecasting, where the goal is to model and forecast multivariate time-series that have hierarchical relations. Previous works assume rigid consistency over the given hierarchies and do not adapt well to real-world data that show deviation from this assumption. Moreover, recent state-of-art neural probabilistic methods also impose hierarchical relations on point predictions and samples of the predictive distribution. This does not account for full forecast distributions being consistent with the hierarchy and leading to poorly calibrated forecasts. We close both these gaps and propose PROFHIT, a probabilistic hierarchical forecasting model that jointly models forecast distributions over the entire hierarchy. PROFHIT (1) uses a flexible probabilistic Bayesian approach and (2) introduces *soft distributional consistency regularization* that enables end-to-end learning of the entire forecast distribution leveraging information from the underlying hierarchy. This enables calibrated forecasts as well as adaptation to real-life data with varied hierarchical consistency. PROFHIT provides 41-88% better performance in accuracy and significantly better calibration over a wide range of dataset consistency. Furthermore, PROFHIT adapts to missing data and can provide reliable forecasts even if up to 10% of input time-series data is missing, whereas other methods' performance severely degrades by over 70%.

CCS CONCEPTS

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

KEYWORDS

Hierarchical Forecasting, Time-series Forecasting, Probabilistic Forecasting

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599547

ACM Reference Format:
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 (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3580305.3599547

1 INTRODUCTION

Time-series forecasting is an important problem that impacts decision-making in a wide range of applications. In many real-world situations, the time-series have inherent hierarchical relations and structures. Examples include forecasting time-series of employment [27] measured at different geographical scales; epidemic forecasting [25] at county, state and country, etc. Given time-series dataset with underlying hierarchical relations, the goal of hierarchical time-series forecasting is to generate an accurate forecast for all time-series leveraging the hierarchical relations between time-series [12].

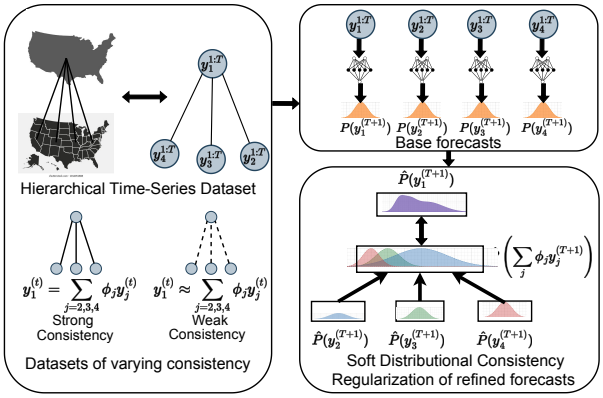


Figure 1: PROFHIT learns to produce accurate and calibrated forecasts from datasets of varying consistency by leveraging underlying hierarchical relations via Soft Distributional Consistency Regularization

Previous hierarchical forecasting methods assume that the dataset is *strongly consistent*: the time-series values of datasets strictly satisfy the underlying hierarchical constraints. Therefore, these models usually impose the generated forecasts to be strongly consistent as

well i.e., forecasts strictly satisfy the hierarchical relations of the dataset. For example, classical *two-step methods* [13] use a bottom-up or top-down approach where all time-series at a single level of the hierarchy are modeled independently and the values of other levels are derived using the aggregation function governing the hierarchy. In contrast, many real-world applications have *weakly consistent* datasets, i.e., the data do not follow the strict constraints of the hierarchy. Such datasets have an underlying data generation process that follows a hierarchical set of constraints but may contain some deviations. These deviations can be caused by factors such as measurement or reporting error, asynchrony in data aggregation and revision pipeline, etc, as frequently observed in epidemic forecasting [1]. Most state-of-the-art methods are designed for applications having strongly consistent datasets by imposing rigid constraints — they thus may not adapt to such deviations and can *provide poor forecasts for application with weakly consistent datasets*

Moreover, previous methods do not focus on providing *calibrated forecasts* with precise uncertainty measures. Traditional methods focus on point predictions only. Recent *post-processing methods* [2, 27, 30] refine base independent forecast distribution as a post-processing step. While these methods can be easily applied to forecasts from any model, this does not enable the models generating the base forecasts to learn from hierarchical relations between time-series of the hierarchy. *End-to-end learning neural methods* directly leverage hierarchical relations as part of the model architecture [24] or learning algorithm [11]. Due to their comprehensive end-to-end approach, they usually outperform post-processing methods by imposing hierarchical constraints on the mean or fixed quantiles of the forecast distributions. However, these methods do not enforce hierarchical consistency on the full distributions. Therefore, the *forecasts may not be well-calibrated* [18] i.e., they produce unreliable prediction intervals that may not match observed probabilities from ground truth [10].

Table 1: Comparison of PROFHIT with state-of-the-art methods.

	Two-step methods	Post-processing methods	End-to-end neural methods	**PROFHIT** (This paper)
Probabilistic Forecasts	×	✓	✓	✓
Strong & Weak Consistency	×	×	×	✓
Distributional Consistency	×	✓	×	✓
End-to-end Learning	×	×	✓	✓

In this work, we fill this gap of learning well-calibrated and accurate forecasts for both strong and weakly consistent datasets leveraging underlying hierarchical relations. We propose PROFHIT (Probabilistic Robust Forecasting for Hierarchical Time-series), a neural probabilistic hierarchical time-series forecasting method that provides an end-to-end Bayesian approach to model the distributions of forecasts of all time-series together (see Table 1 for a comparison). Specifically, we introduce a novel *Soft Distributional Consistency Regularization (SoftDisCoR)* to tackle the challenge. First, SoftDisCoR enables PROFHIT to leverage hierarchical relations over entire forecast distributions to generate calibrated forecast distributions by encouraging the forecast distribution of any parent node to be similar to the aggregation of children nodes' forecast distributions (Figure 1). Second, since SoftDisCoR is a soft constraint, our model is trained to adapt to datasets with varying hierarchical consistency that allows the model to trade-off consistency for better accuracy and calibration on weakly consistent datasets. Our main contributions are:

(1) **Accurate and Calibrated Probabilistic Hierarchical Time-Series Forecasting:** We propose PROFHIT, a deep probabilistic framework for modeling the distributions of each time-series together using the soft distributional consistency regularization (SoftDisCoR). PROFHIT leverages probabilistic deep-learning models to learn priors of individual time-series and refines the priors of all time-series leveraging the hierarchy to provide accurate and well-calibrated forecasts.
(2) **Adaptation to Strong and Weak Consistency via Soft Distributional Consistency Regularization:** SoftDisCoR imposes soft hierarchical constraints on the full forecast distributions to help adapt the model to varying levels of hierarchical consistency. We build a novel refinement module over base forecast priors and leverage multi-task learning over shared parameters that enable PROFHIT to perform consistently well across the hierarchy.
(3) **Evaluation Across Multiple Datasets and with Missing Data:** We show that our method PROFHIT outperforms a wide variety of state-of-the-art baselines on both accuracy and calibration, at all levels of the hierarchy, for both strong and weakly consistent datasets. We also show training using SoftDisCoR enables PROFHIT to leverage hierarchical relations to provide reliable predictions that can handle missing data values in the time-series.

2 RELATED WORK

Probabilistic time-series forecasting Classical probabilistic time-series forecasting methods include exponential smoothing and ARIMA [13]. They are simple but focus on univariate time-series and model each time-series sequence independently. Recently, deep learning based methods have been successfully applied in this area. DeepVAR [26] trains an auto-regressive recurrent network model on a large number of related time series to directly output the mean and variance parameters of the forecast distribution. Other works are inspired from the space-state models and explicitly model the transition and emission components with deep learning modules such as deep Markov models [17] and deep state space models [19, 23] Recently, EpiFNP [14] has achieved state-of-art performance in epidemic forecasting. It learns the stochastic correlations between input data and datapoints to model a flexible non-parametric distribution for univariate sequences.

Hierarchical time-series forecasting Classical works on hierarchical time-series forecasting used a two-step approach [12, 13] and focus on point predictions. They first forecast for time-series only at a single level of the hierarchy and then derive the forecasts for other nodes using the hierarchical relations.

Recent methods like MinT and ERM are post-processing steps applied on the set of forecasts at all levels of hierarchy. MinT [29, 30]

assumes that the base-level forecasts are uncorrelated and unbiased and solve an optimization problem to minimize the variance of forecast errors of past predictions. The unbiased assumption is relaxed in ERM [2]. Corani et al. [6] and [22] use a fully Bayesian bottom-up post-processing approach using base forecasts from full hierarchy. Another line of works projects the base forecasts of all time-series into a subspace of consistent forecasts. [9] use an iterative Game-theoretic approach of minimizing forecast error and projection error. Taieb et al. [27] uses copula method to refine base forecasts to be distributionally consistent as a post-processing step. Recent neural methods perform end-to-end learning that enables the model to leverage hierarchical relations while forecasting. Rangapuram et al. [24] use a deep-learning based end-to-end approach to directly train on the projected forecasts. SHARQ [11] is another recent probabilistic deep-learning based method that uses quantile regression and regularizes for consistency at different quantiles of forecast distribution. However, unlike our approach, these end-to-end methods do not regularize for forecast consistency over the entire distribution (Distributional consistency) but only over fixed quantiles. Most of these methods also are not designed for cases where the hierarchical constraints are not always consistently followed.

3 PRELIMINARIES

3.1 Problem Statement

Consider 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 time-series have a hierarchical relationship denoted as $\mathcal{T} = (G_\mathcal{T}, H_\mathcal{T})$ where $G_\mathcal{T}$ is a tree of N nodes rooted at time-series 1. For a non-leaf node (time-series) i, we denote its children as C_i. The node values are related via set of relations $H_\mathcal{T}$ of 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 known and time-independent real-valued constants.

