

Chronos-2: From Univariate to Universal Forecasting

Abdul Fatir Ansari^{1*}, Oleksandr Shchur^{1*}, Jaris Küken^{1,3*†}, Andreas Auer^{1,4‡}, Boran Han¹, Pedro Mercado¹, Syama Sundar Rangapuram¹, Huibin Shen¹, Lorenzo Stella¹, Xiyuan Zhang¹, Mononito Goswami¹, Shubham Kapoor¹, Danielle C. Maddix¹, Pablo Guerron^{2,5†}, Tony Hu¹, Junming Yin¹, Nick Erickson¹, Prateek Mutalik Desai¹, Hao Wang^{1,6‡}, Huzefa Rangwala¹, George Karypis¹, Yuyang Wang^{1‡}, Michael Bohlke-Schneider^{1‡}

ansarnd@amazon.de

¹Amazon Web Services ²Amazon ³University of Freiburg ⁴Johannes Kepler University Linz ⁵Boston College ⁶Rutgers University

Code: github.com/amazon-science/chronos-forecasting

Abstract

Pretrained time series models have enabled inference-only forecasting systems that produce accurate predictions without task-specific training. However, existing approaches largely focus on univariate forecasting, limiting their applicability in real-world scenarios where multivariate data and covariates play a crucial role. We present Chronos-2, a pretrained model capable of handling univariate, multivariate, and covariate-informed forecasting tasks in a zero-shot manner. Chronos-2 employs a group attention mechanism that facilitates in-context learning (ICL) through efficient information sharing across multiple time series within a group, which may represent sets of related series, variates of a multivariate series, or targets and covariates in a forecasting task. These general capabilities are achieved through training on synthetic datasets that impose diverse multivariate structures on univariate series. Chronos-2 delivers state-of-the-art performance across three comprehensive benchmarks: fev-bench, GIFT-Eval, and Chronos Benchmark II. On fev-bench, which emphasizes multivariate and covariate-informed forecasting, Chronos-2’s universal ICL capabilities lead to substantial improvements over existing models. On tasks involving covariates, it consistently outperforms baselines by a wide margin. Case studies in the energy and retail domains further highlight its practical advantages. The in-context learning capabilities of Chronos-2 establish it as a general-purpose forecasting model that can be used “as is” in real-world forecasting pipelines.

1 Introduction

The advent of pretrained models (also referred to as *foundation models*) has led to a paradigm shift in time series forecasting. Instead of training a model for each time series (*local models*) (Hyndman & Athanasopoulos, 2018) or dataset (*task-specific models*) (Lim et al., 2021; Challu et al., 2023), a single model can be trained once on large-scale time series data and then applied across different forecasting problems (Ansari et al., 2024; Das et al., 2024b). Pretrained models greatly simplify the forecasting pipeline by eliminating the need for training from scratch for each use case. More remarkably, they often match or exceed the forecast accuracy of task-specific models (Aksu et al., 2024).

Despite these advances, a fundamental limitation persists: most pretrained models operate only on univariate data, considering solely the historical observations of a single time series to generate forecasts. Although univariate forecasting is important, the class of real-world forecasting tasks spans far beyond it. In practice, one may encounter tasks where multiple co-evolving time series need to be predicted simultaneously (*multivariate forecasting*) (Bańbura et al., 2010; Cohen et al., 2025) or where forecasts depend on various external factors (*covariate-informed forecasting*). For example, cloud infrastructure metrics such as CPU usage, memory consumption, and storage I/O evolve together and benefit from joint modeling (Cohen et al., 2025). Likewise, retail demand is heavily influenced by promotional

*Equal contribution.

†Jaris Küken and Andreas Auer contributed to this work during their internships at AWS. Hao Wang and Pablo Guerron hold concurrent appointments at Amazon and their corresponding universities, and this report describes work performed at Amazon.

‡Equal advisory contribution.

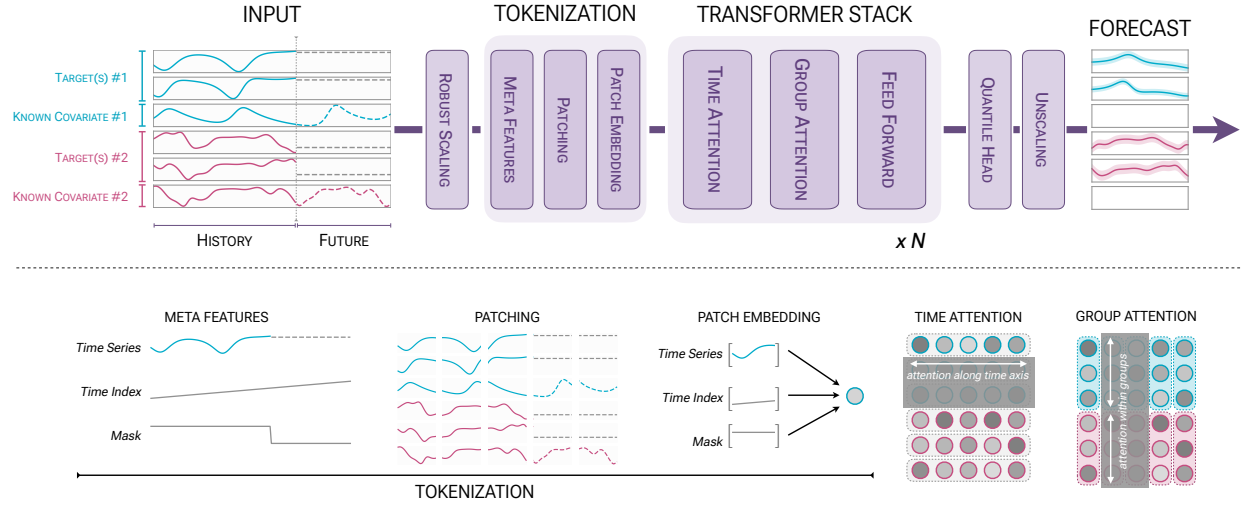


Figure 1: **The complete Chronos-2 pipeline.** Input time series (targets and covariates) are first normalized using a robust scaling scheme, after which time index and mask meta features are added. The resulting sequences are split into non-overlapping patches and mapped to high-dimensional embeddings via a residual network. The core transformer stack operates on these patch embeddings and produces multi-patch quantile outputs corresponding to the masked future patches provided as input. Each transformer block alternates between time and group attention layers: the time attention layer aggregates information across patches within a single time series, while the group attention layer aggregates information across all series within a group at each patch index. A group is a flexible notion of relatedness and may correspond to a single time series, multiple series sharing a source or metadata, variates of a multivariate series, or targets along with associated covariates. The figure illustrates two multivariate time series with one known covariate each, with corresponding groups highlighted in blue and red. This example is for illustration only; Chronos-2 supports arbitrary numbers of targets and optional covariates.

activities, while energy consumption patterns are driven by weather conditions (Petropoulos et al., 2022). The lack of multivariate and covariate-informed forecasting capabilities hinders the widespread adoption of pretrained models in real-world production systems.

Developing *universal* pretrained models that can handle both multivariate dependencies and covariates remains challenging due to two factors. First, the heterogeneity of forecasting problems requires rethinking the model architecture. Each downstream task differs in the number of dimensions and their semantics. Since it is impossible to know a priori how the variables will interact in an unseen task, the model must infer these interactions from the available context. Second, high-quality pretraining data with multivariate dependencies and informative covariates is scarce.

In this work, we present Chronos-2, a pretrained model designed to handle arbitrary forecasting tasks — univariate, multivariate, and covariate-informed — in a *zero-shot* manner. Chronos-2 leverages in-context learning (ICL) to support multivariate forecasting and arbitrary covariates, whether past-only or with known future values, real-valued or categorical. Its enhanced ICL capabilities also improve univariate forecasting by enabling *cross learning*, where the model shares information across univariate time series in the batch, leading to more accurate predictions.

At the core of Chronos-2’s ICL capabilities is the *group attention* mechanism. It enables information exchange within groups of time series, which may represent arbitrary sets of related series, variates of a multivariate series, or targets and covariates (both past-only and known) in a forecasting task. Rather than extending the context by concatenating targets and covariates, the group attention layer shares information within groups across the batch axis, allowing it to scale gracefully with the number of variates. A key innovation of Chronos-2 lies in our training approach: to enable its ICL capabilities, we rely on synthetic time series data generated by imposing multivariate structure on time series sampled from base univariate generators. The complete inference pipeline of Chronos-2 including tokenization and modeling is shown in Figure 1.

Empirical evaluation on comprehensive forecasting benchmarks, including fev-bench (Shchur et al., 2025), GIFT-Eval (Aksu et al., 2024), and Chronos Benchmark II (Ansari et al., 2024), shows that Chronos-2 achieves state-of-the-art performance. On fev-bench, which spans a wide range of forecasting tasks — univariate, multivariate,

Model	Univariate Forecasting	Multivariate Forecasting	Past-Only Covariates	Known Covariates	Categorical Covariates	Cross Learning	Memory Scaling
Chronos-2	✓	✓	✓	✓	✓	✓	$\mathcal{O}(V)$
Toto-1.0	✓	✓	✓	✗	✗	✗	$\mathcal{O}(V)$
TabPFN-TS	✓	✗	✗	✓	✓	✗	$\mathcal{O}(V)$
COSMIC	✓	✗	✓	✓	✗	✗	$\mathcal{O}(V^2)$
Moirai-1.0	✓	✓	✓	✓	✗	✗	$\mathcal{O}(V^2)$
Chronos-Bolt	✓	✗	✗	✗	✗	✗	-
Moirai-2.0	✓	✗	✗	✗	✗	✗	-
Sundial	✓	✗	✗	✗	✗	✗	-
TimesFM-2.5	✓	✗	✗	✗	✗	✗	-
TiRex	✓	✗	✗	✗	✗	✗	-

Table 1: Comparison of capabilities of pretrained forecasting models. *Past-Only Covariates*: support for covariates only observed in the past; *Known Covariates*: support for covariates whose future values are known; *Categorical Covariates*: support for nominal features in the covariates; *Cross Learning*: support for in-context learning across related time series; *Memory Scaling*: inference memory requirements with respect to the total number of variates V (including both targets and covariates).

and covariate-informed – Chronos-2 outperforms baselines across all categories. The largest gains are observed on covariate-informed tasks, demonstrating Chronos-2’s strength in this practically important setting. Chronos-2 offers these new capabilities while maintaining high computational efficiency, running on a single mid-range GPU (NVIDIA A10G) with a throughput of 300 time series per second.¹

The rest of the technical report is organized as follows. Section 2 introduces the background on time series forecasting and existing forecasting methods with a special focus on pretrained models. In Section 3, we describe the architecture of Chronos-2 and discuss its training and inference pipelines. Section 4 briefly discusses the training corpus of Chronos-2. In Section 5, we present our main results on three forecasting benchmarks, case studies on energy and retail domains, and ablations. We conclude the report and discuss potential future work in Section 6.

2 Background and Related Work

Time series forecasting aims to predict future values of a temporal sequence given historical observations. Formally, let $\mathbf{Y}_{1:T} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$ denote a historical time series of length T , where each observation $\mathbf{y}_t \in \mathbb{R}^D$ can either be univariate ($D = 1$) or multivariate ($D > 1$). Given this historical context, the goal is to predict the next H time steps $\mathbf{Y}_{T+1:T+H}$, where H defines the forecast horizon. Forecasts may be supported by covariates (also known as *exogenous variables*) $\mathbf{X}_{1:T+H} = [\mathbf{x}_1, \dots, \mathbf{x}_{T+H}]$, where $\mathbf{x}_t \in \mathbb{R}^M$ represents additional information that can span both historical ($t \leq T$) and future ($t > T$) time steps. The task itself can be defined as either *point forecasting*, where the objective is to predict a single future value at each time step, or *probabilistic forecasting*, where the objective is to estimate the conditional distribution $\mathcal{P}(\mathbf{Y}_{T+1:T+H} \mid \mathbf{Y}_{1:T}, \mathbf{X}_{1:T+H})$ in order to capture forecast uncertainty. *Zero-shot forecasting* refers to the setting in which a model generates forecasts for a previously unseen time series datasets without requiring any additional training, adaptation, or fine-tuning.

Forecasting methods preceding the pretrained model paradigm can be broadly divided into local and global models. Local models fit one set of parameters for each time series in the dataset. These include classical approaches such as ARIMA, Exponential Smoothing (Hyndman & Athanasopoulos, 2018), and Theta (Assimakopoulos & Nikolopoulos, 2000). In contrast, global models share their parameters across all time series within a specific dataset. Deep learning approaches in this category have become increasingly common over the last decade. Notable examples of global models include recurrent neural networks (RNN) like DeepState (Rangapuram et al., 2018), DeepAR (Salinas et al., 2020), and TimeGrad (Rasul et al., 2021); stacked architectures such as N-BEATS (Oreshkin et al., 2020) and N-HITS (Challu et al., 2023); and transformer-based architectures like TFT (Lim et al., 2021) and PatchTST (Nie et al., 2023).

Pretrained forecasting models have recently emerged as a new paradigm in time series forecasting. While earlier work already demonstrated limited transfer learning capabilities for forecasting (Orozco & Roberts, 2020; Oreshkin et al.,

¹Based on inference time for a batch of 1,024 time series with a context length of 2048 and prediction length of 64 times steps.

2021; Jin et al., 2022; Nie et al., 2023), pretrained models adopt principles similar to large language models (LLMs) and enable zero-shot generalization on diverse datasets. Initial attempts focused on directly adapting language models to time series tasks (Gruver et al., 2023; Jin et al., 2024), whereas more recent approaches primarily borrow architectural concepts from LLMs but pretrain them on time series data (Das et al., 2024b; Garza et al., 2024; Ansari et al., 2024).

