Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A foundation model for multivariate time series forecasting

Abdul Fatir Ansari
abdulfatirs@gmail.com
Amazon Web Services
Oleksandr Shchur
Amazon Web Services
Jaris Küken
University of Freiburg
Andreas Auer
Amazon Web Services
Boran Han
Amazon Web Services
Pedro Mercado
Amazon Web Services
Syama Sundar Rangapuram
Amazon Web Services
Huibin Shen
Amazon Web Services
Lorenzo Stella
Amazon Web Services
Xiyuan Zhang
Amazon Web Services
Mononito Goswami
Amazon Web Services
Shubham Kapoor
Amazon Web Services
Danielle Robinson
Amazon Web Services https://orcid.org/0000-0002-2317-4068
Pablo Guerron
Amazon
Florian Saupe
Amazon Web Services
Markus Reichstein

Amazon Web Services

Tony Hu
Amazon Web Services

Junming Yin
Amazon Web Services

Nick Erickson
Amazon Web Services

Prateek Mutalik Desai
Amazon Web Services

Hao Wang
Amazon Web Services

Huzefa Rangwala
Amazon Web Services

George Karypis
Amazon Web Services

Yuyang Wang
Amazon Web Services

Michael Bohlke-Schneider
Amazon Web Services

Physical Sciences - Article

Keywords: multivariate, time series forecasting, foundation models, transformers

Posted Date: April 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9096522/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

A foundation model for multivariate time series forecasting

Abdul Fatir Ansari[1*†], Oleksandr Shchur[1*†], Jaris Küken[2†], Andreas Auer[1], Boran Han[1], Pedro Mercado[1], Syama Sundar Rangapuram[1], Huibin Shen[1], Lorenzo Stella[1], Xiyuan Zhang[1], Mononito Goswami[1], Shubham Kapoor[1], Danielle C. Maddix[1], Pablo Guerron[3,4], Florian Saupe[1], Markus Reichstein[1], Tony Hu[1], Junming Yin[1], Nick Erickson[1], Prateek Mutalik Desai[1], Hao Wang[1,5], Huzefa Rangwala[1], George Karypis[1], Yuyang Wang[1*], Michael Bohlke-Schneider[1*]

[1*]Amazon Web Services.
[2]University of Freiburg.
[3]Amazon.
[4]Boston College.
[5]Rutgers University.

*Corresponding author(s). E-mail(s): ansarnd@amazon.de; shchuro@amazon.de; yuyawang@amazon.com; bohlkem@amazon.de;
Contributing authors: kuekenj@cs.uni-freiburg.de; auerand@amazon.de; boranhan@amazon.com; pedroml@amazon.de; rangapur@amazon.de; huibishe@amazon.de; stellalo@amazon.de; xiyuanz@amazon.com; mononito@amazon.com; kapooshu@amazon.de; dmmaddix@amazon.com; paguerro@amazon.com; fsaupe@amazon.de; mreichst@amazon.de; tonyhu@amazon.com; junmingy@amazon.com; neerick@amazon.com; pmdesai@amazon.com; howngz@amazon.com; rhuzefa@amazon.com; gkarypis@amazon.com;
†These authors contributed equally to this work.

Abstract

Accurate forecasting with uncertainty quantification is critical in a variety of scientific applications ranging from epidemiology and economics to energy systems and earth sciences. Yet most forecasting models still need to be developed and trained separately for each domain-specific dataset. Time series foundation models (TSFMs) aim to remove this limitation and enable the application of the same model across domains through large-scale pretraining. However, existing TSFMs treat each time series independently and therefore cannot model interactions among time series and external drivers. Here, we introduce Chronos-2, a foundation model for zero-shot forecasting that models the interactions between multivariate time series and heterogeneous covariates, which commonly arise in scientific applications. The model infers relationships among time series directly from historical context and generates probabilistic forecasts without task-specific treatment. Across various benchmarks, Chronos-2 achieves state-of-the-art accuracy by improving skill scores by 5.2 percentage points relative to the strongest baselines and increasing win rates by 10 percentage points. Chronos-2 is also computationally efficient, generating forecasts for 100 time series in under one second on a single mid-range GPU, 50× and 1000× faster than the best task-specific point and probabilistic forecasters, respectively. In addition, we demonstrate its effectiveness in diverse scientific settings — including infectious-disease forecasting, macroeconomic modeling, carbon-flux prediction, and runoff simulation — where the model matches or surpasses specialized methods developed for each domain. These results show that a single pretrained model can capture structure shared across temporal systems. Importantly, our work lays a foundation toward a transition from task-specific forecasting to generalist forecasting systems across scientific and industrial domains.

Keywords: multivariate, time series forecasting, foundation models, transformers

1 Introduction

Many natural and human-centric systems evolve as coupled processes. For example, epidemics spread across interconnected populations [1], economic indicators co-evolve through complex causal relationships [2], ecosystems exchange energy and carbon through linked feedback systems [3], and weather events drive river runoff across hydrological basins [4]. These phenomena are typically observed as collections of time series, whose joint evolution encodes the underlying dynamics of the system. Accurate forecasting of this temporal data is central to decision-making across disciplines ranging from public health to macroeconomics to energy systems to earth sciences.

