ChronosX: Adapting Pretrained Time Series Models with Exogenous Variables

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Covariates provide valuable information on external factors that influence time series and are critical in many real-world time series forecasting tasks. For example, in retail, covariates may indicate promotions or peak dates such as holiday seasons that heavily influence demand forecasts. Recent advances in pretraining large language model architectures for time series forecasting have led to highly accurate forecasters. However, the majority of these models do not readily use covariates as they are often specific to a certain task or domain. This paper introduces a new method to incorporate covariates into pretrained time series forecasting models. Our proposed approach incorporates covariate information into pretrained forecasting models through modular blocks that inject past and future covariate information, without necessarily modifying the pretrained model in consideration. In order to evaluate our approach, we introduce a benchmark composed of 32 different synthetic datasets with varying dynamics to evaluate the effectivity of forecasting models with covariates. Extensive evaluations on both synthetic and real datasets show that our approach effectively incorporates covariate information into pretrained models, outperforming existing baselines.

1 Introduction

Many decision and planning processes require an estimate of how historical time series will evolve in the future. There are numerous instances of this problem across many application domains, such as retail [5, 8, 26], energy [11, 34] and traffic [22, 23]. For example, forecasting electricity demand can inform the production capacity that is required to meet the energy demand. Together with forecasts for renewable energy, this can inform the mix of renewable and fossil energy sources that provides stable energy supply but minimizes carbon emissions. Another example is inventory planning in retail, where the customer demand informs the number of items that need to be stocked. This task of estimating the future of a time series from its past is called time series forecasting. Time series forecasting can be addressed by many different models. Statistical local models (ARIMA [6], ETS [19] and Theta [4]) fit their model parameters per time series. Many deep learning methods (DeepAR [32], TFT [24], PatchTST [27]) act as global models and learn parameters for all time series in a dataset from the same domain. Recently, a new class of time series models emerged that are trained *across* datasets from *different* domains. These models are called pretrained forecasting models and have been recently shown to be effective zero-shot forecasters [3, 9, 12, 18, 37].

One important component for time series forecasting –regardless of the forecasting model class– are exogenous variables, also called covariates. Covariates provide external information to the forecasting model and allow the forecasting model to adjust the forecast based on this additional information. Coming back to the renewable energy example, the predicted energy output of a wind farm can be informed by additional weather information like the forecasted wind speed. While it is straightforward to incorporate covariate information into local statistical models and global deep learning (or other machine learning) models, it is less clear how this can be achieved for pretrained forecasting models. Pretrained forecasting models are trained across different datasets from different domains and every dataset might have a different number of covariates with different properties, making it difficult to pretrain these models with covariates. Consequently, the majority of pretrained time series models do not support covariate data [3, 9, 12, 18]. One notable exception is Moirai, which introduces one possible way to address this problem with a mechanism called "any-variate attention" that is able to pretrain on datasets with a varying number of covariates [37]. However, how to include covariates into pretrained models that do not natively support covariates remains an open question.

In this paper, we present an approach called CHRONOSX to include covariate data into the pretrained forecasting model Chronos [3]. CHRONOS is a pretrained model that is trained on a time series dataset corpus without covariates. Drawing inspiration from modular deep learning [31], our proposed approach consists of two modules. The first module updates the pretrained token embeddings with past covariates and the second module uses future covariates to adjust the output distribution. These light-weight adapter modules can be quickly trained for a downstream forecasting task even when the underlying pretrained model is frozen.

To validate our approach CHRONOSX we introduce a novel collection of 32 datasets that emulates different kinds of covariates together with several and time dynamics. We also demonstrate that this model achieves low forecasting error across 20 real-world datasets with covariates. Furthermore, we show that the adapter approach can be used to finetune a pretrained time series model on a target dataset, which is much faster than full finetuning of the entire pretrained model. In summary, we make the following contributions: 1) We introduce CHRONOSX, an adapter approach for incorporating covariates into pretrained models for time series forecasting, 2) we demonstrate that CHRONOSX achieves low forecasting error both for synthetic and real world datasets with covariates, 3) we demonstrate that the adapters also allow fine tuning without covariates and are a fast alternative to full finetuning, 4) we introduce a novel benchmark of synthetic datasets that allows to evaluate forecasting models using covariates under different dynamics.

2 Background and Related Work

Time series forecasting is the task of extrapolating a time series into the future based on its historical values and, optionally, a set of covariates. These covariates are external variables that potentially influence the primary time series and may enhance the forecasting model's accuracy by providing additional context (e.g., weather may influence the sales of fans). In its general form, time series forecasting can be casted as the problem of modeling the conditional distribution,

$$P(\mathbf{z}_{C+1:H}|\mathbf{z}_{1:C}, \mathbf{X}_{1:H}; \Phi), \tag{1}$$

where $\mathbf{z}_{1:C} = [z_1, \dots, z_C]$ is the historical context of the primary time series, $\mathbf{z}_{C+1:H} = [z_{C+1}, \dots, z_H]$ is the future target until horizon H, $\mathbf{X}_{1:H} = [\mathbf{x}_1, \dots, \mathbf{x}_H]$, are covariates from the historical context until horizon H, and Φ denotes a set of learnable parameters. In this paper, we focus on the case where the primary time series is univariate, $z_t \in \mathbb{R}$, while the the covariates may have multiple dimensions, $\mathbf{x}_t \in \mathbb{R}^c$.

A wide variety of time series forecasting models have been proposed in the literature. Based of how they leverage time series data for training, they may be categorized into: *local* and *global* models. Local models such as ARIMA [6] and ETS [19] are fit individually for each time series. On the other hand, global models such as DeepAR [32], TFT [24], PatchTST [27], and NHiTS [7] are trained across multiple time series from a given dataset, and are able to leverage global patterns present within a dataset.

Pretrained time series models. Recent work [3, 9, 18, 37] on large scale training of time series models has given rise to a special category of global models called *pretrained models* (also referred to as "foundation" models). Pretrained models are trained on a large corpus of time series data and can be used for accurate zero-shot forecasting or finetuned on downstream datasets and tasks. Moirai [37], MOMENT [18], and TimesFM [9] are pretrained models based on patching [27]. Specifically, they convert time series into patches before processing these patches with transformer-based models. On the other hand, CHRONOS [3] directly maps time series values into tokens from a fixed vocabulary via scaling and quantization, and trains existing language models architectures on such *time series tokens*. Ansari et al. [3] show that this simple tokenization scheme is surprisingly very effective and

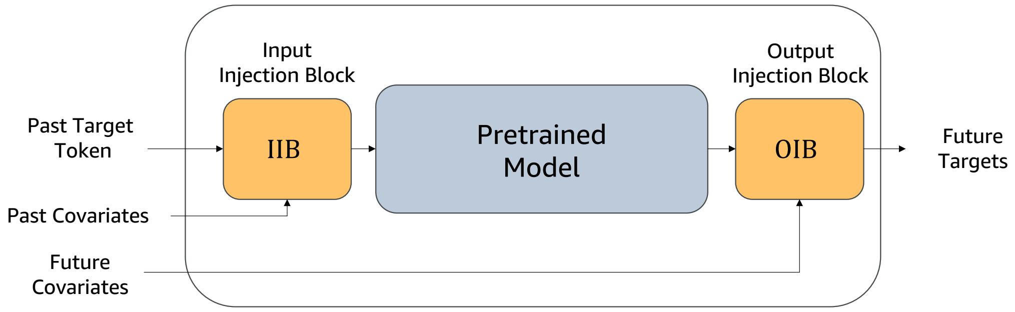


Figure 1: Architecture of CHRONOSX. It consists of two blocks where past and future covariates are added. Observe that the first block add covariates of the past and updates token embeddings whereas the second block is focused on future covariates and updates the estimate distribution of the pretrained model.

CHRONOS models achieve zero-shot performance comparable to deep learning models with access to training data.

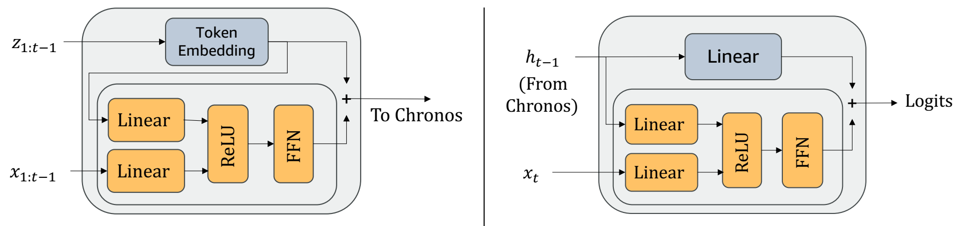
Forecasting with covariates. Covariates are essential to account for external events into forecasting models, such as peak dates in retail, planned strikes that might disrupt traffic, or sports events that generate massive online attendance. Forecasting models incorporate covariates in different ways. For instance, in DeepAR [32] covariates are part of the input to the RNN block, whereas TFT [24] explicitly encodes past and future covariates with different encoding networks whose outputs are concatenated. ARIMA uses extra coefficients to account for covariates, and NBEATSx [28] and N-HiTS [7] concatenate time series in the context with past and future covariates and compute a fixed input size. To the best of our knowledge, the only pretrained model that consumes covariates is Moirai [37], which flattens time series and covariates into a single sequence and assigns a variate ID to distinguish between target time series and covariates. In this paper we introduce CHRONOSX, an adapter approach inspired by modular deep learning [31], for incorporating covariates into the univariate CHRONOS [3] model which does not support covariates natively.

3 CHRONOSX: Covariate Integration in Pretrained Models

In this section we introduce the proposed model to add covariates to pretrained models. We denote by **CHRONOSX** the resulting model of adding covariates to the pretrained model CHRONOS [3]. See Figure 1 for an overview of the model architecture.

CHRONOS is a pretrained model for univariate probabilistic time series forecasting based on the encoder-decoder T5 transformer models. CHRONOS applies a preprocessing scheme where time series are first mean-scaled and then tokenized through a suitable bin-quantization approach. The resulting tokenization is then passed to generate the corresponding pretrained embedding and fed into the encoder. CHRONOS uses the categorical distribution over elements of the vocabulary of tokens as output distribution, and is trained to minimize the cross entropy between the distribution of the quantized ground truth label and the predicted distribution. For more details we refer the reader to [3].

Our novel approach CHRONOSX adds the information of covariates in two different stages: 1) we add covariates from the past to update the corresponding token embeddings, and 2) we add covariates of the future to adjust the logits. Observe that, depending on the user context, one can choose to either work with covariates of the past, the future, or both. Following a modular deep learning paradigm [31], we propose modules to achieve this while minimizing the modifications to the original model. CHRONOSX allows us to freeze any arbitrary portion of the original architecture and just update the proposed modules. In the following sections we refer as Input Injection Block (IIB) and Output Injection Block (OIB) to the adapters injecting covariates from the past and the future, respectively. See Figure 2 for a depiction of these. Additionally, we will overload the notation and use $\mathbf{z}$ for tokenized time series. In what follows we will make use of fully connected feed-forward networks consisting of two linear transfomration with a ReLU activation in between, $\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$.



(a) Input Injection Block (IIB) (b) Output Injection Block (OIB)

Figure 2: Description of the input and output injection blocks used in ChronosX (cfr. Figure 1 for the full model structure). Observe that IIB and OIB take past and future covariates, respectively. Figure 2a shows that the input injection block takes a pretrained tokenized embedding together with the corresponding covariates of the past, whereas Figure 2b takes the pretrained logits (expressed as the final hidden state multiplied by a pretrained matrix) together with covariates of the future. The blue color indicates that the module is taken from the pretrained model.

Input Injection Block. To add information of covariates from the past we update the token embeddings. Specifically, for every time step $\mathbf{z}_{t-1}$ we pass the corresponding token embeddings and covariates through independent linear layers and apply an FFN to the ReLU of the concatenated mappings. Let $h_{\text{emb}}(\mathbf{z}_{t-1}) \in \mathbb{R}^{d_{\text{model}}}$ be the pretrained token embedding, the updated embeddings f_{IIB} are defined as

$$f_{\text{IIB}}\left(\mathbf{z}_{t-1}, \mathbf{x}_{t-1}\right) = h_{\text{emb}}(\mathbf{z}_{t-1}) + g_{\text{IIB}}(h_{\text{emb}}(\mathbf{z}_{t-1}), \mathbf{x}_{t-1}) \tag{2}$$

where

$$g_{\text{IIB}}(h_{\text{emb}}(\mathbf{z}_{t-1}), \mathbf{x}_{t-1}) = \text{FFN}\left(\text{ReLU}\left(h_{\text{emb}}(\mathbf{z}_{t-1})W_{\text{IIB}}^{(\text{emb})} \oplus \mathbf{x}_{t-1}W_{\text{IIB}}^{(\text{cov})}\right)\right) \tag{3}$$

with linear mappings $W_{\text{IIB}}^{(\text{emb})}$ and $W_{\text{IIB}}^{(\text{cov})}$ and $\oplus$ represents the concatenation operator. The function g_{IIB} adjusts the embeddings of the tokenized time series value $\mathbf{z}_{t-1}$ by merging information of covariates from the past $\mathbf{x}_{t-1}$ and the corresponding time series embedding $h_{\text{emb}}(\mathbf{z}_{t-1})$.

Output Injection Block. We leverage information from future covariates $\mathbf{x}_t$ by adjusting the logits of the pretrained model. These are generated through a matrix multiplication between the last hidden state and a pretrained matrix [35]. The adjusted logits are defined as follows: let $h_{\text{out}}(\mathbf{z}_{t-1})$ be the final hidden state corresponding to $\mathbf{z}_{t-1}$, and W_{out} the pretrained matrix producing the logits in the model, then the adjusted logits f_{OIB} are defined as

$$f_{\text{OIB}}\left(\mathbf{z}_{t-1}, \mathbf{x}_{t}\right) = h_{\text{out}}(\mathbf{z}_{t-1})W_{\text{out}} + g_{\text{OIB}}(h_{\text{out}}(\mathbf{z}_{t-1}), \mathbf{x}_{t}) \tag{4}$$

where g_{OIB}

$$g_{\text{OIB}}(h_{\text{out}}(\mathbf{z}_{t-1}), \mathbf{x}_{t}) = \text{FFN}\left(\text{ReLU}\left(h_{\text{out}}(\mathbf{z}_{t-1})W_{\text{OIB}}^{(\text{out})} \oplus \mathbf{x}_{t}W_{\text{OIB}}^{(\text{cov})}\right)\right) \tag{5}$$

with parameter matrices $W_{\text{OIB}}^{(\text{out})}$ and $W_{\text{OIB}}^{(\text{cov})}$, and $\oplus$ represents the concatenation operator. Observe that the inputs considered for this update are covariates from the future and the last hidden state evaluated on time series values from the past.

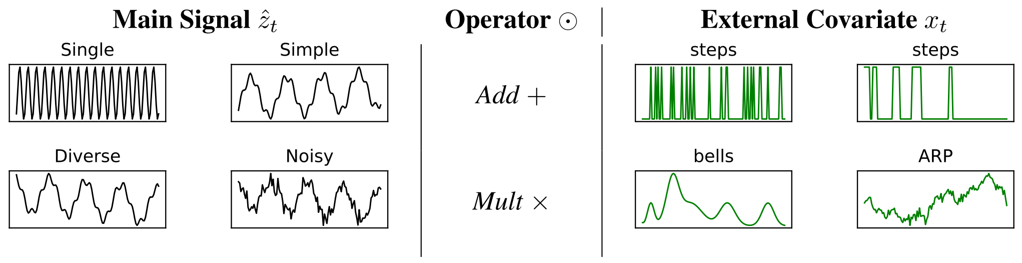
Hidden State Only and Covariates Only (Residual) Injection Block. To understand the impact of the proposed output and input injection blocks, we present the following variants. First, we propose a hidden-state base adapter that does not take covariates into account and works as an alternative to full finetuning without covariates, defined as

$$f_{\text{HS}}(h_{\text{out}}(\mathbf{z}_{1:t-1})) = \text{FFN}\left(\text{ReLU}\left(h_{\text{out}}(\mathbf{z}_{1:t-1})W_{\text{OIB}}^{(\text{out})}\right)\right) \tag{6}$$

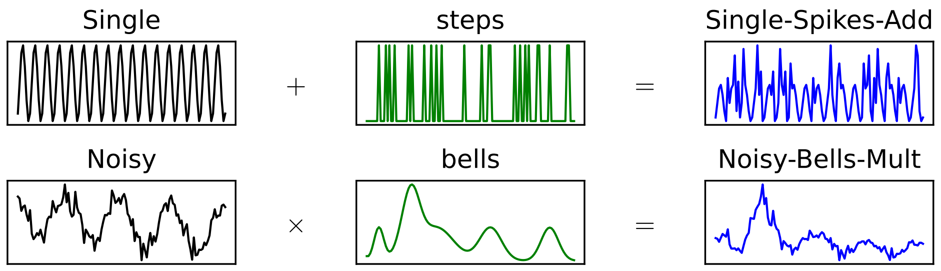
and second, we present an adapter to incorporate covariates of the future by updating logits through a linear transformation.

$$f_{\text{RS}}\left(\mathbf{z}_{1:t-1}, \mathbf{x}_{t}\right) = h_{\text{out}}(\mathbf{z}_{1:t-1})W_{\text{out}} + \mathbf{x}_{t}W_{\text{OIB}}^{(\text{cov})} \tag{7}$$

Modular variants of ChronosX. Due to its modular nature, we can adapt ChronosX depending on the kind of covariates that we want to incorporate. In what follows we will denote by ChronosX



(a) In order to generate the synthetic datasets, we choose one element for each building block. Then, a 100 times series are generated by randomly sampling parameters for each building blocks.



(b) Two examples of generated time-series.

Figure 3: An illustration of the process used to generate synthetic dataset. By considering different combination of the 4 possible main signals modified by one of the 4 possible external covariates through a chosen operator between $+$ and $\times$, we are able to generate realistic time series from the given model.

the case where both past and future covariates are incorporated, i.e. IIB and OIB are used. Yet, we can extend our model to the case where only covariates of the past are available by only considering the input injection block. Similarly, for the case where we only want to consume covariates we can consider only the output injection block. When training CHRONOSX, we train from scratch the parameters of the Feed-Forward Netowrks (FFN) and new matrices $W_{\text{IIB}}^{(\text{emb}}, W_{\text{IIB}}^{(\text{cov}}, W_{\text{OIB}}^{(\text{cov})}, W_{\text{OIB}}^{(\text{out})}$. Additionally, we consider a full-finetuned variant CHRONOSX(FF) where all the parameters, including the pretrained ones, are updated. Whereas in the main paper we will only report results for CHRONOSX, in the appendix we present evaluations of additional variants. For instance, we study the case where either only past or only future covariates are considered, together with the residual-based approach.

4 Synthetic Datasets with Covariates

Although the task of forecasting time series with covariates is a highly relevant task, there is a limited amount of freely available time series datasets with covariates where forecasting models can be evaluated. To overcome this limitation in this paper we introduce a collection of 32 different synthetic datasets of time series with covariates carefully handcrafted to emulate real-life applications where external factors greatly impact the time series.

Each of the datasets contains 100 times series of daily frequency extending over 1827 days[1]. For each dataset, we consider 3 basic elements that allow for the generation of 100 times series with covariates: 1) the **main signal** ($\hat{z}_t$), 2) an **external covariate** (x_t), and 3) a **combination operator** $\odot \in \{+, \times\}$. The generation of time series with these three ingredients is defined by $z_t = \hat{z}_t \odot x_t$. In what follows we present several variations of these three basic elements with the goal of providing a comprehensive evaluation of suitable forecasting models for time series with covariates.

The time series representing the main signal $\hat{z}_t$ are divided in two types: **Simple** and **Complex**. The simple synthetic type includes single sinusoids and simple sinusoids variants, whereas the complex synthetic type includes diverse sinusoids and noisy sinusoids variants. We now describe each of the

[1]This corresponds to the number of days in the fictional time series between 01/01/2025 and 31/12/2029.

four different variants of the main signal $\hat{z}_t$ with increasing complexity with the form of Eq. (8), mainly: 1) *Single sinusoids* considers a standard sine curve with fixed zero phase, unit amplitude and period of 7 days corresponding to one week for the entire dataset; 2) *Simple sinusoids* is the sum of three sine curves each with a weekly, monthly and yearly period, together with a randomly sampled amplitude and zero phase; 3) *Diverse sinusoids* is the result of combining three sine curves with randomly sampled phases and amplitudes and with a global trend, and 4) *Noisy sinusoids* which results from adding gaussian noise (ϵ) to *diverse sinusoids* and allows us to assess model robustness. In Appendix E, we give details on the process to generate the dataset.

$$\hat{z}_t = \sum_{i=1}^{3} a_i \sin\left(f_i t + \phi_i\right) + b_1 t + b_2 + \epsilon \tag{8}$$

We design four different kinds of **external covariates**: 1) *spikes* models short time events such as strikes or power failure, 2) *steps* model longer events with sudden and sustained changes such as discounts in the retail industry, 3) *bells* represent smoother changes such as festive season, and 4) *autoregressive process* with randomly generated parameters that model cases where the value of covariates is determined by previous observations. Finally, the **combination operator** $\odot$ is defined either as the addition $+$ or the multiplication $\times$ between the main signal ($\hat{z}_t$) and the covariates (x_t). An illustration of the generative process for this synthetic datasets along with a few examples of the resulting time series is depicted in fig. 3. In Section 5.1 we show that the resulting 32 different synthetic datasets here introduced are useful to understand how different forecasting model behave under different scenarios. For further details please see the appendix.