The majority of pretrained models are limited to univariate forecasting (Rasul et al., 2023; Das et al., 2024b; Ansari et al., 2024; Liu et al., 2025; Auer et al., 2025b), treating each dimension independently in multivariate scenarios and ignoring covariates. Notable exceptions include Moirai-1 (Woo et al., 2024) and Toto (Cohen et al., 2025), which incorporate multivariate structure into their architectures. Moirai-1 supports multivariate inputs but flattens them internally, which limits scalability to high-dimensional cases. Toto introduces a cross-variate attention mechanism but does not support known or categorical covariates. COSMIC (Auer et al., 2025a) advances covariate utilization through synthetic augmentations but remains restricted to univariate targets. TabPFN-TS (Hoo et al., 2025), a tabular foundation model adapted for time series, can incorporate known covariates but it does not model past-only covariates or multivariate targets. Despite these advances, empirical analyses show that most approaches provide only marginal benefits over univariate models (Żukowska et al., 2024; Auer et al., 2025a), indicating that jointly modeling multiple variates and integrating covariates effectively in a zero-shot setting remains an open challenge.

Our approach addresses this gap with a *group attention* mechanism, which generalizes ideas from cross-attention architectures for multivariate forecasting (Zhang & Yan, 2023; Rao et al., 2021; Arnab et al., 2021) and cross-learning across multiple univariate series (Das et al., 2024a). Unlike prior approaches, group attention operates over groups of related time series and naturally accommodates diverse forecasting setups, including univariate, multivariate, and covariate-informed tasks, within a unified framework without requiring architectural changes or task-specific adaptations. Table 1 compares the capabilities of Chronos-2 with those of existing pretrained models.

3 The Chronos-2 Model

In this section, we introduce the Chronos-2 model. We begin with scaling and tokenization, followed by the model’s architecture including the group attention mechanism which enables Chronos-2’s in-context learning capabilities. Subsequently, we discuss the training and inference pipelines of Chronos-2. The complete inference pipeline of Chronos-2 is visualized in Figure 1.

3.1 Scaling and Tokenization

Input Construction. The model operates on two inputs derived from the target $\mathbf{Y}_{1:T}$ and covariates $\mathbf{X}_{1:T+H}$. We concatenate all historical values into $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_T]$, where each $\mathbf{v}_t \in \mathbb{R}^{D+M}$ consists of the target observation $\mathbf{y}_t$ and the corresponding covariate vector $\mathbf{x}_t$. Similarly, we define the future values as $\mathbf{W} = [\mathbf{w}_{T+1}, \dots, \mathbf{w}_{T+H}]$, where $\mathbf{w}_t \in \mathbb{R}^{D+M}$ contains known future covariate values $\mathbf{x}_t$ when available, while the entries corresponding to targets and past-only covariates are set to missing values.

Categorical covariates in $\mathbf{X}_{1:T+H}$ are transformed into real-valued representations before being concatenated into $\mathbf{V}$ and $\mathbf{W}$. For univariate targets, we apply target encoding (Pedregosa et al., 2011; Micci-Barreca, 2001), which maps each category to a numerical value based on its relationship with the target. For multivariate targets, the model falls back to ordinal encoding, assigning a unique integer to each category.

Robust Scaling. The input values, $\mathbf{V}$ and $\mathbf{W}$, may be at an arbitrary scale, so our tokenization pipeline begins by normalizing the series. We adopt *standardization*, a widely used normalization method in the literature, and introduce an additional step: applying the $\sinh^{-1}$ transformation to the standardized values. This log-like transformation further stabilizes variance and reduces the influence of outliers on the objective function. It has been used in econometrics (Burbidge et al., 1988) and energy price forecasting (Uniejewski & Weron, 2018) literature for handling extreme values. Formally, each historical value $v_{t,d}$ and the future value $w_{t,d}$ are normalized as

$$\tilde{v}_{t,d} = \sinh^{-1} \left(\frac{v_{t,d} - \mu_d}{\sigma_d} \right) \quad \text{for } t \in \{1, \dots, T\}, \quad (1)$$

$$\tilde{w}_{t,d} = \sinh^{-1} \left(\frac{w_{t,d} - \mu_d}{\sigma_d} \right) \quad \text{for } t \in \{T+1, \dots, T+H\}, \quad (2)$$

where μ_d and σ_d are the mean and standard deviation of the historical values $[v_{1,d}, \dots, v_{T,d}]$, respectively. Any missing values in $\mathbf{V}$ are excluded when computing μ_d and σ_d . The normalized historical values $\tilde{\mathbf{V}}$ and future values $\tilde{\mathbf{W}}$ are concatenated to construct the input matrix $\mathbf{U} = [\tilde{\mathbf{V}}, \tilde{\mathbf{W}}] \in \mathbb{R}^{(T+H) \times (D+M)}$.

Meta Features. During tokenization, each dimension of $\mathbf{U}$ is processed independently by the model. To describe the tokenization procedure, consider a single column $\mathbf{u}_d = [u_{1,d}, \dots, u_{T+H,d}]^\top$ corresponding to one target or covariate dimension d . Two additional meta features are appended to each column: a time index and a mask. The time index $\mathbf{j} = [-\frac{T}{C}, -\frac{T-1}{C}, \dots, 0, \dots, \frac{H-1}{C}]$ encodes the relative position of each time step, where C is the maximum context length supported by the model. It provides explicit information about temporal ordering to the model which is beneficial when using patch-based inputs. The mask $\mathbf{m}_d$ is a binary indicator equal to 1 when the value is observed, and 0 otherwise. It serves two purposes: indicating which values are missing in the historical context and specifying which input dimensions correspond to future-known covariates. After construction of the mask, all missing values in $\mathbf{u}_d$ are replaced with zeros.

Patching and Embedding. The input $\mathbf{u}_d$ with the corresponding meta features, $\mathbf{j}$ and $\mathbf{m}_d$, are split into non-overlapping patches of length P (Nie et al., 2023). The context and future sections of the time series and meta features are split into patches separately. When T and H are not multiples of P , zero padding is applied on the left (context) or right (future). Let $\bar{\mathbf{u}}_p$, $\bar{\mathbf{j}}_p$, and $\bar{\mathbf{m}}_p$ denote the p -th patches of the input, time index, and mask, respectively. These are concatenated and mapped into the embedding space using a residual network, $f_\phi^{\text{in}} : \mathbb{R}^{3P} \rightarrow \mathbb{R}^{D_{\text{model}}}$,

$$\mathbf{h}_p = f_\phi^{\text{in}}([\bar{\mathbf{u}}_p, \bar{\mathbf{j}}_p, \bar{\mathbf{m}}_p]), \quad (3)$$

where ϕ denotes parameters of the residual network and D_{model} is the hidden dimension of the transformer model. Between the patch embeddings of the context and future, we include a special REG token which serves both as a separator token and an attention sink (Xiao et al., 2024).

3.2 Architecture

Chronos-2 is an encoder-only transformer (Vaswani et al., 2017) model which closely follows the design of the T5 encoder (Raffel et al., 2020). In the following, we discuss the key architectural components of Chronos-2.

Time Attention. The time attention layer is the usual attention layer found in typical sequence models. It applies self-attention along the temporal axis and aggregates information across patches of the same input dimension. We replace relative position embeddings used in the self-attention layers of the original T5 model with rotary position embeddings (RoPE) (Su et al., 2024) which have become the de-facto standard for position embeddings in modern transformer-based models (Touvron et al., 2023).

Group Attention. We introduce a *group attention* layer into the transformer stack, which is central to enabling the in-context learning capabilities of Chronos-2. This layer aggregates information across time series that belong to the same group at a given patch index. A group refers to a set of related time series and may refer to different things depending on the forecasting task. For example, a group may consist of:

- *a single time series*: the minimal grouping where the model makes univariate predictions without referring to other time series in the batch.
- *a set of time series with shared source or metadata*: this grouping enables the model to perform cross learning across items by making joint predictions for related time series (also referred to as *few-shot learning*) instead of generating univariate forecasts by solely taking the histories of individual time series into account. Sharing information between related time series could be especially helpful when all or some (cold start scenario) time series have short histories and when the characteristics of the downstream dataset differ considerably from the training data distribution.
- *a set of variates with shared dynamics*: this grouping enables multivariate forecasting where the model jointly predicts all variates with shared dynamics.
- *a set of target(s), past-only covariates and known covariates*: the most general case where the model forecasts targets while taking covariates into account.

Within a batch of size B , multiple groups of varying sizes are possible, each identified by group IDs $\mathbf{g}$, a vector of length B . Internally, the group attention layer maps these IDs to a two-dimensional attention mask, ensuring that aggregation occurs only within groups and not across them. Since time series within a group lack a natural ordering, the group attention layer omits positional embeddings.

Quantile Head. After a sequence of alternating time and group attention layers, the embeddings of future patches of the D target dimensions are passed through a residual block to produce the direct multi-step quantile forecast $\hat{\mathbf{Z}} \in \mathbb{R}^{H \times D \times |\mathcal{Q}|}$. By producing forecasts for multiple target patches within a single forward pass, the model can efficiently generate predictions over long forecast horizons. Chronos-2 predicts a set of 21 quantiles $\mathcal{Q} = \{0.01, 0.05, 0.1, \dots, 0.9, 0.95, 0.99\}$. This results in a richer representation of the predictive distribution compared to the 9-quantile grid $\{0.1, 0.2, \dots, 0.9\}$ commonly used in existing pretrained models. The inclusion of extreme quantiles (0.01 and 0.99) improves coverage of rare events and enhances the model’s applicability to tasks such as anomaly detection and risk-aware forecasting.

3.3 Training

During training, batches are constructed to include heterogeneous forecasting tasks: univariate forecasting, multivariate forecasting (which also covers tasks with past-only covariates), and multivariate forecasting with known covariates. Each task is characterized by the number of target dimensions D , the number of covariates M , and the role of each dimension (target, past-only covariate, or known covariate). A unique group ID is assigned to each task, and the combination of group IDs $\mathbf{g}$ with whether the future input $\mathbf{W}$ is observed allows the model to infer the specific forecasting setup.

The model is trained using the quantile regression objective

$$\sum_{q \in \mathcal{Q}} \left(q \cdot \max(z - \hat{z}^q, 0) + (1 - q) \cdot \max(\hat{z}^q - z, 0) \right), \quad (4)$$

where $\hat{z}^q$ is the forecast at quantile level q , and z is the corresponding target value normalized as in Eq. (1). The loss is averaged over all forecast steps and items in the batch and is computed only on target dimensions, with entries corresponding to known covariates or missing target values excluded from the objective. The number of output patches is randomly sampled for each batch during training.

Training proceeds in two stages. First, the model is pretrained with a maximum context length of 2048 and a low number of maximum output patches. In the second stage, the context length is extended to 8192, and the maximum number of sampled output patches is increased. Longer contexts enable the model to capture long-term seasonalities in high-frequency time series, while multi-patch outputs allow for long-horizon forecasts without relying on heuristics.

3.4 Inference

Forecasts are generated by de-normalizing the model predictions $\hat{z}_{t,d}^q$ and inverting Eq. (1). Formally, the quantile head output $\hat{z}_{t,d}^q$ is transformed as

$$\hat{y}_{t,d}^q = \mu_d + \sigma_d \cdot \sinh(\hat{z}_{t,d}^q), \quad (5)$$

to obtain the prediction $\hat{y}_{t,d}^q$ of the quantile level q at time step t along the target dimension d .

During inference, multiple time series in a batch can be grouped to solve different forecasting tasks:

- *univariate forecasting*: each item in the batch is assigned a unique group ID. This ensures that the model makes independent predictions for each time series in the batch.
- *multivariate forecasting*: each variate which belongs to the same multivariate series is assigned the same group ID with variates from different multivariate series having distinct group IDs. This allows the model to share dynamics information between different variates of a multivariate time series.
- *forecasting with covariates*: all target(s), past-only and known covariates belonging to the same task are assigned the same group ID. The future inputs $\mathbf{W}$ corresponding to known covariates contain their known future values. The predictions generated by the model for covariates are ignored.

Task Type	Group IDs g	Future Inputs W
Univariate Forecasting (3 independent series)	$g = (1, 2, 3)$	$W = \begin{bmatrix} * & \dots & * \\ * & \dots & * \\ * & \dots & * \end{bmatrix} \in \mathbb{R}^{3 \times H}$
Multivariate Forecasting (3 targets)	$g = (1, 1, 1)$	$W = \begin{bmatrix} * & \dots & * \\ * & \dots & * \\ * & \dots & * \end{bmatrix} \in \mathbb{R}^{3 \times H}$
Forecasting with Covariates (1 target, 1 past-only covariate, 2 known covariates)	$g = (1, 1, 1, 1)$	$W = \begin{bmatrix} * & \dots & * \\ * & \dots & * \\ x_{T+1,3} & \dots & x_{T+H,3} \\ x_{T+1,4} & \dots & x_{T+H,4} \end{bmatrix} \in \mathbb{R}^{4 \times H}$

Table 2: Diverse forecasting tasks can be solved by specifying group IDs and future inputs appropriately. Here, g and W denote the group IDs and future values provided to the model. Future inputs for targets and past-only covariates are masked as missing values, denoted as *. The examples use fixed numbers of variates for clarity, but Chronos-2 can handle arbitrary dimensions.

Table 2 summarizes how group IDs and future inputs must be specified to solve different forecasting tasks. In addition to these, Chronos-2 can also be used in the *full cross learning* mode where each item in the batch is assigned the same group ID regardless of whether the item is a target, a past-only covariate or a known covariate. Since each item belongs to the same group, the model shares information across items in the batch and makes joint predictions for the entire batch.

4 Training Data

For a generalist pretrained model such as Chronos-2, the training data often plays a more decisive role than the model’s specific architecture. Although recent efforts have expanded the availability of large-scale time series datasets (Woo et al., 2024; Ansari et al., 2024; Aksu et al., 2024), they primarily contain univariate data. To overcome this limitation and endow Chronos-2 with in-context learning capabilities, we relied extensively on synthetic data.