Despite decades of progress, forecasting approaches remain largely application-specific due to the heterogeneity of temporal patterns and task characteristics — including varying granularities, context lengths, missing values, and diverse multivariate interactions. Classical statistical models [5] effectively capture simple temporal structure, but struggle to leverage large-scale datasets or jointly model many related variables. Modern machine learning methods [6–8] can represent more complex dynamics, yet typically require task-specific training and careful tuning for each

dataset. In contrast, recent time series foundation models (TSFMs) enable *zero-shot* forecasting on unseen datasets through large-scale pretraining [9–12]. However, most TSFMs still treat each time series in isolation [9, 10], which prevents them from capturing the multivariate interactions that characterize real-world systems.

To address the aforementioned limitations, we introduce Chronos-2, a multivariate foundation model that forecasts collections of time series in a zero-shot manner. Central to our approach is *in-context learning* (ICL): rather than fitting parameters separately for each task, Chronos-2 infers multivariate dependencies directly from historical context at inference time. A group-attention mechanism enables information aggregation across arbitrary sets of target time series (the primary variables to be forecasted) and heterogeneous covariates (auxiliary variables that provide predictive context), which supports their joint forecasting. Chronos-2 is pretrained on a mixture of real-world univariate time series and synthetic multivariate processes spanning diverse instantaneous and lagged dependencies to encourage generalization across domains.

We evaluate Chronos-2 on comprehensive forecasting benchmarks, where it outperforms state-of-the-art models, i.e., task-specific statistical and machine learning methods, and existing TSFMs, without any task-specific training. Chronos-2 is also 50× and 1000× faster than the best task-specific point and probabilistic forecasters; it can process 100 time series in 0.8 seconds on a mid-range NVIDIA A10G GPU. Additionally, Chronos-2 can be deployed on standard CPU hardware without specialized accelerators due to its lightweight size of 120M parameters. Lastly, we demonstrate the generalization of the model across scientific applications — infectious-disease dynamics, macroeconomic forecasting, ecosystem carbon-flux prediction, and hydrology — where it produces accurate probabilistic forecasts and often matches or surpasses specialized models used in these domains.

To summarize, our results suggest that the dynamics of complex temporal systems can be inferred directly from contextual observations using a single pretrained model. This work shifts the focus of the forecasting field from dataset-specific model fitting toward contextual inference over interacting time series.

2 A unified architecture for multivariate temporal dynamics

Chronos-2 is a unified foundation model for zero-shot forecasting across univariate and multivariate time series problems (Figure 1). The method has four central components: (i) an encoder-only Transformer [13] that alternates temporal attention with group attention to infer dependencies among related series; (ii) a flexible inference procedure dictated by group assignments; (iii) a tokenizer that maps targets and covariates into normalized patch sequences; (iv) a pretraining recipe based on heterogeneous forecasting tasks and a quantile-regression objective (see Methods for specific technical details). We discuss the training corpus that combines real univariate time series with synthetic multivariate systems in Section 3.

Chronos-2 adopts the Transformer architecture [13] — specifically, a 120M-parameter encoder-only model derived from the T5 architecture [14] — as its core

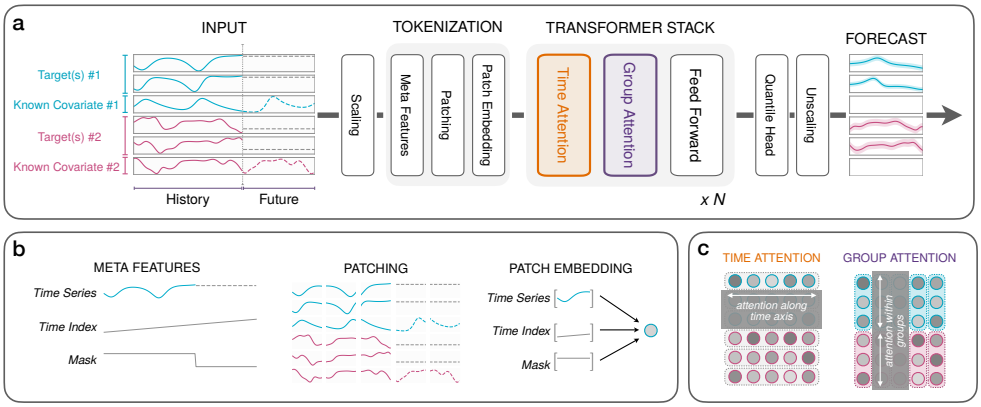


Fig. 1: **The complete Chronos-2 pipeline.** (a) *Architecture overview*: Input time series (targets and covariates) are first normalized (scaling) and then tokenized for downstream processing. The Transformer stack alternates between time and group attention layers. The output from the Transformer stack is mapped to quantile predictions through a neural network. The figure illustrates two multivariate time series with one known (future) covariate each, with corresponding groups highlighted in cyan and pink. (b) *Tokenization*: Meta features are appended to the time series which indicate the position of an observation (time index) and whether it is missing (mask). Time series and meta features are split into non-overlapping patches and mapped to high-dimensional embeddings via a neural network. (c) *Attention layers*: Each Transformer block alternates between time and group attention layers: the time attention layer aggregates information across patches within a single time series, and 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 group attention layer enables the model to infer relationships between different time series in the group based on the available context at inference time (in-context learning).

computational building block. Transformers use the attention mechanism to selectively aggregate information based on underlying relationships. Chronos-2 employs attention along two dimensions: the time attention layer aggregates information along the temporal axis within each series, and the group attention layer aggregates information across related series at a shared time index.