5 Experiments and Results

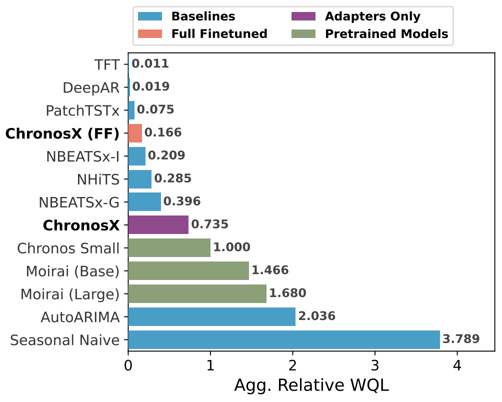
Setup. In all experiments for our models we consider learning rates $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and select the one with the best validation error with a validation window of length equal to the prediction length of the dataset in consideration. A detailed description of the hyperparameters is presented in the appendix.

Metrics. We consider the Weighted Quantile Loss (WQL) to evaluate probabilistic forecasts, which measure the alignment of quantile levels of the predictive distribution with respect to the ground truth [15, 16, 33]. In all experiments the WQL is computed on quantile levels $\{0.1, 0.2, \ldots, 0.9\}$. For methods generating sample forecasts we compute the quantiles based on 100 samples, whereas quantile forecasting methods are trained on the same quantile levels we use for evaluation. Further, we consider the Mean Absolute Scaled Error (MASE), which measures deviations from the median forecast with the ground-truth values [20]. Finally, we follow [3, 13, 37] and report the aggregated WQL and MASE computed as the geometric mean of normalized scores by the corresponding baseline. Please see the appendix for more details.

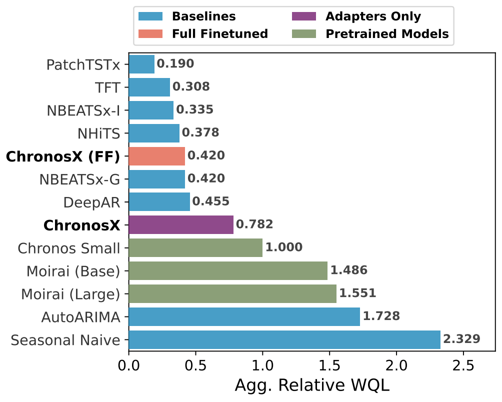
5.1 Finetuning and Adapters on Synthetic Benchmark Datasets with Covariates

In this section we evaluate how effective are different forecasting models in incorporating external information through covariates under different kinds dynamics. For this, we take the 32 different datasets as introduced in Section 4 and follow the suggested splits Simple and Complex dataset collections each consisting of 16 datasets, and for each dataset we generate 100 time series of daily frequency with length 1827 and prediction length of 30. Recall that our adapter-based approach CHRONOSX incorporates both past and future covariates. Further, we consider a full finetuning variant and denote it by CHRONOSX(FF), which updates all parameters including those of the pretrained model. Moreover, we consider baseline that include covariates like DeepAR, PatchTSTx (an extension of PatchTST that supports covariates; see the supplementary appendix for details), NHiTS, and NBEATSx with its interpretable and general variants, together with AutoArima, which incorporates covariates through residuals. For all models we standardize the covariates by the mean of absolute values. All scores are normalized by the scores of CHRONOS SMALL.

Figures 4a and 4b present the aggregated relative WQL on *Simple* and *Complex* synthetic datasets. First, we can see that for both *Simple* and *Complex* synthetic datasets that our proposed approaches CHRONOSX and CHRONOSX(FF) do incorporate covariates in the generated forecasts as their average relative scores are at least 22% and 58% smaller than the zero-shot performance of the



(a) Agg. Rel (WQL) on **Simple** synthetic datasets.



(b) Agg. Rel (WQL) on **Complex** synthetic datasets.

Figure 4: Evaluations on Simple and Complex datasets as introduced in Section 4. Scores are normalized by CHRONOS SMALL. We can see that our proposed models CHRONOSX and CHRONOSX(FF) effectively incorporate covariates.

pretrained model CHRONOS SMALL, respectively. Second, we can see that baseline models like TFT, DeepAR and PatchTSTx perform particularly well in *Simple* datasets, verifying that they effectively incorporate covariate in simple cases, whereas for *Complex* datasets the errors are larger, yet most of baselines outperform the zero-shot performance of CHRONOS SMALL.

Moreover, we can see that our full finetuning model CHRONOSX(FF) outperforms our adapter model CHRONOSX showcasing the capacity provided by updating a larger amount of parameters in a controlled scenario where each dataset has 100 time series with long enough history. Further, we can see that in both cases that our full finetuning model CHRONOSX(FF) belongs to the top-4 and top-5 best models with a small performance gap with the best models. Yet, we can see that Moirai, the first pretrained model designed to handle covariates, underperforms CHRONOS SMALL in both *Simple* and *Complex*, suggesting that providing pretrained models with covariates is still a challenging task.

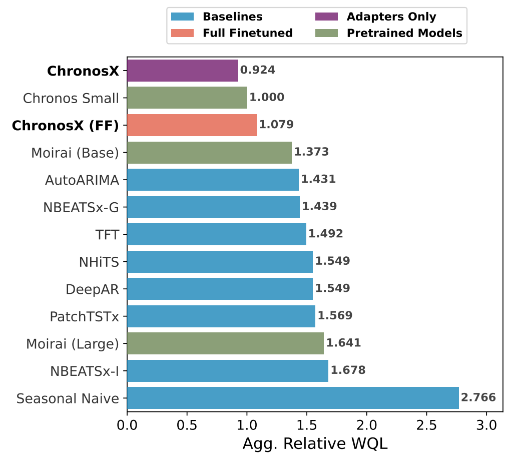
Hence, in this section we have observed that our proposed benchmark of synthetic datasets is useful to discern the effectivity of different models to incorporate covariates in time series. In particular, we have seen that most of the baselines here considered incorporate covariates that yield smaller scores than methods that do not take covariates, and that pretrained models that accept covariates still have room for improvement.

5.2 Finetuning and Adapters on Real Datasets with Covariates.

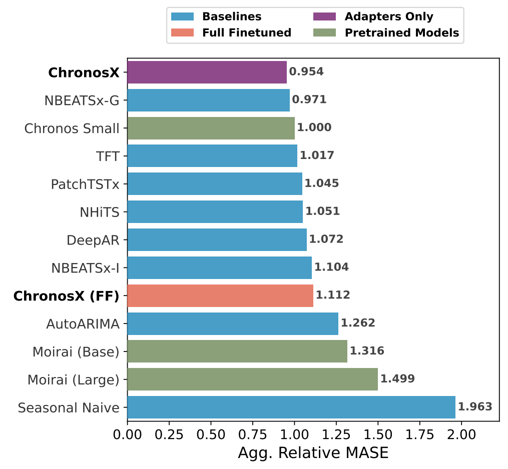
In this section we evaluate our proposed models on 19 real datasets with covariates taken from diverse sources [10, 17, 25, 36–38]. These datasets encompass data from fields such as retail, nature, transport, and mostly energy. Frequencies include hourly, daily, weekly, and 15-minute intervals, with forecast horizons up to 30 steps ahead. Please see the appendix for further details.

One of the main challenges in these collection of datasets is that 10 datasets are conformed by one time series. These cases belong to two major subsets of datasets: ProEnFo [36, 37] and electricity price forecasting (EPF) [21, 28]. To avoid over-representing these cases with one time series we take the weighted average of the scores of each of these subsets weighted by the size of each dataset, and take this as the score to compute the aggregated relative WQL and MASE.

In Figure 5 we can see that our adapter-based approach CHRONOSX performs best and that our full finetuning variant CHRONOSX(FF) underperforms with respect to several of the baselines here considered. This is likely due to the fact that 50% of the considered datasets consist of only one time series, and hence having a model with less trainable parameters provides an advantage in data-sparse regimes. Further, we can see that Moirai (Base) presents a more competitive performance, as it outperforms NHiTS, TFT, and DeepAR in terms of WQL. Moirai's capability of incorporating covariates in these datasets highlights that pretrained models are suitable for data-scarce regimes, although it underperforms in terms of MASE.

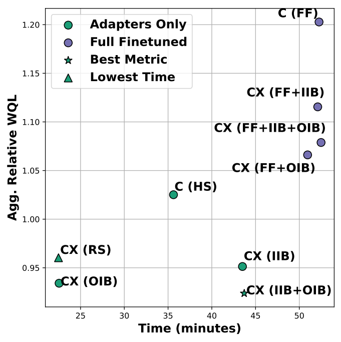


(a) Agg. Rel. WQL on real datasets with covariates.

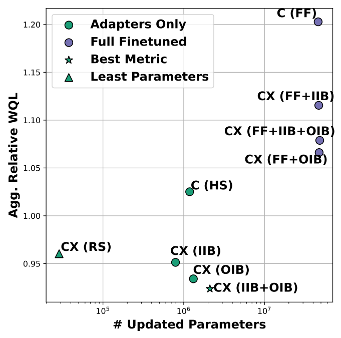


(b) Agg. Rel. MASE on real datasets with covariates.

Figure 5: Evaluations on real datasets with covariates. We report the Aggregated Relative WQL and MASE. We can see our proposed model CHRONOSX performs best and most models underperform the zero-shot model CHRONOS SMALL in WQL whereas for MASE several baselines present better performance.



(a) WQL vs Execution Time



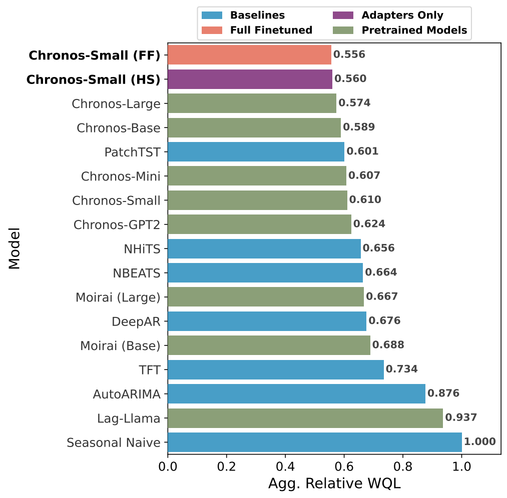
(b) WQL vs Number of Parameters

Figure 6: Average Time Execution and number of updated parameters of our proposed models against Aggregated Relative Weighted Quantile Loss on real datasets with covariates. We can see that full finetuning models have a larger amount of updated parameters and larger execution time than our approaches using only adapters. Overall the model with smallest execution time and number of updated parameters is the residual based approach CHRONOSX(RS).

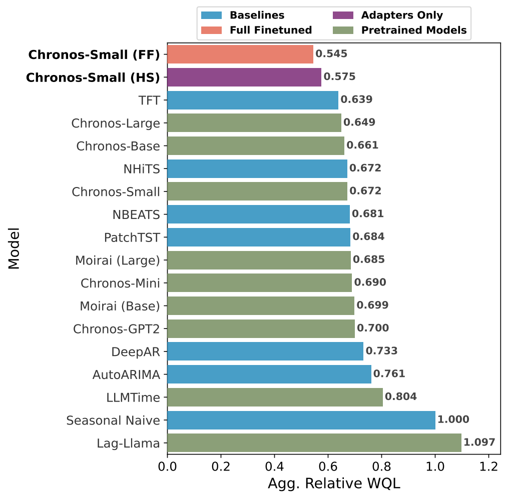
Time and Memory Efficiency of Adapters and Finetuning. We present an empirical analysis of the trade-offs between our models using only adapters and full finetuning against time execution and number of updated parameters. For a better visualization we represent CHRONOSX and CHRONOS with CX and C, respectively. For this, we measure the average computational time consumed per run for training and testing across datasets and compared it to the aggregated WQL metric. As can be seen in Figure 6a, the adapters are among the best performing and fastest variants. This is due to the decreased number of updated parameters as we report in Figure 6b. Overall the most efficient variants in terms of time and number of updated hyperparameters is CHRONOSX(RS), while the best performing and still efficient is CHRONOSX.

5.3 Finetuning and Adapters on Real Datasets Without Covariates.

To further understand the impact of finetuning and adapters, we present evaluations on datasets without covariates. We employ real-world datasets used to pretrain and evaluate Chronos from [3]. This includes Benchmark I and Benchmark II datasets, each composed by 15 and 27 datasets and used to pretrain and for zero-shot evaluations of CHRONOS, respectively. For comparison, we take the evaluations from [3] corresponding to 23 different baselines. Please refer to [3] and the appendix for more details on datasets and baselines. We analyze the performance of the following two approaches: 1) we do full finetuning (FF) on all parameters and denote this by CHRONOS-SMALL (FF); 2) an adapter-only approach where we freeze the parameters of Chronos and apply a feedforward network to its hidden state and only train the weights of the feed-forward network, denoted by CHRONOS-SMALL (HS), as introduced in Equation (6).



(a) Evaluations on **Benchmark I** datasets.



(b) Evaluations on **Benchmark II** datasets.

Figure 7: Aggregated Relative Weighted Quantile Loss of models evaluated on Benchmark I and Benchmark II datasets depicted in Figures. 7a and 7b, respectively. Overall we can see that our proposed models based on full finetuning, CHRONOS-SMALL(FF), and adapters, CHRONOS-SMALL(HS) consistently perform best.

In Figure 7a and Figure 7b we present the corresponding evaluations with Benchmark I and Benchmark II datasets, respectively. The forecasting metrics of each model are normalized using the metrics of Seasonal Naive. We can see that for both Benchmark I and Benchmark II that our proposed adapter-only model with hidden state CHRONOS-SMALL(HS), together with our full finetuning model CHRONOS-SMALL(FF) consistently perform best. In particular, we can see that the improvement gap of our models with respect to the remaining models is larger in Benchmark II than for Benchmark I, suggesting that there is more room for improvement when dealing with data that has not been presented previously to CHRONOS while pretraining. Additional results are presented in the appendix.

6 Conclusion

In this paper we presented CHRONOSX, a novel method to incorporate covariates into pretrained models through well defined modules that inject information from past and future covariates. We study the cases where we finetune the entire model or we only fit the parameters of the covariate related modules, and introduced a family of 32 synthetic datasets with covariates with diverse time dynamics to evaluate the efficacy of forecasting models to integrate covariates. Our experiments in synthetic and real datasets demonstrate that our approach CHRONOSX enables the pretrained model to capture the covariate information and performs favorably when compared to other baselines.

Limitations and Future Work. One of the advantages of pretrained forecasting models is that they simplify forecasting pipelines because zero-shot inference does not require any training loop. One limitation of this work is that the adaptor approach for integrating covariate data requires training of the adaptor module, therefore losing the appealing benefit of pretrained models to perform zero-shot inference (arguable in exchange for lower forecasting error). One interesting future research direction is to integrate covariates into pretrained forecasting models during inference akin to in-context learning. This would allow a pretrained forecasting model to still operate in an inference-only mode while leveraging covariate data to reduce the forecast error. Another limitation of our study is that the ChronosX instantiations that use output injection blocks assume that the covariate data is available in the future, which might not an assumption that holds for specific use cases. However, we also propose variants that only use past covariate data (input injection blocks) and we demonstrate that these still improve over the pretrained Chronos model without any covariate data. One future work direction could be to extend the adaptor approach to multivariate time series data, which would be another way to use covariate data that is only available in the past.

References

[1] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, et al. Gluonts: Probabilistic and neural time series modeling in python. *The Journal of Machine Learning Research*, 21(1):4629–4634, 2020.

[2] A. F. Ansari, K. Benidis, R. Kurle, A. C. Turkmen, H. Soh, A. J. Smola, B. Wang, and T. Januschowski. Deep explicit duration switching models for time series. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang. Chronos: Learning the language of time series, 2024.

[4] V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, 16(4):521–530, 2000.

[5] K. Bandara, C. Bergmeir, and H. Hewamalage. Lstm-msnet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[6] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[7] C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2023.

[8] J. D. Croston. Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303, Sep 1972. ISSN 1476-9360. doi: 10.1057/jors.1972.50. URL https://doi.org/10.1057/jors.1972.50.

[9] A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. *arXiv:2310.10688*, 2023.

[10] E. David, J. Bellot, and S. L. Corff. HERMES: Hybrid error-corrector model with inclusion of external signals for nonstationary fashion time series. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=4ofFo7D5GL.

[11] I. Dimoulkas, P. Mazidi, and L. Herre. Neural networks for GEFCom2017 probabilistic load forecasting. *International Journal of Forecasting*, 35(4):1409 – 1423, 2019.

[12] S. Dooley, G. S. Khurana, C. Mohapatra, S. Naidu, and C. White. ForecastPFN: Synthetically-trained zero-shot forecasting. In *Advances in Neural Information Processing Systems*, 2023.

[13] P. J. Fleming and J. J. Wallace. How not to lie with statistics: the correct way to summarize benchmark results. *Communications of the ACM*, 29(3):218–221, 1986.

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

[15] J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski. Probabilistic forecasting with spline quantile function rnns. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1901–1910. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/gasthaus19a.html.

[16] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[17] R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, and P. Montero-Manso. Monash time series forecasting archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[18] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

[19] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[20] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[21] J. Lago, G. Marcjasz, B. De Schutter, and R. Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2021.116983. URL https://www.sciencedirect.com/science/article/pii/S0306261921004529.

[22] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl. Time-series extreme event forecasting with neural networks at uber. In *International conference on machine learning*, volume 34, pages 1–5. sn, 2017.

[23] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

[24] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[25] S. Makridakis, E. Spiliotis, and V. Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022.

[26] S. Mukherjee, D. Shankar, A. Ghosh, N. Tathawadekar, P. Kompalli, S. Sarawagi, and K. Chaudhury. ARMDN: Associative and recurrent mixture density networks for eretail demand forecasting. *arXiv preprint arXiv:1803.03800*, 2018.

[27] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

[28] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, and A. Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx. *CoRR*, abs/2104.05522, 2021. URL https://arxiv.org/abs/2104.05522.

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

[30] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, and A. Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx. *International Journal of Forecasting*, 39(2):884–900, 2023.

[31] J. Pfeiffer, S. Ruder, I. Vulić, and E. Ponti. Modular deep learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=z9EkXfvxta. Survey Certification.

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

[33] O. Shchur, A. C. Turkmen, N. Erickson, H. Shen, A. Shirkov, T. Hu, and B. Wang. Autogluon–timeseries: Automl for probabilistic time series forecasting. In *International Conference on Automated Machine Learning*, pages 9–1. PMLR, 2023.

[34] S. Smyl and N. G. Hua. Machine learning methods for GEFCom2017 probabilistic load forecasting. *International Journal of Forecasting*, 35(4):1424–1431, 2019.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[36] Z. Wang, Q. Wen, C. Zhang, L. Sun, L. V. Krannichfeldt, and Y. Wang. Benchmarks and custom package for electrical load forecasting, 2023.

[37] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. *arXiv:2402.02592*, 2024.

[38] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pages 11106–11115. AAAI Press, 2021.

A Appendix Structure

In this appendix, we specify details on different aspects of our work. We address the following items:

- In Appendix B, we detail the hyperparameters used for the main method.
- In Appendix C, we mention the computing resource used in the experiments.
- In Appendix D, we describe the Chronos variants.
- In Appendix E, we give more information about the synthetic dataset creation.
- In Appendix F, we explain the datasets without covariates.
- In Appendix G, we detail information on the datasets with covariates.
- In Appendix H, we specify the baselines configurations.
- In Appendix I, we describe the metrics that we used in our experiments.
- In Appendix J, we present additional results: 1) Evaluations on adaption injection on pretrained models on synthetic benchmark datasets (Section J.0.1), 2) Fine-tuning and Adapters in real datasets with covariates (Section J.0.2, and 3) Probabilistic forecasts (Section J.0.3).

B Hyperparameter Details

For all the experiments, we use the output dimension of the linear layers and the hidden dimension of the Feed Forward Networks is 256. We train the models for 5000 steps, and checkpointing the best observed model every 100 steps. The rest of the training setup such as the optimizer and learning rate scheduler were set following previous work [3].

C Compute Resource Information

For the experiments, we use GPU Nvidia A10G. More expensive experiments used NVIDIA Tesla V100. For baselines we used cpu instances with 16 virtual cpus and 32 GiB of memory.

D Chronos Variants

Here we explain the baselines, for more information please refer to the section 3. In the notation below, we refer to the IIB (Input Injection Block) as the network that receives the past covariates and updates the token embedding, while the OIB (Output Injection Block) indicates the network that receives the hidden state and the future covariates.