4.1 Univariate Data

We incorporated select datasets from the Chronos (Ansari et al., 2024) and GIFT-Eval (Aksu et al., 2024) pretraining corpora into Chronos-2’s training corpus. The full list of datasets is provided in Table 6 (Appendix). To further enhance data diversity, we generated synthetic data using two approaches:

- **TSI (Trend, Seasonality, and Irregularity)**: based on Bahrpeyma et al. (2021), this generator produces diverse synthetic series by randomly constructing and combining different trend, seasonality, and irregularity components.
- **TCM (Temporal Causal Model)**: this generator samples random causal graphs from a temporal causal model (Runge et al., 2023), from which time series are generated via autoregression.

4.2 Multivariate Data

For multivariate and covariate-informed tasks, we relied entirely on synthetic data. To enable a broad class of multivariate structures, we introduce the concept of *multivariatizers*. A multivariatizer samples multiple time series from base univariate generators and imposes dependencies among them to create multivariate dynamics. As base univariate generators, we employed a diverse set including autoregressive (AR) models, exponential smoothing (ETS) models, TSI, and KernelSynth (Ansari et al., 2024).

We used two broad classes of multivariatizers:

- *Cotemporaneous multivariatizers* apply linear or nonlinear transformations at the same time step across time series sampled from the base univariate generators. This introduces instantaneous correlations between the time series resulting in a multivariate time series.
- *Sequential multivariatizers* induce dependencies across time, generating richer multivariate properties such as lead-lag effects and cointegration.

The multivariate time series generated from the multivariatizers were used to construct both multivariate tasks (where all variates must be predicted) and covariate-informed tasks, where a subset of variates was randomly designated as known covariates.

5 Experiments

In this section, we present empirical results, beginning with an evaluation of Chronos-2 against state-of-the-art approaches across three comprehensive benchmarks (Section 5.1). We then demonstrate the gains achieved through in-context learning on univariate, multivariate, and covariate-informed forecasting tasks (Section 5.2). Next, we examine Chronos-2’s performance on tasks from the energy and retail domains, where covariates are often important for accurate forecasting (Section 5.3). Finally, we report results for ablated variants of Chronos-2 (Section 5.4), including a smaller model, a version trained only on synthetic data, and the model prior to long-context post-training.

5.1 Benchmark Results

Model	Avg. Win Rate (%)	Skill Score (%)	Median runtime (s)	Leakage (%)	#Failures
Chronos-2	90.7	47.3	3.6	0	0
TiRex	80.8	42.6	1.4	1	0
TimesFM-2.5	75.9	42.3	16.9	8	0
Toto-1.0	66.6	40.7	90.7	8	0
COSMIC	65.6	39.0	34.4	0	0
Moirai-2.0	61.1	39.3	2.5	28	0
Chronos-Bolt	60.3	38.9	1.0	0	0
TabPFN-TS	59.3	39.6	305.5	0	2
Sundial	41.0	33.4	35.6	1	0
Stat. Ensemble	40.4	20.2	690.6	0	11
AutoARIMA	35.2	20.6	186.8	0	10
AutoETS	29.1	-26.8	17.0	0	3
AutoTheta	21.8	5.5	9.3	0	0
SeasonalNaive	14.5	0.0	2.3	0	0
Naive	7.8	-45.4	2.2	0	0

Table 3: **fev-bench results.** The average win rate and skill score are computed with respect to the scaled quantile loss (SQL) metric. Higher values are better for both. Chronos-2 outperforms all existing pretrained models by a substantial margin on this benchmark that includes univariate, multivariate, and covariate-informed forecasting tasks. Baseline results and the imputation strategy for handling data leakage in certain tasks are both taken from Shchur et al. (2025). Results for additional forecasting metrics are provided in Tables 7 to 9 (Appendix).

We evaluated the *base* Chronos-2 model with 120M parameters on three comprehensive forecasting benchmarks: fev-bench (Shchur et al., 2025), GIFT-Eval (Aksu et al., 2024), and Chronos Benchmark II (Ansari et al., 2024). To contextualize its performance, we compared it against state-of-the-art time series foundation models that achieved the strongest results on these benchmarks. These include TiRex (Auer et al., 2025b), TimesFM-2.5 (Das et al., 2024b), Toto-1.0 (Cohen et al., 2025), Moirai-2.0 (Woo et al., 2024), TabPFN-TS (Hoo et al., 2025), COSMIC (Auer et al., 2025a), Sundial (Liu et al., 2025), and Chronos-Bolt (Ansari et al., 2024), the latest publicly released version of Chronos. As additional baselines, we also included AutoARIMA, AutoETS, AutoTheta, and their ensemble (Petroopoulos & Svetunkov, 2020), representing well-established methods from the statistical forecasting literature (Hyndman & Athanasopoulos, 2018). We compare Chronos-2 only with the aforementioned models and exclude task-specific deep learning models from our evaluation, as prior studies (Aksu et al., 2024; Ansari et al., 2024) — which include

GIFT-Eval and Chronos Benchmark II, two of the three benchmarks considered in our work — have shown that pretrained models perform comparably to or better than task-specific models on average.

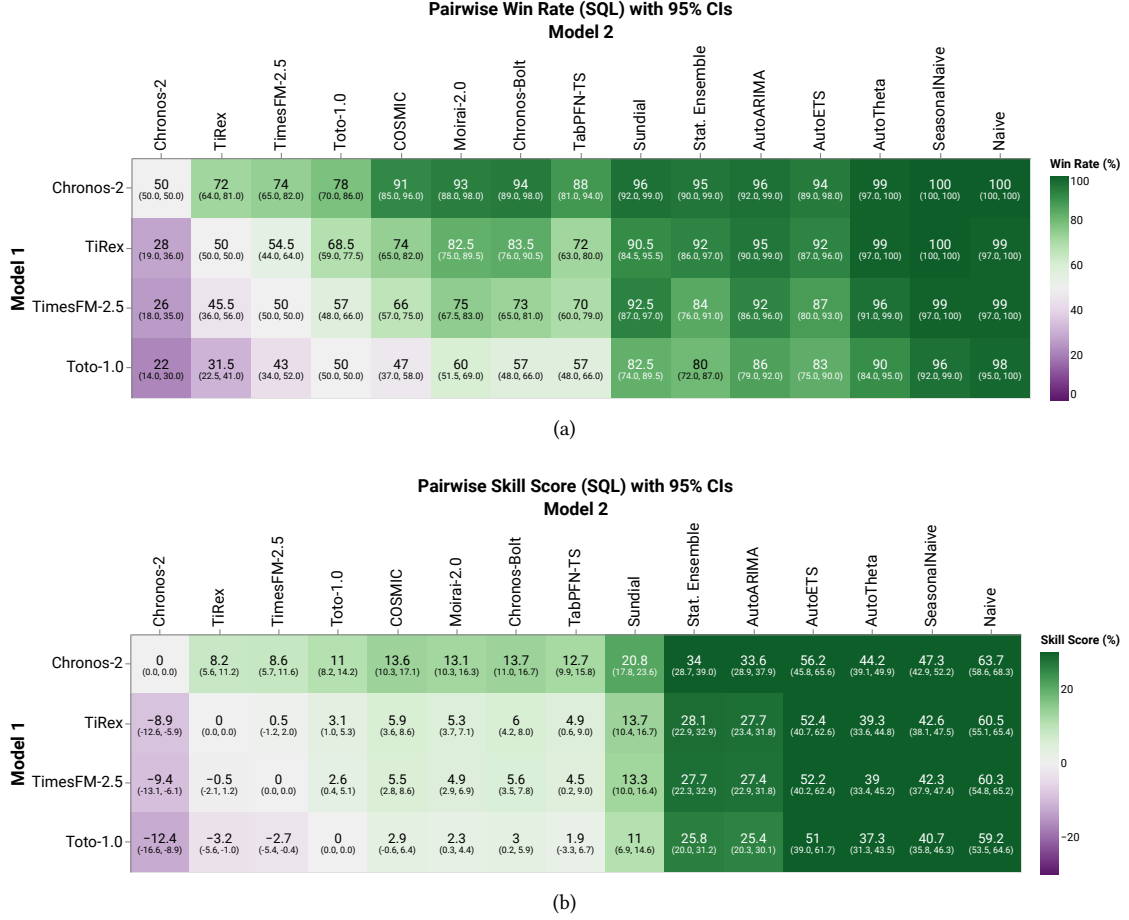


Figure 2: The pairwise win rates (a) and skill scores (b) of the top-4 pretrained models on fev-bench with 95% confidence intervals (CIs) obtained through bootstrapping. Chronos-2 outperforms the next best models (TiRex and TimesFM) by a statistically significant margin on both metrics. The complete plot and results for other forecasting metrics can be found in Figures 12 to 19 (Appendix).

Following Shchur et al. (2025), we report both average win rates (W) and skill scores (S) for all models. These metrics are mathematically equivalent to the average rank (R) and *geometric mean relative error* (G) metrics used in prior work (Ansari et al., 2024; Aksu et al., 2024). Specifically, $R = 1 + (1 - \frac{W}{100})(N - 1)$ and $G = 1 - \frac{S}{100}$, where N is the number of evaluated models. However, win rates and skill scores provide more interpretable summaries. The win rate measures the proportion of pairwise comparisons in which a model outperforms other models, while the skill score reflects the average percentage improvement over a baseline — in our case, the Seasonal Naive model. For a detailed discussion, we refer the reader to Shchur et al. (2025).

fev-bench. This benchmark consists of 100 forecasting tasks and offers the most comprehensive coverage of diverse real-world scenarios, including tasks with covariates. None of these datasets or tasks were seen by Chronos-2 during training. Table 3 reports results on fev-bench with respect to the scaled quantile loss (SQL) metric which evaluates the probabilistic forecasting performance. Chronos-2 outperforms existing time series foundation models by a significant margin, both in win rate and skill score. fev-bench also provides tooling to answer questions like: “Does Model A outperform Model B in a statistically significant way?”. These pairwise comparisons with 95% confidence intervals (CIs), shown in Figure 2, further confirm that Chronos-2 surpasses the next best models (TiRex and TimesFM-2.5) by a statistically significant margin. Specifically, the CIs of the pairwise win rates and skill scores of Chronos-2 against any baseline do not include 50% and 0%, respectively.

Model	Avg. Win Rate (%)	Skill Score (%)	Model	Avg. Win Rate (%)	Skill Score (%)
Chronos-2	81.9	51.4	Chronos-2	83.8	30.2
TimesFM-2.5	77.5	51.0	TimesFM-2.5	77.7	29.5
TiRex	76.5	50.2	TiRex	71.9	27.6
Toto-1.0	67.4	48.6	Moirai-2.0	64.3	27.2
Moirai-2.0	64.4	48.4	Toto-1.0	61.3	25.2
COSMIC	56.4	44.5	Chronos-Bolt	58.4	19.2
Chronos-Bolt	53.8	42.6	Sundial	53.4	25.0
TabPFN-TS	53.5	43.1	COSMIC	51.9	20.8
Sundial	49.1	44.1	TabPFN-TS	45.4	16.6
AutoARIMA	21.8	8.8	AutoARIMA	24.4	-7.4
Seasonal Naive	16.6	0.0	AutoETS	19.5	-21.2
AutoTheta	16.0	-24.4	Seasonal Naive	19.4	0.0
AutoETS	15.2	-648.9	AutoTheta	18.5	-9.0

(a)
(b)

Table 4: **GIFT-Eval results.** The average win rate and skill score with respect to the (a) weighted quantile loss (WQL) and (b) mean absolute scaled error (MASE) metrics. Higher values are better for both. Chronos-2 outperforms previous best models, TimesFM-2.5 and TiRex. Baseline results have been taken from the GIFT-Eval leaderboard (Aksu et al., 2024).

GIFT-Eval. The GIFT-Eval benchmark comprises 97 tasks derived from 55 datasets, with a particular emphasis on high-frequency time series and long-horizon forecasting. The results in Table 4 show that Chronos-2 surpasses the previously leading models (TiRex and TimesFM-2.5) in win rate and skill score under both the weighted quantile loss (WQL) and mean absolute scaled error (MASE) metrics. When constructing the pretraining corpus for Chronos-2, we carefully ensured that it did not overlap with the test portions of any GIFT-Eval task at any sampling frequency. Nonetheless, the corpus does include partial overlap with the training portions of some GIFT-Eval datasets. For strictly zero-shot results, we refer the reader to Section 5.4, where we evaluate a variant of Chronos-2 trained exclusively on synthetic data.

Model	Avg. Win Rate (%)	Skill Score (%)	Model	Avg. Win Rate (%)	Skill Score (%)
Chronos-2	79.8	46.6	Chronos-2	81.5	26.5
TiRex	70.4	41.7	TimesFM-2.5	71.6	23.3
TimesFM-2.5	70.0	42.4	TiRex	67.1	22.2
Toto-1.0	60.9	41.9	Toto-1.0	58.0	22.3
Moirai-2.0	56.0	40.9	Moirai-2.0	53.5	19.8
Chronos-Bolt	49.4	39.3	Chronos-Bolt	50.6	20.4
TabPFN-TS	46.3	32.6	COSMIC	42.0	18.1
COSMIC	42.8	36.7	TabPFN-TS	40.1	10.5
Sundial	14.4	24.1	Sundial	21.8	9.5
Seasonal Naive	10.1	0.0	Seasonal Naive	13.8	0.0

(a)
(b)

Table 5: **Chronos Benchmark II results.** The average win rate and skill score with respect to the (a) weighted quantile loss (WQL) and (b) mean absolute scaled error (MASE) metrics. Higher values are better for both. Chronos-2 achieves the best results across all metrics.

Chronos Benchmark II. Originally proposed in Ansari et al. (2024) to evaluate the first Chronos models, this benchmark comprises 27 tasks, the majority of which involve short histories (fewer than 300 time steps on average). None of these datasets were included in the training corpus of Chronos-2. On this benchmark, Chronos-2 consistently outperforms existing models in terms of the win rate and skill score under both probabilistic (WQL) and point (MASE) forecasting metrics, as shown in Table 5.

Taken together, these results show that Chronos-2 not only outperforms all competing models across the three benchmarks but also substantially improves over Chronos-Bolt, its predecessor, highlighting the impact of the architectural and training improvements in Chronos-2.