Group attention allows Chronos-2 to model complex relationships between groups of time series. Here, a *group* is a flexible notion of relatedness that enables diverse downstream forecasting scenarios. Changing group identities allows the model to seamlessly switch between univariate and multivariate forecasting modes at inference time.

Time series foundation models need the flexibility to handle real-world data formats, e.g., differing scales, covariate types and missing values. To stabilize the heterogeneous scales encountered in practice, Chronos-2 represents targets and covariates

in a common token space where each input dimension is normalized using historical statistics. Additionally, future values of certain covariates (e.g., planned events or holidays, promotions or weather forecasts) are known in advance and can be useful in several forecasting scenarios. Chronos-2 integrates these known covariates by taking both the historical context and the future horizon of each time series as input. We use a binary mask to indicate unknown future values (targets and past-only covariates), which are zero-filled. This masking strategy also accommodates missing values, which are ubiquitous in scientific and industrial forecasting.

Chronos-2 splits the time series into non-overlapping patches to improve memory usage and inference speed [8, 10, 11]. The patches are embedded via a neural network into a sequence of high-dimensional vectors for processing. Importantly, Chronos-2 decodes multiple forecast patches in a single forward pass rather than relying on expensive, autoregressive step-by-step generation. This patch-based design enables the model to process long contexts and horizons, handling up to 8,192 historical time steps and predicting up to 1,024 future steps at once, while remaining efficient enough to run on consumer-grade GPUs or standard CPU machines.

Lastly, Chronos-2 is probabilistic by design to support reliable downstream decision-making. A quantile head outputs a set of quantiles rather than a single point estimate, which enables probabilistic forecasts without imposing a strict parametric form (e.g., a normal distribution) on the predictive distribution. Quantile regression [15, 16] captures distributions with complex or multimodal shapes. The inclusion of extreme quantiles (0.01 and 0.99) improves the coverage of rare events, which is critical in scientific applications [17].

3 Synthesizing diverse multivariate dynamics

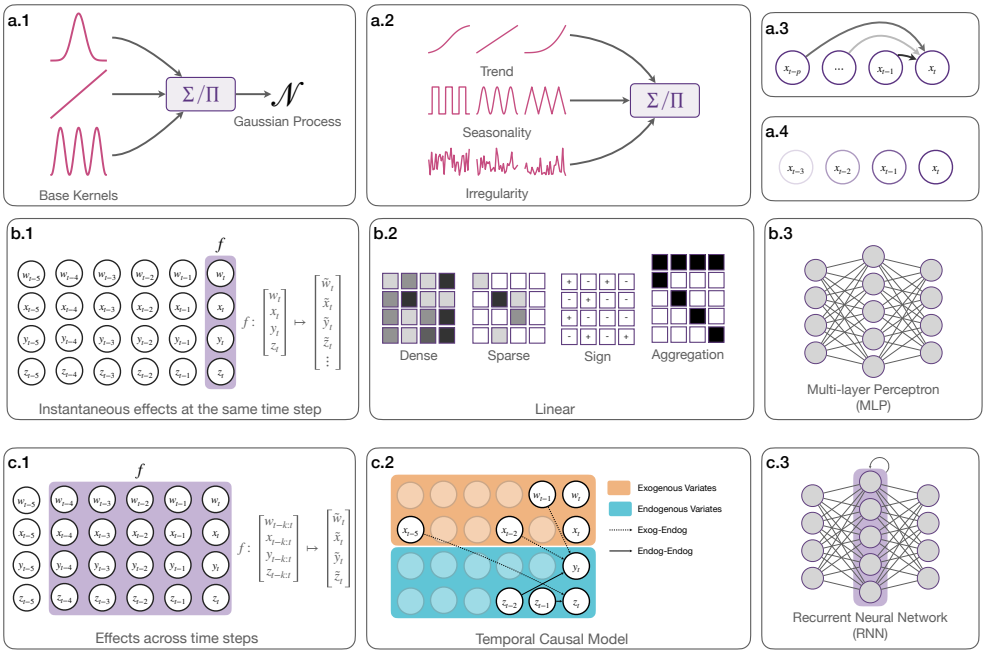


Fig. 2: **Generation of multivariate synthetic training data**. We generate multivariate synthetic time series by applying random transformation functions to time series sampled from univariate generators. *Univariate generators* produce diverse temporal patterns to serve as the building blocks for multivariate data. These include: (a.1) KernelSynth [9], which draws samples from Gaussian processes defined by the random composition of basic kernels (e.g., linear, periodic); (a.2) Trend-Seasonality-Irregularity (TSI), which stochastically combines structural components; and parametric simulations using random (a.3) Autoregressive (AR) and (a.4) Exponential Smoothing (ETS) processes. (b.1) *Instantaneous transformations* impose coupling between variables without time delays. A mixing function transforms univariate samples at each time step. We employ (b.2) random linear operators (e.g., random dense, spare, sign or aggregation matrices) and (b.3) non-linear mappings via random Multi-Layer Perceptrons (MLPs) to simulate complex static dependencies. (c.1) *Sequential transformations* introduce time-delayed relationships between time series. A random function maps a look-back window of univariate series to multivariate outputs. We generate (c.2) linear lead-lag relationships via a Temporal Causal Model, which samples sparse directed edges across time and variates (simulating Vector Autoregression), while (c.3) non-linear dynamic correlations are introduced via random Recurrent Neural Networks (RNNs).