Definition 1 (Consistency Error - CE). *Given a dataset $\mathcal{D}$ of N time-series over the time horizon $1, 2, \ldots, T$ and aggregation relations $H_\mathcal{T}$ as above, the dataset consistency error (CE) is defined as*

$$E_\mathcal{T}(\mathcal{D}) = \sum_{i \in \{1,2,\ldots N\}, C_i \neq \emptyset} \left(\mathbf{y}_i - \sum_{j \in C_i} \phi_{ij} \mathbf{y}_j \right)^2 . \tag{1}$$

(Intuitively, datasets with lower CE have time-series values which more strictly follow relations $H_\mathcal{T}$).

Definition 2 (Strong and weak consistency). *A dataset $\mathcal{D}$ is strongly consistent if $E_\mathcal{T}(\mathcal{D}) = 0$. Otherwise, $\mathcal{D}$ is said to be weakly consistent.*

Let the current time-step be t. For any $1 \leq t_1 < t_2 \leq t$, we denote $\mathbf{y}_i^{(t_1:t_2)} = \{y_i^{(t_1)}, y_i^{(t_1+1)}, \ldots, y_i^{(t_2)}\}$. Given the data $\mathcal{D}^t = [\mathbf{y}_1^{1:t}, \mathbf{y}_2^{1:t}, \ldots, \mathbf{y}_N^{1:t}]$ and hierarchical relations $H_\mathcal{T}$, a model M is trained to predict the marginal forecast distributions at time $t+\tau$ for all time-series of hierarchy leveraging past values of all time-series: $\{p_M(y_1^{(t+\tau)}|\mathcal{D}^t), \ldots p_M(y_N^{(t+\tau)}|\mathcal{D}^t)\}$. Along with the accuracy of probabilistic forecasts, we also evaluate forecast distributions for *calibration*. We define calibration of model forecasts based on previous works [14, 18]:

Definition 3. *(Calibration Score of a Model) Given a model M we define a calibration function $k_M : [0, 1] \rightarrow [0, 1]$ as follows: Given a confidence c, $k_M(c)$ is the fraction of the predictions for which the ground truth lies within c-confidence interval. The calibration score $CS(M)$ is the total deviation between c and $k_M(c)$: $CS(M) = \int_0^1 |k_M(c) - c| dc$. A perfectly calibrated model is such that $\forall c : k_M(c) \approx c$.*

Given a dataset $\mathcal{D}$ with underlying hierarchical relations $H_\mathcal{T}$, the goal of *Calibrated Probabilistic Hierarchical Forecasting* is to design a model M that provides *accurate* and *well-calibrated* forecast distributions $\{p_M(y_1^{(t+\tau)}|\mathcal{D}^t), \ldots p_M(y_N^{(t+\tau)}|\mathcal{D}^t)\}$ across all levels of the hierarchy for both weakly and strongly consistent datasets.

3.2 Functional Neural Process for Base Forecasts

PROFHIT first derives base forecasts for all the node from any differentiable *base forecasting model* such that we can use backpropagation on the loss function to update the parameters of the base forecasting model as well. Formally, the base forecasting model outputs the base forecast distribution parameters $\{\mu_i, \sigma_i\}_{i=1}^N$ from input time-series of all nodes as $P(\{\mu_i, \sigma_i\}_{i=1}^N | \{\mathbf{y}_i^{1:t}\}_{i=1}^N)$.

We leverage the recent advances in using Functional Neural Process [20] based non-parametric probabilistic sequential models that have provided state-of-art accurate and calibrated predictions in many domains [14, 15]. These models model the uncertainty of the input time-series as well as its correlation with time-series in training data to provide calibrated forecast distribution. Specifically, we use a slightly modified version of the model proposed in [14] and denote it as TSFNP. The only difference between [14] and TSFNP is that instead of modeling correlations with past values of the same time-series as in univariate time-series forecasting case, we model correlation with the time-series from all nodes' past. We provide a detailed description of TSFNP in the Appendix. For our further discussion, we can view TSFNP as a stochastic model with some latent variables:

$$P(\{\mu_i, \sigma_i\}_{i=1}^N | \mathcal{D}^t) = \int P\left(\mathcal{Z} | \{\mathbf{y}_i^{1:t}\}_{i=1}^N\right) \left(\prod_{i=1}^N P(\mu_i, \sigma_i | \mathcal{Z})\right) d\mathcal{Z} \tag{2}$$

where $\mathcal{Z}$ denotes the full set of latent variables of TSFNP.

4 METHODOLOGY

Overview. PROFHIT models the forecast distributions of all time-series nodes of the hierarchy $\{P(y_i^{(t+\tau)}|\mathcal{D}^t)\}_{i=1}^N$ by leveraging the relations from the hierarchy to provide accurate and well-calibrated forecasts that are adaptable to varying hierarchical consistency. Most existing methods do not attempt to model the entire probabilistic distribution but focus on the consistency of point forecasts or samples or fixed quantiles of the distribution [11, 24]. This approach does not fully capture the uncertainty of the forecasts and in turn, does not provide calibrated predictions. Methods like PEMBU, MinT and ERM are post-processing steps that can be applied to base forecasts from any model and provide theoretical guarantees. However, they do not allow the forecasting model to learn from relations across the hierarchy. Moreover, most methods assume

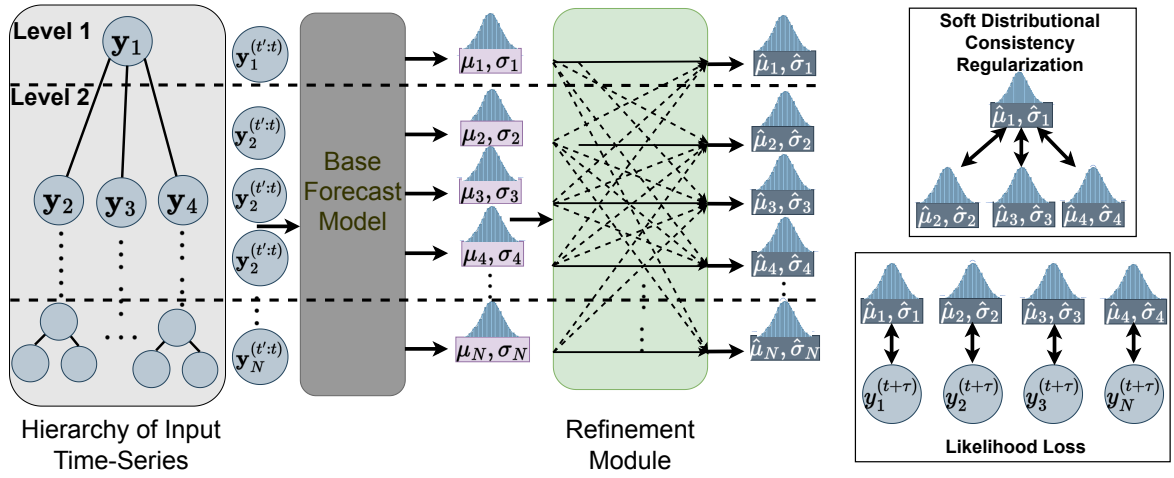


Figure 2: Overview of pipeline of PROFHIT. The input time series is ingested by TSFNP, a Functional Neural Process based probabilistic forecasting model, to output the base forecast distribution. The parameters of base forecasts are refined by the Hierarchy-aware Refinement module using predictions from all the time-series. The training is driven by a likelihood loss that learns from ground truth and Soft Distributional Consistency Regularization that regularizes the forecast distribution to follow the hierarchical relations.

that the datasets are strongly consistent over hierarchical relations. However, many real-world datasets are weakly consistent with time-series values of all nodes of the hierarchy observed simultaneously and may not follow the hierarchical relations strictly due to noise and discrepancies in collecting data at different levels. Therefore, most previous works may not adapt well to such deviations from these constraints.

PROFHIT, on the other hand, reconciles the need to model consistency between entire forecast distributions as well as induce a soft adaptable constraint to enforce consistency via a two-step process that is trained in an *end-to-end* manner. The first component of PROFHIT is a differentiable neural probabilistic model such as TSFNP that produces a *base forecast* distribution for each node parameterized by $\{(\mu_i, \sigma_i)\}_{i=1}^N$. Base forecasts of all nodes are used as priors to derive a refined set of forecast distributions parameterized by $\{(\hat{\mu}_i, \hat{\sigma}_i)\}_{i=1}^N$ via the *Hierarchy-aware Refinement Module* described in Section 4.1. We introduce the novel Soft Distributional Consistency Regularization (SoftDisCoR) that enables PROFHIT to produce refined forecast distributions that are distributionally consistent with the hierarchical relation $H_{\mathcal{T}}$ as described in Section 4.2. The full probabilistic process of PROFHIT is depicted in Figure 2.

4.1 Hierarchy-aware Refinement Module

The base forecast distributions $P(\{\mu_i, \sigma_i\}|\mathcal{D}^t)$ produced by TSFNP (or any other model that can be used in its place) do not leverage the underlying hierarchical relations $H_{\mathcal{T}}$. This may lead to sub-optimal forecasting performance and inconsistent forecasts. The refinement module is a differentiable module that aims to fuse the information from base forecasts of all nodes to output refined forecast distributions that can leverage SoftDisCoR to be consistent.

Formally, given the parameters of *base forecast* distributions $\{\mu_i, \sigma_i\}_{i=1}^N$ derived from TSFNP for all time-series $\{\mathbf{y}_i^{(t':t)}\}_{i=1}^N$, the refinement module derives the refined forecast distributions denoted by parameters $\{\hat{\mu}_i, \hat{\sigma}_i\}_{i=1}^N$ as functions of parameters of base forecasts of all time-series. Let $\mu = [\mu_1 \ldots, \mu_N]$ and $\sigma = [\sigma_1 \ldots, \sigma_N]$ be vectors of means and standard deviations of base distributions. Since each of the node's refined distribution parameters depends on all N node's base forecast parameters, the refinement process must be efficient in fusing the information from all the base forecasts for each node. Moreover, since we require that PROFHIT should be adaptable to datasets of both strong and weak consistency, the refinement process should automatically learn to trade-off between the influence of base forecast distribution for each node and the fused information from all the nodes. Considering these objectives, we derive the mean $\hat{\mu}_i$ of refined distribution as a weighted sum of two terms: a) μ_i, the mean of base time-series, and b) linear combination of all base mean of all time-series:

$$\gamma_i = \text{sigmoid}(\hat{w}_i), \quad \hat{\mu}_i = \gamma_i \mu_i + (1 - \gamma_i)\mathbf{w}_i^T \mu. \tag{3}$$

$\{\hat{w}_i\}_{i=1}^N$ and $\{\mathbf{w}_i\}_{i=1:N}$ are both learnable set of parameters of the model. sigmoid$(\cdot)$ denotes the sigmoid function. The operations in Equation 3 have a total computational complexity of $O(N)$ for each node and therefore $O(N^2)$ in total. This is on par with previous state-of-art end-to-end refinement methods like HierE2E [24] and more efficient than post-processing methods like MinT and ERM during inference. The learnable parameter γ_i allows the refinement module to trade-off between the influence of the base distribution of node i and the influence of the other nodes of the hierarchy making PROFHIT automatically adapt to datasets with varying hierarchical consistency.