5.2 Improvements with In-context Learning

The results in Section 5.1 correspond to Chronos-2 with in-context learning (ICL) enabled, specifically in the *full cross learning* mode described in Section 3.4. In this section, we disentangle the gains from ICL compared to univariate inference. To this end, we split fev-bench into three subsets: the *univariate subset* with 32 tasks involving a single target time series without covariates, the *multivariate subset* with 26 tasks containing multiple targets but no covariates, and the *covariates subset* with 42 tasks that include at least one past-only or known covariate. We compare Chronos-2 with ICL to its univariate inference mode on these three subsets, as well as on GIFT-Eval and Chronos Benchmark II. In the univariate mode, each time series in the batch is forecast independently, and covariates, if present, are ignored.

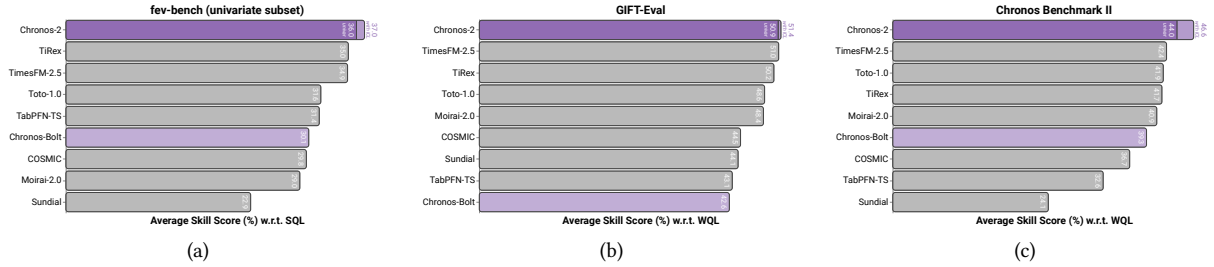


Figure 3: Chronos-2’s probabilistic forecasting results in univariate mode and the corresponding improvements from in-context learning (ICL), shown as stacked bars on (a) the univariate subset of fev-bench, (b) GIFT-Eval, and (c) Chronos Benchmark II. For these univariate benchmarks, ICL enables cross-learning, allowing the model to share information across items within a batch and thereby generate more accurate forecasts than univariate inference alone. Results for point forecasting metrics are available in Figure 9 (Appendix).

Univariate Tasks. ICL provides improvements in skill score on univariate tasks, as shown in Figure 3. The effect is especially strong on Chronos Benchmark II (Figure 3 (b)), which contains many tasks with short contexts. This demonstrates that Chronos-2 can leverage information from related time series to improve predictions when ICL is enabled, particularly when limited time series history is available.

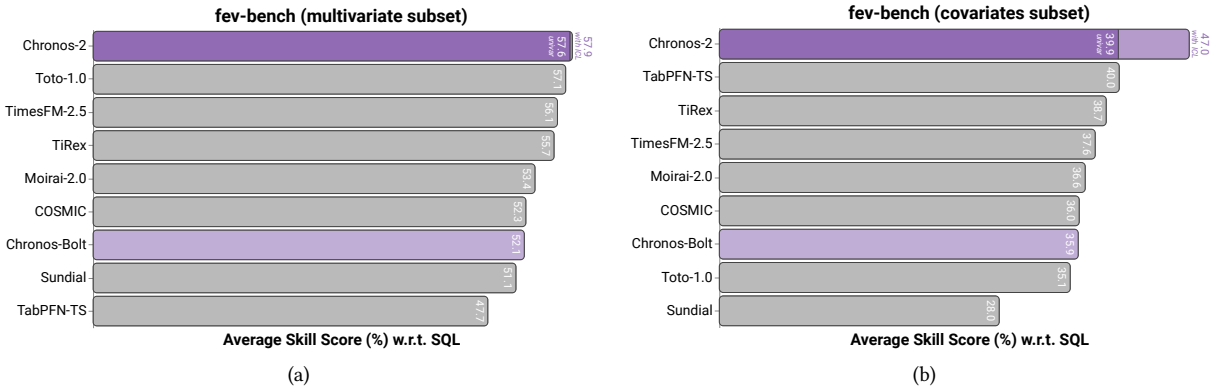


Figure 4: Chronos-2’s probabilistic forecasting results in univariate mode and the corresponding gains from in-context learning (ICL), shown as stacked bars on the multivariate and covariates subsets of fev-bench. On multivariate tasks, ICL provides only modest improvements, though Chronos-2 in univariate mode already surpasses the multivariate Toto-1.0 model. On the covariates subset, however, ICL delivers the largest gains, demonstrating Chronos-2’s ability to effectively use covariates. Besides Chronos-2, only TabPFN-TS and COSMIC support covariates, and Chronos-2 outperforms all baselines (including TabPFN-TS and COSMIC) by a wide margin. Results for point forecasting metrics are available in Figures 10a and 10b (Appendix).

Multivariate Tasks. On the multivariate subset of fev-bench, ICL yields only modest gains over univariate inference (Figure 4a (a)). Interestingly, in univariate mode, Chronos-2 even outperforms Toto-1.0, a model which natively supports multivariate forecasting. This suggests that while these tasks involve multiple variates with potentially shared dynamics, the benefits of explicit multivariate modeling can be limited. One possible intuition comes from Takens’s Embedding Theorem (Takens, 2006), which implies that the dynamics of a system can often

be reconstructed from delayed observations of a single variable. In practice, this means that with sufficiently long histories, a strong univariate model may capture much of the same structure as a multivariate model. Similar empirical findings have been reported elsewhere; for example, [Nie et al. \(2023\)](#) observed that univariate (“channel-independent”) models often perform on par with multivariate (“channel-dependent”) models, albeit on a different benchmark.

Tasks with Covariates. The largest gains with ICL are observed on tasks with covariates (Figure 4a (b)). Here, the performance margin clearly demonstrates that Chronos-2 with ICL can effectively exploit covariates to improve predictions compared to univariate inference, which ignores them. Chronos-2 outperforms baselines by a large margin on this subset. Unsurprisingly, the second spot is taken by TabPFN-TS, another model which supports (known) covariates. These results underscore both the strength of Chronos-2 and the limitations of existing pretrained models, most of which lack covariate support — a capability of immense practical importance.

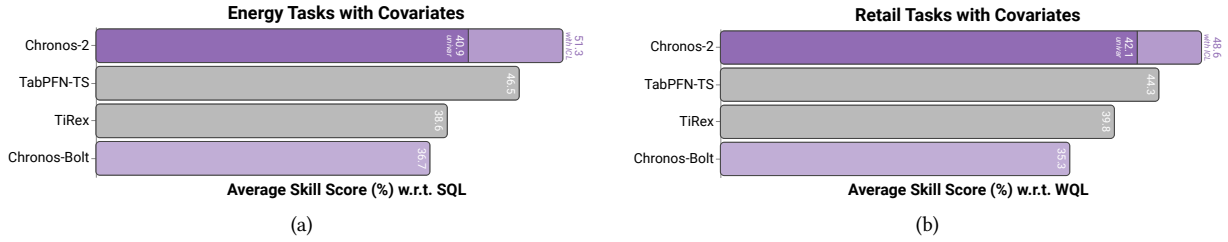


Figure 5: Comparison of Chronos-2 against baselines on tasks which include dynamic covariates from the energy and retail domains. Chronos-2 outperforms all baselines by a wide margin, including TabPFN-TS and TiRex, the strongest baselines on the covariates subset of fev-bench (Figure 4b). For retail, we consider the domain-appropriate WQL metric. Results for point forecasting metrics are available in Figures 11a and 11b (Appendix).

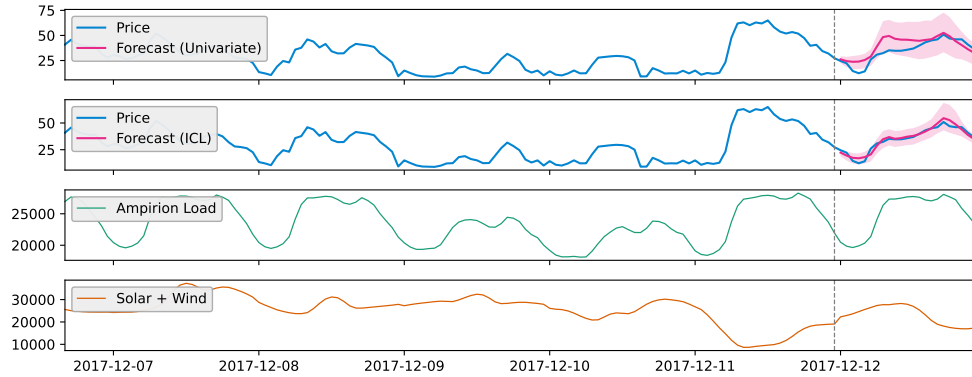


Figure 6: Forecasts generated by Chronos-2 in univariate mode (top), i.e., without covariates, and with in-context learning (second from top) on the energy price forecasting task. The dashed vertical gray line indicates the forecast start date and the shaded region represents 80% prediction interval around the median forecast. With ICL, Chronos-2 leverages Ampirion Load and Solar + Wind covariates to produce a more accurate prediction.

5.3 Domain Case Studies

We conducted further analysis on tasks from the *energy* and *retail* domains, where covariates often provide crucial information for accurate forecasting. For both domains, we selected all tasks with dynamic covariates from fev-bench resulting in 16 and 17 tasks for energy and retail, respectively (see Tables 10 and 11 in the Appendix for details). As baselines, we used TabPFN-TS and TiRex, the two strongest models on the covariates subset of fev-bench, as shown in Figure 4b. The results in Figures 5a and 5b demonstrate that Chronos-2 consistently outperforms these baselines by a wide margin. Incorporating covariates provides a substantial boost in performance for Chronos-2, reinforcing their critical role in real-world forecasting tasks. Consistent with Figure 4b, the second-best results are achieved by TabPFN-TS, another model capable of leveraging covariates.

To illustrate how Chronos-2 with ICL uses covariates, we compared forecasts produced in univariate mode versus with ICL. We selected one task from each domain where ICL delivers the largest gains.

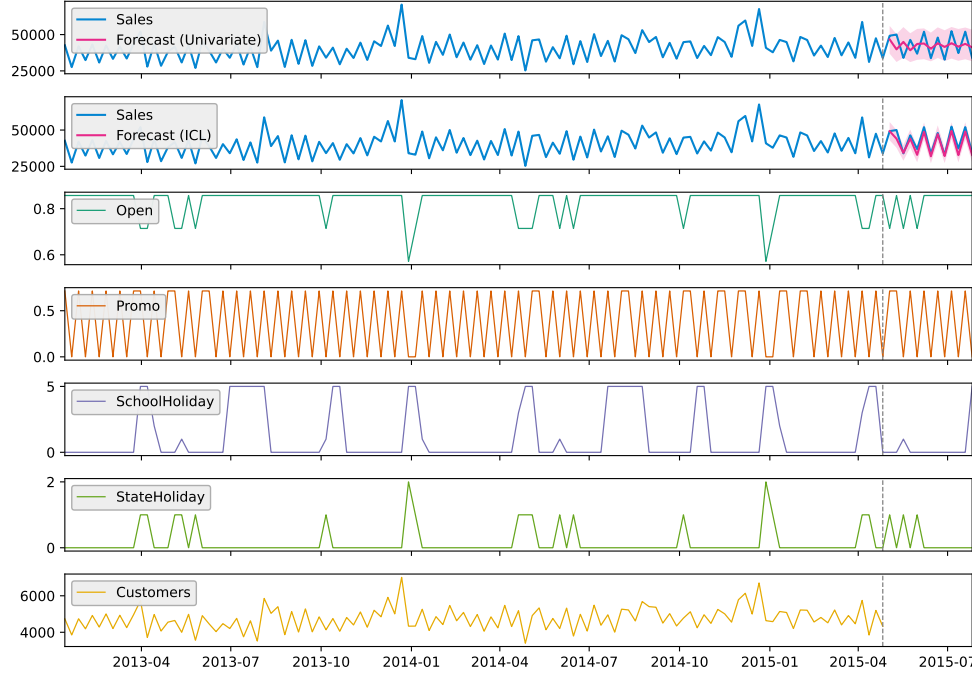


Figure 7: Forecasts generated by Chronos-2 in univariate mode (top), i.e., without covariates, and with in-context learning (second from top) on the Rossmann sales forecasting task. The dashed vertical gray line indicates the forecast start date and the shaded region represents 80% prediction interval around the median forecast. With ICL, Chronos-2 produces a substantially more accurate forecast by capturing the influence of promotion and holiday covariates on future sales.

Figure 6 shows forecasts on the energy price forecasting task for Germany (EPF-DE), where the goal is to predict the hourly energy price for the next day using historical prices, day-ahead forecasts of the load and renewable (solar and wind) energy generation. In the univariate mode, Chronos-2 makes reasonable but imprecise predictions. However, with ICL, Chronos-2 effectively uses the covariates, producing significantly more accurate predictions.

The retail task in Figure 7 involves predicting next quarter’s weekly store sales of Rossmann, a European drug store chain, using historical sales and covariates: historical customer footfall plus known covariates indicating store operation, promotion periods, and holidays. Chronos-2’s univariate forecast is nearly flat with high uncertainty. In contrast, the ICL forecast leverages covariates — particularly promotion and holiday information — to capture the true sales dynamics over the forecast horizon.