- CHRONOS is the model with the pretrained weights (not finetuned).
- CHRONOS(FF) is the pretrained model full-finetuned on the corresponding dataset.
- CHRONOSX, represented as well as CHRONOSX(IIB+OIB), is the frozen pretrained model with the additional Input and Output Injection Blocks.
- CHRONOSX(FF), represented as well as CHRONOSX(FF+IIB+OIB), is the full-finetuned CHRONOS model with the additional Input and Output Injection Block.
- CHRONOSX(FF+IIB) is the full-finetuned CHRONOS model with Input Injection Block.
- CHRONOSX(FF+OIB) is the full-finetuned CHRONOS model with Output Injection Block.
- CHRONOSX(OIB) is the frozen pretrained model only with Output Injection Block.
- CHRONOSX(IIB) is the frozen pretrained model only with Input Injection Block.
- CHRONOSX(HS) is the frozen pretrained model only with hidden state, as in Equation 6.
- CHRONOSX(RS) is the frozen pretrained model only with modules accepting covariates, as in Equation 7.

E Synthetic Dataset Creation

We created a suite of 32 synthetic datasets comprising a 100 time series with length $L = 1827$ and 30 steps for the prediction length, assuming a "fictional" daily frequency. Each dataset is created by combining a **main signal** $\hat{z}_t$ with a **covariate** x_t using a **combination operator** according to Eq. 9.

$$z_t = \hat{z}_t \odot x_t, \quad t = 1, ..., L \tag{9}$$

Once a combination is chosen, we randomly sample the parameters for the main signal appendix E.1 and the covariates appendix E.2 to get 100 different time series. All possible combinations are illustrated in fig. 8 and fig. 9.

E.1 Main signal generation

The main signal to be affected by the covariates can be generated in different ways. We describe four variants in an increasing order of complexity by changing the set of possible parameters in Eq. 10.

$$\hat{z}_t = \sum_{i=1}^{3} a_i \sin\left(f_i t + \phi_i\right) + b_1 t + b_2 + \epsilon \tag{10}$$

- **Single Sinuoid**: This represents the simplest time series, keeping the same amplitude, phase and a weekly frequency among time series.

$$\hat{z}_t^{\text{(single)}} = \sin\left(2\pi \frac{t}{7}\right)$$

- **Simple Sinusoids**: Every time series in the dataset is the superposition of three sinusoids with different frequencies, simulating weekly, monthly and yearly seasonal patterns. For each time series, a different amplitude is sampled from $a_i \sim U(1, 5)$.

$$\hat{z}_t^{\text{(sinusoids)}} = a_1 \sin\left(2\pi \frac{t}{7}\right) + a_2 \sin\left(2\pi \frac{t}{30}\right) + a_3 \sin\left(2\pi \frac{t}{365}\right)$$

- **Diverse Sinusoids**: Every time series is similar to *simple sinusoids*, but we sample a different phase for every sinusiodal component with a random amplitude $a_i \sim U(1, 5)$ and a random phase $\phi_i \sim U(-\pi, \pi)$. A trend component is also added with random coefficient $b_1, b_2 \sim U(-1, 1)$.

$$\hat{z}_t^{\text{(diverse)}} = a_1 \sin\left(2\pi \frac{t}{7} + \phi_1\right) + a_2 \sin\left(2\pi \frac{t}{30} + \phi_2\right) + a_3 \sin\left(2\pi \frac{t}{365} + \phi_3\right) + b_1 \frac{t}{365} + b_2$$

- **Noisy Sinusoids**. The Noisy Sinusoid are *Diverse sinusoids* with noise added at every step. The variance is chosen to be proportional to the scale $s = \frac{1}{L} \sum_t^L |\hat{z}_t|$ of the main diverse sinusoid $\epsilon \sim \mathcal{N}\left(0, \frac{s}{4}\right)$.

$$\hat{z}_t^{\text{(noisy)}} = \hat{z}_t^{\text{(diverse)}} + \epsilon$$

E.2 Covariate

Similarly to the main signal, the parameters of the covariates are randomly chosen depending on the types and the scale of the signal.

- **Spikes** are used to represent short time events by considering large value for single "active" time steps as in Eq. 11. The $N = 500$ times steps $\mathbb{T} = \{t_1, \ldots, t_N\}$ are uniformly sampled without replacement. The strength of all the spikes is randomly sampled in an interval limited to 5 times the scale $\gamma \sim U(1, 5s)$.

$$x_t = \begin{cases} \gamma & \text{if } t \in \mathbb{T}_N \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

- **Steps** are used to study the influence of sudden but sustained events. Steps covariates share the same formula as the spikes in Eq. 12 but the generation of the "active" index is different. We consider $N = 125$ non-overlapping intervals $T_i = \{t_i, t_{i+1}, \ldots, t_i + \delta_i\}$ by uniformly sampling the starting steps $t_i \sim U(0, L)$ and the associated duration $\delta_i \sim U(1, 30)$ with $T_i \cap T_j = \emptyset$. The resulting signal is then composed of a union of the different intervals : $\mathbb{T} = \cup_i^N T_i$. All steps have the save amplitude γ, randomly sampled in the same manner as the spikes.

$$x_t = \begin{cases} \gamma & \text{if } t \in \mathbb{T} \\ 1 & \text{if otherwise} \end{cases} \tag{12}$$

- **Bells** are considered to represent trends with smoother changes. In this case, the signal is a mixture of gaussian bells as in Eq. 13. We consider mixtures of $N = 125$ bells with randomly sampled mean $\hat{\mu}_i \sim U(0, L)$ and randomly sampled standard deviation $\sigma_i \sim U(1, 15)$. The scale of the bells is randomly sampled in an interval limited to 5 times the scale $\gamma \sim U(1, 5s)$.

$$x_\text{t} = \gamma \sum_i^N \exp\left(\frac{-\|t - \hat{\mu}_i\|^2}{\sigma_i^2}\right) \tag{13}$$

- **ARP** is created following Eq. 14. Only second order auto-regressive processes are created with coefficients $a_1 \sim U(0, 1)$ and $a_2 = 1 - a_1$, and noise amplitude equal to one. The scale of the final covariates is randomly sampled in an interval limited to 5 times the scale $\gamma \sim U(1, 5s)$.

$$x_t = a_1 \cdot x_{t-1} + a_2 \cdot x_{t-2} + \epsilon \tag{14}$$

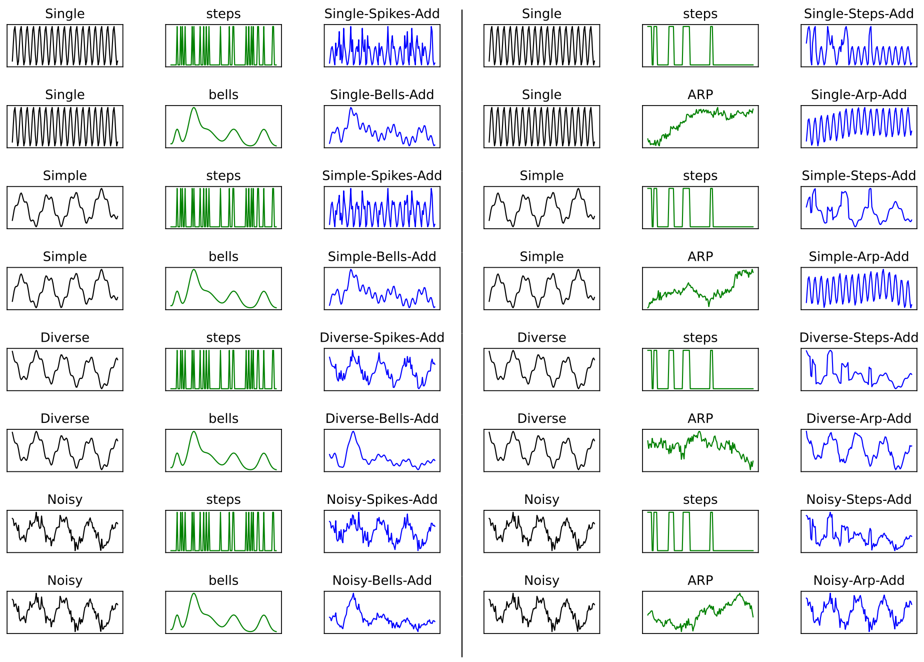


Figure 8: This figure provides a single time series example from all possible combinations of **covariates** and **main signal** with the add + **operator.**

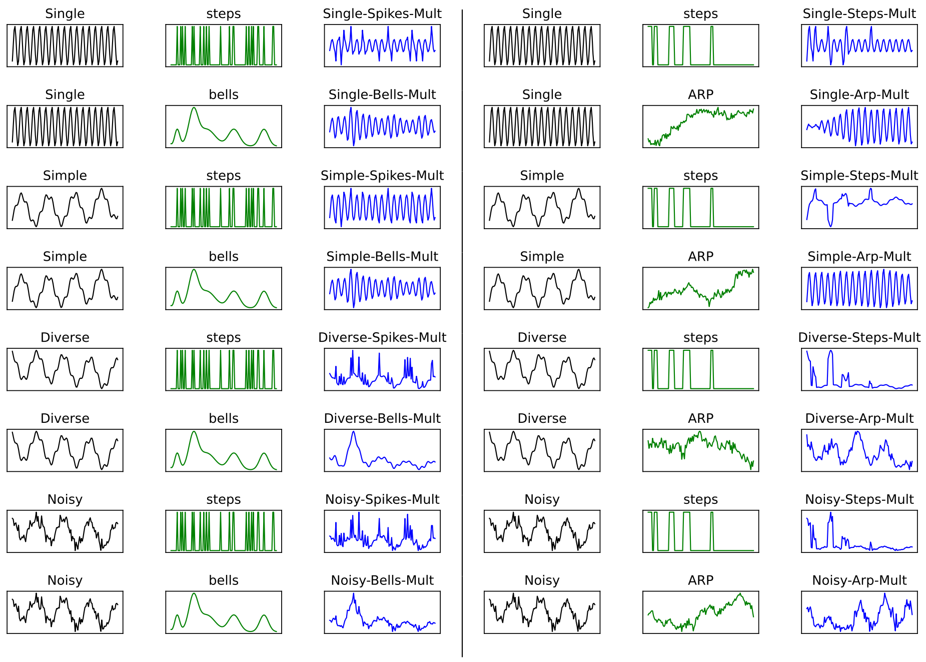


Figure 9: This figure provides a single time series example from all possible combinations of **covariates** and **main signal** with the mult $\times$ **operator.**

F Datasets Without Covariates

The following table is taken from [3] and is added here for completeness.

Table 1: All datasets that were used for experiments. The datasets are partitioned according to how they were used for training and evaluation of models: *Benchmark I* data was used for training models and other task-specific baselines, except for the H observations that were held out for in-domain testing only; *Benchmark II* data was *not* used in training models, but only for evaluation (final H observations), as well as for training task-specific baselines (excluding the final H observations).

Dataset	**Domain**	**Freq.**	**Num. Series**	**Series Length**			**Prediction**
				min	**avg**	**max**	**Length (H)**
Benchmark I							
Electricity (15 Min.)	energy	15min	370	16032	113341	140256	24
Electricity (Hourly)	energy	1H	321	26304	26304	26304	24
Electricity (Weekly)	energy	1W	321	156	156	156	8
KDD Cup 2018	nature	1H	270	9504	10897	10920	48
London Smart Meters	energy	30min	5560	288	29951	39648	48
M4 (Daily)	various	1D	4227	107	2371	9933	14
M4 (Hourly)	various	1H	414	748	901	1008	48
M4 (Monthly)	various	1M	48000	60	234	2812	18
M4 (Weekly)	various	1W	359	93	1035	2610	13
Pedestrian Counts	transport	1H	66	576	47459	96424	48
Rideshare	transport	1H	2340	541	541	541	24
Taxi (30 Min.)	transport	30min	2428	1469	1478	1488	48
Temperature-Rain	nature	1D	32072	725	725	725	30
Uber TLC (Daily)	transport	1D	262	181	181	181	7
Uber TLC (Hourly)	transport	1H	262	4344	4344	4344	24
Benchmark II							
Australian Electricity	energy	30min	5	230736	231052	232272	48
CIF 2016	banking	1M	72	28	98	120	12
Car Parts	retail	1M	2674	51	51	51	12
Covid Deaths	healthcare	1D	266	212	212	212	30
Dominick	retail	1D	100014	201	296	399	8
ERCOT Load	energy	1H	8	154854	154854	154854	24
ETT (15 Min.)	energy	15min	14	69680	69680	69680	24
ETT (Hourly)	energy	1H	14	17420	17420	17420	24
Exchange Rate	finance	1B	8	7588	7588	7588	30
FRED-MD	economic	1M	107	728	728	728	12
Hospital	healthcare	1M	767	84	84	84	12
M1 (Monthly)	various	1M	617	48	90	150	18
M1 (Quarterly)	various	3M	203	18	48	114	8
M1 (Yearly)	various	1Y	181	15	24	58	6
M3 (Monthly)	various	1M	1428	66	117	144	18
M3 (Quarterly)	various	3M	756	24	48	72	8
M3 (Yearly)	various	1Y	645	20	28	47	6
M4 (Quarterly)	various	3M	24000	24	100	874	8
M4 (Yearly)	various	1Y	23000	19	37	841	6
M5	retail	1D	30490	124	1562	1969	28
NN5 (Daily)	finance	1D	111	791	791	791	56
NN5 (Weekly)	finance	1W	111	113	113	113	8
Tourism (Monthly)	various	1M	366	91	298	333	24
Tourism (Quarterly)	various	1Q	427	30	99	130	8
Tourism (Yearly)	various	1Y	518	11	24	47	4
Traffic	transport	1H	862	17544	17544	17544	24
Weather	nature	1D	3010	1332	14296	65981	30

G Datasets With Covariates

In this section we present a summary of the datasets with covariates used for evaluations.

Table 2: Real Datasets Descriptors.

Dataset Name	Num. Series	Num. Covariates	Freq.	Prediction Length	Source
ETT (15 Min.)	2	5	15min	24	[38]
ETT (Hourly)	2	5	1H	24	[38]
Hermes	10000	1	1W	28	[10]
M5	30490	12	1D	28	[25]
Electricity-BE	1	2	1H	24	[21, 28]
Electricity-DE	1	2	1H	24	[21, 28]
Electricity-FR	1	2	1H	24	[21, 28]
Electricity-NP	1	2	1H	24	[21, 28]
Electricity-PJM	1	2	1H	24	[21, 28]
BDG-2 Hog	24	5	1H	24	[36, 37]
BDG-2 Bull	41	3	1H	24	[36, 37]
BDG-2 Cockatoo	1	5	1H	24	[36, 37]
Covid19Energy	1	6	1H	24	[36, 37]
GEF12	20	1	1H	24	[36, 37]
GEF14	1	1	1H	24	[36, 37]
GEF17	8	1	1H	24	[36, 37]
PDB	1	1	1H	24	[36, 37]
Spanish	1	20	1H	24	[36, 37]
Rideshare	156	8	1H	24	[17]

Table 3: Baseline models and hyperparameter choices. Hyperparameters not specified are set to defaults in their respective implementations. C stands for context length, d_h for hidden layer dimension, n_L for number of layers, n_H for number of heads, and η for learning rate.

Model	Model Type	Implementation	Probabilistic	Hyperparameters
SeasonalNaive	Local	StatsForecast	Yes	N/A
DeepAR	Task-specific	GluonTS	Yes	$d_h = 40, n_L = 2$
TFT	Task-specific	GluonTS	Yes	$d_h = 32, n_H = 4$
PatchTST	Task-specific	GluonTS	Yes	Patch length: 16, Stride: 8, $d_h = 32, n_L = 2, n_H = 4$
N-BEATSx-I	Task-specific	NeuralForecast	No	Input size multiplier: 5, scaler: robust_scaler, stack_types: Interpretable
N-BEATSx-G	Task-specific	NeuralForecast	No	Input size multiplier: 5, scaler: robust_scaler, stack_types: Generalized
N-HiTS	Task-specific	NeuralForecast	No	Input size multiplier: 5, scaler: robust_scaler
Moirai	Zero-shot	Reference	Yes	$C = 1024$, Patch length: auto, batch size:4

H Baselines

For task-specific deep learning architectures DeepAR [32], PatchTST [27], TFT [24] we based evaluations on implementations in GluonTS [1]. However, NBEATSx-I, , NBEATSx-G [30] and NHiTS [7] experiments were based on implementations in the NeuralForecast [29] library. The original PatchTST model does not naturally accept additional time features as input, so we extend it in the following way. Similar to target time series, covariates are converted to patches and appended to the target patches. Since PatchTST is encoder only model, we combine past and future covariates (if available) and (left) shift them by prediction length so that the covariates are aligned with the target input. We call this version PatchTSTx.

The baselines DeepAR, PatchTST, TFT, NBEATSx-I, NBEATSx-G, , and N-HiTS were trained and evaluated three times and their performance averaged in order to account for high variance inherent in their optimization.

Statistical baselines SeasonalNaive was used with their default hyperparameters in StatsForecast [14], but with season length implied by their frequencies. For example, daily frequency data had season length set to 7, hourly data 24, etc. For this heuristic, we used the helper function `get_seasonality` from GluonTS.

Default hyperparameter configurations provided in baseline implementations were kept as is, and no dataset specific or global hyperparameter tuning was performed. GluonTS-based implementations were optimized with a batch size of 128.

I Evaluation Metrics

For the evaluation metrics we follow the approach exposed in [2]. Let $\{\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,C+H}]\}_{i=1}^N$ be a dataset with N time series where C and H are the context length and prediction length, respectively.

The mean absolute scaled error (MASE, [20]) is defined as:

$$\mathrm{MASE}(\hat{\mathbf{x}}_i, \mathbf{x}_i) = \frac{C-S}{H} \frac{\sum_{t=C+1}^{C+H} |\hat{x}_{i,t} - x_{i,t}|}{\sum_{t=1}^{C-S} |x_{i,t} - x_{i,t+S}|},$$

where S is a seasonality parameter.

The Weighted Quantile Loss is defined as

$$\mathrm{WQL} = \frac{1}{K} \sum_{j=1}^{K} \mathrm{WQL}_{\alpha_j}.$$

where K is the number of quantiles, α is the set of quantile levels to evaluate, and

$$\mathrm{WQL}_{\alpha} = \frac{2 \sum_{i,t} \mathrm{QL}_{\alpha}(q_{i,t}^{(\alpha)}, x_{i,t})}{\sum_{i,t} |x_{i,t}|}.$$

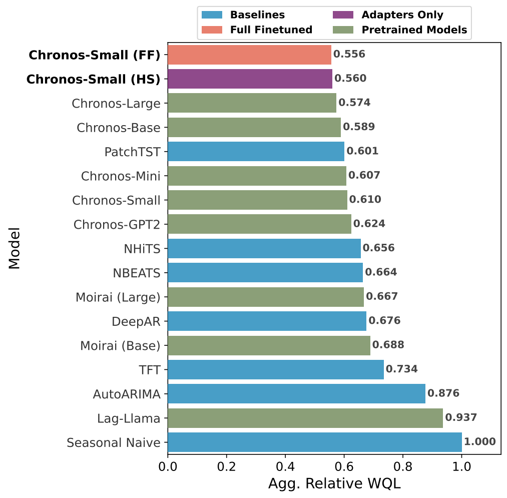
where $\mathbf{q}_i^{(\alpha)} = [q_{i,C+1}^{(\alpha)}, \ldots, q_{i,C+H}^{(\alpha)}]$ are the predicted quantiles at levels $\alpha \in (0,1)$, and

$$\mathrm{QL}_{\alpha}(q, x) = \begin{cases} \alpha(x - q), & \text{if } x > q, \\ (1-\alpha)(q - x), & \text{otherwise.} \end{cases} \tag{15}$$

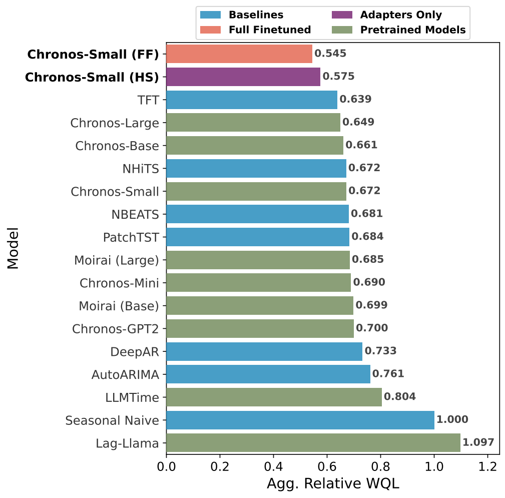
J Additional Results

We present the following results:

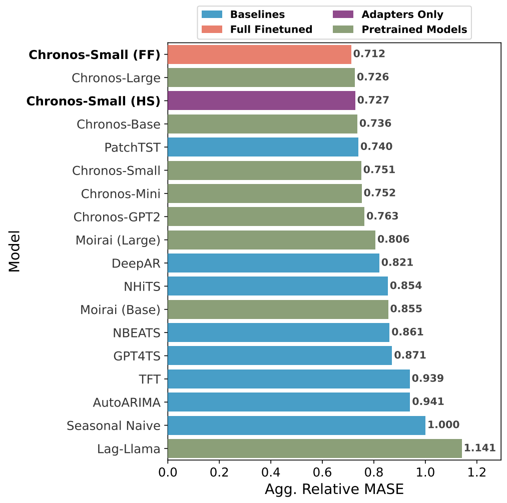
- In Fig. 10 we present the Aggregated Relative WQL and MASE,
- In Fig. 11 we present the Average Rank with WQL and MASE,
- In Table 4 and Table 5 we present WQL scores on Benchmark I and Benchmark II datasets,
- In Table 6 and Table 7 we present MASE scores on Benchmark I and Benchmark II datasets,



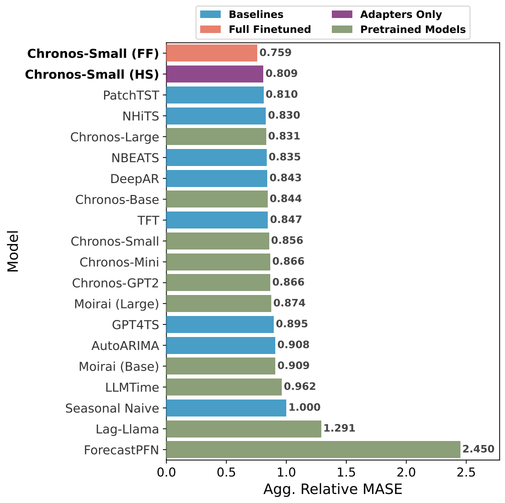
(a) Agg. Rel. WQL on **Benchmark I** datasets.