In addition to the model architecture, the pretraining data plays an important role when developing a foundation model [18, 19]. While recent efforts have expanded the availability of large-scale time series datasets, they remain primarily univariate [9, 11]. To overcome this limitation, we developed a scalable pipeline for generating synthetic multivariate data that covers a comprehensive space of temporal patterns and interactions. We trained Chronos-2 on this dataset to equip it with the capability to understand multivariate interactions through ICL.

Our generation process mimics the complexity of real-world systems by hierarchically composing simple signals into complex interactions. We begin by sampling univariate time series from a diverse suite of generators (Figure 2a), ranging from structural models that capture trends and seasonality [20] to stochastic processes, e.g., exponential smoothing [21] and Gaussian processes [9]. These act as the fundamental building blocks, which provides a rich variety of univariate temporal dynamics.

We construct multivariate time series by coupling these univariate time series through two distinct interaction mechanisms: instantaneous and sequential.

Instantaneous interactions commonly occur in real-world applications, e.g., multiple sensors in a machine responding to a load change or retail sales across different store locations responding to the same holiday or weather event. These interactions introduce relationships among time series where there is no time delay. To capture these dynamics, we apply transformations across the univariate series at the same time step (Figure 2b). We model linear dependencies using mixing matrices to generate dense or sparse correlations. Non-linear interactions are synthesized via multi-layer perceptrons (MLPs) that map concurrent observations to a new multivariate space, creating non-linear relationships between variables at each time step.

Sequential interactions also occur in real-world systems, e.g., infection rates leading to hospitalization spikes or marketing spending preceding sales growth. These interactions often exhibit time-delayed relationships, where an event in one variable triggers a response in another after a specific duration. To capture these dynamics, we introduce lagged dependencies using two distinct mechanisms (Figure 2c). For linear temporal relationships, we synthesize Vector Autoregressive (VAR) processes [22] based on randomly sampled causal graphs by explicitly defining which variables drive others over time. For non-linear temporal dynamics, we apply Recurrent Neural Networks (RNNs) [23, 24] to the univariate history to create systems where the future evolution depends on complex, history-dependent interactions between variables that cannot be captured by simple linear lags.

In summary, we scale up pretraining data and introduce multivariate relationships through multivariate synthetic data generation. We train Chronos-2 on this corpus to enable it to infer the relationship between time series from the available context and react to them at inference time.

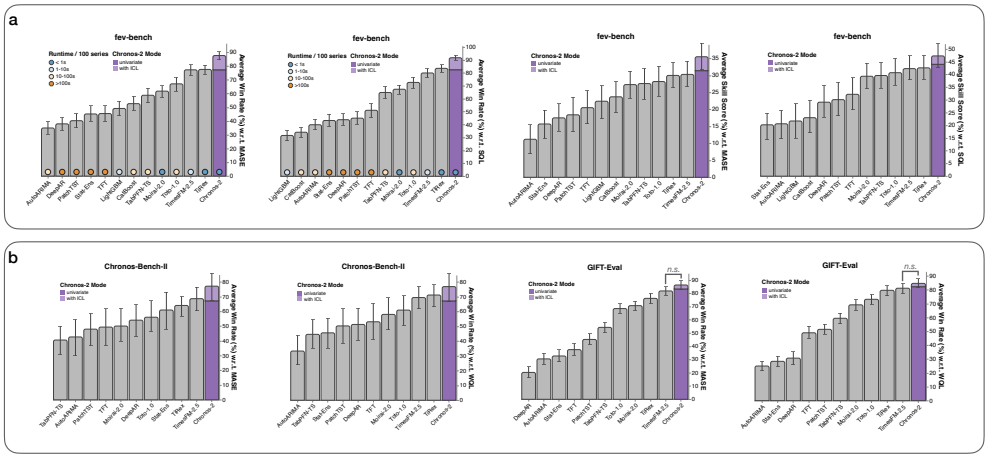


Fig. 3: **Evaluation of Chronos-2 on time series forecasting benchmarks** (*best viewed in color on a screen*). Performance is evaluated using the Mean Absolute Scaled Error (MASE) and Scaled/Weighted Quantile Loss (SQL/WQL) for point and probabilistic forecasts, respectively. The Win Rate measures the proportion of pairwise comparisons where a model outperforms competitors, while the Skill Score reflects the average percentage improvement over a Seasonal Naive baseline. Dark purple bars denote Chronos-2 in univariate mode (group attention disabled); light purple segments highlight the performance gain attributed to in-context learning (ICL) via group attention. Error bars indicate 95% confidence intervals obtained through bootstrapping. (a) Results on `fev-bench` ($n = 100$ tasks), a rigorous benchmark where 68 tasks involve multivariate dependencies. Chronos-2 outperforms all baselines, including statistical ensembles, tree-based methods, task-specific deep learning models, and existing foundation models, while maintaining fast inference speed; Wilcoxon signed-rank test $p < 10^{-6}$ against all baselines (see Tables 2 and 3 in the Appendix for results in tabular format and Table 4 for the p-values). (b) Results on `Chronos-Bench-II` ($n = 27$ tasks) and `GIFT-Eval` ($n = 97$ tasks), which focus primarily on univariate forecasting. Chronos-2 outperforms or ties state-of-the-art methods; notably, enabling ICL yields performance improvements even in these univariate settings by retrieving patterns across related series. *n.s.* indicates that the performance difference relative to Chronos-2 is not statistically significant ($p \geq 0.05$) based on the Wilcoxon signed-rank test; see Tables 5, 6, 8 and 9 in the Appendix for results in tabular format and Tables 7 and 10 for the p-values.