5.4 Ablation Studies

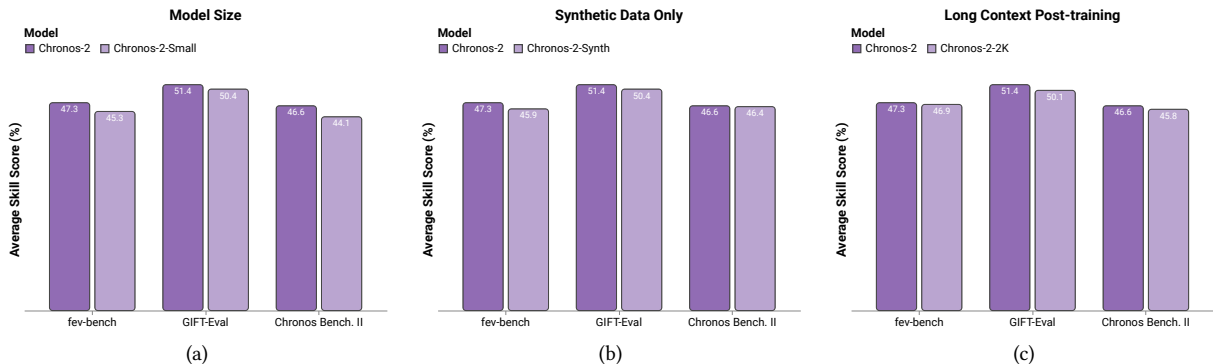


Figure 8: Comparison of the main Chronos-2 model (120M parameters) with (a) a smaller 28M-parameter model, (b) a model trained exclusively on synthetic data, and (c) the main model prior to long-context post-training.

In this section, we present additional experiments and ablations that disentangle the impact of different design choices. We investigate the performance of Chronos-2 across different parameter counts, evaluate models trained exclusively on synthetic data, and demonstrate the importance of post-training on long-context scenarios.

Model Size. We trained a *small* model with 28M parameters to understand the impact of model size on forecasting performance. As shown in Figure 8a, the small model delivers strong performance despite its reduced size. On GIFT-Eval, for instance, its skill score lags the base model by as little as 1% points, while offering nearly $2\times$ faster inference. This makes it particularly suitable for low-resource environments, such as CPU-only settings, or applications where inference speed is prioritized over maximum forecast accuracy.

Synthetic Data Only. Synthetic time series data has played a pivotal role in advancing pretrained forecasting models (Ansari et al., 2024; Das et al., 2024b). TabPFN-TS (Hoo et al., 2025) demonstrated that strong performance is achievable even when training relies exclusively on synthetic data. To examine the limits of this approach, we trained a version of Chronos-2 using only synthetic data. On Chronos Benchmark II and GIFT-Eval, this model (Chronos-2-Synth) performs only slightly below the version with real data in its pretraining corpus (Figure 8b). It also delivers strong results on fev-bench, though with a larger performance gap. These results underscore the importance of synthetic data, suggesting that with further research, real data may not even be required for effective pretraining.

Long context Post-training. As described in Section 3.3, Chronos-2 is initially trained with a context length of 2,048 time steps and then post-trained with an extended context of 8,192 steps. Figure 8c compares the base model (denoted Chronos-2-2K) with the post-trained variant. Extending the context length yields gains, particularly on the GIFT-Eval benchmark, which contains many high-frequency datasets with long seasonal periods.

6 Discussion

We introduced Chronos-2, a pretrained time series model designed to handle a wide range of forecasting scenarios — including univariate, multivariate, and covariate-informed tasks — in a zero-shot manner. Across three comprehensive benchmarks, Chronos-2 consistently outperforms existing foundation models, demonstrating that in-context learning enhances forecasting performance across diverse task types.

A particularly large performance gap appears on covariate-informed tasks, where Chronos-2 substantially surpasses prior foundation models. This highlights both the limitations of existing models and the critical role contextual information (e.g., covariates) plays in accurate forecasting. While Chronos-2 supports only numeric and categorical covariates, extending pretrained models to incorporate multimodal inputs, such as text, represents a promising direction for future research (Zhang et al., 2025).

Our results further emphasize the importance of synthetic data in enabling generalist forecasting. The abilities of Chronos-2 beyond univariate forecasting rely entirely on synthetic data, and ablation studies show that models trained solely on synthetic data perform only slightly worse than those trained on a mixture of real and synthetic datasets. We expect synthetic data to play an increasingly central role in advancing pretrained time series models.

Finally, the flexible group attention mechanism in Chronos-2 opens opportunities for further applications. For instance, time series could be grouped using sparse metadata or dense embeddings to enable *retrieval-augmented forecasting*, potentially improving performance in small-data or cold-start scenarios.

Acknowledgements

We thank the developers of open-source libraries used in the development of Chronos-2, including but not limited to torch (Paszke et al., 2019), numpy (Harris et al., 2020), pandas (pandas development team, 2020; Wes McK-inney, 2010), statsmodels (Seabold & Perktold, 2010), transformers (Wolf et al., 2020), gluonts (Alexandrov et al., 2020), autogluon (Shchur et al., 2023), statsforecast (Garza et al., 2022), einops (Rogozhnikov, 2022) and scikit-learn (Pedregosa et al., 2011). We also thank our colleagues at Amazon for their invaluable support in releasing Chronos-2: Kevin Ormiston, Jenna Larson, Larry Hardesty, Divya Sukumar, Lahari Chowtoori and Henri Yandell. Finally, we are grateful to our fellow researchers for insightful discussions and their contributions to the field: Andrew Gordon Wilson, Michael Mahoney, Dmitry Efimov, Christoph Bergmeir, Valentin Flunkert, David Salinas,

Imry Kissos, Devamanyu Hazarika, Tim Januschowski, Jan Gasthaus, William Gilpin, Annan Yu, Zelin He, Kashif Rasul, Rajat Sen, Yichen Zhou, Chenghao Liu, Taha Aksu, Gerald Woo, Emaad Khwaja and Ben Cohen.

References

- Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024. [1](#), [2](#), [7](#), [8](#), [9](#), [10](#)
- Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, et al. GluonTS: Probabilistic and Neural Time Series Modeling in Python. *The Journal of Machine Learning Research*, 21(1):4629–4634, 2020. [14](#)
- Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. [1](#), [2](#), [4](#), [7](#), [8](#), [9](#), [10](#), [14](#), [20](#)
- Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021. [4](#)
- V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, 16(4):521–530, 2000. [3](#)
- Andreas Auer, Raghul Parthipan, Pedro Mercado, Abdul Fatir Ansari, Lorenzo Stella, Bernie Wang, Michael Bohlke-Schneider, and Syama Sundar Rangapuram. Zero-shot time series forecasting with covariates via in-context learning. *arXiv preprint arXiv:2506.03128*, 2025a. [4](#), [8](#)
- Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. In *Advances in Neural Information Processing Systems*, 2025b. [4](#), [8](#)
- Fouad Bahrpeyma, Mark Roantree, Paolo Cappellari, Michael Scriney, and Andrew McCarren. A methodology for validating diversity in synthetic time series generation. *MethodsX*, 8:101459, 2021. [7](#)
- Marta Bańbura, Domenico Giannone, and Lucrezia Reichlin. Large bayesian vector auto regressions. *Journal of applied Econometrics*, 25(1):71–92, 2010. [1](#)
- John B Burbidge, Lonnie Magee, and A Leslie Robb. Alternative transformations to handle extreme values of the dependent variable. *Journal of the American statistical Association*, 83(401):123–127, 1988. [4](#)
- Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2023. [1](#), [3](#)
- Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, et al. This time is different: An observability perspective on time series foundation models. In *Advances in Neural Information Processing Systems*, 2025. [1](#), [4](#), [8](#)
- Abhimanyu Das, Matthew Faw, Rajat Sen, and Yichen Zhou. In-context fine-tuning for time-series foundation models. *arXiv preprint arXiv:2410.24087*, 2024a. [4](#)
- Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024b. [1](#), [4](#), [8](#), [14](#)
- Patrick Emami, Abhijeet Sahu, and Peter Graf. BuildingsBench: A Large-Scale Dataset of 900K Buildings and Benchmark for Short-Term Load Forecasting. *arXiv:2307.00142*, 2023. [20](#)

- FiveThirtyEight. uber-tlc-foil-response: Uber trip data from a freedom of information request to NYC’s Taxi & Limousine Commission. <https://github.com/fivethirtyeight/uber-tlc-foil-response>, 2025. Accessed: 2025-09-26. 20
- Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1, 2024. 4
- Federico Garza, Max Mergenthaler Canseco, Cristian Challú, and Kin 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>. 14
- Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash Time Series Forecasting Archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 20
- Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large Language Models Are Zero-Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*, 2023. 4
- Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL <https://doi.org/10.1038/s41586-020-2649-2>. 14
- Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: How TabPFN-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945*, 2025. 4, 8, 14
- Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018. 1, 3, 8
- Jiawei Jiang, Chengkai Han, Wenjun Jiang, Wayne Xin Zhao, and Jingyuan Wang. Libcity: A unified library towards efficient and comprehensive urban spatial-temporal prediction. *arXiv preprint arXiv:2304.14343*, 2023. 20
- Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 4
- Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pp. 10280–10297. PMLR, 2022. 4
- Bryan Lim, Serkan Ö Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. 1, 3
- Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *arXiv:2306.08259*, 2023. 20
- Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *International Conference on Machine Learning*, 2025. 4, 8
- Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. 20
- Daniele Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD explorations newsletter*, 3(1):27–32, 2001. 4
- Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. 3, 4, 5, 12
- Boris N. Oreshkin, Dmitri Carpo, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. 3

- Boris N. Oreshkin, Dmitri Carпов, Nicolas Chapados, and Yoshua Bengio. Meta-learning framework with applications to zero-shot time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 3
- Bernardo Pérez Orozco and Stephen J. Roberts. Zero-shot and few-shot time series forecasting with ordinal regression recurrent neural networks. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 503–508, 2020. 3
- The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL <https://doi.org/10.5281/zenodo.3509134>. 14
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 14
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 4, 14
- Fotios Petropoulos and Ivan Svetunkov. A simple combination of univariate models. *International journal of forecasting*, 36(1):110–115, 2020. 8
- Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J Bessa, Jakub Bijak, John E Boylan, et al. Forecasting: theory and practice. *International Journal of forecasting*, 38(3):705–871, 2022. 2
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 5
- Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018. 3
- Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International conference on machine learning*, pp. 8844–8856. PMLR, 2021. 4
- Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weather-bench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020. 20
- Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021. 3
- Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, Sahil Garg, Alexandre Drouin, Nicolas Chapados, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for time series forecasting, 2023. 4
- Alex Rogozhnikov. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*, 2022. 14
- Jakob Runge, Andreas Gerhardus, Gherardo Varando, Veronika Eyring, and Gustau Camps-Valls. Causal inference for time series. *Nature Reviews Earth & Environment*, 4(7):487–505, 2023. 7
- David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32, 2019. 20

- David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. [3](#)
- Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010. [14](#)
- Oleksandr Shchur, Ali Caner Turkmen, Nick Erickson, Huibin Shen, Alexander Shirkov, Tony Hu, and Bernie Wang. Autogluon-timeseries: Automl for probabilistic time series forecasting. In *International Conference on Automated Machine Learning*, pp. 9–1. PMLR, 2023. [14](#)
- Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Bernie Wang. fev-bench: A realistic benchmark for time series forecasting models. *arXiv preprint arXiv*, 2025. [2](#), [8](#), [9](#)
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. [5](#)
- Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pp. 366–381. Springer, 2006. [11](#)
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023. [5](#)
- Bartosz Uniejewski and Rafał Weron. Efficient forecasting of electricity spot prices with expert and lasso models. *Energies*, 11(8):2039, 2018. [4](#)
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 2017. [5](#)
- Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman (eds.), *Proceedings of the 9th Python in Science Conference*, pp. 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a. [14](#)
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. [14](#)
- Gerald Woo, Chenghao Liu, Akshat Kumar, and Doyen Sahoo. Pushing the limits of pre-training for time series forecasting in the cloudops domain. *arXiv preprint arXiv:2310.05063*, 2023. [20](#)
- Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, 2024. [4](#), [7](#), [8](#)
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. [5](#)
- Xiyuan Zhang, Boran Han, Haoyang Fang, Abdul Fatir Ansari, Shuai Zhang, Danielle C Maddix, Cuixiong Hu, Andrew Gordon Wilson, Michael W Mahoney, Hao Wang, et al. Does multimodality lead to better time series forecasting? *arXiv preprint arXiv:2506.21611*, 2025. [14](#)

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023. [4](#)

Nina Żukowska, Mononito Goswami, Michał Wiliński, Willa Potosnak, and Artur Dubrawski. Towards long-context time series foundation models. *arXiv preprint arXiv:2409.13530*, 2024. [4](#)

A Training Data

Dataset Name	Frequencies	# Time Series	Domain	Source
Electricity	15min, 1H, 1W, 1D	370	Energy	Godahehwa et al. (2021)
KDD Cup (2018)	1H, 1D	270	Nature	Godahehwa et al. (2021)
M4 (Daily)	1D	4227	Various	Makridakis et al. (2020)
M4 (Hourly)	1H	414	Various	Makridakis et al. (2020)
M4 (Monthly)	1M	48000	Various	Makridakis et al. (2020)
M4 (Weekly)	1W	359	Various	Makridakis et al. (2020)
Mexico City Bikes	1H, 1D, 1W	494	Transport	Ansari et al. (2024)
Pedestrian Counts	1H, 1D, 1W	66	Transport	Godahehwa et al. (2021)
Solar	5min, 10min, 1H	5166	Energy	Ansari et al. (2024)
Taxi	30min, 1H	2428	Transport	Salinas et al. (2019)
Uber TLC	1H, 1D	262	Transport	FiveThirtyEight (2025)
USHCN	1D, 1W	225280	Nature	Ansari et al. (2024)
Weatherbench	1H, 1D, 1W	225280	Nature	Rasp et al. (2020)
Wiki	1H, 1D, 1W	100000	Web	Ansari et al. (2024)
Wind Farms	1H, 1D	337	Energy	Godahehwa et al. (2021)
Temperature-Rain	1D	32072	Nature	Godahehwa et al. (2021)
London Smart Meters	30min, 1D	5560	Energy	Godahehwa et al. (2021)
Alibaba Cluster Trace (2018)	5min, 1H	100000	Cloud Ops	Woo et al. (2023)
Azure VM Traces (2017)	5min, 1H	100000	Cloud Ops	Woo et al. (2023)
Borg Cluster Data (2011)	5min, 1H	100000	Cloud Ops	Woo et al. (2023)
LargeST (2017)	1H, 1D	8196	Transport	Liu et al. (2023)
Q-Traffic	15min, 1H	45148	Transport	Jiang et al. (2023)
Buildings 900K	1H, 1D	100000	Energy	Emami et al. (2023)

Table 6: Real univariate datasets used for pretraining Chronos-2.