Similarly, we assume the variance of the refined distribution depends on the base mean and variance of all the time-series. The variance parameter $\hat{\sigma}_i$ of the refined distribution is derived from

the base distribution parameters μ and σ as

$$\hat{\sigma}_i = c\sigma_i \text{sigmoid}(\mathbf{v}_{1i}^T\mu + \mathbf{v}_{2i}^T\sigma + b_i) \tag{4}$$

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. Note that the complexity of Equation 4 is also $O(N^2)$.

4.2 Soft Distributional Consistency Regularization

While the refinement module helps aggregate information from base forecasts to refine the distribution parameters, we also need to design the loss function such that parameters of the refinement module and TSFNP utilize the underlying hierarchical relations $H_{\mathcal{T}}$ to provide hierarchically consistent forecast distributions by effectively utilizing information from all nodes of the time-series. For the full distribution of the refined forecasts to be consistent, we use a *Distributional Consistency Error*(DCE) as part of the loss function and regularize the full distribution of all nodes.

The Distributional Consistency Error (DCE) is defined as follows:

Definition 4. *(Distributional Consistency Error - DCE) Given the forecasts at time $t+\tau$ as $\{p_M(y_1^{(t+\tau)}|\mathcal{D}^t), \dots p_M(y_N^{(t+\tau)}|\mathcal{D}^t)\}$ distributional consistency error (DCE) is defined as*

$$\sum_{i\in\{1,\dots,N\}, C_i\neq\emptyset} Dist\left(p_M(y_i^{(t+\tau)}|\mathcal{D}^t), p_M(\sum_{j\in C_i}\phi_{i,j}y_j^{(t+\tau)}|\mathcal{D}^t)\right) \tag{5}$$

where Dist is a distributional distance metric.

Leveraging distributional consistency error as a soft regularizer enforces forecast distributions to be well-calibrated while adaptively adhering to the hierarchical relations of the dataset.

For the distance metric Dist in Equation 5, we use the Jensen-Shannon Divergence [8] (JSD) as the distance metric since it is a symmetric and bounded variant of the popularly used KL-Divergence distance. Moreover, it assumes a closed form for many widely used distributions including for the Gaussian used in PROFHIT. While we can replace JSD with other distance measures for capturing distributional similarity, we observed that JSD was sufficient for providing good forecast performance in our applications. We derive the *distributional consistency error* on Gaussian parameters $\{(\hat{\mu}_i, \hat{\sigma}_i)\}_{i=1}^N$ as

$$\mathcal{L}_{\text{dist}} = \sum_{i=1}^N \text{JSD}\left(P(y_i^{(t+\tau)}|\hat{\mu}_i, \hat{\sigma}_i), P\left(\sum_{j\in\mathbf{C}_i}\phi_{ij}y_j^{(t+\tau)}|\{\hat{\mu}_j, \hat{\sigma}_j\}_{j\in\mathbf{C}_i}\right)\right). \tag{6}$$

The computation of JSD is generally intractable. However, in our case, due to parameterization of each time-series distribution as a Gaussian we get a closed-form differentiable expression:

$$\mathcal{L}_{\text{dist}} = \sum_{i=1}^N \frac{\hat{\sigma}_i^2 + \left(\hat{\mu}_i - \sum_{j\in C_i}\phi_{ij}\hat{\mu}_j\right)^2}{4\sum_{j\in C_i}\phi_{ij}^2\hat{\sigma}_j^2} + \sum_{i=1}^N \frac{\sum_{j\in C_i}\phi_{ij}^2\hat{\sigma}_j^2 + \left(\hat{\mu}_i - \sum_{j\in C_i}\phi_{ij}\hat{\mu}_j\right)^2}{4\hat{\sigma}_i^2} - \frac{1}{2}. \tag{7}$$

Derivation of Distributional Consistency Error. To derive Equation 7, we use the following well-known result for JSD of two Gaussian Distributions [21]: Given two univariate Normal distributions $P_1 = \mathcal{N}_1(\hat{\mu}_1, \hat{\sigma}_1)$ and $P_2 = \mathcal{N}_2(\hat{\mu}_2, \hat{\sigma}_2)$, the JSD is

$$\text{JSD}(P_1, P_2) = \frac{1}{2}\left[\frac{\hat{\sigma}_1^2 + (\hat{\mu}_1 - \hat{\mu}_2)^2}{2\hat{\sigma}_2^2} + \frac{\hat{\sigma}_2^2 + (\hat{\mu}_1 - \hat{\mu}_2)^2}{2\hat{\sigma}_1^2} - 1\right] \tag{8}$$

Consider each JSD term of the summation in Equation 6. Note that

$$P(y_i^{t+\tau}|\mu_i, \sigma_i) = \mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i) \tag{9}$$

and $P(\sum_{j\in\mathbf{C}_i}\phi_{ij}y_j^{t+\tau}|\{|\mu_j, \sigma_j\}_{j\in\mathbf{C}_i}))$ is weighted sum of Gaussian variables $\{\mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j)\}_{j\in C_i}$. Therefore,

$$P\left(\sum_{j\in\mathbf{C}_i}\phi_{ij}y_j^{t+\tau}|\{\mu_j, \sigma_j\}_{j\in\mathbf{C}_i}\right) = \mathcal{N}\left(\sum_{j\in C_i}\phi_{ij}\hat{\mu}_j, \sqrt{\sum_{j\in C_i}\phi_{ij}^2\hat{\sigma}_j^2}\right). \tag{10}$$

Using Equation 8 along with Equations 9,10 we get the desired result in Equation 7.

We use the distributional consistency error as a soft regularization term to enable PROFHIT to leverage constraints $H_{\mathcal{T}}$ when generating forecast distributions. We do not make DCE a hard constraint since the model needs to adapt to datasets of varying consistency. Particularly, for weakly consistent datasets, we do not require PROFHIT to strictly adhere the hierarchical relations $H_{\mathcal{T}}$ which may result in sub-optimal forecast accuracy and calibration, since the ground truth does not follow $H_{\mathcal{T}}$ as well. Therefore, using DCE as a soft-regularizer allows the model to adapt to varying strictness of $H_{\mathcal{T}}$ across different domains.

4.3 Details on Training

Training loss. Along with the SoftDisCoR loss $\mathcal{L}_{\text{dist}}$ which informs PROFHIT of hierarchical relations $H_{\mathcal{T}}$ optimizes for distributional consistency, we derive the *Likelihood Loss* $\mathcal{L}_{\text{ll}}$ to optimize for the accuracy and calibration of the forecasts. Using TSFNP (or any other base forecasting model with latent variables) as the, the full probabilistic process of PROFHIT can be summarized as:

$$P(\{y_i^{(t+\tau)}\}_{i=1}^N|\mathcal{D}^t) = \int \underbrace{P\left(\mathcal{Z}|\{\mathbf{y}_i^{1:t}\}_{i=1}^N\right)\left(\prod_{i=1}^N P(\mu_i, \sigma_i|\mathcal{Z})\right)}_{\text{TSFNP (Base forecasts)}} \underbrace{\prod_{i=1}^N P(\hat{\mu}_i, \hat{\sigma}_i|\{\mu_j, \sigma_j\}_{j=1}^N)P(y_i^{(t+\tau)}|\hat{\mu}_i, \hat{\sigma}_i)}_{\text{Refinement}} d\mathcal{Z}. \tag{11}$$

Integrating over the latent variables $\mathcal{Z}$ in Equation 11 is highly intractable. Therefore, we use variational inference by approximating the posterior over the latent variables $P(\mathcal{Z}|\{y_i^{(t+\tau)}\}_{i=1}^N)$ and derive an ELBO $\mathcal{L}_{\text{ll}}$ which we use as the optimization objective. The details of the derivation of ELBO loss are in the Appendix. Since the refinement module is a deterministic mapping from base to refined distribution parameters, the ELBO derivation is very similar to that in [14]. Therefore, our framework is flexible to adapt to a

wide range of neural forecasting models with different learning algorithms.

Thus, the total loss for training is given as $\mathcal{L} = \mathcal{L}_{\text{ll}} + \lambda \mathcal{L}_{\text{dist}}$ where the hyperparameter λ controls the trade-off in importance between data likelihood and distributional consistency. We also use the reparameterization trick to make the sampling process differentiable and we learn the parameters of all training modules via Stochastic Variational Bayes [16]. The full pipeline of PROFHIT is summarized in Figure 2.

Parameter sharing across nodes. Since PROFHIT's TSFNP module forecasts for multiple nodes, we leverage the hard-parameter sharing paradigm of multi-task learning [3] and use a different set of parameters for Predictive Distribution Decoder (i.e., weights of Θ_3) whereas the parameters of other components of TSFNP are shared across all nodes (Figure 2). Sharing parameters for Probabilistic Neural Encoder drastically lowers the number of learnable parameters since datasets can have a large number of nodes (up to 512 nodes in our experiments).

Pre-training on individual time-series. Before we start training for refined forecasts, we pre-train the parameters of TSFNP on given training dataset to model base forecast distribution accurately. We pre-train using only log-likelihood loss to learn parameters $\{\mu_i, \sigma_i\}_{i=1}^{N}$.