4 Zero-shot generalization across benchmarks

We assess the performance of Chronos-2 on `fev-bench` [25], a comprehensive benchmark comprising 100 forecasting tasks designed to evaluate time series models in a standardized setting. The `fev-bench` benchmark contains tasks spanning domains such as energy, retail, cloud computing, economics, healthcare, and mobility. It emphasizes real-world heterogeneity with 68 of the tasks involving multivariate dependencies.

These tasks require the model to handle multiple co-evolving target variables, past-only and known (future) covariates, and diverse data types (numeric, boolean, and categorical), often in the presence of missing values. To complement this multivariate focus, we also evaluated Chronos-2 on `Chronos-Bench-II` [9] and `GIFT-Eval` [26], which are two established benchmarks comprising 27 and 97 primarily univariate tasks, respectively.

We benchmarked Chronos-2 against a diverse spectrum of forecasting methods, including statistical models (AutoETS, AutoARIMA, AutoTheta, and their ensembles) [5, 27], tree-based methods (LightGBM [28], CatBoost [29]), task-specific deep learning models (DeepAR [6], TFT [7], PatchTST [8]), and state-of-the-art TSFMs (TiRex [30], TimesFM-2.5 [10], Toto-1.0 [12], Moirai-2.0 [31], and TabPFN-TS [19, 32]). Chronos-2 and other foundation models are pretrained and used directly without fitting on individual tasks in the benchmarks (zero-shot inference).

We evaluated the performance using Mean Absolute Scaled Error (MASE) for point forecasting and Scaled/Weighted Quantile Loss (SQL/WQL) for probabilistic forecasting. We report two complementary metrics: the win rate, representing the proportion of pairwise comparisons, where a model outperforms its competitors, and the skill score, which reflects the average percentage improvement over a baseline (Seasonal Naive).

Figure 3(a) illustrates the strong performance of Chronos-2 on `fev-bench`. The model establishes a new state-of-the-art by achieving an absolute MASE (SQL) win rate increment of 41.6 (47.8) percentage points over the best statistical model (Statistical Ensemble), 34.4 (56.7) over the best tree-based method (CatBoost), 41.8 (40.7) over the leading deep learning model (TFT), and 10 (8.3) over the strongest TSFM baseline (TiRex). On MASE (SQL) skill scores, Chronos-2 yields improvements of 11.8 (15.1) percentage points over the best task-specific model (CatBoost/TFT) and 5.2 (4.7) over the strongest TSFMs (TimesFM-2.5/TiRex). Notably, Chronos-2 achieves this accuracy while maintaining high throughput, with an average runtime of less than 1 second per 100 time series on an A10G GPU. These results also demonstrate the gains attributable to in-context learning: enabling group attention yields absolute improvements of 10.7 percentage points in the win rate and 4.1 in the skill score compared to univariate inference.

Lastly, we demonstrate that in addition to its multivariate capabilities, Chronos-2 is also effective at univariate forecasting. Figure 3(b) shows that Chronos-2 leads on both `Chronos-Bench-II` and `GIFT-Eval`. These benchmarks stress distinct capabilities: `Chronos-Bench-II` focuses on short-history tasks, and `GIFT-Eval` focuses on long-context and long-horizon prediction. Interestingly, enabling in-context learning yields improvements even on these univariate tasks. This effect is particularly pronounced on `Chronos-Bench-II`. These results suggest that when historical data is scarce, the group attention mechanism allows the model to "cross-learn" and stabilize predictions by retrieving relevant patterns from other time series.