B Additional Results

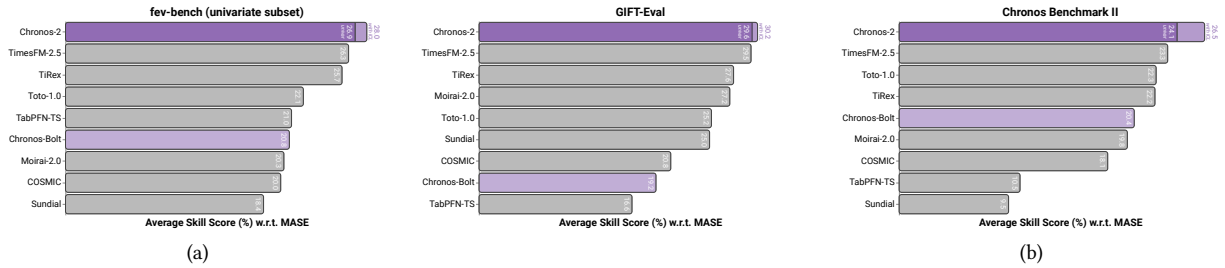


Figure 9: Chronos-2's point forecasting results in univariate mode and the corresponding improvements from in-context learning (ICL), shown as stacked bars on (a) the univariate subset of fev-bench, (b) GIFT-Eval, and (c) Chronos Benchmark II.

Model	Avg. Win Rate (%)	Skill Score (%)	Median runtime (s)	Leakage (%)	#Failures
Chronos-2	87.9	35.5	3.6	0	0
TiRex	75.1	30.0	1.4	1	0
TimesFM-2.5	74.4	30.3	16.9	8	0
Toto-1.0	64.3	28.2	90.7	8	0
Moirai-2.0	58.7	27.3	2.5	28	0
COSMIC	58.6	25.7	34.4	0	0
Chronos-Bolt	57.9	26.5	1.0	0	0
TabPFN-TS	55.7	27.6	305.5	0	2
Sundial	49.8	24.7	35.6	1	0
Stat. Ensemble	44.2	15.7	690.6	0	11
AutoARIMA	32.1	11.2	186.8	0	10
AutoTheta	30.3	11.0	9.3	0	0
AutoETS	30.2	2.3	17.0	0	3
SeasonalNaive	16.7	0.0	2.3	0	0
Naive	14.0	-16.7	2.2	0	0

Table 7: **fev-bench results**. The average win rate and skill score are computed with respect to the mean absolute scaled error (MASE) metric on fev-bench. Higher values are better for both.

Model	Avg. Win Rate (%)	Skill Score (%)	Median runtime (s)	Leakage (%)	#Failures
Chronos-2	88.5	51.5	3.6	0	0
TiRex	79.0	46.7	1.4	1	0
TimesFM-2.5	76.8	46.8	16.9	8	0
Toto-1.0	67.6	45.0	90.7	8	0
COSMIC	65.2	43.7	34.4	0	0
TabPFN-TS	64.8	45.8	305.5	0	2
Moirai-2.0	62.8	43.9	2.5	28	0
Chronos-Bolt	60.5	43.2	1.0	0	0
Sundial	41.9	37.4	35.6	1	0
Stat. Ensemble	38.3	21.8	690.6	0	11
AutoARIMA	34.6	23.4	186.8	0	10
AutoETS	26.8	-27.0	17.0	0	3
AutoTheta	21.3	7.8	9.3	0	0
SeasonalNaive	14.1	0.0	2.3	0	0
Naive	7.8	-39.1	2.2	0	0

Table 8: **fev-bench results**. The average win rate and skill score are computed with respect to the weighted quantile loss (WQL) metric on fev-bench. Higher values are better for both.

Model	Avg. Win Rate (%)	Skill Score (%)	Median runtime (s)	Leakage (%)	#Failures
Chronos-2	85.4	39.4	3.6	0	0
TimesFM-2.5	74.1	33.8	16.9	8	0
TiRex	73.7	33.6	1.4	1	0
Toto-1.0	65.1	31.5	90.7	8	0
TabPFN-TS	61.5	33.4	305.5	0	2
COSMIC	60.5	30.1	34.4	0	0
Moirai-2.0	59.6	30.7	2.5	28	0
Chronos-Bolt	58.0	29.8	1.0	0	0
Sundial	47.7	27.3	35.6	1	0
Stat. Ensemble	43.0	17.7	690.6	0	11
AutoETS	30.8	4.3	17.0	0	3
AutoARIMA	30.8	13.3	186.8	0	10
AutoTheta	27.2	13.8	9.3	0	0
Naive	17.5	-6.1	2.2	0	0
SeasonalNaive	15.2	0.0	2.3	0	0

Table 9: **fev-bench results**. The average win rate and skill score are computed with respect to the weighted absolute percentage error (WAPE) metric on fev-bench. Higher values are better for both.

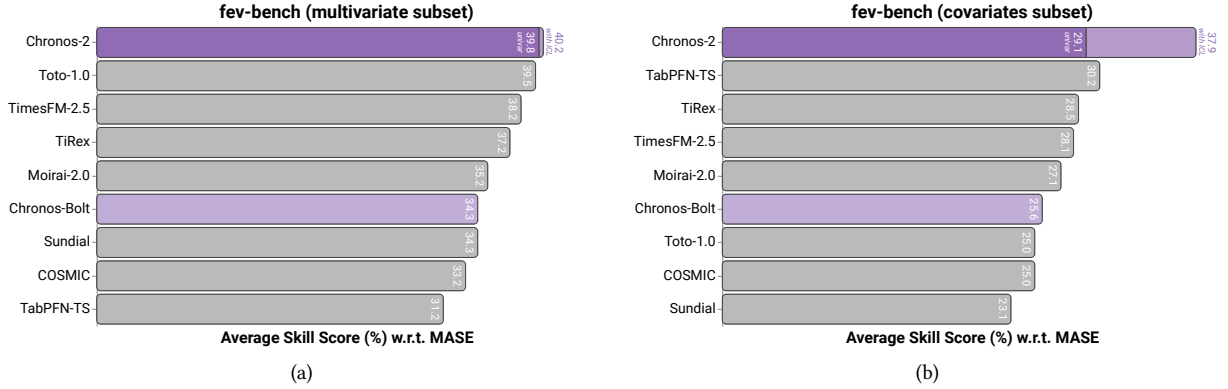


Figure 10: Chronos-2’s point forecasting results in univariate mode and the corresponding gains from in-context learning (ICL), shown as stacked bars on the multivariate and covariates subsets of fev-bench.

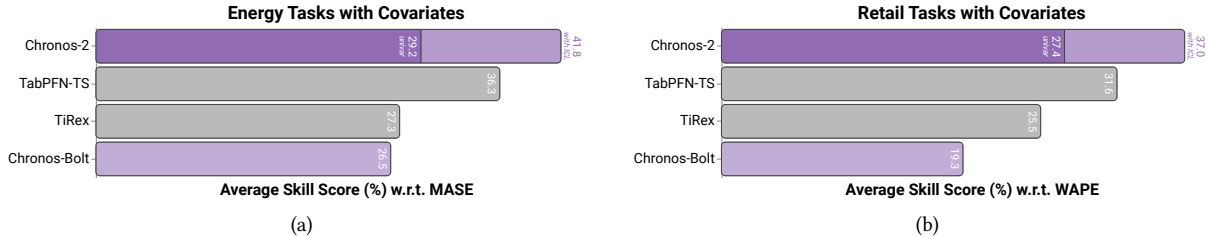


Figure 11: Comparison of Chronos-2 against baselines on tasks which include dynamic covariates from the energy and retail domains. For retail, we consider the domain-appropriate WAPE metric.

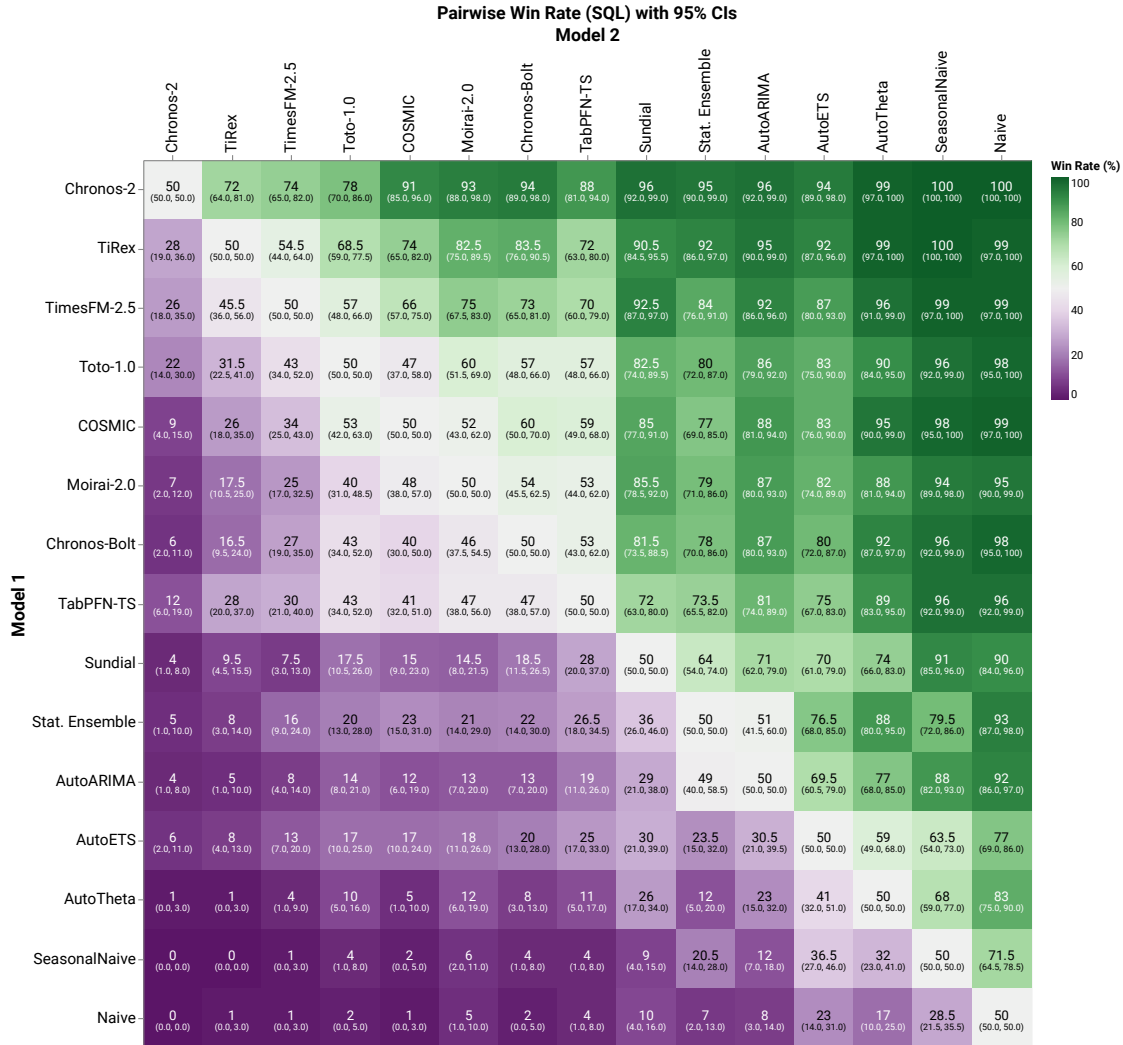


Figure 12: The pairwise win rates for all models on fev-bench with 95% confidence intervals (CIs) with respect to SQL metric.

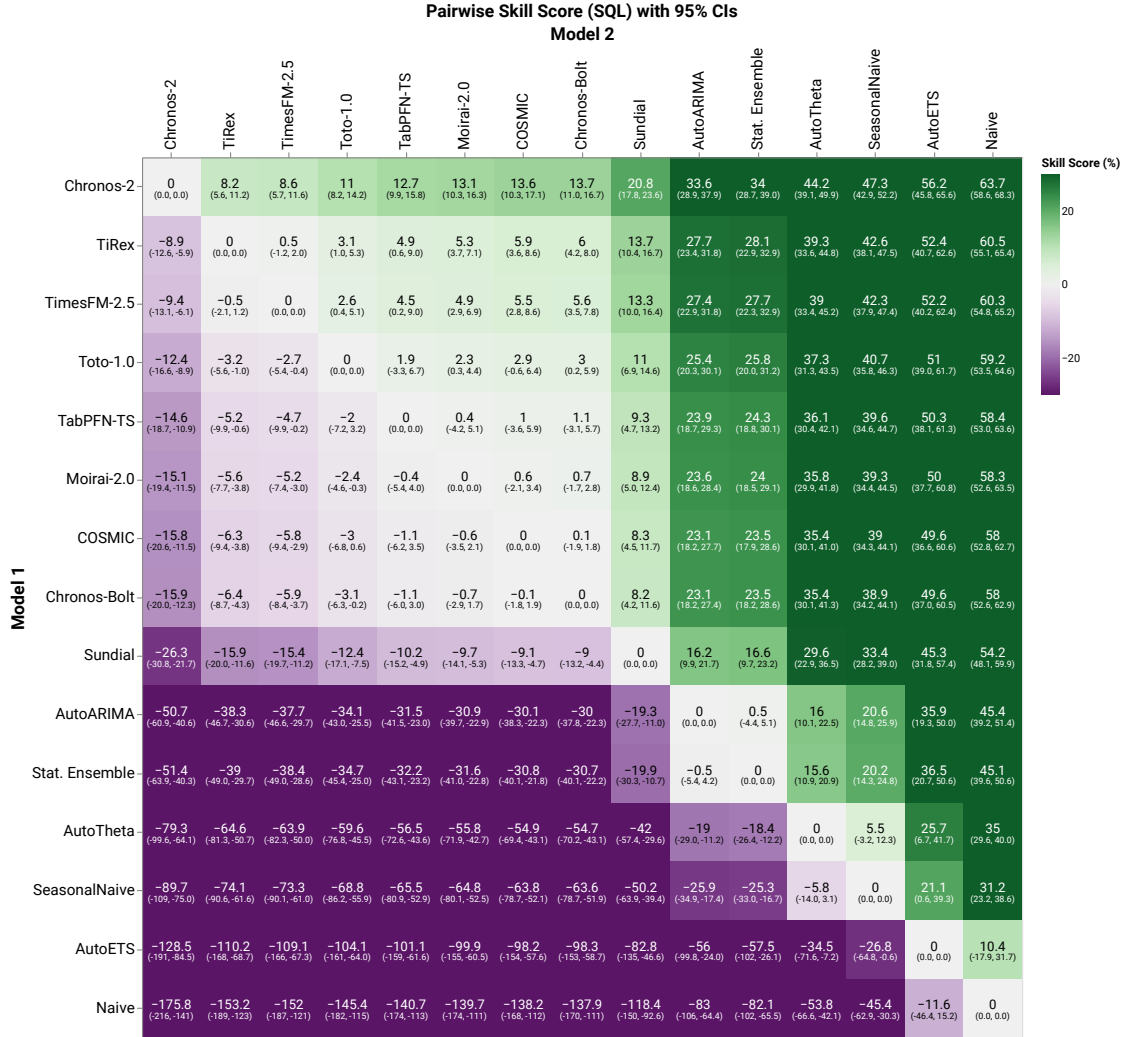


Figure 13: The pairwise skill scores for all models on fev-bench with 95% confidence intervals (CIs) with respect to SQL metric.