5 EXPERIMENTS

We evaluate PROFHIT over multiple datasets and compare it with state-of-the-art baselines[1].

5.1 Setup

Baselines: We compare PROFHIT's performance against state-of-the-art hierarchical forecasting methods. We also compare against state-of-the-art general probabilistic forecasting methods to study the importance of modeling the hierarchy for both weak and strongly consistent datasets.

(1) **TSFNP** [14]: a neural forecasting model for accurate and calibrated forecasts described in Section 3.2
(2) **DeepAR** [26]: another state-of-the-art deep probabilistic forecasting models which do not exploit hierarchy relations.
(3) **MinT** [30]: a post-processing method for reconciliation of base forecasts
(4) **ERM** [2]: another post-processing method like MinT that relaxes unbiased assumptions of base forecasts
(5) **HierE2E** [24] is a recent state-of-the-art 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.
(6) **SHARQ** [11] is another state-of-the-art deep learning based approach that reconciles forecast distributions by using quantile regressions and making the quantile values consistent.
(7) **PEMBU** [27] is a post-processing method that refines base forecasts to be distributionally consistent.

[1]Code and datasets: https://github.com/AdityaLab/Profhit

Note: In our experiments, ee performed ERM and MinT on Monte Carlo samples of TSFNP predictive distribution since TSFNP provided better results compared to DeepAR. We also use the mean forecast from MinT and ERM as input forecasts for PEMBU.

Table 2: Dataset Characteristics and Consistency

Dataset	No. of Nodes	Levels of Hierarchy	τ	Obs. per node	Consistency (CE)
Tourism-L	555	4,5	12	228	Strong(0)
Labour	57	4	8	514	Strong(0)
Wiki	207	5	1	366	Strong(0)
Flu-Symptoms	61	3	4	544	Weak(3.37)
FB-Survey	61	3	4	257	Weak(2.44)

Datasets: We evaluate on a diverse set of publicly available datasets (Table 2) from different domains with varied hierarchical relations and consistency. The benchmarking dataset and evaluation setup is replicated from recent and past literature related to general hierarchical forecasting as well as epidemic forecasting.

(1) Labour dataset contains monthly employment data from Feb 1978 to Dec 2020 collected from the Australian Bureau of Statistics.
(2) Tourism-L [30] contains tourism flows in different regions in Australia grouped via region and demographic. It has two sets of hierarchies (with four and five levels), one for the mode of travel and the other for geography with the top node being the only common node of both hierarchies.
(3) Wiki dataset collects the number of daily views of 145000 Wikipedia articles aggregated into 150 groups [27]. These 150 groups are leaf nodes of a four-level hierarchy with groups of similar topics aggregated together.
(4) Flu-Symptoms contains flu incidence values called *weighted influenza-like incidence* (wILI) values [25] at multiple spatial scales for USA for period of 2004-2020. The scales used are states, HHS and National level (US states are grouped into 10 HHS regions by CDC).
(5) FB-Survey provides an aggregated anonymized daily indicator for the prevalence of Covid-19 symptoms based on online surveys conducted on Facebook [7] from Dec 2020 to Aug 2021 for each state and national level. We use the state-level values to find aggregates at HHS levels.

Tourism-L, Labour and Wiki are constructed by collecting values of leaf nodes and deriving the values of the time series of other nodes of the hierarchy. Hence, they are strongly consistent with zero CE (Definition 1). The values of each node of the hierarchy in the case of Flu-Symptoms and FB-Survey are directly collected or measured. For example, the values of Flu-Symptoms dataset are collected from public health agencies at the state, HHS and national levels and aggregated by CDC. Due to factors like reporting discrepancies and noise, they contain values in time series that may deviate from the given hierarchical relations [4]. Therefore, these datasets are weakly consistent with significant CE (Table 2). We also provide level wise consistency errors for all the datasets in the Appendix.

Evaluation metrics. We evaluate our model and baselines using carefully chosen metrics that are widely used in the literature to measure accuracy and calibration. We also measure the distributional consistency of the output forecast to study how well the model trade-off accuracy and consistent for datasets of varying consistency errors. 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. •**Mean Absolute Percentage Error (MAPE)** is a commonly used score for point-predictions calculated as

$$MAPE = \frac{1}{N}\sum_{t=t_1}^{t_N} |\frac{y^{(t)} - \hat{y}^{(t)}}{y^{(t)}}|$$

•**Log Score (LS)** is a standard score used to measure the accuracy of probabilistic forecasts in epidemiology [25]. LS measures the negative log likelihood of a fixed size interval around the ground truth under the predictive distribution:

$$LS(\hat{p}_y, y) = -\int_{y-L}^{y+L} \log \hat{p}_y(\hat{y}) d\hat{y}.$$

Similar to [25], the log-likelihood of a forecast is capped at -10. The calculation of LS is tractable due to the gaussian assumption on the forecast distribution.

•**Calibration Score (CS):** To measure the calibration of forecasts, we use the calibration score defined in Section 3.1. We approximate the integral via Riemann sum over $[0, 1]$ with step-size 0.05.

•**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.

•**Distributional Consistency Error (DCE):** We calculate the Distributional Consistency Error (Equation 6) on output forecast distributions during inference to study how PROFHIT and baselines leverage SoftDisCoR to learn from hierarchical relations across datasets of varying consistency and trade-off consistent, calibration and accuracy, especially for weakly consistent data (Section 5.2 Q3).

5.2 Results

We comprehensively evaluate PROFHIT through the following questions: **Q1**: Does PROFHIT predict accurate calibrated forecasts? **Q2**: Does PROFHIT provide consistently better performance across all levels of the hierarchy? **Q3**: Does SoftDisCoR help PROFHIT outperform baselines on both strongly and weakly consistent datasets? **Q4**: What impact do various modeling choices have on the model's overall performance? **Q5**: How does improved calibration and forecast consistency help PROFHIT deal with missing values in data?

Accuracy and calibration performance (Q1). We evaluate all baselines, PROFHIT and its variants for all the datasets over 5 independent runs. The average scores across all levels hierarchy are shown in Tables 3. PROFHIT significantly outperforms all baselines in MAPE score by 13%. It also outperforms the baselines in LS and CS significantly in most cases. Finally, PROFHIT shows 41-88% better CRPS scores. Thus, PROFHIT adapts well to varied kinds of datasets and outperforms all baselines in both accuracy and calibration.

Performance across the hierarchy (Q2). Next, we look at the performance of all models across each level of the hierarchy. We compared the performance of PROFHIT with best-performing baselines HierE2E and SHARQ for all datasets. PROFHIT significantly outperforms the best baselines both in terms of accuracy and calibration. The performance improvement is consistent across all levels of the hierarchy in most of the benchmarks. We show detailed results in the Appendix. This shows that the model effectively leverages hierarchical relations across all nodes to provide significantly more accurate and calibrated forecasts across the hierarchy.

Effect of SoftDisCoR on datasets of varying consistency (Q3). Since most previous state-of-the-art models assume datasets to be strongly consistent, deviations from this assumption can cause under-performance when used with weakly consistent datasets. This is evidenced in Table 3 where most of the baselines explicitly optimize for hierarchical consistency as a hard constraint on the forecasts. For example, PEMBU's forecasts have better distributional consistency error (DCE) for weakly consistent datasets. However, they perform much worse in both accuracy and calibration than even TSFNP, which does not even leverage hierarchical relations. Since we use SoftDisCoR as a soft learning constraint, PROFHIT can learn to trade-off consistency for accuracy and calibration. Therefore, PROFHIT provides 93% better CRPS and significantly better calibration over the best baselines. These improvements are more pronounced at non-leaf nodes of hierarchy where PROFHIT's performance is significanly larger for the weakly consistent `Flu-Symptoms` and `FB-Survey` datasets. In the case of strongly consistent datasets, PROFHIT provides 54% better CRPS and better calibration while having comparable DCE to PEMBU. We provide further analysis of these observations in the Appendix.

Ablation study on various modeling choices (Q4). We evaluate the efficacy and contribution of our various modeling choices including the usefulness of SoftDisCoR, refinement module, hard-parameter sharing, and using TSFNP as our model of choice for base forecasts. We perform an ablation study using the following variants of PROFHIT:

• **P-NoConsistency:** This variant is trained by completely removing the SoftDisCoR from the training. Note that unlike P-FineTune which was initially trained with SoftDisCoR before fine-tuning, P-NoConsistency never uses the SoftDisCoR at any point of the training routine. Therefore P-NoConsistency measures the importance of explicitly regularizing over the information from the hierarchy.

• **P-NoRefine**: We remove the hierarchical refinement module and optimize the base forecasts for both likelihood and SoftDisCoR loss.

• **P-DeepAR:** We evaluate our choice of using TSFNP, a previous state-of-the-art univariate forecasting model for accurate and calibrated forecasts by replacing it with DeepAR[26], another popular probabilistic forecasting model that was used by HierE2E.

• **P-NoParamShare:** We study the effect of our multi-tasking hard-parameter sharing approach (Section 4.3) by training a variant

Table 3: Average scores (across 5 runs) across all levels of hierarchy for all baselines, PROFHIT and its variants. PROFHIT provides top performance in terms of all evaluation metrics in most of the benchmarks.