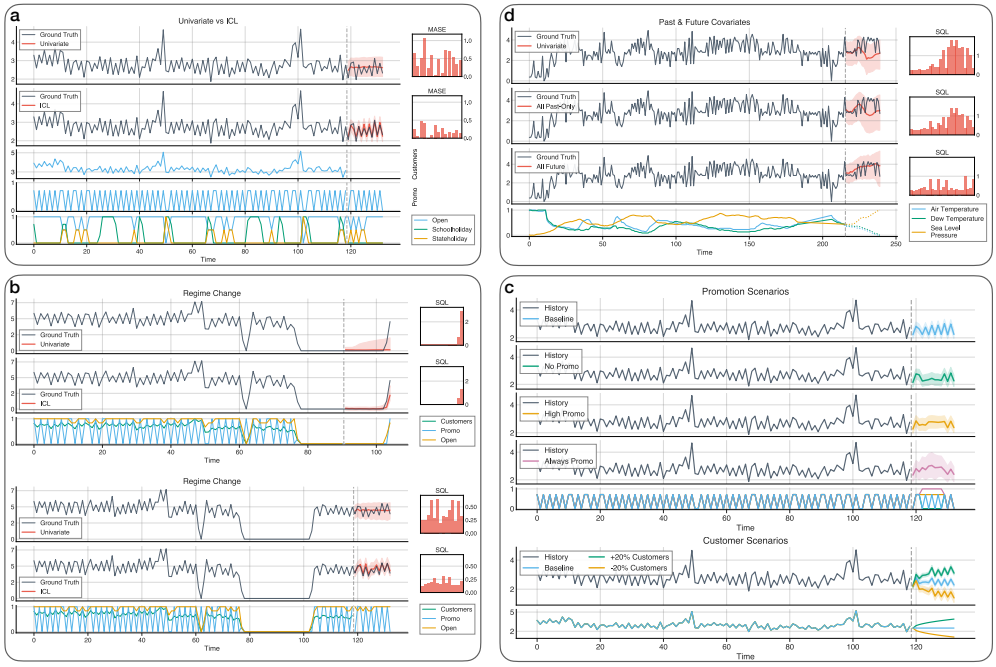


Fig. 4: **Qualitative analysis of in-context learning** (*best viewed in color on a screen*). Probabilistic forecasts show the 10% to 90% quantile range; bar charts show MASE and SQL forecast errors per time step. (a) Retail sales forecasting: univariate inference versus in-context learning with covariates including promotions, customer traffic, and holiday indicators. Covariates are shown in the lower panels. (b) Regime change: store closure at different positions relative to the forecast horizon. The top and bottom rows show the same series with different forecast origins, so that the closure falls within the forecast horizon (top) or entirely within the historical context (bottom). (c) Scenario analysis with simulated future covariates: promotion intensity (top four panels) and customer count trajectories (bottom panel) varied across hypothetical trajectories. (d) Energy load forecasting: univariate, past-only covariates, and full covariates (air temperature, dew temperature, sea-level pressure). The three rows show progressively more covariate information available to the model.

5 Qualitative analysis on in-context learning

Here, we aim to understand the mechanisms driving the strong performance on the quantative benchmarks, which requires taking a deeper look at how the model utilizes context. In particular, we present a qualitative analysis of scenarios from the retail and energy domains to demonstrate how in-context learning allows the model to understand multivariate interactions, respond meaningfully to counterfactual scenarios, and resolve ambiguity during regime shifts.

In several forecasting tasks, the target variable is heavily modulated by external factors. Figure 4(a) compares Chronos-2 in univariate mode against its default

in-context learning-based multivariate mode on a retail sales task from `fev-bench`. While the univariate model captures the level of the time series, it fails to anticipate fluctuations tied to external drivers such as promotions and holidays. By conditioning on these covariates via in-context learning, the multivariate model aligns closely with the ground truth, which results in sharpened prediction intervals and reduced error metrics.

A critical challenge in forecasting is handling regime changes — abrupt transitions between distinct behavioral states, such as store closures or system failures [33, 34]. Observable covariates often carry information about the latent regime state, which enable covariate-driven regime identification [35, 36]. Figure 4(b) illustrates how Chronos-2 exploits this principle: observed covariates allow it to identify regimes that are invisible to a univariate model. In scenarios where a store closes for an extended period, a univariate model is unaware of the cause of the zero-sales period, which results in inflated uncertainty. In contrast, Chronos-2 leverages the "Open" covariate to disambiguate the context. The group attention mechanism correctly identifies the closure regime, maintains tight confidence intervals during the inactive period, and anticipates the reopening at the boundary. This demonstrates the ability of Chronos-2 to leverage covariates to switch effectively between dynamic modes.

In addition to passive conditioning, we can also probe the model's understanding of the dynamics by feeding it hypothetical future covariate trajectories. Figure 4(c) visualizes this by modifying the promotion schedule for a retail sample. When the promotion activity is set to zero (no promo), the forecast settles to the baseline lower bound; when the promotion is set to high, the forecast shifts to the upper bound. Notably, when tested with an always-on promotion schedule — a scenario never seen in the history — the model predicts high sales but significantly widens its uncertainty bands. The model behavior reflects its epistemic uncertainty when forced to extrapolate to out-of-distribution conditions.

Chronos-2 can also use past-only drivers, such as customer counts, to simulate different scenarios where we do not know the future values of the covariate. While typically unavailable at forecast time, we can provide the model with synthetic future trajectories (stable average, 20% increased and 20% decreased) to simulate potential business scenarios. The model produces coherent forecasts, where sales volume tracks the simulated footfall intensity. This indicates that Chronos-2 has internalized the structural coupling between footfall and sales, which allows it to function as a forecaster and a simulator for strategic planning.

Finally, real-world deployment often involves scenarios where future values of certain covariates are unavailable. Figure 4(d) analyzes an electricity load forecasting task under varying information availability. We see that while known (future) covariates provide the highest accuracy, Chronos-2 extracts significant signal even from past-only covariates, which results in reduced error metrics compared to the univariate baseline. This shows the model can leverage historical covariate information to inform future dynamics of the target variable, yielding improvements over univariate forecasting.

6 Case studies

In this section, we apply Chronos-2 to various scientific problems — infectious disease dynamics, consumer price index forecasting, carbon flux prediction, and rainfall-runoff simulation. For each case study, we adopt the rigorous evaluation protocols of the respective discipline. In particular, we compare Chronos-2 directly against the specialized models and operational ensembles typically employed by domain experts. In addition, we report performance using the standard metrics of each field — e.g., the Weighted Interval Score in epidemiology or the Nash–Sutcliffe Efficiency in hydrology — to ensure our results are directly interpretable within the context of the domain-specific literature.

6.1 Epidemiology: Infectious disease dynamics

Accurate forecasting of acute respiratory infection (ARI) and influenza-like illness (ILI) is fundamental to public health preparedness, guiding resource allocation and intervention strategies during peak transmission periods [37]. We leveraged data from the ECDC RespiCast Hub [38], a collaborative platform that aggregates predictions from independent modeling teams to support decision-making at the European Centre for Disease Prevention and Control.