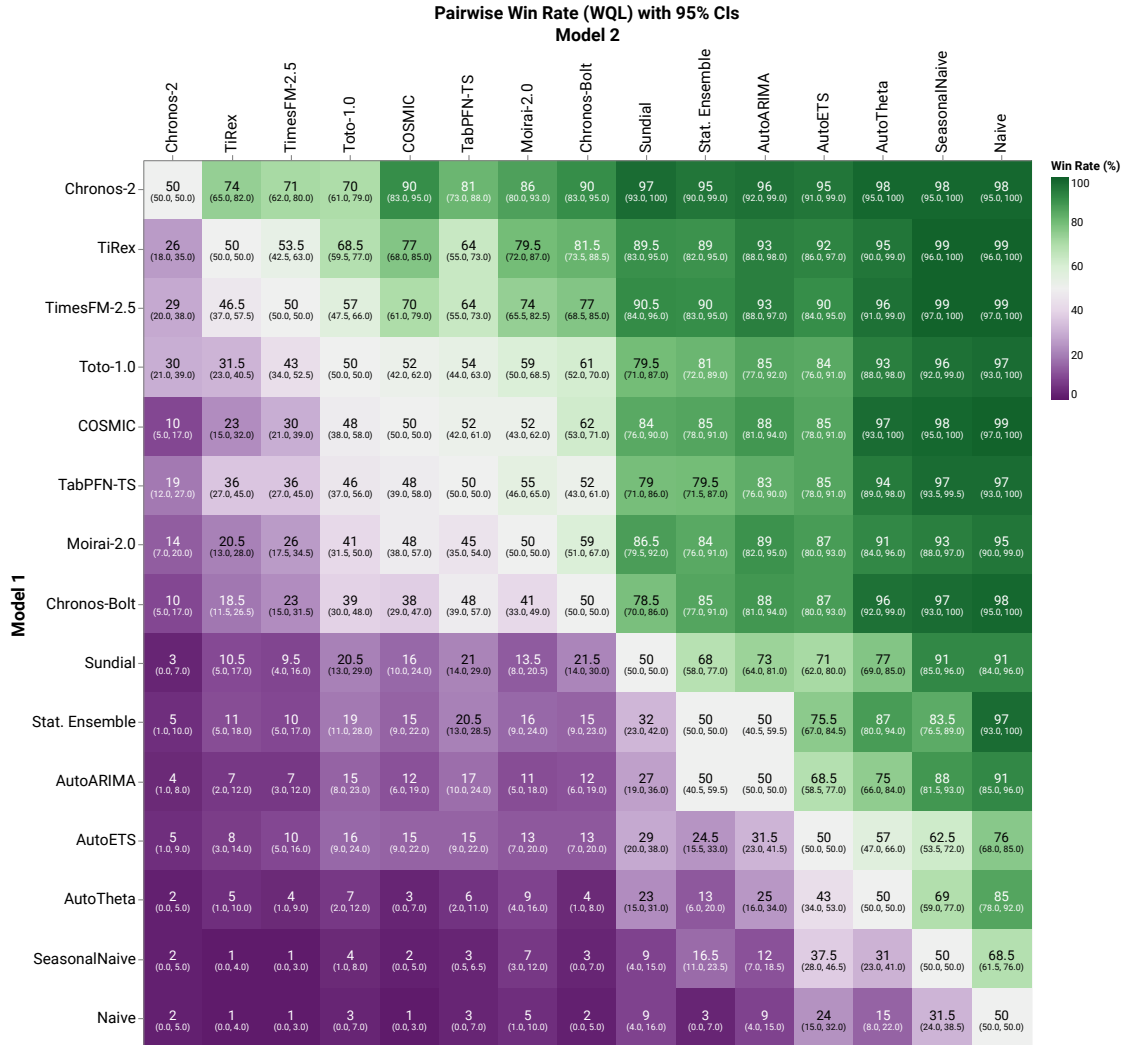


Figure 14: The pairwise win rates for all models on fev-bench with 95% confidence intervals (CIs) with respect to WQL metric.

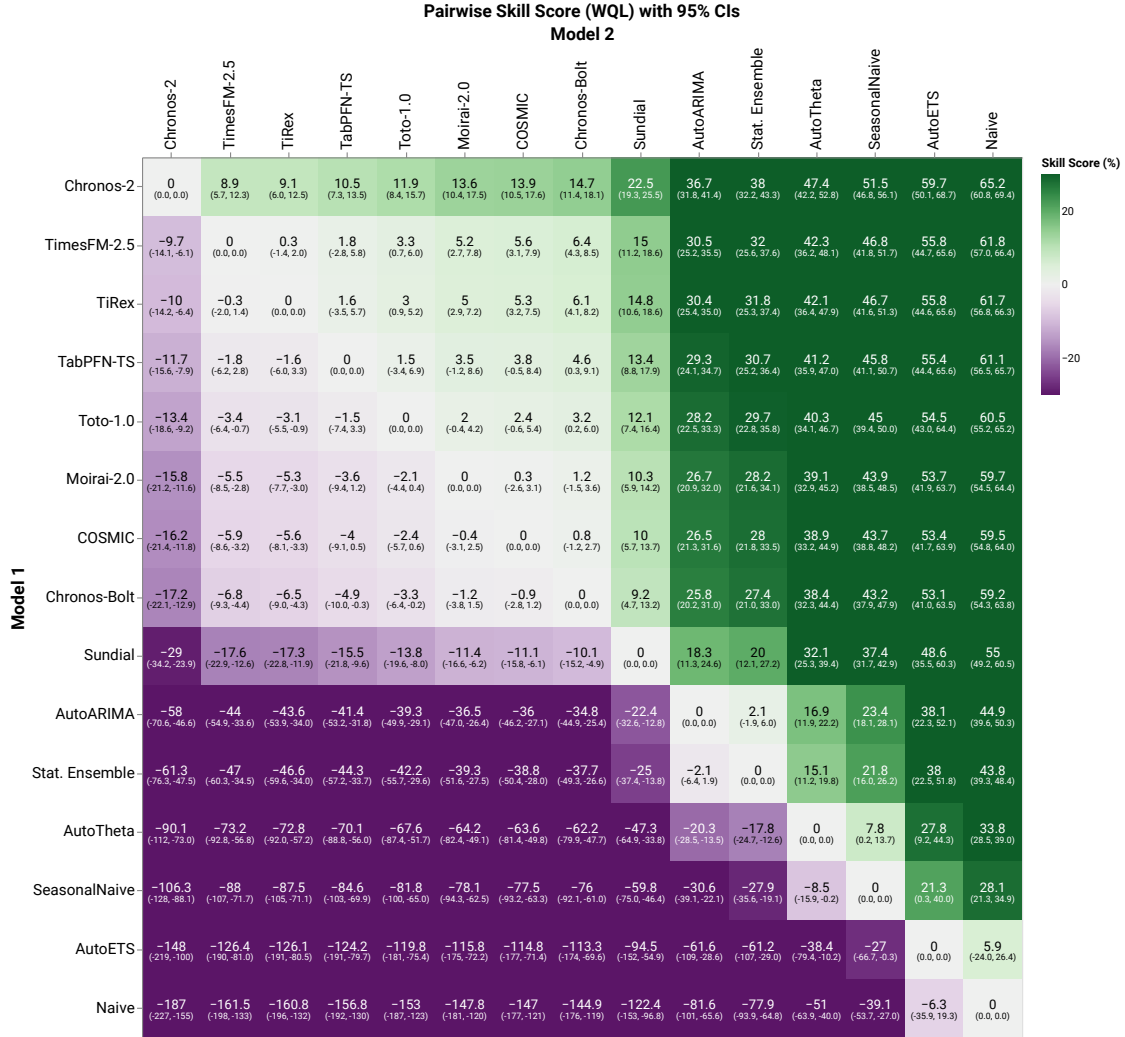


Figure 15: The pairwise skill scores for all models on fev-bench with 95% confidence intervals (CIs) with respect to WQL metric.

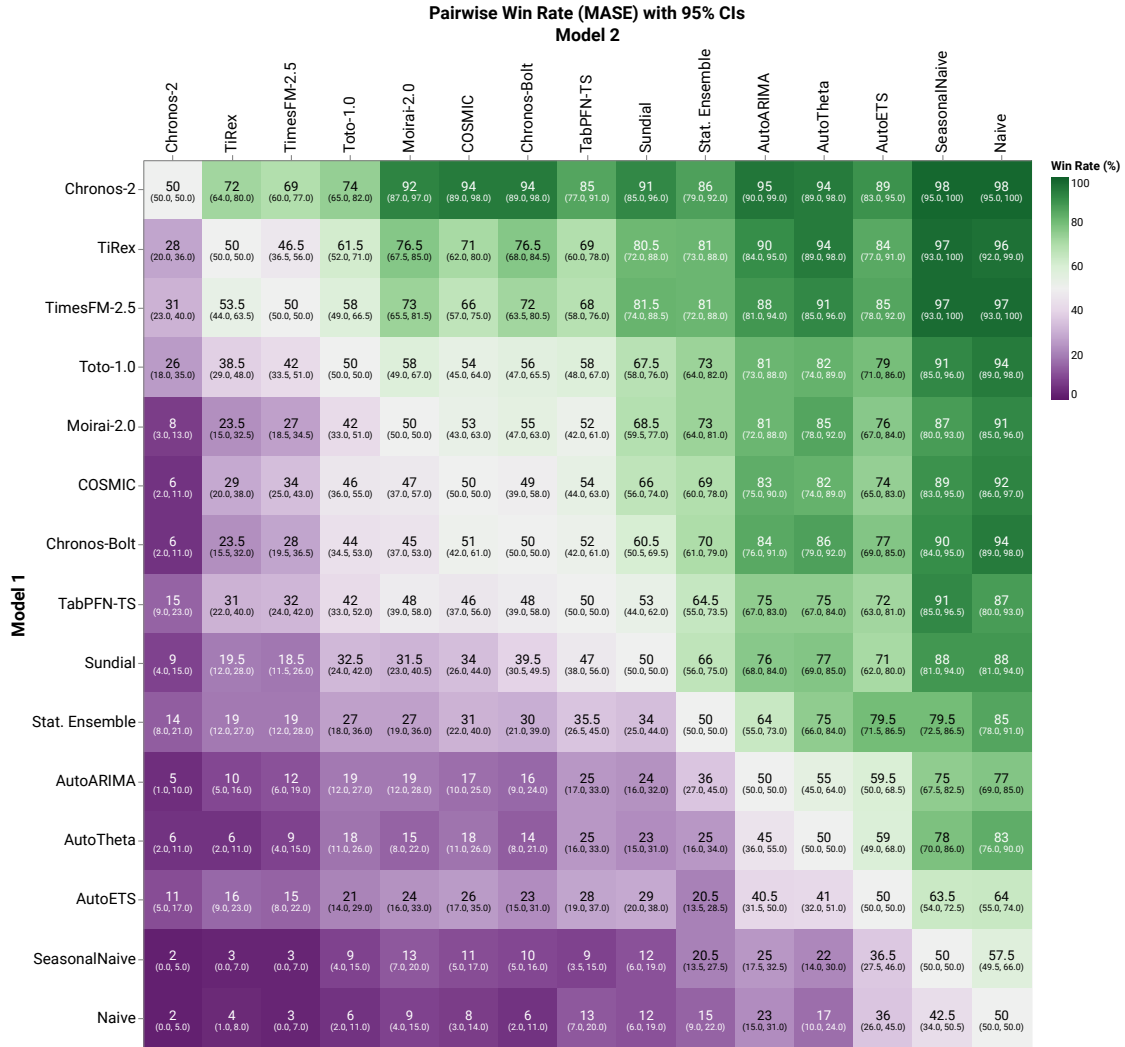


Figure 16: The pairwise win rates for all models on fev-bench with 95% confidence intervals (CIs) with respect to MASE metric.