	Tourism-L					Labour					Wiki				
Models/Data	**MAPE%**	**CRPS**	**LS**	**CS**	**DCE**	**MAPE%**	**CRPS**	**LS**	**CS**	**DCE**	**MAPE%**	**CRPS**	**LS**	**CS**	**DCE**
DeepAR	3.12	0.17	0.61	0.19	0.32	18.27	0.045	0.75	0.25	0.34	16.52	0.232	0.83	0.27	0.26
TSFNP	2.28	0.21	1.19	0.14	0.39	14.52	0.071	1.41	0.21	0.22	15.63	0.287	0.86	0.21	0.39
TSFNP-MinT	1.17	0.5	0.58	0.15	0.24	16.46	0.045	4.12	0.26	0.12	13.79	0.243	0.78	0.18	0.18
TSFNP-ERM	**1.42**	0.56	0.53	0.11	0.18	13.57	0.045	3.63	0.23	0.19	17.74	0.221	0.74	0.19	0.21
HierE2E	1.67	0.15	0.38	0.17	0.21	**12.53**	0.034	0.51	0.25	0.15	17.05	0.211	0.46	0.23	0.12
SHARQ	1.63	0.17	0.41	0.12	0.13	14.21	0.054	0.47	0.18	0.09	16.13	0.241	0.52	0.16	0.16
PEMBU-MinT	1.77	0.15	0.46	0.24	0.03	13.55	0.039	0.56	0.22	0.11	14.66	0.279	0.58	0.21	0.05
PEMBU-ERM	1.63	0.16	0.43	0.21	**0.02**	13.19	0.042	0.61	0.25	**0.03**	15.79	0.268	0.54	0.18	**0.02**
PROFHIT	1.47	**0.12**	**0.33**	**0.09**	**0.02**	12.79	**0.026**	**0.21**	**0.14**	0.05	**12.47**	**0.184**	**0.35**	**0.13**	0.04

	Flu-Symptoms					FB-Survey				
Models/Data	**MAPE%**	**CRPS**	**LS**	**CS**	**DCE**	**MAPE%**	**CRPS**	**LS**	**CS**	**DCE**
DeepAR	31.27	0.610	3.25	0.065	0.31	17.39	7.32	5.32	0.17	0.29
TSFNP	12.8	0.460	0.93	0.034	0.42	15.35	5.53	7.84	0.11	0.37
TSFNP-MinT	10.56	0.630	3.18	0.082	0.18	12.24	5.39	6.35	0.14	0.24
TSFNP-ERM	11.85	0.620	2.75	0.075	0.12	13.16	6.14	4.23	0.12	0.19
HierE2E	15.67	0.420	0.81	0.12	0.32	12.63	4.12	1.13	0.19	0.26
SHARQ	18.34	0.470	1.42	0.071	0.21	12.82	3.12	0.81	0.15	0.19
PEMBU-MinT	15.44	0.621	2.55	0.18	**0.05**	13.75	5.78	4.22	0.22	**0.07**
PEMBU-ERM	17.57	0.688	2.74	0.15	0.07	12.99	6.31	5.18	0.18	0.1
PROFHIT	**8.85**	**0.250**	**0.28**	**0.042**	0.14	9.67	**1.43**	**0.45**	**0.08**	0.16

where all the parameters for each TSFNP forecasting module are shared across all the nodes.

• **P-FineTune:** We also look at the efficacy of our soft regularization using both losses that adapt to optimize for both consistency and training accuracy by comparing it with a variant where the predictive distribution decoder parameters are further fine-tuned for individual nodes using only the likelihood loss.

We compare the performance of PROFHIT with its variants in CRPS, CS and DCE in table 4 (Rest of the metrics are in Appendix). We observe that PROFHIT is comparable to or better than the best-performing variant in most cases. We observe that the best-performing variant for strongly consistent datasets is P-NoParamShare which is trained with both likelihood loss and SoftDisCoR (Table 3). But its performance severely degrades for weakly consistent datasets since sharing all model parameters across all time-series makes it inflexible to model patterns and deviations specific to individual nodes. In contrast, P-FineTune and P-NoConsistency performs the best among variants for weakly consistent datasets since they train separate sets of decoder parameters for each node. But they perform poorly for strongly consistent datasets since they don't leverage Distributional Consistency effectively. PROFHIT combines the flexible parameter learning of P-FineTune and leverage Distributional Consistency to jointly optimize the parameters like P-NoParamShare providing comparable performance to best variants over all datasets.

Adapting to missing data (Q5). Accurate and well-calibrated models that can effectively leverage the knowledge of the hierarchy can intuitively allow models to better adapt to noise/missing data. Hence, we introduce the task of *Hierarchical Forecasting with Missing Values* and study the adaptation of models when there are missing values in time-series. We model a situation that is encountered in many real-world applications such as Epidemic Forecasting where the past few values of time-series are missing due to various factors like data reporting delays [4].

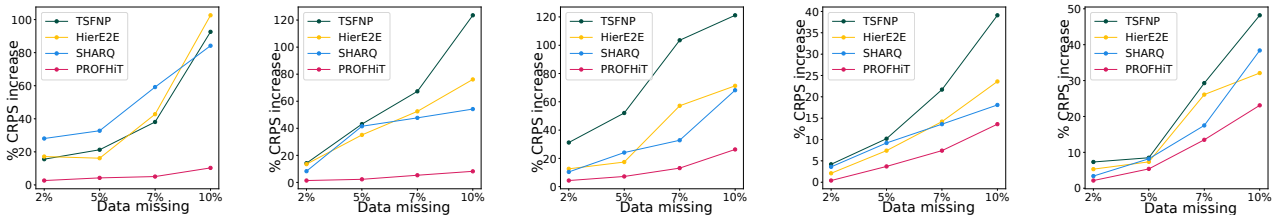
Formally, at time-period t, we are given full data up to time $t - \rho$. We set $\rho = 5$ since it is the average forecast horizon of all datasets. For sequence values in the period between $t - \rho$ and t, we randomly remove k% of these values across all time-series. The models are trained on the complete time-series dataset till time $t' = t - \rho$. Models' predictions are then used to fill in missing values for time t' to t. Finally, we input the filled time-series to generate the forecasts for future time-steps.

We measure the relative decrease in performance of PROFHIT and baselines with an increase in the percentage of missing data k (Figures 3). We observe that PROFHIT's performance decrease as the fraction of missing values increases is much slower compared to other baselines. Even at $k = 10$%, PROFHIT's performance decreases by 10.45-26.8% compared to other baselines that typically decrease by over 70%. Thus, PROFHIT effectively uses hierarchical relations to generate reliable predictions on strong and weakly consistent datasets.

We compare relative performance decrease with an increase in the percentage of missing data for PROFHIT and its variants in Figure 4. We observe that P-NoConsistency's performance deteriorates very rapidly in most benchmarks, showing the importance of SoftDisCoR for learning provides calibrated and consistent forecasts. The second worst-performing variant across all datasets is P-FineTune which also relies less on the hierarchical relations due to fine-tuning of parameters for specific time-series. This is followed by P-NoRefine which performs particularly worse in strongly consistent datasets due to the absence of the refinement

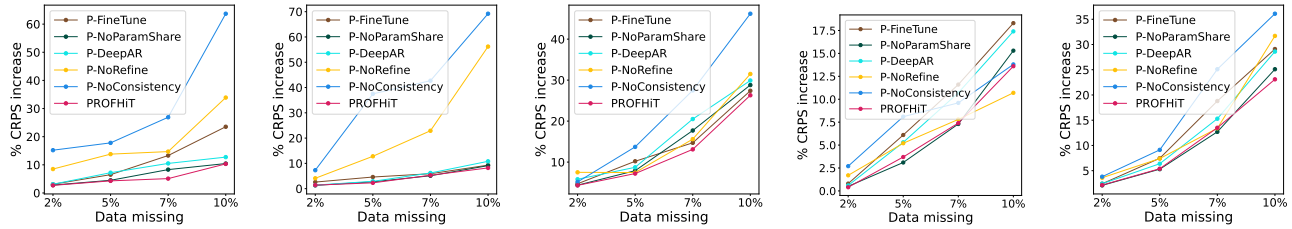
Table 4: Average scores (across 5 runs) across all levels of hierarchy for PROFHIT and its ablation variants. The best score is bolded and the second best is underlined.

	Tourism-L			Labour			Wiki			Flu-Symptoms			FB-Survey		
Models/Data	**CRPS**	**CS**	**DCE**	**CRPS**	**CS**	**DCE**	**CRPS**	**CS**	**DCE**	**CRPS**	**CS**	**DCE**	**CRPS**	**CS**	**DCE**
PROFHIT	**0.12**	0.09	0.02	**0.026**	**0.14**	0.05	**0.184**	**0.13**	**0.04**	0.250	0.042	0.14	1.43	0.08	0.16
P-NoConsistency	0.18	0.21	0.35	0.043	0.26	0.17	0.227	0.35	0.14	0.248	0.16	0.22	1.17	0.24	0.22
P-NoRefine	0.16	0.14	0.19	0.037	0.18	0.15	0.219	0.19	0.09	0.256	0.097	0.17	**1.15**	0.12	0.18
P-DeepAR	0.13	0.12	0.04	0.029	0.17	0.08	0.201	0.24	0.07	0.361	0.083	0.15	2.13	0.18	0.15
P-FineTune	0.16	0.14	0.25	0.031	0.21	0.13	0.216	0.21	0.08	**0.240**	**0.039**	0.17	1.18	**0.07**	0.19
P-NoParamShare	0.13	**0.06**	**0.01**	0.027	0.16	**0.04**	0.185	0.16	**0.04**	0.350	0.086	**0.09**	2.64	0.14	**0.11**



(a) Tourism-L (b) Labour (c) Wiki (d) Flu-Symptoms (e) FB-Survey

Figure 3: % increase in CRPS for all models with increase in proportion of missing data.



(a) Tourism-L (b) Labour (c) Wiki (d) Flu-Symptoms (e) FB-Survey

Figure 4: % increase in CRPS for PROFHIT and variants with an increase in the proportion of missing data.

module to directly learn refined distributions by combining information from base forecasts. Finally, we observe that PROFHIT and P-NoParamShare suffer the least degradation in performance since both these models prioritize integrating hierarchical consistency information which enables them to provide better estimates for imputed data for missing input and use them to generate more accurate and calibrated forecasts.