We benchmarked the zero-shot performance of Chronos-2 against the RespiCast Hub-Ensemble [38], an established operational benchmark that synthesizes outputs from diverse epidemiological approaches ranging from mechanistic compartmental models (SEIR) to statistical baselines. Following the official evaluation protocol, we employed the relative Weighted Interval Score (WIS) to assess the quality of probabilistic forecasts [39]. This comparison is inherently demanding, as the Hub-Ensemble benefits from the "wisdom of the crowd" and creates a consensus from multiple domain-specific and statistical signals [40].

Chronos-2 achieves competitive short-term forecasting performance solely by analyzing historical incidence trajectories (Figure 5(a)). This suggests that it effectively captures the statistical patterns characteristic of viral transmission without explicit knowledge of pathogen biology or social mixing patterns. Importantly, Chronos-2 surpasses the performance of the Hub-Ensemble at longer forecast horizons, especially in the case of ARI incidence. The in-context learning mechanism yields further gains here: by jointly predicting incidence rates across European nations, the model leverages cross-country correlations, thereby refining the forecast for the entire continent. These results suggest that general-purpose foundation models can complement specialized epidemiological modeling. Rather than replacing mechanistic approaches, Chronos-2 offers a rapidly deployable, high-quality option that can enhance existing ensembles, particularly in scenarios where specialized modeling teams are scarce or require significant lead time for calibration.

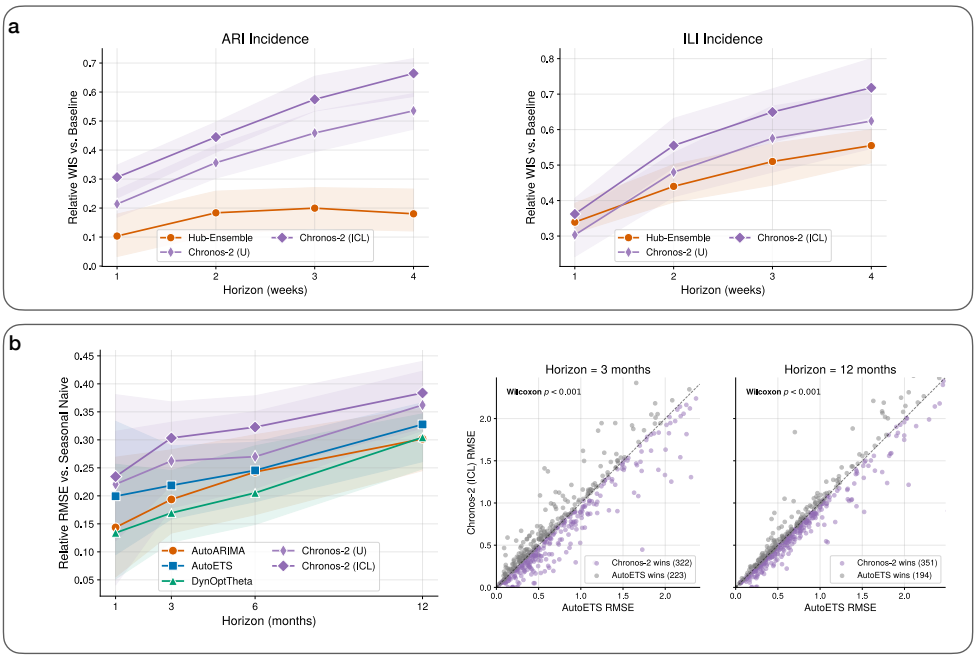


Fig. 5: **Zero-shot forecasting of complex human-centric systems.** (a) Forecasting incidence of Acute Respiratory Infection (ARI) and Influenza-Like Illness (ILI) across 31 European countries. Performance is evaluated using the Weighted Interval Score (WIS) relative to a simple quantile baseline provided by RespiCast [38], calculated as $\log_2(\text{WIS}_{\text{baseline}}/\text{WIS}_{\text{model}})$; values greater than 0 indicate improvement over the baseline. Plots show the median relative WIS (solid lines) and 95% confidence intervals (shaded bands) over 67 rolling evaluation windows (Oct 2024–Jan 2026). In univariate mode (U), Chronos-2 is competitive at short horizons (1 week) against the operational RespiCast Hub-Ensemble (a consensus of mechanistic and statistical models). At longer horizons ($>$ 1 week), Chronos-2 significantly outperforms the ensemble, with in-context learning yielding further gains by exploiting cross-country correlations (Wilcoxon signed-rank test $p < 0.001$). (b) Joint forecasting of German Consumer Price Index (CPI) components (2021–2025). Performance is measured by the Root Mean Squared Error (RMSE) relative to a Seasonal Naive baseline, $\log_2(\text{RMSE}_{\text{baseline}}/\text{RMSE}_{\text{model}})$. While univariate inference (U) performs comparably to robust statistical baselines, enabling in-context learning significantly improves accuracy by leveraging shared structural patterns within CPI groups. Pairwise comparisons (right) confirm that Chronos-2 outperforms the strongest baseline (AutoETS) on the majority of item-window pairs (Wilcoxon signed-rank test $p < 0.001$ for both 3-month and 12-month horizons).

6.2 Macroeconomics: Consumer price index forecasting

Forecasting the components of the Consumer Price Index (CPI) is fundamental to the conduct of monetary policy [41]. Challenges in macroeconomic forecasting include lower frequency (monthly) data, limited historical samples, and significant signal-to-noise challenges. In this regime, complex machine learning models frequently fail to generalize, often underperforming simpler statistical baselines [42]. Methods such as ARIMA and ETS are often difficult to surpass, as they capture trends without overfitting to short-term volatility [5].