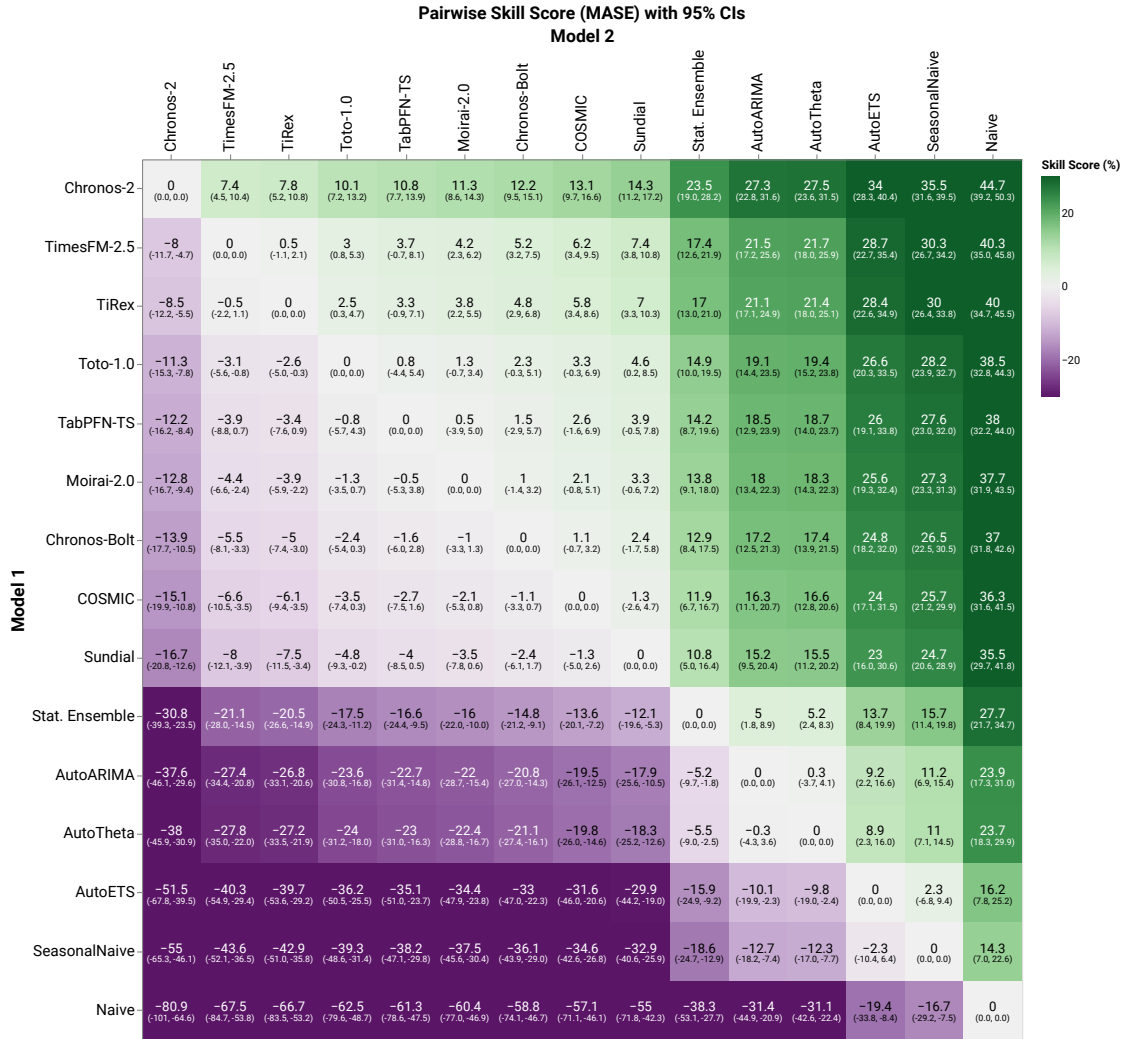


Figure 17: The pairwise skill scores for all models on fev-bench with 95% confidence intervals (CIs) with respect to MASE metric.

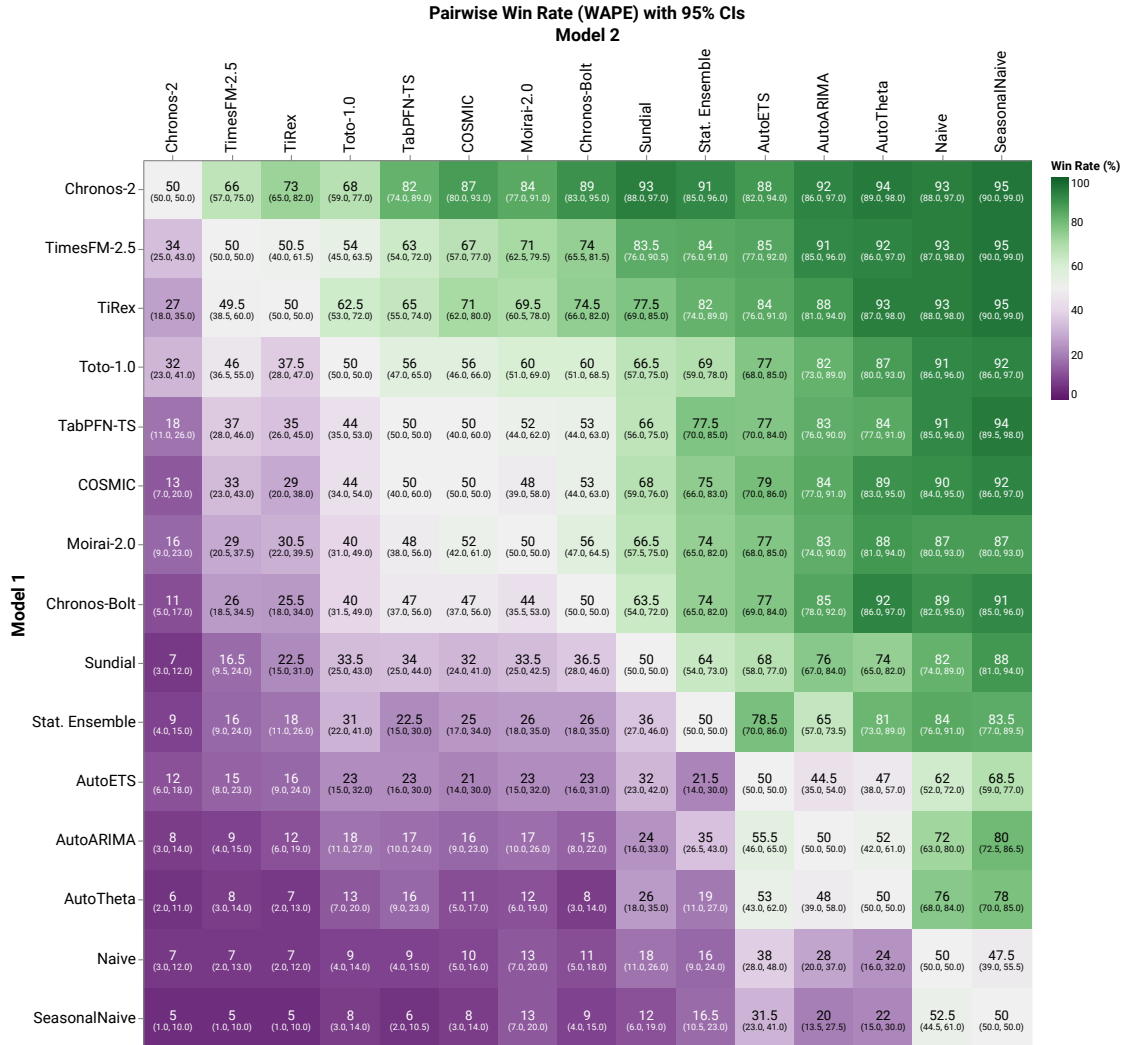


Figure 18: The pairwise win rates for all models on fev-bench with 95% (CIs) with respect to WAPe metric.

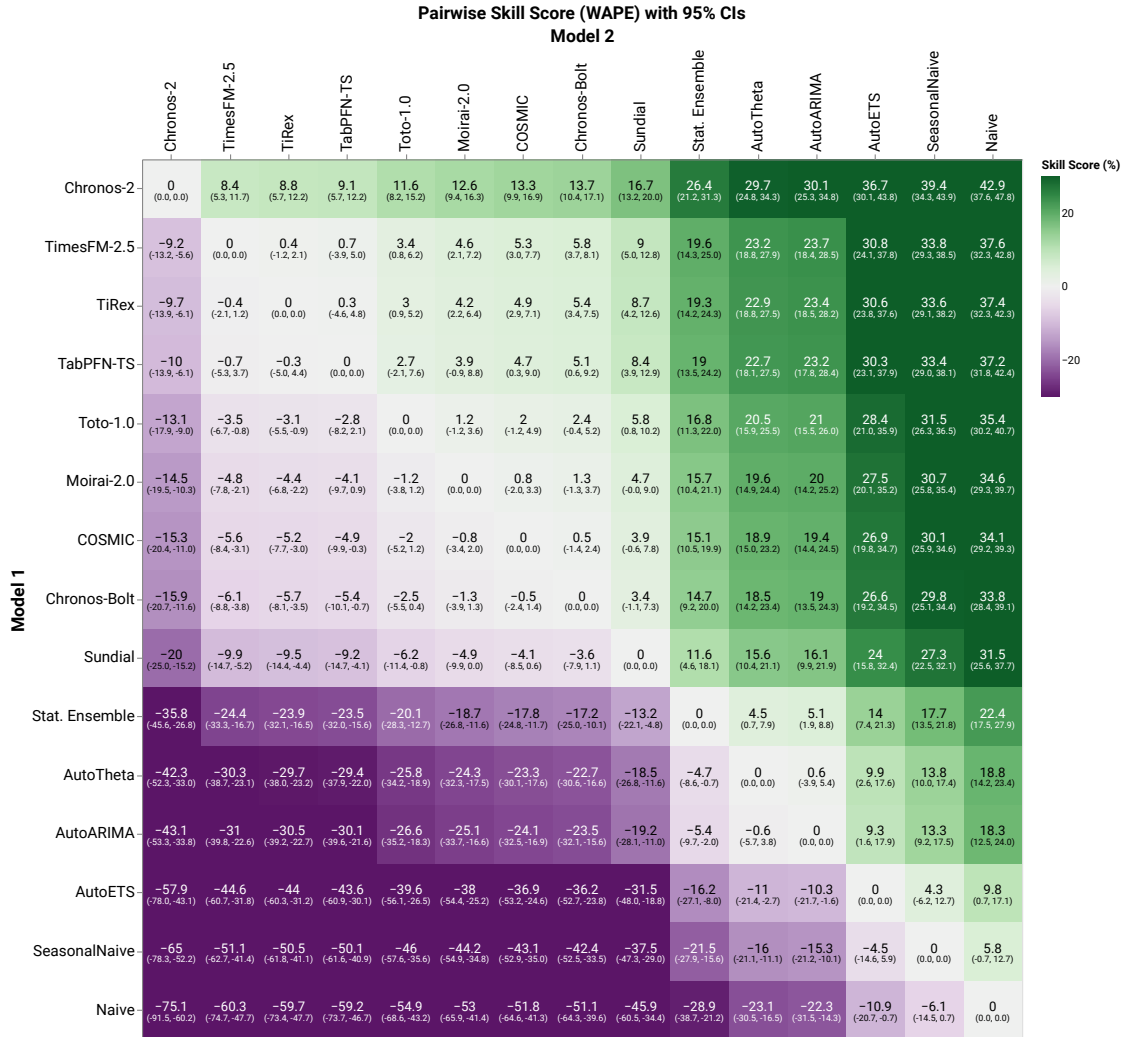


Figure 19: The pairwise skill scores for all models on fev-bench with 95% confidence intervals (CIs) with respect to WAPE metric.

Task	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
ENTSO-e Load	15T	96	20	175,292	6	1	0	3	0
ENTSO-e Load	30T	96	20	87,645	6	1	0	3	0
ENTSO-e Load	H	168	20	43,822	6	1	0	3	0
EPF-BE	H	24	20	52,416	1	1	0	2	0
EPF-DE	H	24	20	52,416	1	1	0	2	0
EPF-FR	H	24	20	52,416	1	1	0	2	0
EPF-NP	H	24	20	52,416	1	1	0	2	0
EPF-PJM	H	24	20	52,416	1	1	0	2	0
GFC12	H	168	10	39,414	11	1	0	1	0
GFC14	H	168	20	17,520	1	1	0	1	0
GFC17	H	168	20	17,544	8	1	0	1	0
Solar with Weather	15T	96	20	198,600	1	1	2	7	0
Solar with Weather	H	24	20	49,648	1	1	2	7	0
KDD Cup 2022	D	14	10	243	134	1	9	0	0
KDD Cup 2022	10T	288	10	35,279	134	1	9	0	0
KDD Cup 2022	30T	96	10	11,758	134	1	9	0	0

Table 10: Subset of datasets from fev-bench with dynamic covariates for the energy domain case study.

Task	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
Favorita Store Sales	M	12	2	54	1,579	1	1	1	6
Favorita Store Sales	W	13	10	240	1,579	1	1	1	6
Favorita Store Sales	D	28	10	1,688	1,579	1	1	2	6
Favorita Transactions	M	12	2	54	51	1	1	0	5
Favorita Transactions	W	13	10	240	51	1	1	0	5
Favorita Transactions	D	28	10	1,688	51	1	1	1	5
M5	M	12	1	58	30,490	1	0	8	5
M5	W	13	1	257	30,490	1	0	8	5
M5	D	28	1	1,810	30,490	1	0	8	5
Rohlik Orders	W	8	5	170	7	1	9	4	0
Rohlik Orders	D	61	5	1,197	7	1	9	4	0
Rohlik Sales	W	8	1	150	5,243	1	1	13	7
Rohlik Sales	D	14	1	1,046	5,390	1	1	13	7
Rossmann	W	13	8	133	1,115	1	1	4	10
Rossmann	D	48	10	942	1,115	1	1	5	10
Walmart	W	39	1	143	2,936	1	0	10	4
Hermes	W	52	1	261	10,000	1	0	1	2

Table 11: Subset of datasets from fev-bench with dynamic covariates for the retail domain case study.