(b) Agg. Rel. WQL on **Benchmark II** datasets.

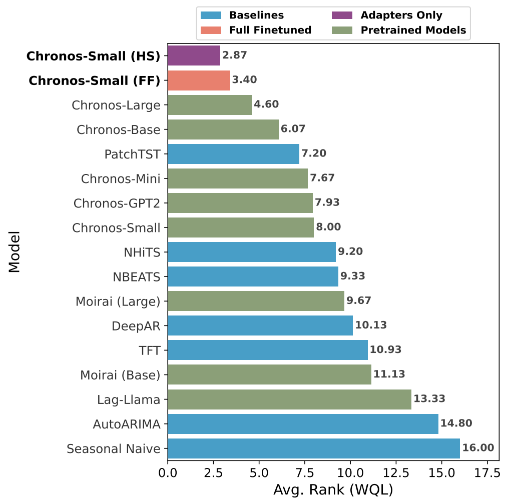


(c) Agg. Rel. MASE on **Benchmark I** datasets.

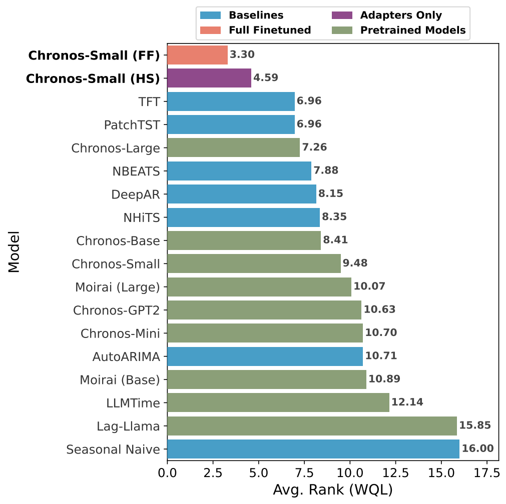


(d) Agg. Rel. MASE on **Benchmark II** datasets.

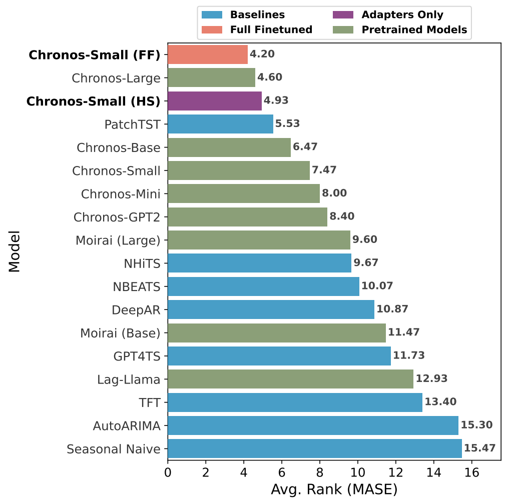
Figure 10: Aggregated Relative WQL and MASE of models evaluated on Benchmark I and Benchmark II datasets.



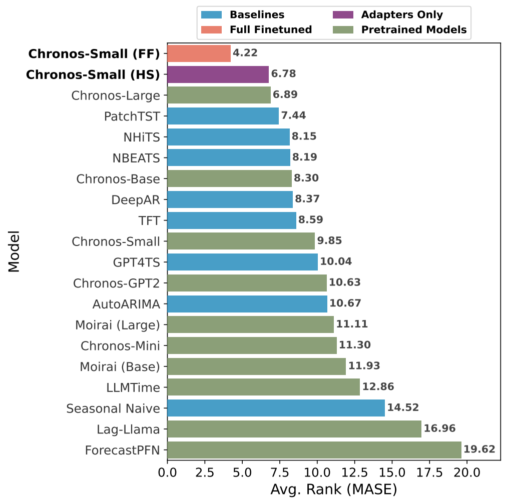
(a) Avg Rank (WQL) on **Benchmark I** datasets.



(b) Avg Rank (WQL) on **Benchmark II** datasets.



(c) Avg Rank (MASE) on **Benchmark I** datasets.



(d) Avg Rank (MASE) on **Benchmark II** datasets.

Figure 11: Average Rank with WQL and MASE of models evaluated on Benchmark I and Benchmark II datasets.

Table 4: Weighted Quantile Loss scores per model on datasets in Benchmark I with 15 datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported for our proposed models CHRONOS SMALL (FF) and CHRONOS SMALL (HS) together with considered baselines are averaged over three random seeds.

	Full Finetuned	Adapters Only	Pretrained Models								Baselines						
	Chronos-Small (FF)	Chronos-Small (HS)	Chronos-Large	Chronos-Base	Chronos-Small	Chronos-Mini	Chronos-GPT2	Lag-Llama	Moirai (Large)	Moirai (Base)	PatchTST	DeepAR	TFT	NHiTS	NBEATS	AutoARIMA	Seasonal Naive
Electricity (15 Min.)	**0.069**	**0.073**	0.076	0.076	0.081	0.081	0.077	0.319	0.103	0.105	0.082	0.090	0.189	0.081	0.084	-	0.117
Electricity (Hourly)	0.104	0.101	0.102	0.115	0.107	**0.092**	0.121	0.104	0.116	0.122	**0.089**	0.106	0.125	0.128	0.127	0.126	0.147
Electricity (Weekly)	**0.059**	**0.064**	0.064	0.067	0.077	0.078	0.069	0.147	0.159	0.110	0.069	0.116	0.106	0.098	0.097	0.138	0.198
KDD Cup 2018	**0.224**	0.260	0.273	0.272	0.294	0.271	0.359	0.369	0.277	0.286	**0.252**	0.330	0.571	0.302	0.315	0.528	0.556
London Smart Meters	0.410	0.419	0.423	0.426	0.430	0.433	0.431	0.384	**0.350**	0.358	**0.346**	0.405	0.365	0.358	0.357	-	0.541
M4 (Daily)	**0.020**	**0.020**	0.021	0.021	0.021	0.021	0.020	0.043	0.023	0.023	0.023	0.023	0.023	0.022	0.022	0.023	0.028
M4 (Hourly)	0.029	**0.025**	0.025	0.028	0.026	0.025	0.038	0.111	**0.022**	0.025	0.027	0.038	0.033	0.040	0.045	-	0.048
M4 (Monthly)	**0.091**	0.094	0.102	0.103	0.104	0.104	0.110	0.153	0.100	0.102	0.095	0.101	0.097	0.094	**0.093**	-	0.146
M4 (Weekly)	**0.037**	**0.037**	0.038	0.039	0.041	0.042	0.040	0.078	0.046	0.049	0.039	0.046	0.051	0.039	0.040	0.050	0.063
Pedestrian Counts	0.218	0.216	**0.198**	0.203	0.234	0.241	**0.175**	0.262	0.259	0.274	0.257	0.229	0.261	0.254	0.241	0.340	0.319
Rideshare	**0.132**	0.136	0.141	0.140	0.141	0.135	0.141	0.158	0.159	0.164	0.135	**0.130**	0.134	0.152	0.172	0.157	0.186
Taxi (30 Min.)	**0.255**	**0.261**	0.267	0.273	0.312	0.311	0.335	0.357	0.368	0.513	0.363	0.395	0.382	0.306	0.305	-	0.471
Temperature-Rain	0.676	**0.644**	0.663	0.670	0.686	0.704	0.669	0.717	0.685	**0.655**	0.804	0.718	0.670	0.780	0.798	0.869	1.424
Uber TLC (Daily)	0.097	**0.094**	**0.096**	0.097	0.099	0.106	0.099	0.176	0.108	0.116	0.100	0.110	0.111	0.116	0.108	0.151	0.231
Uber TLC (Hourly)	**0.141**	**0.145**	0.155	0.154	0.156	0.161	0.161	0.176	0.167	0.176	0.167	0.176	0.179	0.166	0.161	0.311	0.299
Agg. Relative Score	**0.556**	**0.560**	0.574	0.589	0.610	0.607	0.624	0.937	0.667	0.688	0.601	0.676	0.734	0.656	0.664	0.876	1.000
Avg. Rank	**3.400**	**2.867**	4.600	6.067	8.000	7.667	7.933	13.333	9.667	11.133	7.200	10.133	10.933	9.200	9.333	14.800	16.000

Table 5: Weighted Quantile Loss scores per model on datasets in Benchmark II with 27 datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported for our proposed models CHRONOS SMALL (FF) and CHRONOS SMALL (HS) together with considered baselines are averaged over three random seeds.

	Full Finetuned	Adapters Only	Pretrained Models									Baselines						
	Chronos-Small (FF)	Chronos-Small (HS)	Chronos-Large	Chronos-Base	Chronos-Small	Chronos-Mini	Chronos-GPT2	LLMTime	Lag-Llama	Moirai (Large)	Moirai (Base)	PatchTST	DeepAR	TFT	NHiTS	NBEATS	AutoARIMA	Seasonal Naive
Australian Electricity	**0.027**	0.071	0.068	0.079	0.077	0.069	0.082	0.069	0.097	0.046	0.054	0.037	0.087	0.036	**0.034**	0.038	0.073	0.084
CIF 2016	**0.011**	0.016	0.013	0.013	0.014	0.015	0.014	0.014	0.041	0.013	0.011	0.140	0.136	**0.011**	0.032	0.039	0.017	0.015
Car Parts	0.878	0.946	1.067	1.058	1.033	1.029	1.031	-	1.011	1.626	1.652	0.998	0.967	**0.871**	0.880	**0.877**	-	1.600
Covid Deaths	**0.031**	0.032	0.045	0.046	0.060	0.078	0.089	0.032	0.276	0.034	0.038	0.065	0.108	0.034	0.038	0.056	**0.029**	0.133
Dominick	**0.307**	0.322	0.337	0.338	0.344	0.351	0.341	-	0.443	0.345	0.361	0.345	0.364	0.320	0.313	**0.312**	-	0.453
ERCOT Load	**0.016**	0.017	0.016	**0.015**	0.017	0.018	0.016	0.053	0.033	0.021	0.019	0.017	0.032	0.023	0.020	0.020	0.052	0.037
ETT (15 Min.)	0.064	0.059	0.064	0.065	0.060	0.065	0.068	0.088	0.080	0.070	0.074	0.054	0.069	0.075	**0.051**	**0.053**	0.073	0.141
ETT (Hourly)	**0.069**	0.073	0.074	0.075	0.077	0.083	0.076	0.122	0.106	0.083	0.094	**0.071**	0.081	0.082	0.081	0.074	0.105	0.122
Exchange Rate	0.013	0.015	0.017	0.019	0.020	0.017	0.017	0.015	0.011	0.012	0.010	**0.010**	**0.009**	0.011	0.010	0.011	0.011	0.013
FRED-MD	0.023	**0.017**	0.026	0.024	**0.017**	0.019	0.029	0.041	0.389	0.046	0.050	0.042	0.043	0.112	0.057	0.061	0.056	0.122
Hospital	0.059	0.056	0.057	0.057	0.058	0.059	0.059	0.066	0.093	0.057	0.059	0.070	0.056	0.053	**0.052**	**0.050**	0.058	0.073
M1 (Monthly)	0.152	0.130	**0.129**	**0.126**	0.138	0.141	0.135	0.181	0.196	0.149	0.155	0.165	0.150	0.175	0.189	0.187	0.146	0.191
M1 (Quarterly)	**0.063**	**0.059**	0.105	0.099	0.103	0.101	0.115	0.115	0.141	0.106	0.106	0.078	0.089	0.122	0.111	0.085	0.091	0.150
M1 (Yearly)	**0.104**	**0.099**	0.176	0.183	0.169	0.179	0.208	0.144	0.293	0.207	0.195	0.165	0.139	0.124	0.198	0.182	0.160	0.209
M3 (Monthly)	**0.088**	**0.092**	0.096	0.097	0.099	0.099	0.104	0.108	0.155	0.101	0.102	0.113	0.099	0.096	0.097	0.101	0.102	0.149
M3 (Quarterly)	**0.070**	0.072	0.073	0.075	0.078	0.080	0.078	0.084	0.134	0.084	0.080	0.074	0.073	**0.071**	0.076	0.080	0.079	0.101
M3 (Yearly)	0.134	**0.125**	0.150	0.151	0.153	0.158	0.149	0.148	0.192	0.170	0.165	0.133	**0.122**	0.130	0.182	0.181	0.162	0.167
M4 (Quarterly)	**0.073**	0.078	0.082	0.083	0.083	0.085	0.086	-	0.132	0.080	0.081	0.074	0.080	0.080	0.073	**0.073**	0.082	0.119
M4 (Yearly)	**0.109**	0.113	0.133	0.135	0.135	0.139	0.147	-	0.178	0.139	0.121	**0.106**	0.111	0.110	-	-	0.130	0.161
M5	**0.537**	**0.549**	0.588	0.587	0.591	0.596	0.600	-	0.635	0.584	0.692	0.597	0.657	0.560	0.563	0.560	0.624	1.024
NN5 (Daily)	0.154	0.161	0.155	0.160	0.170	0.172	0.165	0.242	0.261	0.162	0.181	0.149	0.155	**0.145**	0.149	**0.147**	0.312	0.425
NN5 (Weekly)	**0.082**	0.083	0.089	0.090	0.090	0.089	0.094	0.092	0.111	0.092	0.092	**0.081**	0.087	0.086	0.098	0.114	0.090	0.123
Tourism (Monthly)	**0.073**	0.092	0.099	0.099	0.112	0.108	0.096	0.125	0.213	0.114	0.125	0.092	0.092	0.096	0.092	**0.084**	0.093	0.104
Tourism (Quarterly)	**0.058**	0.068	**0.062**	0.068	0.070	0.077	0.068	0.071	0.202	0.085	0.099	0.074	0.072	0.074	0.077	0.063	0.098	0.119
Tourism (Yearly)	0.115	**0.096**	0.185	0.199	0.197	0.217	0.187	0.163	0.238	0.164	0.167	0.136	0.127	**0.102**	0.139	0.154	0.156	0.209
Traffic	0.253	0.250	0.254	0.261	0.261	0.262	0.252	0.287	0.256	**0.232**	**0.225**	0.246	0.233	0.264	0.263	0.270	-	0.362
Weather	**0.133**	0.133	0.139	0.140	0.143	0.149	0.145	-	0.164	**0.132**	0.135	0.143	0.147	0.151	0.143	0.144	0.185	0.217
Agg. Relative Score	**0.545**	**0.575**	0.649	0.661	0.672	0.690	0.700	0.804	1.097	0.685	0.699	0.684	0.733	0.639	0.672	0.681	0.761	1.000
Avg. Rank	**3.296**	**4.593**	7.259	8.407	9.481	10.704	10.630	12.143	15.852	10.074	10.889	6.963	8.148	6.963	8.346	7.885	10.708	16.000

Table 6: MASE scores per model on datasets in Benchmark I with 15 datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported for our proposed models CHRONOS SMALL (FF) and CHRONOS SMALL (HS) together with considered baselines are averaged over three random seeds.

	Full Finetuned	Adapters Only	Pretrained Models								Baselines							
	Chronos-Small (FF)	Chronos-Small (HS)	Chronos-Large	Chronos-Base	Chronos-Small	Chronos-Mini	Chronos-GPT2	Lag-Llama	Moirai (Large)	Moirai (Base)	PatchTST	DeepAR	TFT	NHiTS	NBEATS	GPT4TS	AutoARIMA	Seasonal Naive
Electricity (15 Min.)	**0.393**	0.414	**0.403**	0.409	0.430	0.453	0.406	1.169	0.626	0.706	0.450	0.515	1.108	0.579	0.567	0.508	-	0.498
Electricity (Hourly)	1.503	1.485	1.457	1.602	1.500	**1.370**	1.653	1.573	1.665	1.713	**1.349**	1.528	1.789	1.880	1.848	1.487	1.715	1.840
Electricity (Weekly)	**1.564**	1.792	1.829	1.888	2.007	2.044	1.879	2.979	2.759	2.848	**1.631**	2.517	2.800	1.975	2.035	1.880	3.009	3.037
KDD Cup 2018	**0.575**	0.684	0.663	0.673	0.688	0.656	0.771	0.844	0.656	0.661	**0.616**	0.779	1.022	0.674	0.731	0.737	1.023	0.994
London Smart Meters	0.850	0.865	0.843	0.852	0.859	0.868	0.856	0.792	**0.755**	0.770	**0.733**	0.832	0.788	0.777	0.781	0.794	-	0.966
M4 (Daily)	**3.044**	**3.046**	3.099	3.109	3.112	3.108	3.058	8.038	3.376	3.443	3.450	3.305	3.292	3.143	3.155	5.109	3.257	3.278
M4 (Hourly)	0.893	**0.868**	0.927	0.894	0.881	**0.879**	0.977	3.807	0.951	1.209	0.967	1.215	1.833	3.231	3.457	1.511	-	1.193
M4 (Monthly)	**0.928**	0.968	0.976	0.985	1.000	1.006	1.051	2.090	1.005	1.033	0.962	1.040	1.009	0.994	**0.942**	0.979	-	1.260
M4 (Weekly)	2.283	2.221	2.201	2.221	2.270	2.309	2.377	5.658	2.391	2.466	**1.996**	2.346	2.745	2.094	**1.976**	3.040	2.373	2.777
Pedestrian Counts	0.302	0.291	0.289	**0.286**	0.304	0.309	**0.277**	0.342	0.330	0.355	0.339	0.311	0.364	0.324	0.315	0.393	0.383	0.369
Rideshare	0.858	0.871	0.892	0.892	0.880	**0.857**	0.897	0.891	0.899	0.911	**0.827**	0.996	1.067	0.933	0.919	1.088	1.028	1.250
Taxi (30 Min.)	**0.840**	0.857	**0.828**	0.849	0.938	0.939	1.032	1.069	1.089	1.374	1.077	1.158	1.113	0.950	0.934	1.113	-	1.160
Temperature-Rain	1.034	0.983	**0.982**	0.991	1.013	1.033	0.984	1.031	0.988	**0.963**	1.250	1.015	0.994	1.232	1.343	1.226	1.524	2.243
Uber TLC (Daily)	0.831	0.843	**0.819**	0.836	0.871	0.906	0.846	1.289	0.878	0.935	**0.813**	0.905	0.916	0.877	0.879	0.838	1.114	1.378
Uber TLC (Hourly)	0.707	0.711	0.727	0.729	0.727	0.743	0.740	0.711	0.717	0.727	**0.696**	**0.703**	0.746	0.716	0.751	0.754	0.982	0.931
Agg. Relative Score	**0.712**	0.727	**0.726**	0.736	0.751	0.752	0.763	1.141	0.806	0.855	0.740	0.821	0.939	0.854	0.861	0.871	0.941	1.000
Avg. Rank	**4.200**	4.933	**4.600**	6.467	7.467	8.000	8.400	12.933	9.600	11.467	5.533	10.867	13.400	9.667	10.067	11.733	15.300	15.467

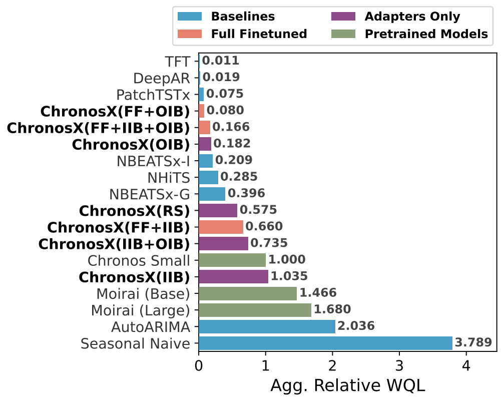
Table 7: MASE scores per model on datasets in Benchmark II with 27 datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported for our proposed models Chronos Small (FF) and Chronos Small (HS) together with considered baselines are averaged over three random seeds.