6 CONCLUSION AND DISCUSSION

We introduced PROFHIT, a probabilistic hierarchical forecasting model that produces accurate and well-calibrated forecasts using soft distributional consistency regularization (SoftDisCoR). This enables PROFHIT to adapt to datasets with varying levels of hierarchical consistency. We evaluated PROFHIT against previous state-of-the-art hierarchical forecasting baselines over a wide variety of datasets and observed 41-88% improvement average improvement in accuracy and significantly better calibration scores. PROFHIT provided the best performance across the entire hierarchy as well as significantly outperformed other models in providing robust predictions when it encountered missing data where other baselines' performance degraded by over 70%. We also showed the efficacy of various design choices of PROFHIT including using TSFNP for generating raw forecasts, multi-tasking approach of partial parameter sharing, refinement module, and introducing the novel distributional consistency loss as a soft regularizer.

Our work opens new possibilities like extending to various domains where time-series values across the hierarchy may not be continuous real numbers, can not be modeled as Gaussian distributions or may have different sampling rates. We can also explore modeling more complex structures between time-series with different aggregation relations. PROFHIT can also be used to study anomaly detection in time-series, especially in time-periods where there are deviations from assumed consistency relations. Similar to Kamarthi et al. [15], we can extend our work to include multiple sources of features and modalities of data both specific to each time-series and global to the entire hierarchy.

Acknowledgments: This work was supported in part by the NSF (Expeditions CCF-1918770, CAREER IIS-2028586, RAPID IIS-2027862, Medium IIS-1955883, Medium IIS-2106961, IIS-2008334, CCF-2115126, PIPP CCF-2200269, CAREER IIS-2144338), CDC MInD program, faculty research award from Facebook and funds/computing resources from Georgia Tech.

REFERENCES

[1] Bijaya Adhikari, Xinfeng Xu, Naren Ramakrishnan, and B. Aditya Prakash. 2019. EpiDeep: Exploiting Embeddings for Epidemic Forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 577–586.

[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] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[4] Prithwish Chakraborty, Bryan Lewis, Stephen Eubank, John S Brownstein, Madhav Marathe, and Naren Ramakrishnan. 2018. What to know before forecasting the flu. *PLOS Computational Biology* 14, 10 (2018), e1005964.

[5] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[6] Giorgio Corani, Dario Azzimonti, Joao PSC Augusto, and Marco Zaffalon. 2020. Probabilistic Reconciliation of Hierarchical Forecast via Bayes' Rule.. In *ECML/PKDD (3)*. 211–226.

[7] CMU Delphi Research Group. 2021. COVID-19 Trends and Impact Survey. (2021). https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/fb-survey.html

[8] Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information theory* 49, 7 (2003), 1858–1860.

[9] Tim van Erven and Jairo Cugliari. 2015. Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts. In *Modeling and stochastic learning for forecasting in high dimensions*. Springer, 297–317.

[10] Adam Fisch, Tommi Jaakkola, and Regina Barzilay. 2022. Calibrated Selective Classification. *arXiv preprint arXiv:2208.12084* (2022).

[11] 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.

[12] 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.

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

[14] 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).

[15] 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).

[16] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[17] Rahul Krishnan, Uri Shalit, and David Sontag. 2017. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[18] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. 2018. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*. PMLR, 2796–2804.

[19] Longyuan Li, Junchi Yan, Xiaokang Yang, and Yaohui Jin. 2021. Learning interpretable deep state space model for probabilistic time series forecasting. *arXiv preprint arXiv:2102.00397* (2021).

[20] Christos Louizos, Xiahan Shi, Klamer Schutte, and Max Welling. 2019. The functional neural process. *arXiv preprint arXiv:1906.08324* (2019).

[21] Frank Nielsen. 2019. On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy* 21, 5 (2019), 485.

[22] Julie Novak, Scott McGarvie, and Beatriz Etchegaray Garcia. 2017. A Bayesian model for forecasting hierarchically structured time series. *arXiv preprint arXiv:1711.04738* (2017).

[23] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. 2018. Deep state space models for time series forecasting. *Advances in neural information processing systems* 31 (2018).

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[29] Shanika L Wickramasuriya. 2021. Probabilistic forecast reconciliation under the Gaussian framework. *arXiv preprint arXiv:2103.11128* (2021).

[30] 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.

Appendix for When Rigidity Hurts: Soft Consistency Regularization for Probabilistic Hierarchical Time Series Forecasting

A DETAILS ON TSFNP

We briefly describe the components of TSFNP here and direct the readers to the [14] for more details.

1) *Probabilistic Neural Encoder*: It models the temporal patterns of the input time-series and the uncertainty of its latent representation. It encodes the input univariate time-series into a latent stochastic embedding via a GRU [5] followed by a self-attention layer [28]:

$$[\mu_{\mathbf{u}_i}, \log \sigma_{\mathbf{u}_i}] = \text{Self-Atten}(\text{GRU}(\mathbf{y}_i^{(t':t)})), \quad \mathbf{u}_i \sim \mathcal{N}(\mu_{\mathbf{u}_i}, \sigma_{\mathbf{u}_i}). \tag{12}$$

2) *Stochastic Data Correlation Graph*: We next model the correlations between the input time-series and other time-series of the dataset to capture contextual representation and uncertainty of the input data point with respect to training data distribution. These contextual representations are called **local latent variable**. The only difference in our approach compared to [14] is that, unlike [14] which uses past time-series information from the same node, in our multi-variate case TSFNP uses past information from all nodes. Formally, for input sequence $\mathbf{y}_i^{(t':t)}$ we sample sequences from the the past training sequences $\mathbf{y}_j$ where $j \in \{1, \ldots, N\}$ using similarity between their latent stochastic embeddings $\{\mathbf{u}_i\}_{i=1}^N$. For input time-series of node i and each of the past training sequences $\mathbf{y}_j$, we sample $\mathbf{y_j}$ with probability $\exp(-\gamma||\mathbf{u}_i - \mathbf{u}_j||_2^2)$ into set N_i. Then, we derive the local latent variable as

$$\mathbf{z}_i \sim \mathcal{N}\left(\sum_{j \in N_i} \Theta_1(\mathbf{u}_j), \exp(\sum_{j \in N_i} \Theta_2(\mathbf{u}_j))\right) \tag{13}$$

where Θ_1 and Θ_2 are feed-forward networks.

3) *Predictive Distribution Decoder*: The final step of TSFNP's stochastic process involves combining the representations from Probabilistic Neural Encoder and Stochastic Data Correlation Graph that capture relevant sequential and contextual representation and uncertainty of input time-series. We combine the latent stochastic embedding, local latent variable and combined information of all past sequences to derive the parameters of the output distribution via a simple feed-forward network. We first derive a *global latent variable* that combines the information from local latent embeddings of all past sequences as $\mathbf{z} = \text{Self-Atten}(\{\mathbf{u}_i\}_{i=1}^N)$ via a self-attention layer over $\{\mathbf{u}_i\}_{i=1}^N$ and summation of self-attention layer's output.

Finally, we combine the latent embedding of input time-series, local latent variable and global latent variable to derive the base forecast distribution modeled as a Gaussian $\mathcal{N}(\mu_i, \sigma_i)$ as:

$$\mathbf{e} = \text{concat}(\mathbf{u}_i, \mathbf{z}_i, \mathbf{z}), \quad [\mu_i, \log \sigma_i] = \Theta_3(\mathbf{e}) \tag{14}$$

where Θ_3 is a feed-forward network.

The full stochastic process of TSFNP can be summarized as:

$$P(\{\mu_i, \sigma_i\}_{i=1}^N | \mathcal{D}^t) = \int \underbrace{\left(\prod_{i=1}^N P(\mathbf{u}_i | \mathbf{y}_i^{(1:t)})\right)}_{\text{Probabilistic Encoder}} \underbrace{\left(\prod_{i=1}^N P(N_i | \{\mathbf{u}_i\}_{i=1}^N) P(\mathbf{z}_i | N_i, \{\mathbf{u}_j\}_{j=1}^N)\right)}_{\text{SDCG}} \underbrace{P(\mathbf{z} | \{\mathbf{u}_i\}_{i=1}^N)}_{\text{Global Latent variable}} \underbrace{\left(\prod_{i=1}^N P(\mu_i, \sigma_i | \mathbf{z}, \mathbf{z}_i, \mathbf{u}_i)\right)}_{\text{Raw forecasts}} d\{\mathbf{u}_i\}_{i=1}^N d\{\mathbf{z}_i\}_{i=1}^N d\{N_i\}_{i=1}^N. \tag{15}$$

Note that in the main paper we note the set of all latent variables $\{\mathbf{u}_i, \mathbf{z}_i, N_i\}_{i=1}^N, \mathbf{z}$ as $\mathcal{Z}$.

Note on running time. The novel component of PROFHIT is the Hierarchy-aware refinement module that facilitates the integration of base forecast distributions. As described in lines 398-403, the total computational complexity of obtaining the refined distributional parameters is $O(N^2)$ (N is the number of nodes in the hierarchy), which is comparable to the reconciliation step of end-to-end methods like HIERE2E. Post-processing techniques such as MINT, ERM, and PEMBU have an even higher time complexity of $O(N^3)$.

Note that the other portion of the pipeline that may add to the time-complexity is the base forecasting models. Models like DEEPAR and RNN used by HIERE2E, SHARQ as well as the post-processing methods and TSFNP (used by PROFHIT) scale linearly with respect to the length of the time-series and linearly with the number of nodes N. Therefore all these baselines and PROFHIT use models with similar time-complexity for base forecasts with respect to the size of the hierarchy N.

B CODE AND DATASET

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 PROFHIT along with the datasets used at https://github.com/AdityaLab/Profhit.

C HYPERPARAMETERS

C.1 Data Preprocessing

Most datasets used in our work assume the aggregation function to be a simple summation (i.e, $\phi_{ij} = 1$ for all weights). We first normalize the values of leaf time-series training data to have 0 mean and variance of 1. Since the aggregation of values at higher levels of the hierarchy can lead to very large values in time-series, we instead divide each non-leaf time-series by the number of children. Then the weights of hierarchical relations become $\phi_{ij} = \frac{1}{|C_i|}$ where C_i is the set of all children nodes of time-series i. For the remaining datasets (Flu-Symptoms, FB-Symptoms) the time-series values are normalized by default and thus require no extra pre-processing.