	Full Finetuned	Adapters Only	Pretrained Models										Baselines							
	Chronos-Small (FF)	Chronos-Small (HS)	Chronos-Large	Chronos-Base	Chronos-Small	Chronos-Mini	Chronos-GPT2	LLMTime	ForecastPFN	Lag-Llama	Moirai (Large)	Moirai (Base)	PatchTST	DeepAR	TFT	NHiTS	NBEATS	GPT4TS	AutoARIMA	Seasonal Naive
Australian Electricity	**0.584**	1.412	1.306	1.333	1.403	1.212	1.370	1.186	2.158	1.635	0.981	1.241	0.871	1.473	0.810	**0.794**	0.828	1.161	1.393	1.253
CIF 2016	**0.929**	0.983	0.985	0.995	1.016	1.052	1.065	1.384	3.588	2.235	1.115	1.218	1.537	1.363	1.553	1.389	1.440	**0.960**	1.006	1.289
Car Parts	0.805	0.841	0.911	0.897	0.891	0.893	0.879	-	2.657	0.816	1.538	1.734	0.803	**0.798**	**0.799**	0.803	0.803	0.891	-	1.201
Covid Deaths	36.328	40.448	42.762	42.641	42.689	43.525	48.028	32.143	91.515	78.456	33.019	33.010	36.465	38.203	**30.635**	31.771	31.730	75.909	**31.705**	46.912
Dominick	0.800	0.844	0.837	0.838	0.838	0.853	0.841	-	3.274	1.250	0.845	0.880	0.867	0.851	0.800	**0.782**	**0.782**	1.813	-	0.871
ERCOT Load	**0.538**	0.572	0.556	**0.541**	0.577	0.579	0.568	1.319	3.975	0.834	0.653	0.596	0.553	1.197	0.690	0.615	0.648	0.558	1.284	0.761
ETT (15 Min.)	0.737	0.702	0.712	0.710	0.669	0.732	0.753	1.042	1.138	0.967	0.777	0.956	0.652	0.874	0.962	**0.643**	0.659	**0.574**	0.879	1.169
ETT (Hourly)	0.740	0.782	**0.738**	0.749	0.769	0.794	0.750	1.232	1.833	1.002	0.833	0.887	**0.729**	0.814	0.875	0.811	0.782	0.768	0.977	0.932
Exchange Rate	2.563	2.798	3.231	3.460	3.357	3.223	3.206	1.743	7.583	3.087	1.937	**1.565**	**1.540**	1.615	2.361	2.041	2.149	2.709	1.882	1.740
FRED-MD	0.669	0.626	0.592	0.584	0.576	0.537	0.569	**0.513**	2.621	2.283	0.603	0.611	0.745	0.621	0.929	0.696	0.635	0.693	**0.473**	1.101
Hospital	0.823	0.809	0.815	0.820	0.819	0.821	0.833	0.861	1.775	0.939	0.823	0.821	0.859	0.804	0.799	**0.781**	**0.760**	0.793	0.820	0.921
M1 (Monthly)	**1.066**	1.202	**1.086**	1.119	1.163	1.172	1.164	1.415	2.172	1.875	1.243	1.271	1.208	1.122	1.326	1.333	1.236	1.198	1.153	1.314
M1 (Quarterly)	**1.706**	1.725	**1.699**	1.735	1.776	1.799	1.767	1.802	9.931	3.036	1.818	1.858	1.920	1.741	2.144	2.061	2.043	1.958	1.770	2.078
M1 (Yearly)	**3.215**	**3.397**	4.296	4.582	4.616	4.898	4.674	4.077	23.089	7.149	4.707	4.635	4.042	3.685	4.316	5.568	6.212	3.675	3.870	4.894
M3 (Monthly)	**0.820**	**0.852**	0.853	0.861	0.883	0.898	0.925	0.996	2.240	1.846	0.924	0.948	1.225	0.943	0.916	0.899	0.883	0.950	0.933	1.146
M3 (Quarterly)	1.210	1.241	1.170	1.185	1.250	1.270	1.230	1.450	10.176	2.886	1.449	1.439	1.264	1.209	**1.160**	1.202	**1.147**	1.448	1.419	1.425
M3 (Yearly)	**2.798**	**2.728**	3.094	3.186	3.238	3.348	3.112	3.140	18.728	5.114	3.824	3.647	2.949	2.827	2.860	3.432	3.547	3.418	3.165	3.172
M4 (Quarterly)	**1.112**	1.181	1.203	1.213	1.228	1.252	1.285	-	6.927	2.663	1.259	1.284	1.150	1.254	1.248	1.157	**1.129**	1.215	1.276	1.602
M4 (Yearly)	3.126	3.209	3.569	3.641	3.613	3.705	3.894	-	-	5.866	4.173	3.603	**3.072**	3.178	**3.119**	-	-	3.374	3.730	3.974
M5	**0.903**	0.915	0.946	0.942	0.942	0.946	0.972	-	1.530	0.965	0.930	1.439	0.919	0.956	**0.909**	0.917	0.917	0.935	1.057	1.399
NN5 (Daily)	0.577	0.603	**0.570**	0.584	0.620	0.638	0.612	0.953	1.375	0.992	0.627	0.702	0.575	0.585	**0.556**	0.571	0.571	0.720	1.214	1.292
NN5 (Weekly)	0.899	0.898	0.917	0.925	0.930	0.927	0.962	0.968	1.349	1.141	0.979	1.003	**0.877**	0.920	**0.896**	0.919	1.014	1.268	0.995	1.063
Tourism (Monthly)	**1.455**	1.636	1.741	1.817	1.891	1.937	1.772	2.139	4.348	3.030	1.913	2.042	1.572	1.529	1.686	1.514	**1.486**	1.573	1.573	1.631
Tourism (Quarterly)	**1.479**	1.685	1.660	1.705	1.727	1.822	1.814	1.916	5.595	3.695	2.331	2.702	1.723	1.586	1.729	**1.585**	1.618	1.750	1.661	1.699
Tourism (Yearly)	**3.032**	**2.729**	3.729	3.858	3.879	4.049	3.839	3.309	12.093	3.755	3.267	3.074	3.138	3.702	3.047	3.448	3.564	-	4.043	3.552
Traffic	0.818	0.829	0.798	0.823	0.833	0.847	0.810	0.973	1.909	0.829	0.760	**0.725**	0.790	**0.737**	0.880	0.927	0.968	0.787	-	1.077
Weather	**0.818**	0.819	0.827	0.829	0.839	0.855	0.871	-	2.003	1.001	**0.808**	0.831	0.860	0.911	0.913	0.910	0.888	0.972	0.907	1.004
Agg. Relative Score	**0.759**	**0.809**	0.831	0.844	0.856	0.866	0.866	0.962	2.450	1.291	0.874	0.909	0.810	0.843	0.847	0.830	0.835	0.895	0.908	1.000
Avg. Rank	**4.222**	**6.778**	6.889	8.296	9.852	11.296	10.630	12.857	19.615	16.963	11.111	11.926	7.444	8.370	8.593	8.154	8.192	10.038	10.667	14.519

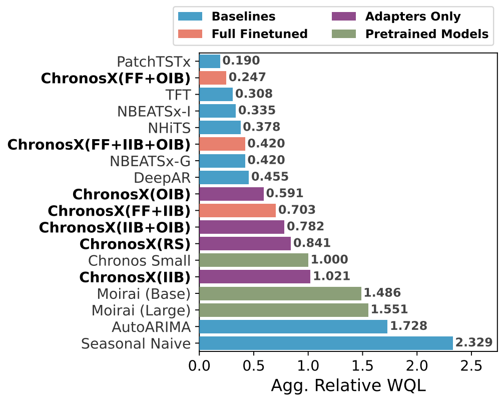
J.0.1 Evaluations on adaption injection on pretrained models on synthetic benchmark datasets - Additional results

We present the following results:

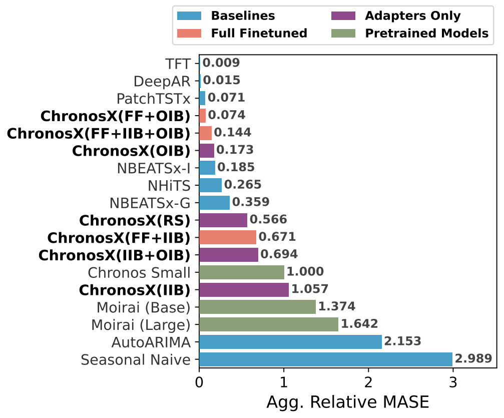
- In Fig. 12 we present the Aggregated Relative WQL and MASE on simple and complex datasets,
- In Fig. 13 we present the Average Rank with WQL and MASE on simple and complex datasets,
- In Table 8 and Table 9 we present the WQL on simple and complex datasets, respectively,
- In Table 10 and Table 11 we present the MASE on simple datasets, respectively.



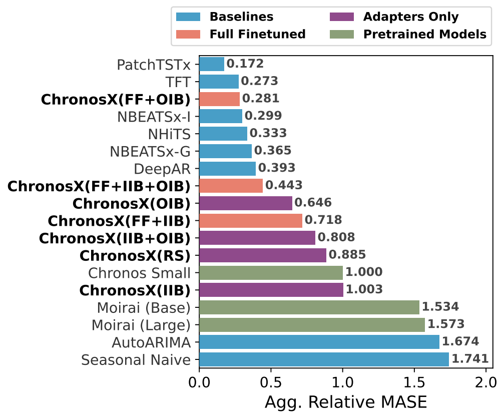
(a) Agg. Rel. WQL on **Simple** synthetic datasets.



(b) Agg. Rel. WQL on **Complex** synthetic datasets.

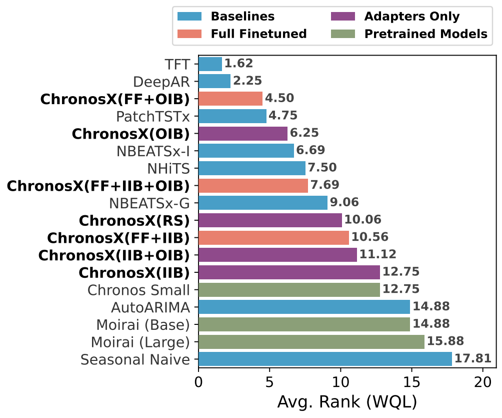


(c) Agg. Rel. MASE on **Simple** synthetic datasets

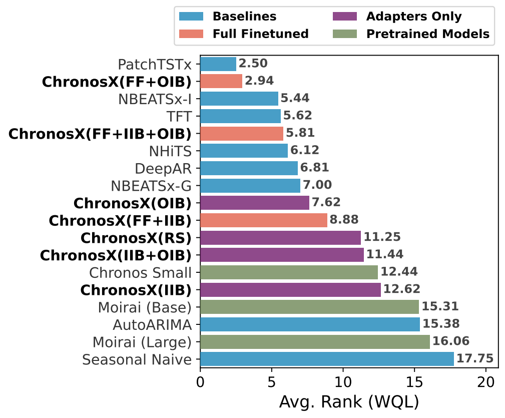


(d) Agg. Rel. MASE on **Complex** synthetic datasets

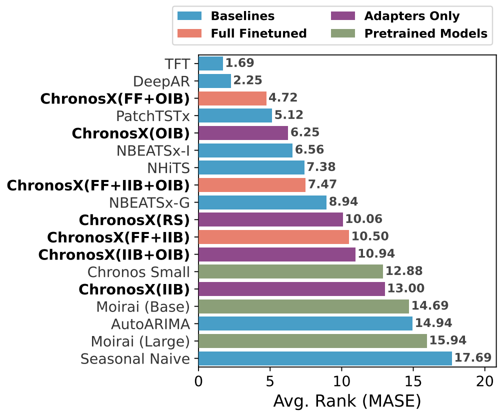
Figure 12: Aggregated Relative WQL and MASE on simple and complex datasets.



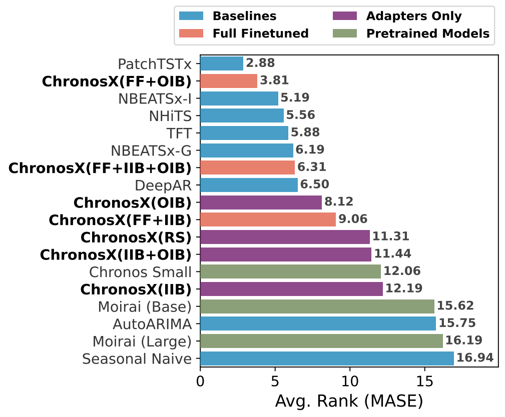
(a) Avg Rank (WQL) on **Simple** synthetic datasets.



(b) Avg Rank (WQL) on **Complex** synthetic datasets.



(c) Avg Rank (MASE) on **Simple** synthetic datasets.



(d) Avg Rank (MASE) on **Complex** synthetic datasets.

Figure 13: Average rank on simple and complex datasets.

Table 8: WQL scores per model on datasets in 16 **simple** synthetic datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported are averaged over three random seeds.

	Full Finetuned			Adapters Only				Pretrained Models			Baselines							
	ChronosX(FF+IIB+OIB)	ChronosX(FF+OIB)	ChronosX(FF+IIB)	ChronosX(IIB+OIB)	ChronosX(OIB)	ChronosX(IIB)	ChronosX(RS)	Chronos Small	Moirai (Large)	Moirai (Base)	NHiTS	AutoARIMA	TFT	NBEATSx-I	NBEATSx-G	DeepAR	PatchTSTx	Seasonal Naive
Simple ARP (Add)	0.164	0.018	0.045	0.168	0.043	0.060	0.055	0.056	0.148	0.099	0.026	0.198	0.010	0.027	0.031	**0.004**	**0.002**	0.402
Simple ARP (Mult.)	0.042	0.007	0.028	0.136	0.024	0.041	0.037	0.037	0.153	0.114	0.058	0.313	**0.002**	0.049	0.050	**0.003**	0.007	0.572
Simple Bells (Add)	0.287	0.271	0.288	0.225	0.185	0.252	0.318	0.319	0.383	0.346	0.066	0.176	**0.008**	0.063	0.066	0.061	**0.003**	0.582
Simple Bells (Mult.)	0.041	**0.008**	0.040	0.176	0.022	0.083	0.089	0.089	0.156	0.136	0.034	0.273	**0.002**	0.034	0.047	0.010	0.009	0.417
Simple Spikes (Add)	0.007	0.007	0.338	0.250	0.065	0.458	0.157	0.431	0.515	0.498	0.073	0.381	**0.003**	0.030	0.342	0.004	**0.003**	0.978
Simple Spikes (Mult.)	0.007	0.007	0.117	0.022	0.008	0.143	0.026	0.142	0.212	0.196	0.113	0.446	**0.001**	0.117	0.114	**0.001**	0.093	0.440
Simple Steps (Add)	0.007	0.007	0.273	0.185	0.039	0.412	0.165	0.415	0.455	0.458	0.092	0.293	0.014	0.035	0.104	**0.005**	**0.007**	0.835
Simple Steps (Mult.)	0.007	0.007	0.078	0.030	0.007	0.116	0.064	0.169	0.275	0.269	0.042	0.486	**0.001**	0.036	0.056	**0.001**	0.130	0.577
Single ARP (Add.)	0.130	0.011	0.048	0.175	0.041	0.066	0.061	0.062	0.152	0.106	0.028	0.222	**0.002**	0.027	0.030	0.006	**0.004**	0.416
Single ARP (Mult.)	0.038	0.007	0.025	0.157	0.023	0.231	0.037	0.038	0.131	0.109	0.010	0.410	**0.001**	0.009	0.013	**0.001**	0.003	0.538
Single Bells (Add.)	0.252	0.159	0.283	0.261	0.188	0.268	0.368	0.367	0.410	0.383	0.008	0.181	**0.003**	0.011	0.013	0.005	**0.002**	0.592
Single Bells (Mult.)	0.034	0.008	0.034	0.214	0.022	0.078	0.104	0.105	0.163	0.136	0.014	0.289	**0.001**	0.011	0.027	**0.002**	0.004	0.461
Single Spikes (Add.)	0.008	0.008	0.303	0.181	0.051	0.409	0.161	0.391	0.479	0.436	0.204	0.310	**0.001**	0.030	0.323	**0.001**	0.262	0.954
Single Spikes (Mult.)	0.008	0.008	0.120	0.031	0.008	0.150	0.031	0.145	0.230	0.197	0.093	0.463	**0.000**	0.101	0.104	**0.001**	0.082	0.462
Single Steps (Add.)	0.008	0.008	0.268	0.151	0.020	0.407	0.214	0.390	0.471	0.449	0.052	0.262	**0.001**	0.025	0.086	**0.002**	0.002	0.900
Single Steps (Mult.)	0.007	0.007	0.087	0.018	0.007	0.125	0.059	0.186	0.291	0.283	0.059	0.622	**0.001**	0.042	0.082	**0.001**	0.249	0.596
Agg. Relative Score	0.166	0.080	0.660	0.735	0.182	1.035	0.575	1.000	1.680	1.466	0.285	2.036	**0.011**	0.209	0.396	**0.019**	0.075	3.789
Avg. Rank	7.688	4.500	10.562	11.125	6.250	12.750	10.062	12.750	15.875	14.875	7.500	14.875	**1.625**	6.688	9.062	**2.250**	4.750	17.812

Table 9: WQL scores per model on datasets in 16 **complex** synthetic datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported are averaged over three random seeds.

	Full Finetuned			Adapters Only				Pretrained Models			Baselines							
	ChronosX(FF+IIB+OIB)	ChronosX(FF+OIB)	ChronosX(FF+IIB)	ChronosX(IIB+OIB)	ChronosX(OIB)	ChronosX(IIB)	ChronosX(RS)	Chronos Small	Moirai (Large)	Moirai (Base)	NHiTS	AutoARIMA	TFT	NBEATSx-I	NBEATSx-G	DeepAR	PatchTSTx	Seasonal Naive
Diverse ARP (Add)	0.092	0.034	0.037	0.117	0.054	0.062	0.063	0.063	0.167	0.138	0.026	0.265	0.059	**0.025**	0.027	0.086	**0.020**	0.250
Diverse ARP (Mult.)	0.036	**0.031**	**0.036**	0.077	0.048	0.060	0.059	0.059	0.153	0.137	0.049	0.269	0.055	0.056	0.046	0.085	0.059	0.221
Diverse Bells (Add)	0.208	0.192	0.211	0.271	0.210	0.247	0.259	0.257	0.317	0.313	**0.017**	0.168	0.023	0.024	0.030	0.138	**0.006**	0.464
Diverse Bells (Mult.)	0.051	0.020	0.052	0.116	0.074	0.097	0.088	0.089	0.162	0.155	0.033	0.233	**0.006**	0.019	0.035	0.044	**0.006**	0.213
Diverse Spikes (Add)	0.016	**0.008**	0.195	0.075	0.069	0.246	0.108	0.230	0.288	0.279	0.033	0.352	0.014	0.016	0.041	0.053	**0.005**	0.460
Diverse Spikes (Mult.)	0.021	**0.008**	0.071	0.049	0.043	0.093	0.065	0.084	0.134	0.131	0.055	0.200	**0.008**	0.053	0.061	0.018	0.036	0.202
Diverse Steps (Add)	0.046	0.014	0.194	0.108	0.101	0.276	0.194	0.273	0.296	0.293	0.036	0.326	**0.011**	0.021	0.047	0.046	**0.003**	0.490
Diverse Steps (Mult.)	0.042	**0.013**	0.093	0.087	0.066	0.145	0.123	0.126	0.185	0.188	0.103	0.264	**0.039**	0.085	0.153	0.095	0.039	0.273
Noisy ARP (Add)	0.063	0.059	0.063	0.102	0.065	0.077	0.071	0.072	0.169	0.152	0.055	0.185	0.062	**0.049**	**0.052**	0.058	0.056	0.261
Noisy ARP (Mult.)	0.064	**0.049**	0.067	0.077	0.059	0.076	0.066	0.066	0.144	0.134	0.091	0.253	0.061	0.077	0.084	0.072	**0.049**	0.219
Noisy Bells (Add)	0.198	0.070	0.170	0.265	0.155	0.205	0.243	0.249	0.299	0.290	**0.049**	0.108	0.253	0.056	0.060	0.087	**0.035**	0.439
Noisy Bells (Mult.)	**0.049**	**0.043**	0.050	0.101	0.072	0.093	0.094	0.094	0.163	0.157	0.076	0.205	0.071	0.083	0.082	0.064	0.061	0.213
Noisy Spikes (Add)	**0.050**	**0.043**	0.211	0.087	0.082	0.259	0.117	0.239	0.277	0.274	0.058	0.208	0.083	0.058	0.056	0.060	0.056	0.433
Noisy Spikes (Mult.)	0.038	**0.034**	0.083	0.059	0.054	0.104	0.075	0.100	0.149	0.146	0.046	0.197	0.049	0.042	0.043	0.036	**0.035**	0.216
Noisy Steps (Add)	0.065	0.042	0.128	0.127	0.110	0.243	0.202	0.277	0.266	0.267	0.041	0.197	0.109	**0.041**	0.043	0.043	**0.038**	0.425
Noisy Steps (Mult.)	0.044	**0.038**	0.062	0.072	0.068	0.117	0.117	0.120	0.158	0.154	0.090	0.223	0.044	0.071	0.091	0.043	**0.036**	0.259
Agg. Relative Score	0.420	**0.247**	0.703	0.782	0.591	1.021	0.841	1.000	1.551	1.486	0.378	1.728	0.308	0.335	0.420	0.455	**0.190**	2.329
Avg. Rank	5.812	**2.938**	8.875	11.438	7.625	12.625	11.250	12.438	16.062	15.312	6.125	15.375	5.625	5.438	7.000	6.812	**2.500**	17.750

Table 10: MASE scores per model on datasets in 16 **simple** synthetic datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported are averaged over three random seeds.

	Full Finetuned			Adapters Only				Pretrained Models			Baselines							
	ChronosX(FF+IIB+OIB)	ChronosX(FF+OIB)	ChronosX(FF+IIB)	ChronosX(IIB+OIB)	ChronosX(OIB)	ChronosX(IIB)	ChronosX(RS)	Chronos Small	Moirai (Large)	Moirai (Base)	NHiTS	AutoARIMA	TFT	NBEATSx-I	NBEATSx-G	DeepAR	PatchTSTx	Seasonal Naive
Simple ARP (Add)	0.947	0.139	0.262	0.951	0.261	0.344	0.317	0.322	0.849	0.571	0.103	1.103	0.038	0.112	0.129	**0.010**	**0.006**	1.788
Simple ARP (Mult.)	0.161	0.033	0.127	0.500	0.110	0.176	0.156	0.157	0.642	0.454	0.201	1.616	**0.006**	0.175	0.180	**0.014**	0.025	1.930
Simple Bells (Add)	3.203	3.044	3.227	2.459	2.119	2.805	3.474	3.483	4.069	3.689	0.564	1.993	**0.078**	0.553	0.611	0.594	**0.027**	4.951
Simple Bells (Mult.)	0.220	**0.046**	0.214	0.858	0.130	0.462	0.472	0.473	0.847	0.691	0.236	1.698	**0.011**	0.246	0.294	0.068	0.058	1.882
Simple Spikes (Add)	0.010	0.010	0.515	0.372	0.101	0.733	0.292	0.653	0.823	0.756	0.120	0.644	**0.005**	0.050	0.590	0.007	**0.005**	1.258
Simple Spikes (Mult.)	0.024	0.024	0.469	0.089	0.024	0.600	0.093	0.581	0.781	0.664	0.421	1.930	**0.002**	0.445	0.425	**0.004**	0.416	1.397
Simple Steps (Add)	0.023	0.022	1.163	0.742	0.141	1.691	0.780	1.653	1.862	1.856	0.251	1.079	0.039	0.095	0.277	**0.012**	**0.020**	2.577
Simple Steps (Mult.)	0.023	0.023	0.306	0.113	0.023	0.490	0.254	0.720	1.067	1.014	0.172	2.262	**0.003**	0.165	0.216	**0.006**	0.649	1.808
Single ARP (Add.)	0.601	0.058	0.275	0.811	0.232	0.358	0.340	0.344	0.823	0.548	0.081	1.200	**0.005**	0.076	0.097	**0.011**	0.012	1.774
Single ARP (Mult.)	0.148	0.034	0.111	0.593	0.106	1.084	0.162	0.166	0.540	0.418	0.033	1.914	**0.002**	0.035	0.046	**0.003**	0.011	1.877
Single Bells (Add.)	2.675	1.759	3.126	2.599	1.946	2.762	3.735	3.728	4.079	3.835	0.095	1.945	0.044	0.115	0.141	**0.028**	**0.021**	4.779
Single Bells (Mult.)	0.165	0.046	0.168	1.122	0.127	0.426	0.539	0.539	0.851	0.684	0.077	1.611	**0.003**	0.059	0.152	**0.009**	0.020	1.989
Single Spikes (Add.)	0.010	0.010	0.496	0.309	0.078	0.722	0.264	0.653	0.826	0.690	0.443	0.540	**0.002**	0.043	0.562	**0.001**	0.532	1.242
Single Spikes (Mult.)	0.023	0.023	0.466	0.115	0.023	0.586	0.089	0.575	0.831	0.629	0.388	1.909	**0.001**	0.381	0.384	**0.002**	0.386	1.415
Single Steps (Add.)	0.019	0.019	0.969	0.514	0.059	1.367	0.702	1.273	1.566	1.501	0.207	0.886	**0.002**	0.065	0.309	**0.006**	0.008	2.281
Single Steps (Mult.)	0.023	0.023	0.361	0.061	0.023	0.486	0.215	0.735	1.067	0.998	0.207	2.672	**0.002**	0.154	0.290	**0.006**	1.101	1.841
Agg. Relative Score	0.144	0.074	0.671	0.694	0.173	1.057	0.566	1.000	1.642	1.374	0.265	2.153	**0.009**	0.185	0.359	**0.015**	0.071	2.989
Avg. Rank	7.469	4.719	10.500	10.938	6.250	13.000	10.062	12.875	15.938	14.688	7.375	14.938	**1.688**	6.562	8.938	**2.250**	5.125	17.688

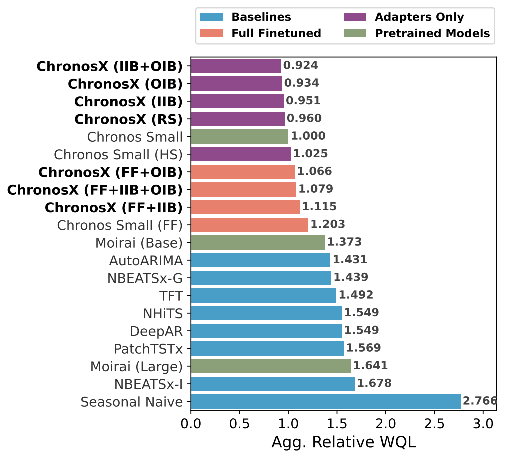
Table 11: MASE scores per model on datasets in 16 **complex** synthetic datasets. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported are averaged over three random seeds.

	Full Finetuned			Adapters Only				Pretrained Models			Baselines							
	ChronosX(FF+IIB+OIB)	ChronosX(FF+OIB)	ChronosX(FF+IIB)	ChronosX(IIB+OIB)	ChronosX(OIB)	ChronosX(IIB)	ChronosX(RS)	Chronos Small	Moirai (Large)	Moirai (Base)	NHiTS	AutoARIMA	TFT	NBEATSx-I	NBEATSx-G	DeepAR	PatchTSTx	Seasonal Naive
Diverse ARP (Add)	1.004	0.461	0.488	1.282	0.687	0.758	0.769	0.768	2.056	1.848	**0.351**	2.801	0.797	0.373	0.372	1.101	**0.273**	1.947
Diverse ARP (Mult.)	0.500	**0.446**	**0.484**	0.987	0.655	0.754	0.767	0.772	1.928	1.791	0.775	3.091	0.796	0.835	0.698	1.311	0.890	1.921
Diverse Bells (Add)	3.392	3.178	3.508	4.066	3.378	3.838	4.000	3.975	4.634	4.713	**0.133**	2.234	0.203	0.210	0.245	1.249	**0.045**	5.563
Diverse Bells (Mult.)	0.925	0.456	0.929	1.959	1.414	1.731	1.591	1.620	2.834	2.755	0.264	3.453	**0.053**	0.170	0.305	0.346	**0.045**	2.698
Diverse Spikes (Add)	0.062	0.036	0.681	0.294	0.283	0.808	0.439	0.767	1.048	0.999	0.084	1.203	**0.032**	0.042	0.095	0.143	**0.015**	1.286
Diverse Spikes (Mult.)	0.173	**0.072**	0.587	0.451	0.410	0.786	0.604	0.691	1.247	1.201	0.347	1.826	**0.066**	0.320	0.387	0.128	0.279	1.427
Diverse Steps (Add)	0.393	0.124	1.624	0.896	0.870	2.196	1.583	2.099	2.387	2.388	0.352	2.182	**0.101**	0.201	0.405	0.204	**0.030**	2.851
Diverse Steps (Mult.)	**0.495**	**0.185**	1.091	1.054	0.847	1.508	1.446	1.464	2.110	2.195	2.156	3.165	0.791	1.696	2.980	2.001	0.770	2.329
Noisy ARP (Add)	0.613	0.577	0.614	0.967	0.642	0.747	0.703	0.713	1.580	1.473	0.202	1.760	0.205	**0.183**	**0.196**	0.223	0.214	1.713
Noisy ARP (Mult.)	0.750	0.583	0.750	0.884	0.716	0.779	0.783	0.778	1.647	1.619	0.540	2.510	**0.399**	0.463	0.495	0.419	**0.302**	1.665
Noisy Bells (Add)	2.370	0.962	2.089	3.301	2.125	2.689	3.274	3.328	3.851	3.721	**0.736**	1.472	3.454	0.852	0.866	1.332	**0.530**	4.558
Noisy Bells (Mult.)	0.693	0.626	0.704	1.425	1.079	1.333	1.369	1.360	2.211	2.166	0.510	2.819	**0.426**	0.581	0.532	0.448	**0.384**	2.155
Noisy Spikes (Add)	**0.210**	**0.184**	0.675	0.344	0.329	0.822	0.456	0.776	0.995	0.974	0.283	0.842	0.394	0.287	0.275	0.300	0.283	1.194
Noisy Spikes (Mult.)	**0.362**	**0.335**	0.655	0.547	0.501	0.815	0.679	0.777	1.281	1.265	0.500	1.717	0.596	0.439	0.472	0.471	0.446	1.401
Noisy Steps (Add)	0.524	**0.347**	0.971	0.958	0.865	1.752	1.540	1.994	2.030	2.071	**0.422**	1.479	1.251	0.433	0.455	0.432	0.423	2.424
Noisy Steps (Mult.)	0.497	**0.447**	0.694	0.806	0.741	1.192	1.172	1.190	1.597	1.574	0.992	2.123	0.523	0.760	0.977	0.459	**0.412**	2.052
Agg. Relative Score	0.443	0.281	0.718	0.808	0.646	1.003	0.885	1.000	1.573	1.534	0.333	1.674	**0.273**	0.299	0.365	0.393	**0.172**	1.741
Avg. Rank	6.312	**3.812**	9.062	11.438	8.125	12.188	11.312	12.062	16.188	15.625	5.562	15.750	5.875	5.188	6.188	6.500	**2.875**	16.938

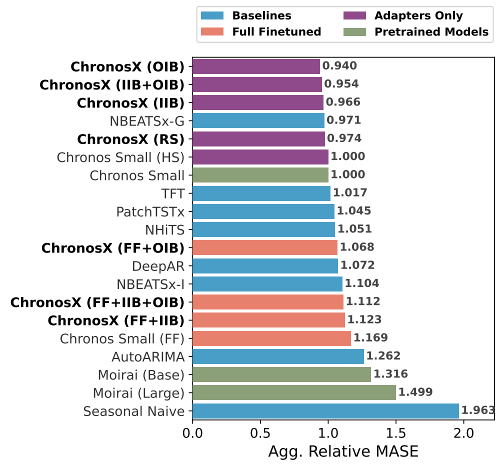
J.0.2 Fine-tuning and Adapters in real datasets with covariates

We present the following results:

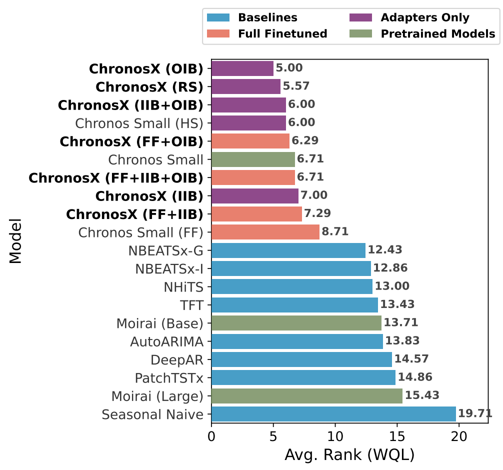
- In Figure 14 we present the Aggregated Relative and Average Ranks with WQL and MASE on real datasets with covariates,
- In Table 12 and Table 13 we present the WQL and MASE scores on real datasets with covariates.



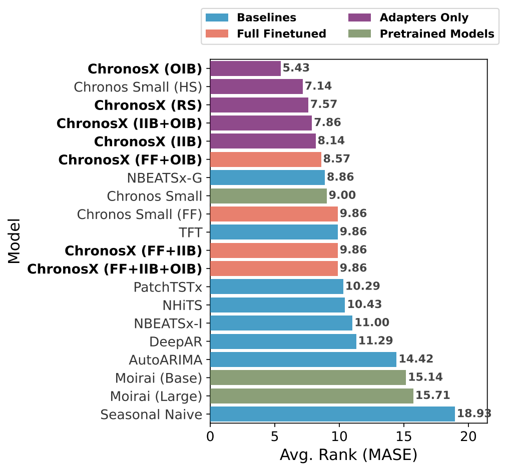
(a) Agg. Rel. WQL on real datasets with covariates.



(b) Agg. Rel. MASE on real datasets with covariates.



(c) Avg. Rank. WQL on real datasets with covariates.



(d) Avg. Rank. MASE on real datasets with covariates.

Figure 14: Aggregated Relative WQL and MASE on real datasets with covariates.

Table 12: WQL scores per model on datasets in real datasets with covariates. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported are averaged over three random seeds.

	Full Finetuned				Adapters Only					Pretrained Models			Baselines								
	ChronosX (FF+IIB+OIB)	ChronosX (FF+OIB)	ChronosX (FF+IIB)	Chronos Small (FF)	ChronosX (IIB+OIB)	ChronosX (OIB)	ChronosX (IIB)	ChronosX (RS)	Chronos Small (HS)	Chronos Small	Moirai (Base)	Moirai (Large)	AutoARIMA	TFT	DeepAR	PatchTSTx	NBEATSx-G	NBEATSx-I	NHiTS	Seasonal Naive
ETT (15 Min.)	0.022	0.020	0.026	0.023	**0.016**	0.017	0.016	**0.015**	0.017	0.016	0.019	0.040	0.020	0.028	0.042	0.034	0.043	0.063	0.064	0.065
ETT (Hourly)	0.025	0.024	0.025	0.024	0.025	**0.023**	0.024	0.024	0.023	**0.022**	0.037	0.076	0.045	0.073	0.053	0.081	0.069	0.097	0.067	0.101
Hermes	0.090	**0.090**	0.090	**0.090**	0.100	0.098	0.101	0.099	0.098	0.100	0.188	0.172	0.217	0.122	0.114	0.132	0.102	0.097	0.099	0.414
M5	0.538	0.538	**0.537**	**0.537**	0.557	0.551	0.552	0.551	0.549	0.549	0.586	0.576	-	0.721	0.934	0.944	0.785	0.795	0.813	0.934
Electricity	0.133	0.164	0.138	0.188	**0.077**	0.086	0.082	0.083	0.108	0.097	0.142	0.125	0.123	0.117	0.094	0.089	**0.069**	0.113	0.086	0.266
ProEnFo	0.057	0.049	0.059	0.088	**0.041**	**0.041**	0.045	0.054	0.063	0.064	0.093	0.094	0.077	0.086	0.106	0.082	0.086	0.081	0.077	0.120
Rideshare	**0.177**	**0.177**	0.177	0.178	0.178	0.179	0.191	0.179	0.177	0.182	0.192	0.194	0.257	0.190	0.194	0.197	0.191	0.185	0.201	0.325
Agg. Relative Score	1.079	1.066	1.115	1.203	**0.924**	**0.934**	0.951	0.960	1.025	1.000	1.373	1.641	1.431	1.492	1.549	1.569	1.439	1.678	1.549	2.766
Avg. Rank	6.714	6.286	7.286	8.714	6.000	**5.000**	7.000	**5.571**	6.000	6.714	13.714	15.429	13.833	13.429	14.571	14.857	12.429	12.857	13.000	19.714

Table 13: MASE scores per model on datasets in real datasets with covariates. Models achieving the **first**, **second**, and third best scores have been highlighted. Scores reported are averaged over three random seeds.

	Full Finetuned				Adapters Only					Pretrained Models			Baselines								
	ChronosX (FF+IIB+OIB)	ChronosX (FF+OIB)	ChronosX (FF+IIB)	Chronos Small (FF)	ChronosX (IIB+OIB)	ChronosX (OIB)	ChronosX (IIB)	ChronosX (RS)	Chronos Small (HS)	Chronos Small	Moirai (Base)	Moirai (Large)	AutoARIMA	TFT	DeepAR	PatchTSTx	NBEATSx-G	NBEATSx-I	NHiTS	Seasonal Naive
ETT (15 Min.)	0.338	0.297	0.352	0.337	0.213	0.207	0.212	**0.206**	0.211	0.224	0.259	0.513	0.225	**0.172**	0.242	0.211	0.241	0.386	0.378	0.677
ETT (Hourly)	0.354	0.341	0.346	0.333	0.352	0.331	0.331	0.338	**0.313**	**0.297**	0.489	0.900	0.469	0.443	0.323	0.473	0.363	0.507	0.366	0.950
Hermes	0.766	0.763	0.766	0.762	0.850	0.831	0.845	0.845	0.834	0.853	1.412	1.266	1.610	0.524	0.500	0.601	**0.433**	**0.421**	0.463	2.382
M5	0.907	0.907	**0.904**	**0.903**	0.925	0.918	0.925	0.918	0.915	0.919	0.959	0.925	-	1.436	1.522	1.524	1.515	1.518	1.520	1.268
Electricity	0.609	0.672	0.628	0.715	**0.420**	0.453	0.431	0.450	0.525	0.497	0.734	0.567	0.489	0.563	0.553	0.421	**0.412**	0.505	0.440	0.781
ProEnFo	0.779	0.631	0.795	1.003	**0.533**	**0.507**	0.575	0.633	0.701	0.706	0.990	1.006	0.918	0.628	0.868	0.614	0.620	0.578	0.597	1.016
Rideshare	**0.461**	**0.460**	0.462	0.463	0.467	0.466	0.498	0.467	0.463	0.472	0.473	0.476	0.630	0.478	0.492	0.499	0.480	0.472	0.478	0.630
Agg. Relative Score	1.112	1.068	1.123	1.169	**0.954**	**0.940**	0.966	0.974	1.000	1.000	1.316	1.499	1.262	1.017	1.072	1.045	0.971	1.104	1.051	1.963
Avg. Rank	9.857	8.571	9.857	9.857	7.857	**5.429**	8.143	7.571	**7.143**	9.000	15.143	15.714	14.417	9.857	11.286	10.286	8.857	11.000	10.429	18.929

544 **J.0.3 Probabilistic Forecasts of Chronos Variants**

545 From Figure 15 to 32, we plot the probabilistic forecast of different Chronos variants.

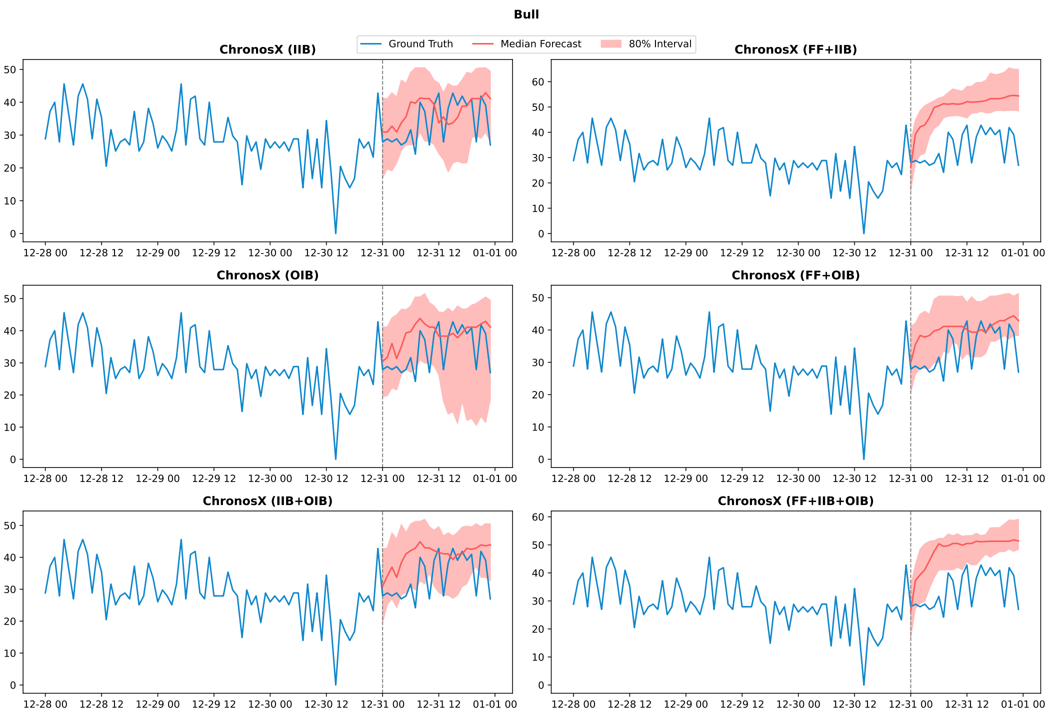


Figure 15: Forecasts on Bull

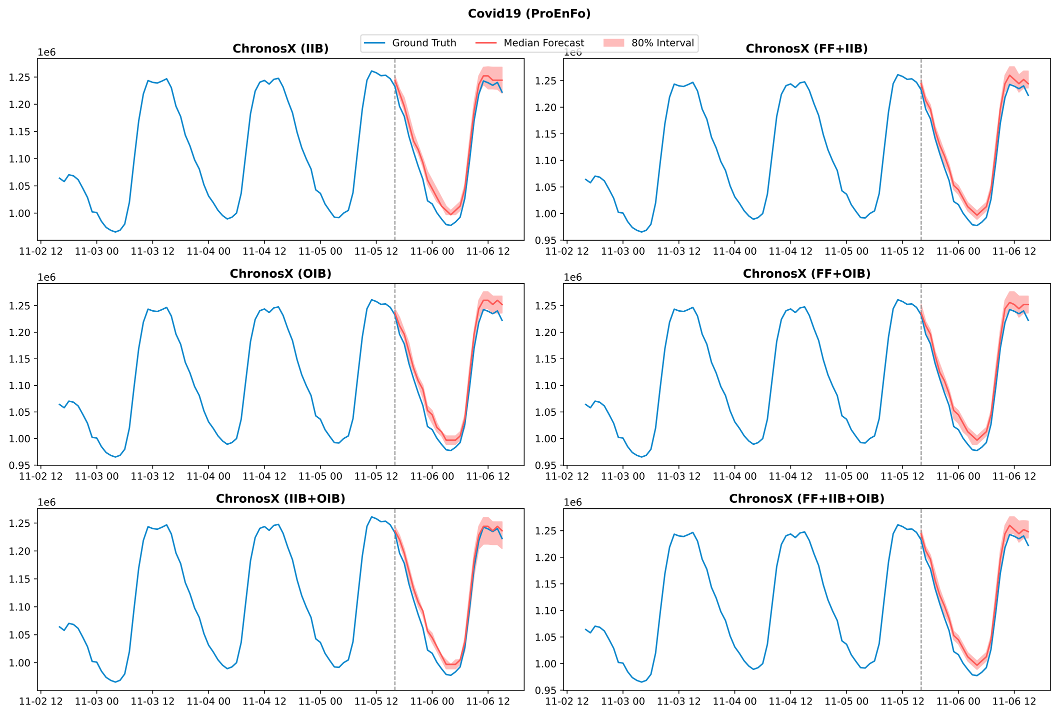


Figure 16: Forecasts on Covid19 (ProEnFo)