C.2 Model Architecture

The architecture of TSFNP used in PROFHIT is similar to that used in the original implementation [14]. The GRU unit contains 60 hidden units and is bi-directional. Thus the local latent variable is also of dimension 60. NN_1 and NN_2 are both 2-layered neural networks with the first layer shared between both. Both layers have 60 hidden units. Finally, NN_3 is a three-layer neural network with the input layer having 180 units (for the concatenated input of three 60-dimensional vectors) and the last two layers having 60 hidden units. We found that the value of c in Equation 4 is not very sensitive and is usually set to 5.

Note that we do not explicitly model covariance between every pair of time series (like MINT, ERM) and use a weighted combination of base forecast parameters to derive refined forecasts. Therefore the refinement module complexity (Section 4.1) is $O(N^2)$ which is on par with previous methods like HIERE2E.

C.3 Training and Evaluation

Given the training dataset $\mathcal{D}_t$ we extract the training dataset for each node as the set of prefix sequences $\{(\mathbf{y}_i^{(t1:t2)}, y_i^{(t2+1)}) : 1 \leq t1 \leq t2 < t - \tau\}$ and train the full model (TSFNP and refinement module). We tune the hyperparameter using backtesting by validating on window $t - \tau$ to t. Finally, we train for entire training set with the best hyperparameters.

For each benchmark, we used the validation set to mainly find the optimal batch size and learning rate. We searched over batch-size of $\{10, 50, 100, 200\}$ and the optimal learning rate was usually around 0.001. We also found the optimal λ to be around 0.01 for strongly consistent datasets and 0.001 for weakly consistent datasets. We used early stopping with the patience of 150 epochs to prevent overfitting. For each independent run of a model, we initialized the random seeds from 0 to 5 for PyTorch and NumPy. We didn't observe large variations due to randomness for PROFHIT and all baselines.

During evaluation, we sampled 2000 Monte-Carlo samples of the forecast distribution and used it to estimate the mean for MAPE. We also used the samples mean and variance to evaluate LS and CS whereas used ensemble scoring to evaluate CRPS directly from the samples using `properscoring` package [2].

[2] https://github.com/properscoring/properscoring

D DERIVATION OF LIKELIHOOD ELBO LOSS

The full predictive distribution of PROFHIT from Equation 11 can be further expanded as:

$$P(\{y_i^{(t+\tau)}\}_{i=1}^N | \mathcal{D}_t) = \int \underbrace{\left(\prod_{i=1}^N P(\mathbf{u}_i | \mathbf{y}_i^{(1:t)})\right)}_{\text{Probabilistic Encoder}} \underbrace{\left(\prod_{i=1}^N P(N_i | \{\mathbf{u}_i\}_{i=1}^N) P(\mathbf{z}_i | N_i, \{\mathbf{u}_j\}_{j=1}^N)\right)}_{\text{SDCG}} \underbrace{P(\mathbf{z} | \{\mathbf{u}_i\}_{i=1}^N)}_{\text{Global Latent variable}} \underbrace{\left(\prod_{i=1}^N P(\mu_i, \sigma_i | \mathbf{z}, \mathbf{z}_i, \mathbf{u}_i)\right)}_{\text{Raw forecasts}} \underbrace{\prod_{i=1}^N P(\hat{\mu}_i, \hat{\sigma}_i | \{\mu_j, \sigma_j\}_{j=1}^N) P(y_i^{(t+\tau)} | \hat{\mu}_i, \sigma_i)}_{\text{Refinement Module}} d\{\mathbf{u}_i\}_{i=1}^N d\{\mathbf{z}_i\}_{i=1}^N. \tag{16}$$

To minimize the data likelihood $P(\{y_i^{(t+\tau)}\}_{i=1}^N | \mathcal{D}_t)$ requires integration over latent variables $\{\mathbf{u}_i\}_{i=1}^N$ and $\{\mathbf{z}_i\}_{i=1}^N$. We instead perform amortized variational inference on the latent variables similar to VAE [16].

We approximate the posterior of latent variables

$$P(\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N, \{N_i\}_{i=1}^N | \{y_i^{(t+\tau)}\}_{i=1}^N)$$

with a variational distribution

$$Q(\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N, \{N_i\}_{i=1}^N | \{y_i^{(t+\tau)}\}_{i=1}^N)$$

expressed as:

$$Q(\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N, \{N_i\}_{i=1}^N | \{y_i^{(t+\tau)}\}_{i=1}^N) = \left(\prod_{i=1}^N P(\mathbf{u}_i | \mathbf{y}_i^{(1:t)})\right) \left(\prod_{i=1}^N P(N_i | \{\mathbf{u}_i\}_{i=1}^N)\right) \left(\prod_{i=1}^N q_\phi(\mathbf{z}_i | \mathbf{y}_i^{(1:t)})\right) \tag{17}$$

where q_ϕ is a feed-forward network over GRU embeddings of Probabilistic Neural Encoder that parameterizes to a gaussian distribution of $\mathbf{z}_i$.

The ELBO loss

$$\begin{aligned} -\mathbb{E}_{Q(\{\mathbf{u}_i, \mathbf{z}_i, N_i\}_{i=1}^N | \{y_i^{(t+\tau)}\}_{i=1}^N)} \\ [\log P(\{y_i^{(t+\tau)}\}_{i=1}^N | \{\mathbf{u}_i, \mathbf{z}_i, N_i\}_{i=1}^N) \\ + \log P(\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N, \{N_i\}_{i=1}^N | \{y_i^{(t+\tau)}\}_{i=1}^N) \\ - \log Q(\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N, \{N_i\}_{i=1}^N | \{y_i^{(t+\tau)}\}_{i=1}^N)] \end{aligned} \tag{18}$$

get simplified to:

$$\mathcal{L}_1 = -E_{Q(\{\mathbf{u}_i,\mathbf{z}_i,N_i\}_{i=1}^N,\mathbf{z}|\{y_i^{(t+\tau)}\}_{i=1}^N)}[\log P(\{y_i^{(t+\tau)}\}_{i=1}^N|\{\mathbf{u}_i,\mathbf{z}_i,N_i\}_{i=1}^N) + \sum_{i=1}^{N}\log P(\mathbf{z}_i|\{\mathbf{u}_j\}_{j=1}^N,N_i) - \log q_i(\mathbf{z}_i|\mathbf{y}_i^{(t':t)})]. \tag{19}$$

by canceling similar terms between the variational and true distribution of latent variables.

E CONSISTENCY OF DATASETS

We noted in Section 5.2 Q4 that `Flu-Symptoms` and `FB-Survey` are weakly consistent datasets since they do not strictly follow the aggregation relations $H_{\mathcal{T}}$ unlike strongly consistent datasets `Tourism-L, Labour, Wiki`.

Table 5: *Average deviation of observed values in time-series from hierarchical relations.*

Data	Flu	FB-Survey	Tourism-L	Labour	Wiki
Level 1	0.043	1.27	0	0	0
Level 2	3.41	2.83	0	0	0
Average across hierarchy	3.37	2.44	0	0	0

We empirically observe this by measuring Consistency errors of all datasets (Definition 1) for the entire hierarchy and at each level of the hierarchy. The results are in Table 5. As expected there are no deviations for strongly consistent datasets whereas there is a significant deviation in weakly consistent data.

F PERFORMANCE ACROSS EACH LEVEL OF HIERARCHY

We compared the performance of PROFHIT with best-performing baselines HierE2E and SHARQ for each level of hierarchy of all datasets. PROFHIT significantly outperforms the best baselines as well as the variants. At the leaf nodes, which contains most data, PROFHIT outperforms best baselines by 7% in `Wiki` to 100% in `FB-Survey`. For the top node of time-series the performance improvement is largest at 35% (`Wiki`) to 962% (`FB-Survey`). We show detailed results in Tables 6 and 7.

Table 6: ***Average CRPS scores at each level of hierarchy. PROFHIT significantly outperforms best baselines across all benchmarks. Note that P-Finetune's performance decreases at higher levels of hierarchy compared to other variants whereas P-Global's performance is worse at lower levels.***

Models/Data	Tourism-L								Labour			
Hierarchy Levels	1	2(Travel)	3(Travel)	4(Travel)	5(Travel)	2(Geo)	3(Geo)	4(Geo)	1	2	3	4
HierE2E	0.081	0.103	0.141	0.205	0.272	0.103	0.136	0.175	0.031	0.034	0.034	0.038
SHARQ	0.093	0.131	0.163	0.218	0.295	0.131	0.138	0.152	0.097	0.124	0.133	0.149
PEMBU-MinT	0.112	0.121	0.139	0.203	0.185	0.116	0.128	0.167	0.063	0.033	0.042	0.085
PROFHIT (Ours)	**0.051**	**0.095**	**0.12**	**0.17**	**0.264**	**0.083**	**0.106**	**0.142**	**0.023**	**0.019**	**0.023**	**0.029**
P-FineTune	0.072	0.136	0.083	0.16	0.278	0.124	0.124	0.158	0.024	0.022	0.026	0.035
P-NoParamShare	0.093	0.113	0.122	0.13	**0.261**	0.093	0.113	0.147	**0.021**	0.027	0.028	**0.027**
P-DeepAR	0.075	0.097	0.136	0.183	0.281	0.095	0.122	0.159	0.025	0.027	0.031	0.033
P-NoConsistency	0.086	0.142	0.107	0.18	0.265	0.132	0.138	0.147	0.027	0.031	0.029	0.026
Models/Data	Wiki					Flu-Symptoms			FB-Survey			
Hierarchy Levels	1	2	3	4	5	1	2	3	1	2	3	
HierE2E	0.042	0.105	0.229	0.272	0.372	0.272	0.421	0.458	4.14	4.04	4.13	
SHARQ	0.039	0.136	0.235	0.291	0.378	0.258	0.376	0.381	3.08	3.21	3.13	
PEMBU-MinT	0.031	0.171	0.241	0.385	0.433	0.337	0.567	0.773	4.82	5.53	6.15	
PROFHIT (Ours)	**0.031**	**0.074**	**0.133**	**0.216**	**0.252**	**0.216**	**0.133**	**0.338**	**0.32**	**0.43**	**1.89**	
P-FineTune	0.034	0.086	0.153	0.232	0.275	0.222	0.175	**0.293**	0.43	0.65	**1.83**	
P-NoParamShare	0.048	0.103	0.187	0.265	**0.186**	0.269	0.213	0.376	0.37	**0.37**	2.11	
P-DeepAR	0.035	0.094	0.193	0.251	0.285	0.242	0.217	0.328	0.44	0.61	2.01	
P-NoConsistency	0.49	0.117	0.93	0.258	0.167	0.227	0.193	0.381	0.42	**0.36**	2.18	

Table 7: ***Average CS scores at each level of hierarchy. PROFHIT significantly outperforms best baselines across all benchmarks.***

Models/Data	Tourism-L								Labour			
Hierarchy Levels	1	2	3	4	5	2(Geo)	3(Geo)	4(Geo)	1	2	3	4
HierE2E	0.15	0.18	0.17	0.21	0.24	0.19	0.18	0.22	0.21	0.23	0.22	0.27
SHARQ	0.09	0.08	0.12	0.11	0.14	0.11	0.12	0.16	0.16	0.16	0.15	0.21
PEMBU-MinT	0.14	0.21	0.22	0.21	0.26	0.18	0.23	0.25	0.21	0.22	0.24	0.21
PROFHIT	**0.05**	**0.06**	**0.04**	**0.06**	**0.11**	**0.06**	**0.06**	**0.1**	**0.17**	**0.11**	**0.15**	**0.16**
P-FineTune	0.09	0.12	0.13	0.17	0.13	0.11	0.13	0.15	0.24	0.21	0.24	0.22
P-NoParamShare	0.06	**0.04**	**0.03**	0.08	**0.05**	**0.05**	**0.03**	**0.04**	**0.14**	0.18	0.19	**0.15**
P-DeepAR	0.11	0.09	0.09	0.14	0.13	0.15	0.14	0.13	0.14	0.19	0.17	0.14
P-NoConsistency	0.18	0.19	0.17	0.19	0.22	0.18	0.19	0.24	0.24	0.22	0.25	0.31
Models/Data	Wiki					Flu-Symptoms			FB-Survey			
Hierarchy Levels	1	2	3	4	5	1	2	3	1	2	3	
HierE2E	0.15	0.21	0.26	0.22	0.24	0.11	0.13	0.11	0.21	0.19	0.18	
SHARQ	0.13	0.14	0.14	0.17	0.15	0.58	0.052	0.085	0.16	0.14	0.15	
PEMBU-MinT	0.12	0.11	0.12	0.13	0.14	0.17	0.22	0.17	0.2	0.19	0.16	
PROFHIT	**0.11**	**0.15**	**0.12**	**0.14**	**0.11**	**0.031**	**0.044**	**0.052**	**0.09**	**0.07**	**0.06**	
P-FineTune	0.19	0.18	0.23	0.22	0.24	0.033	**0.031**	**0.042**	**0.05**	**0.06**	0.09	
P-NoParamShare	0.16	**0.15**	0.16	0.17	0.15	0.065	0.072	0.096	0.11	0.13	0.17	
P-DeepAR	0.21	0.24	0.26	0.22	0.23	0.064	0.077	0.083	0.15	0.19	0.17	
P-NoConsistency	0.29	0.28	0.35	0.33	0.37	0.22	0.18	0.14	0.22	0.25	0.21	

Table 8: ***Std. dev of CRPS and LS (accros 5 runs) across all levels for all baselines, PROFHIT and its variants. PROFHIT performs significantly better than all baselines as noted using t-test with*** $\alpha = 1\%$.

	Models/Data	Tourism-L		Labour		Wiki		Flu-Symptoms		FB-Survey	
		CRPS	LS	CRPS	LS	CRPS	LS	CRPS	LS	CRPS	LS
Baselines	DeepAR	0.011	0.040	0.004	0.038	0.002	0.044	0.018	0.098	0.482	0.434
	TSFNP	0.006	0.021	0.003	0.018	0.015	0.069	0.019	0.004	0.251	0.217
	MinT	0.005	0.019	0.002	0.121	0.018	0.006	0.014	0.111	0.468	0.213
	ERM	0.044	0.005	0.002	0.110	0.016	0.069	0.018	0.133	0.148	0.209
	HierE2E	0.001	0.038	0.003	0.049	0.019	0.018	0.005	0.051	0.325	0.109
	SHARQ	0.000	0.011	0.001	0.046	0.017	0.007	0.002	0.116	0.133	0.048
	PROFHIT (Ours)	0.001	0.017	0.001	0.003	0.001	0.030	0.005	0.009	0.040	0.008
Ablation	P-FineTune	0.016	0.031	0.003	0.003	0.016	0.014	0.001	0.006	0.090	0.004
	P-NoParamShare	0.012	0.033	0.000	0.013	0.002	0.001	0.033	0.024	0.248	0.119
	P-DeepAR	0.006	0.026	0.001	0.028	0.005	0.043	0.035	0.030	0.103	0.065
	P-NoConsistency	0.005	0.012	0.001	0.009	0.015	0.043	0.012	0.025	0.110	0.053

G DETAILS ON DATA IMPUTATION EXPERIMENT

Motivation: During real-time forecasting in real-world applications such as Epidemic or Sales forecasting, we encounter situations where the past few values of time-series are missing or unreliable for some of the nodes. This is observed specifically at lower levels, due to discrepancies or delays during reporting and other factors [4]. Therefore, one approach to performing forecasting in such a situation is first by imputation of missing values based on past data and then using the predicted missing values as part of the input for forecasting.

Task: To simulate such scenarios of missing data and evaluate the robustness of PROFHIT and all baselines, we design a task called *Hierarchical Forecasting with Missing Values* (HFMV). Formally, at time-period t, we are given full data for up to time $t - \rho$. We show results here for $\rho = 5$ which is the average forecast horizon of all tasks. For sequence values in the time period between $t - \rho$ and t, we randomly remove $k\%$ of these values across all time-series. The goal of HFMV task is to use the given partial dataset from $t - \rho$ to t as input along with complete dataset for time-period before $t - \rho$ to predict future values at $t + \tau$. Therefore, success in HFMV implies that models are robust to missing data from the recent past by effectively leveraging hierarchical relations.

Setup: We first train PROFHIT and baselines on complete dataset till time t' and then fill in the missing values of input sequence using the trained model. Using the predicted missing values, we again forecast the output distribution. For each baseline and PROFHIT, we perform multiple iterations of Monte-Carlo sampling for missing values followed by forecasting future values to generate the forecast distribution. We estimate the evaluation scores using sample forecasts from all sampling iterations.

H ADAPTING TO VARYING DATASET CONSISTENCY

Observation 1. *The average improvement in performance of PROFHIT over best forecasting baselines is 72% higher for weakly consistent datasets over its improvement for strongly consistent datasets.*

Since most previous state-of-art models assume datasets to be strongly consistent, deviations to this assumptions can cause underperformance when used with weakly consistent datasets. This is evidenced in Table 3 where some of the baselines like MinT and ERM that explicitly optimize for hierarchical consistency perform worse than even TSFNP, which does not leverage hierarchical relations, in `Flu-Symptoms` and `FB-Survey`. Overall, we found that for weakly consistent datasets, PROFHIT provides a much larger 93% average improvement in CRPS scores over the best baselines compared to 54% average improvement for strongly consistent datasets. These improvements are more pronounced at non-leaf nodes of hierarchy where PROFHIT improves by 2.8 times for `Flu-Symptoms` and 9.2 times for `FB-Survey`. This is because the baselines which assume strong consistency do not adapt to noise at leaf nodes that compound to errors at higher levels of hierarchy.

Observation 2. *PROFHIT's approach to parameter sharing and soft consistency regularization helps adapt to varying hierarchical consistency.*

We observe that that best performing variant for strongly consistent datasets in P-NoParamShare which is trained with both likelihood loss and SoftDisCoR (Table 3). But its performance severely degrades for weakly consistent datasets since sharing all model parameters across all time-series makes it inflexible to model patterns and deviations specific to individual nodes. In contrast, P-FineTune and P-NoConsistency performs the best among variants for weakly consistent datasets since they train separate sets of decoder parameters for each node. But they perform poorly for strongly consistent datasets since they don't leverage Distributional Consistency effectively. PROFHIT combines the flexible parameter learning of P-FineTune and leverage Distributional Consistency to jointly optimize the parameters like P-NoParamShare providing comparable performance to best variants over all datasets.

Table 9: Average value of γ_i for all datasets. Note that weakly consistent datasets have higher γ_i (depends mode on past data of same time-series) where as strongly-consistent data have lower γ_i (leverages the hierarchical relations).

Consistency	Dataset	Average value of γ_i
Strong	Tourism-L	0.420 ± 0.096
	Labour	0.348 ± 0.091
	Wiki	0.313 ± 0.057
Weak	Symp	0.759 ± 0.152
	Fbsymp	0.789 ± 0.180

Observation 3. *PROFHIT's Refinement module automatically learns to adapt to varying hierarchical consistency.*

The design choices of the refinement module help PROFHIT to adapt to datasets of different levels of hierarchical consistency. Specifically, by optimizing for values of $\{\gamma_i\}_{i=1}^{N}$ of Equation 3, PROFHIT aims to learn a good trade-off between leveraging prior forecasts for a time-series and hierarchical relations of forecasts from the entire hierarchy. We study the learned values of $\{\gamma_i\}_{i=1}^{N}$ of Equation 3 used to derive refined mean. Note that higher values of γ_i indicate larger dependence on base forecasts of node and smaller dependence of forecasts of the entire hierarchy. We plot the average values of γ_i for each of the datasets in Table 9. We observe that strongly consistent datasets have lower values of γ_i indicating that PROFHIT's refinement module automatically learns to strongly leverage the hierarchy for these datasets compared to weakly consistent datasets.