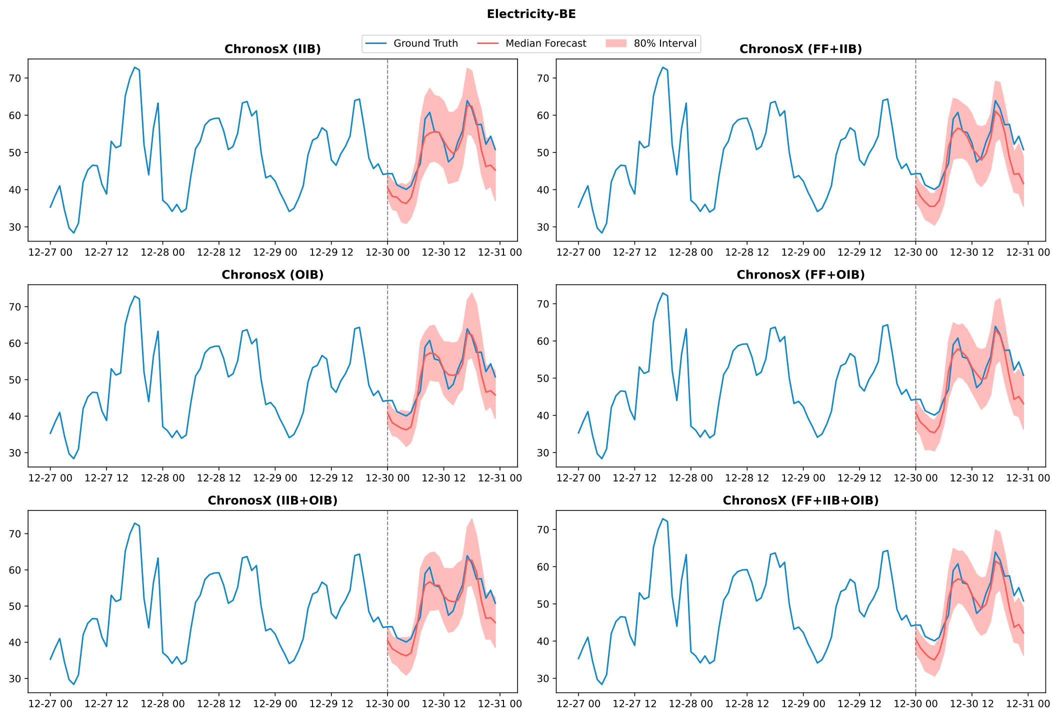


Figure 17: Forecasts on Electricity-BE

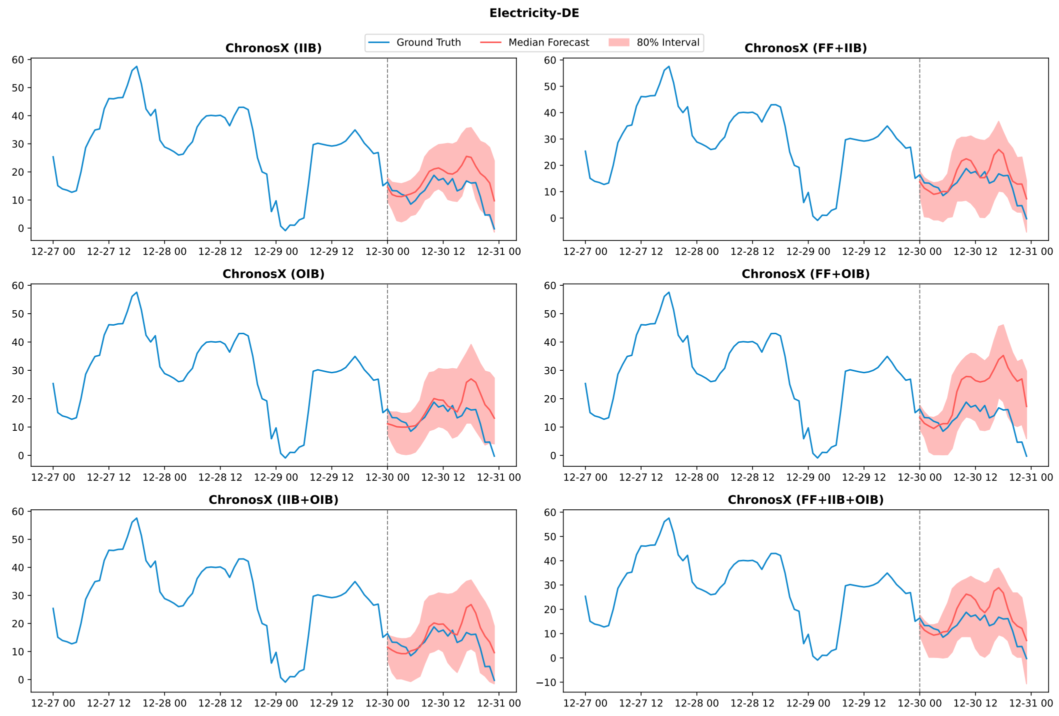


Figure 18: Forecasts on Electricity-DE

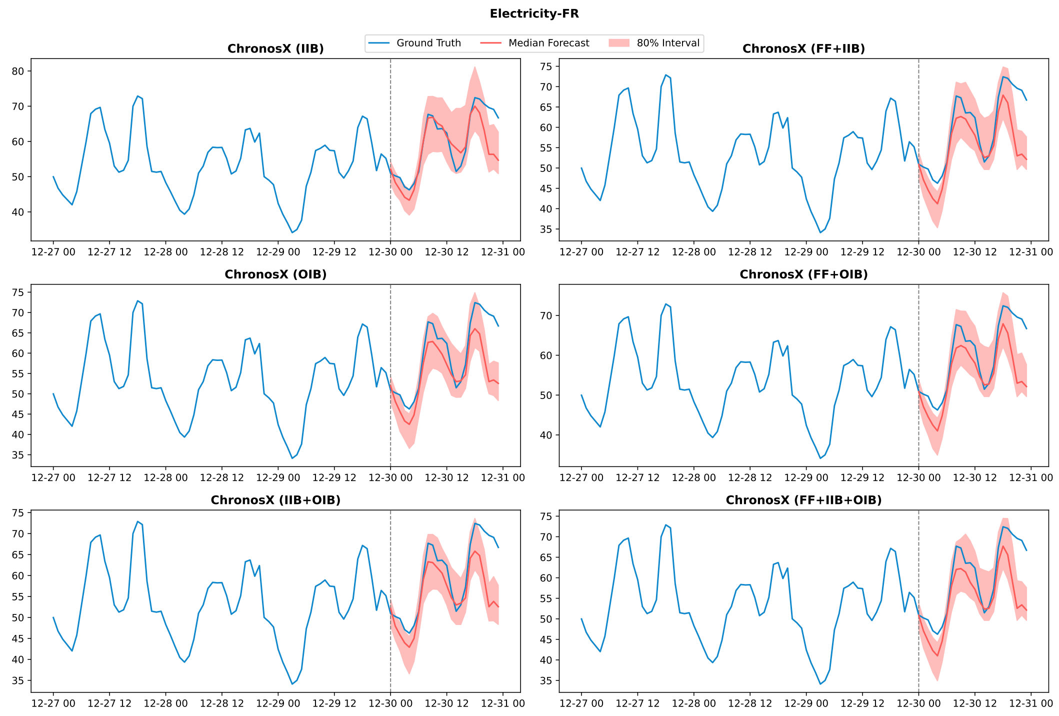


Figure 19: Forecasts on Electricity-FR

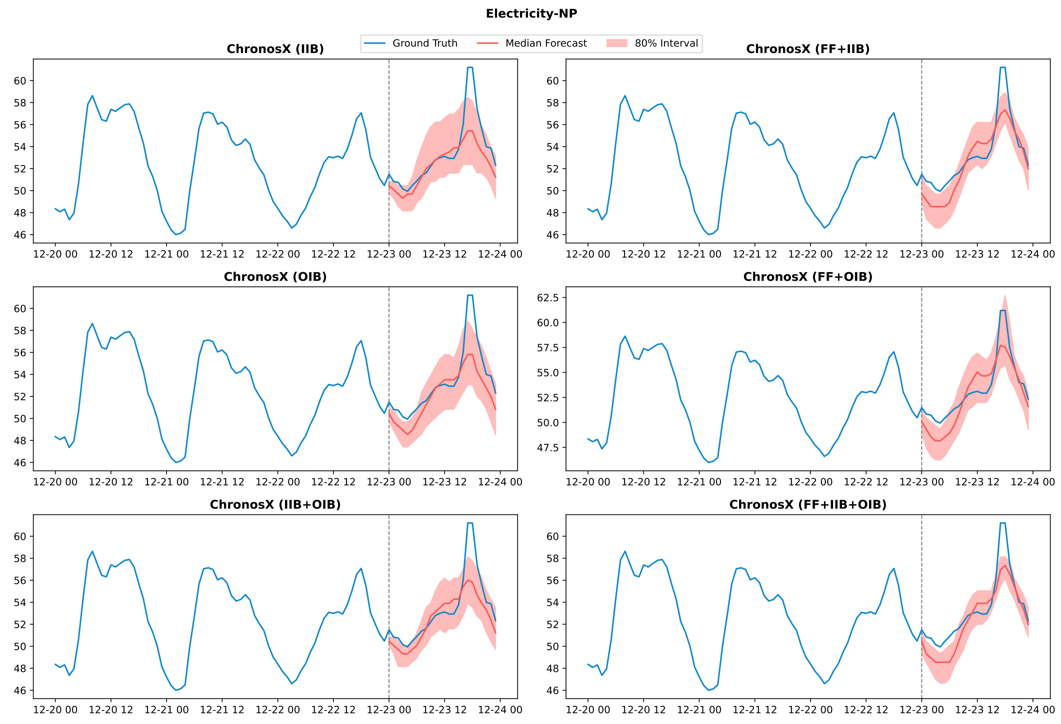


Figure 20: Forecasts on Electricity-NP

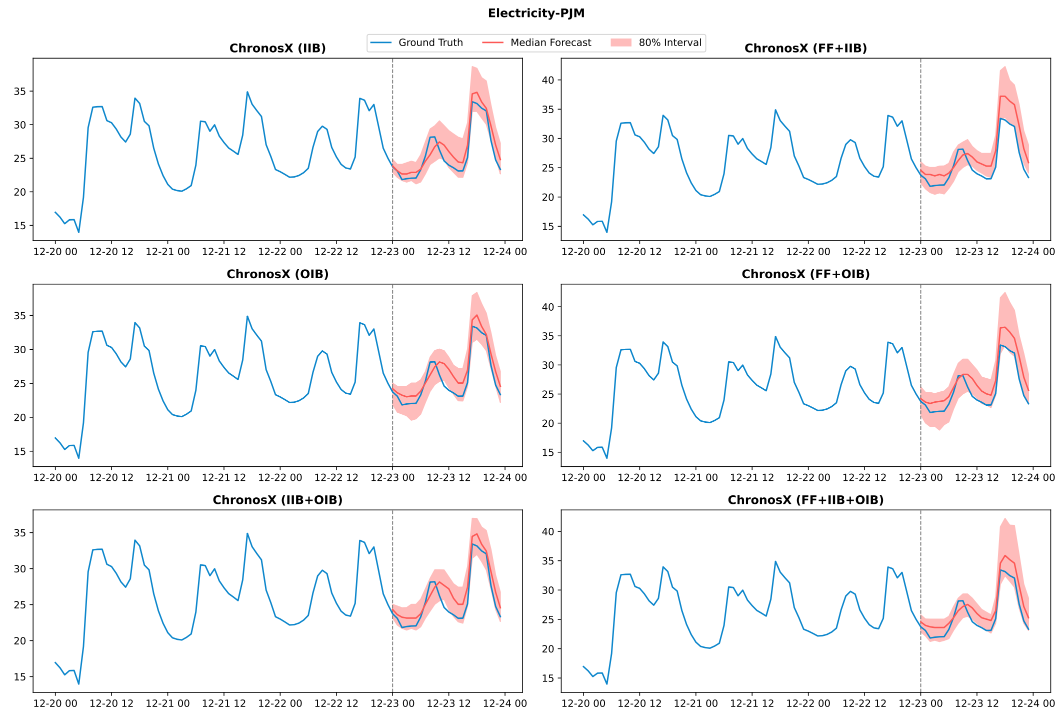


Figure 21: Forecasts on Electricity-PJM

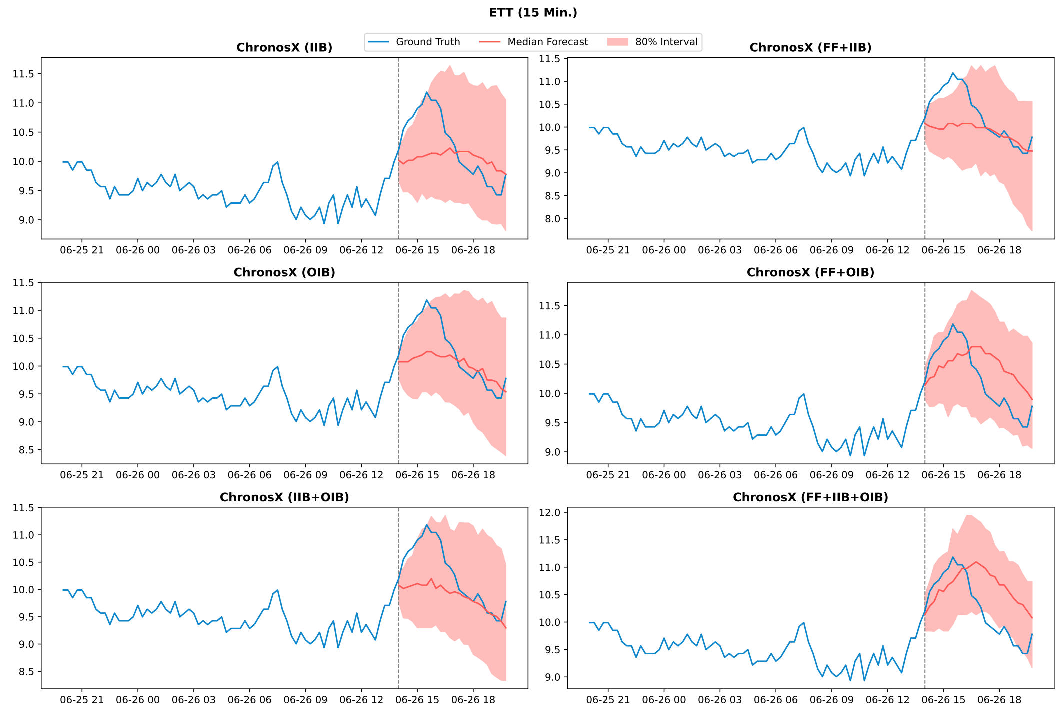


Figure 22: Forecasts on ETT (15 Min.)

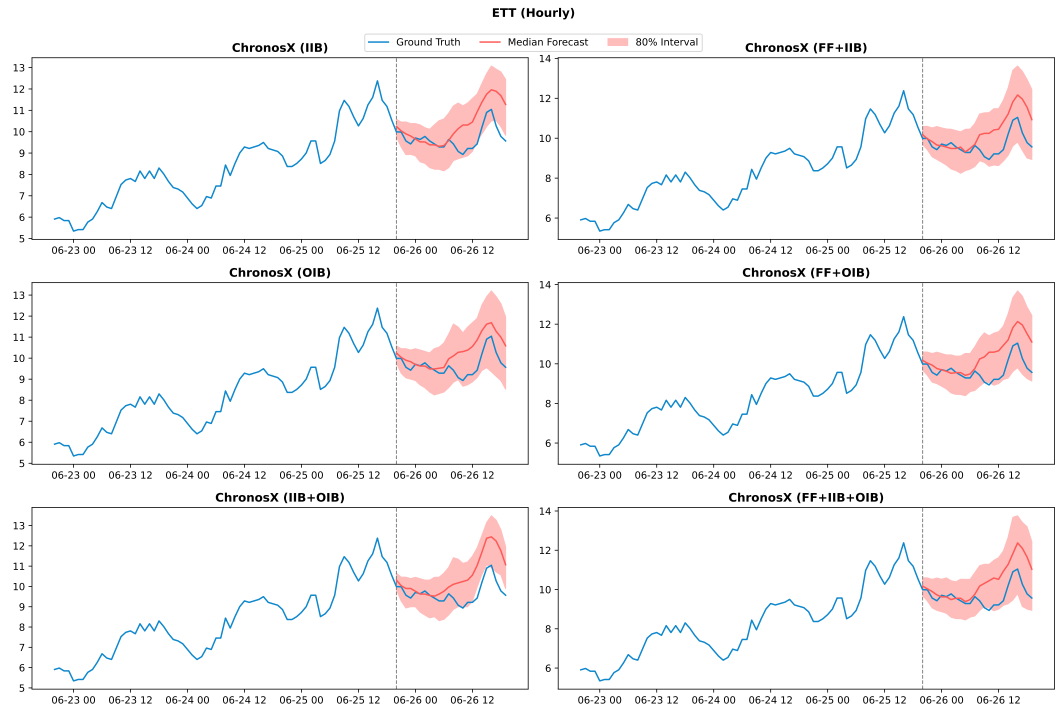


Figure 23: Forecasts on ETT (Hourly)

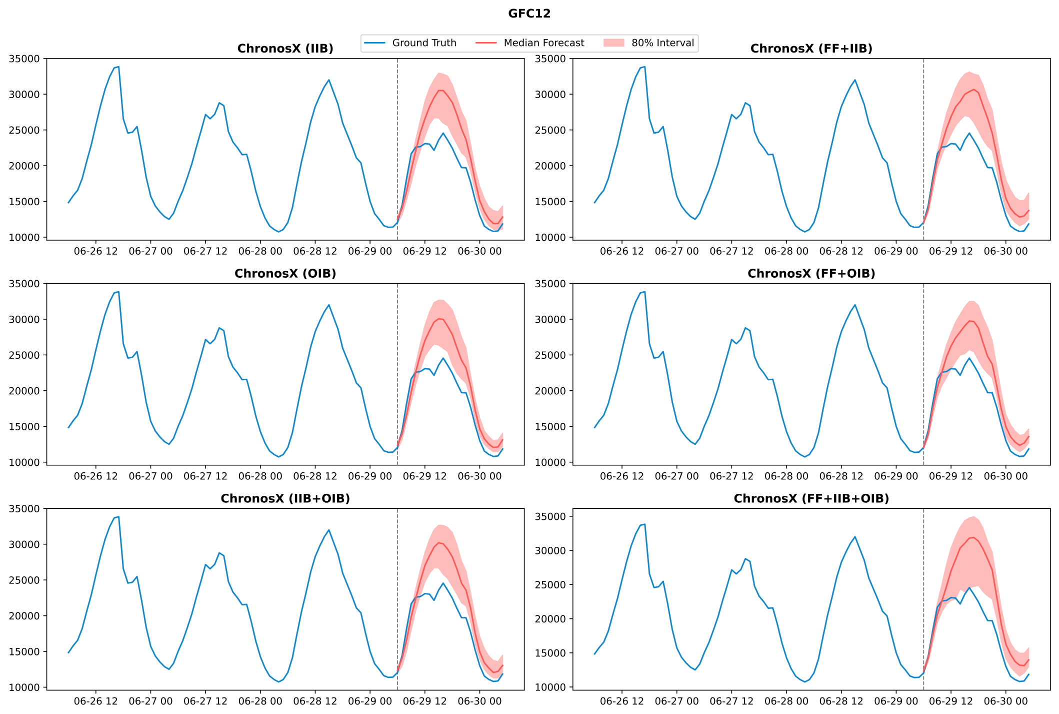


Figure 24: Forecasts on GFC12

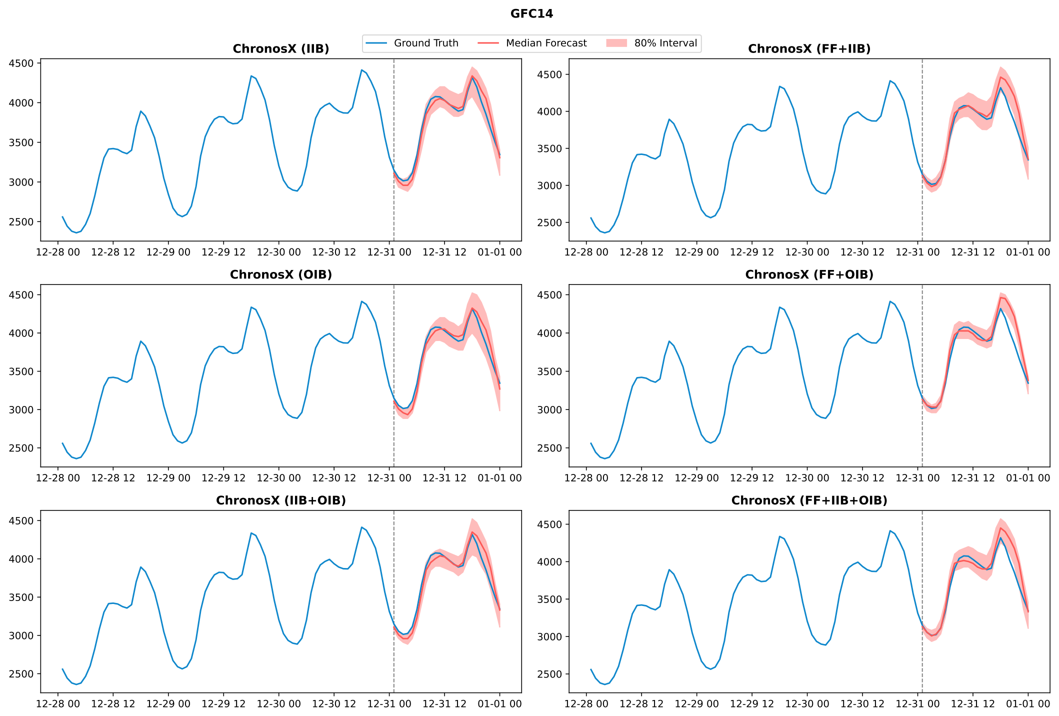


Figure 25: Forecasts on GFC14

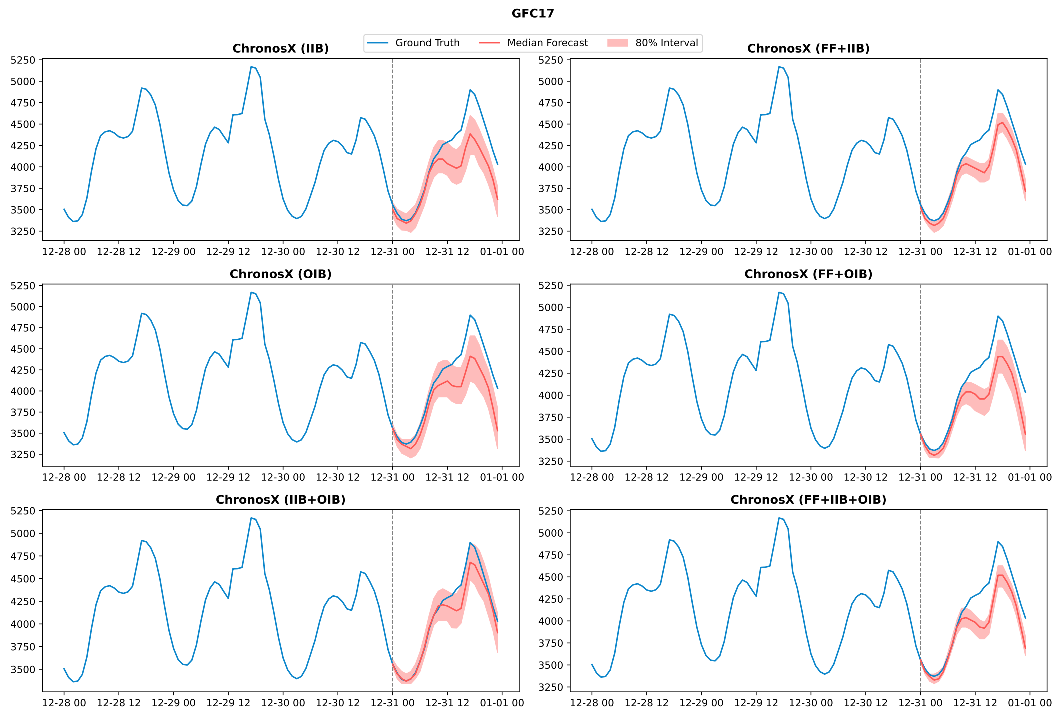


Figure 26: Forecasts on GFC17

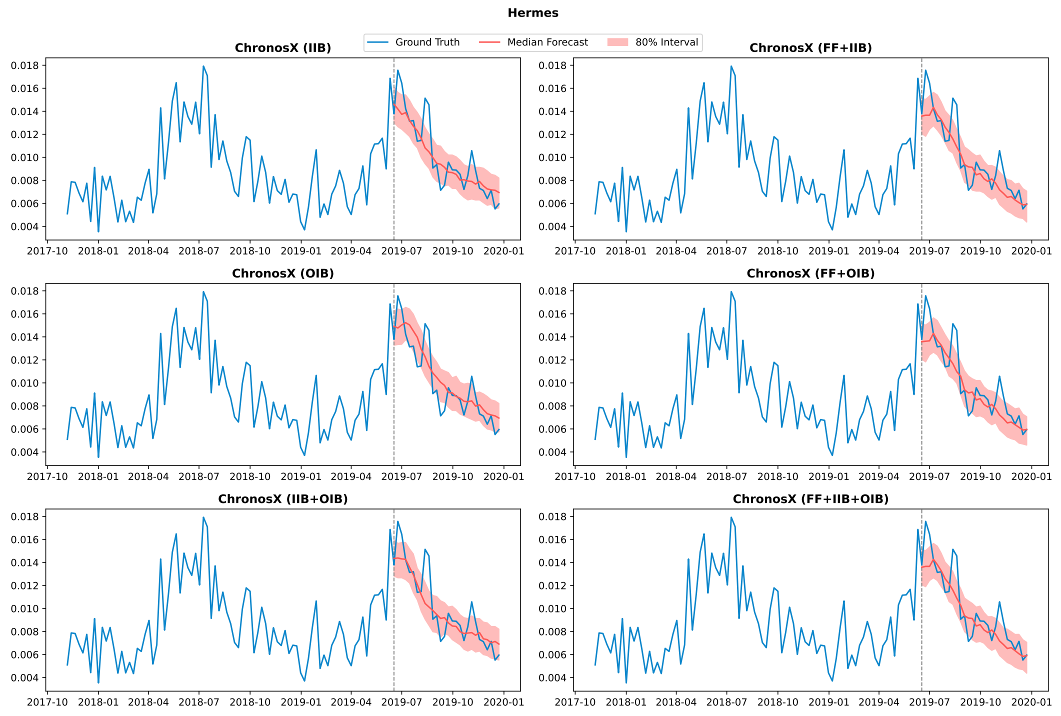


Figure 27: Forecasts on Hermes

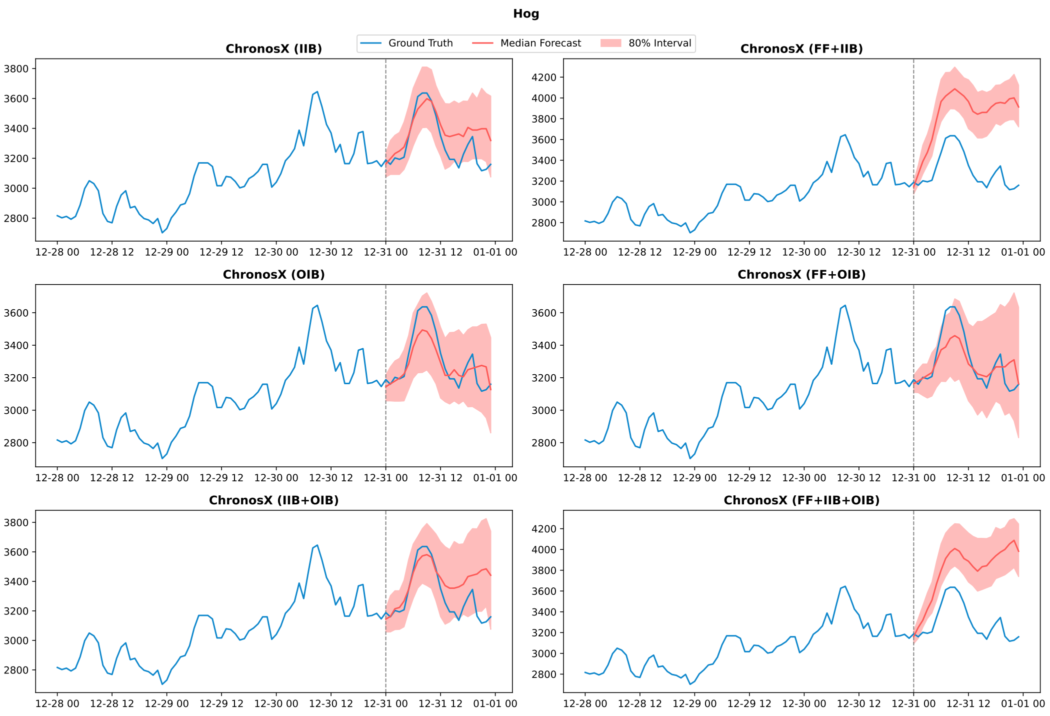


Figure 28: Forecasts on Hog

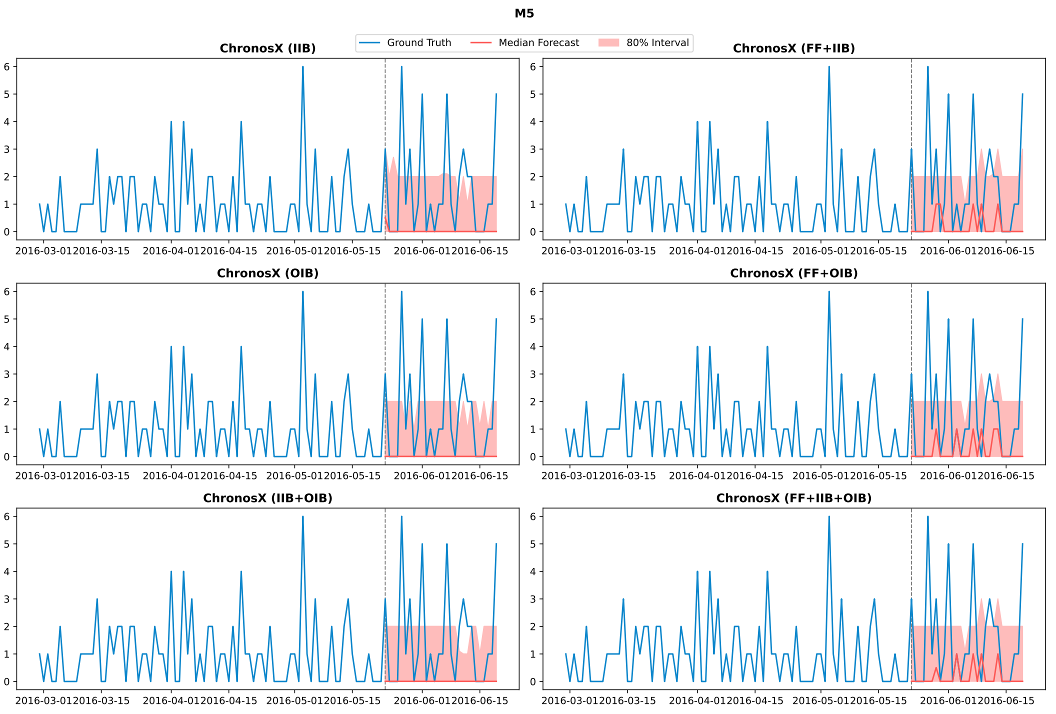


Figure 29: Forecasts on M5

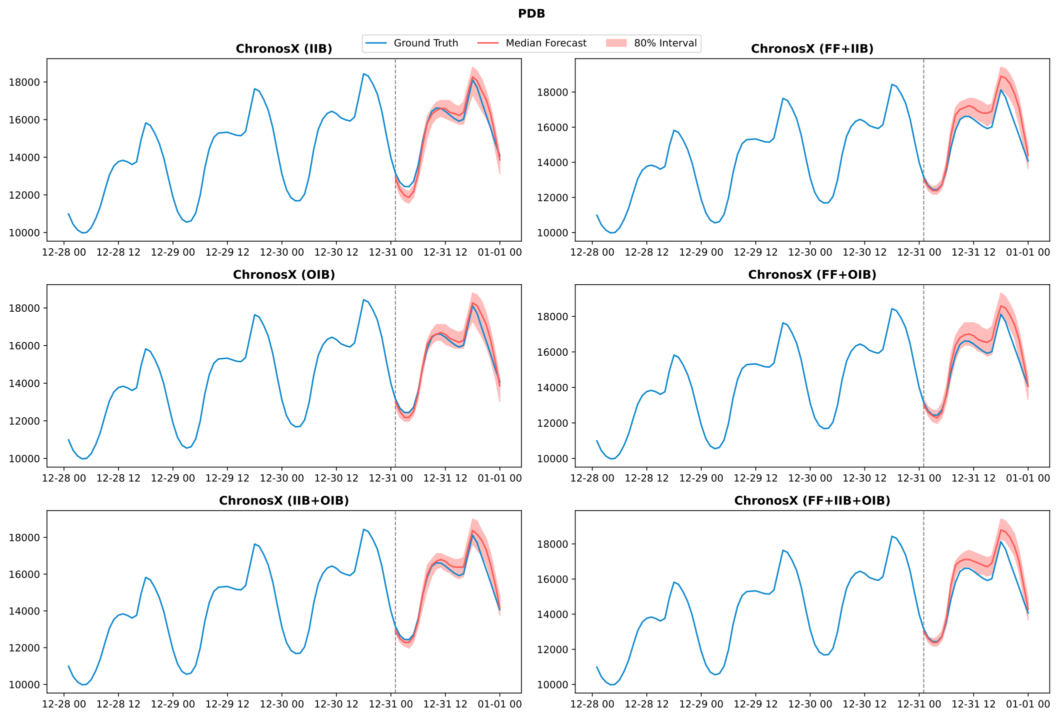


Figure 30: Forecasts on PDB

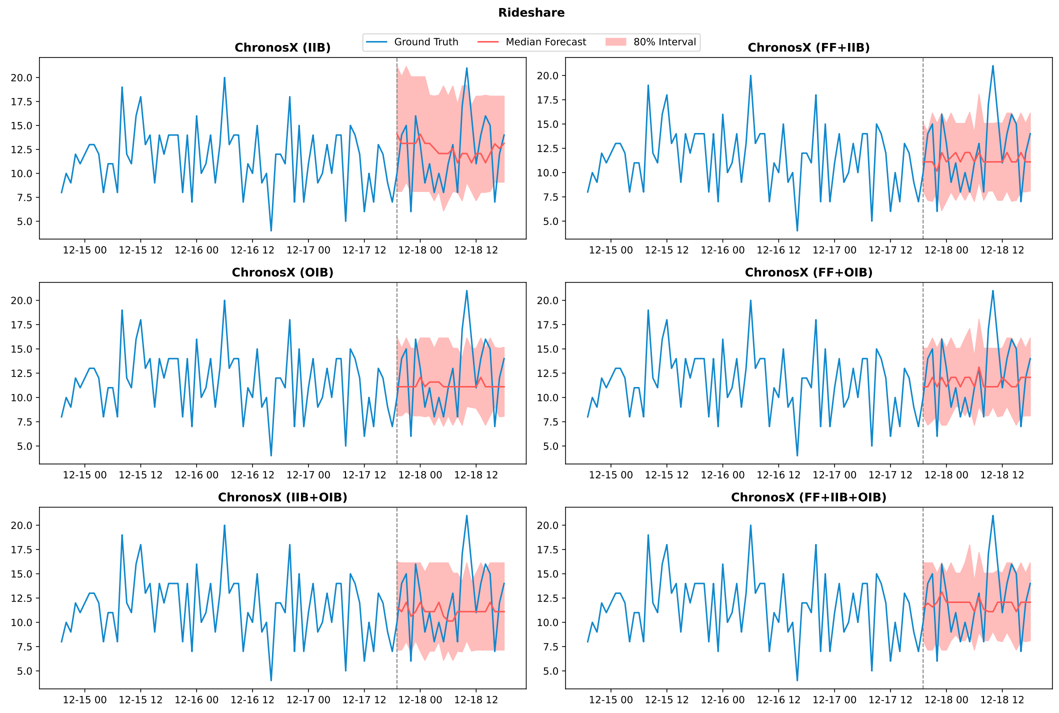


Figure 31: Forecasts on Rideshare

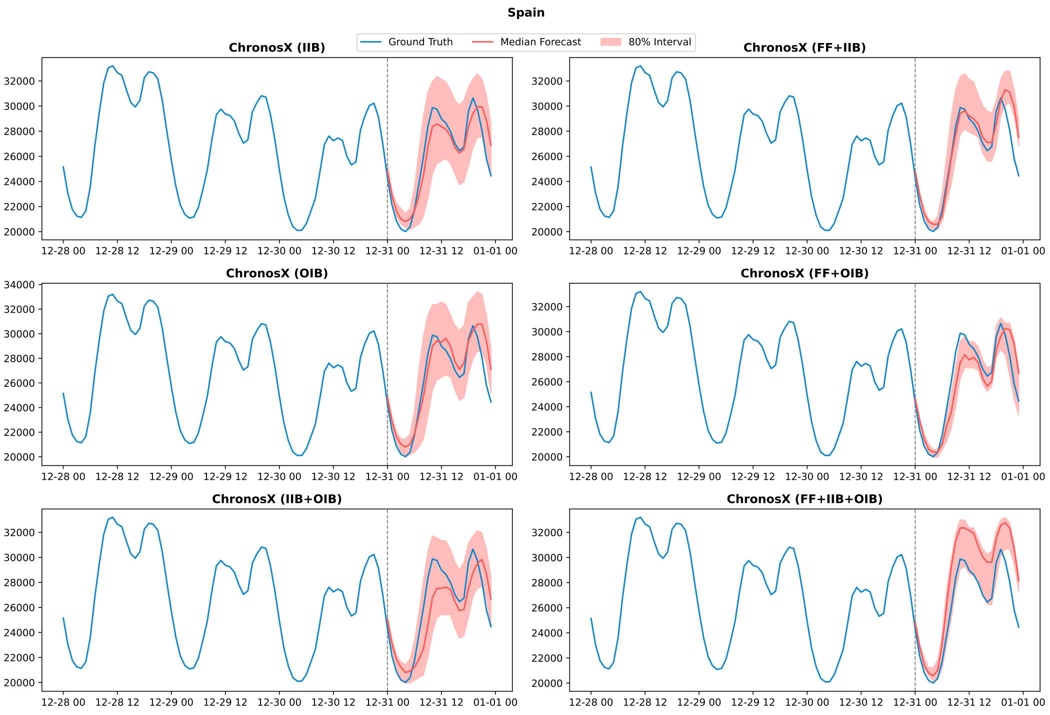


Figure 32: Forecasts on Spain

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explain how to do the covariate integration in Section 3, following what was exposed in the abstract. We validate our approach to integrate covariates into pretrained time series models by evaluating our method on synthetic datasets and real-world datasets with covariates.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in a dedicated section.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There are no theoretical results in this paper.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We discuss in detail the setup in Sections 5 and B.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All the datasets we used were public. This submission does not include the code but we aim to release the code with the camera-ready version of the manuscript.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify the data splits, hyperparameters and other details in Sections 5 and B.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report relative scores that are aggregated across all datasets using the geometric mean. We use the geometric mean preserves the scale of the relative scores and therefore is more meaningful to aggregate relative scores than an arithmetic mean, which does not preserve this property (see Fleming & Wallace 1986: "How no to lie with statistics"). However, the standard deviation or standard error for the geometric mean are not straightforward to interpret in terms of statistical significance. Therefore, we have not decided to not report the error bar to not confuse the reader with errors that do not signal statistical significance. Note that we report the average scores over three runs throughout the paper to improve robustness of our results.

8. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the ethics guidelines. We do not foresee potentially harmful consequences.

9. **Broader Impacts**

   Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

   Answer: [Yes]

   Justification: We motivate our work and its possible impact in the abstract and in the introduction. Time series forecasting is a generalist technology and could be used by actors with malicious or ethically questionable intend. However, this is true for any generalist technology and we do not explicitly discuss this in the paper.

10. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

11. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes] ,

    Justification: Yes, we cite and acknowledge all assets that we use in the paper in the supplementary material H.

12. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce a new synthetic dataset and describe it in Section 4.

13. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

14. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]