Here, we tackle the challenging problem of jointly forecasting its granular components, where existing literature has typically focused on predicting headline inflation as a single aggregate. We evaluate Chronos-2 on the German CPI, decomposed into 110 product groups at Level 4 of the COICOP standard — a granularity sufficient to isolate sector-specific drivers while smoothing idiosyncratic item-level noise.

The challenge of achieving performance gains in this regime is evident in Figure 5(b), which illustrates the tight performance margins among methods. Chronos-2 in univariate mode performs comparably against robust statistical baselines (AutoARIMA, AutoETS, and DynOptTheta). In-context learning significantly improves the model's capability. Rather than processing the 110 components individually, ICL jointly forecasts items within each Level 2 COICOP group (e.g., "Food and non-alcoholic beverages"), exploiting natural clusters of economically related products. As shown in the pairwise comparisons, Chronos-2 (ICL) consistently outperforms AutoARIMA and AutoETS, achieving significantly lower errors on the majority of items (Wilcoxon signed-rank test $p < 0.001$) at both 3-month and 12-month horizons. These results show that foundation models yield gains even in regimes traditionally dominated by classical statistical models.

6.3 Earth Science: Carbon flux prediction

Eddy-covariance networks provide the observational backbone for understanding land–atmosphere feedbacks and the terrestrial carbon cycle. We evaluated Chronos-2 on the prediction of Net Ecosystem Exchange (NEE) and turbulent energy fluxes — Latent Heat (LE) and Sensible Heat (H) — using the standardized FLUXNET collection [43]. While traditional methodologies focus on gap-filling (imputing missing historical data) [44], we address the problem of forecasting fluxes based on meteorological drivers. These forward-looking estimates have broad operational utility, ranging from optimizing irrigation in precision agriculture to monitoring natural vegetation stress under increasing atmospheric aridity [45].

We structured this problem as a daily multivariate forecasting task, using air temperature (TA), incoming shortwave radiation (SW_IN), and vapour pressure deficit (VPD) as covariates. We focused on the boreal summer (June–August 2020), the peak period of biological activity and surface-atmosphere exchange. Chronos-2 operates entirely zero-shot, whereas conventional baselines are fully supervised, trained on all available historical observations prior to the test period. As the supervised baseline, we selected XGBoost [46] which is a standard in flux modeling [44]. To reflect common

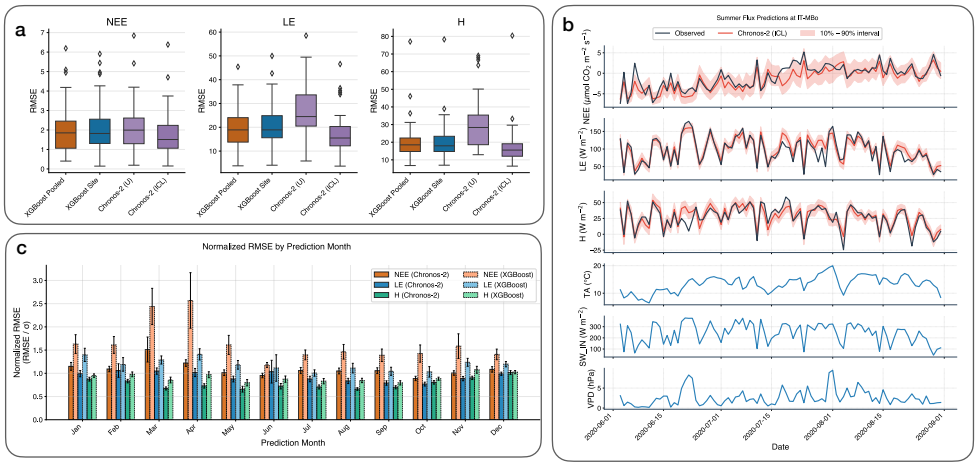


Fig. 6: **Ecosystem carbon and energy flux prediction with Chronos-2.** We evaluate the performance of Chronos-2 on daily eddy-covariance targets from the FLUXNET collection [43]. (a) Comparison of the distribution of site-level RMSE across 65 sites for net ecosystem exchange (NEE), latent heat flux (LE), and sensible heat flux (H) over the June–August 2020 evaluation window between supervised regression baselines (XGBoost, trained either as a single pooled model across sites or as site-specific models) and Chronos-2 in zero-shot mode. Box plots show the median (center line), interquartile range (box limits), and 1.5× interquartile range (whiskers); points denote outliers. (b) Representative summer time series at IT-MBo, showing observed fluxes (black) and Chronos-2 predictions from the meteorological covariates with 10–90% prediction intervals (red shaded); the corresponding covariates (TA, SW_IN and VPD) are shown below. (c) Rolling one-month-ahead forecasting over the full year 2020. For each evaluation month, models condition on the preceding 12 months of context and predict the subsequent month (step size: 1 month). Bars report normalized RMSE (RMSE divided by the target standard deviation) stratified by prediction month for Chronos-2 and XGBoost, highlighting seasonal variations in predictability across NEE, LE, and H.

modeling choices in heterogeneous multi-site datasets, we considered both a pooled XGBoost model (trained across all sites) and site-specific variants.

Figure 6(a) shows that Chronos-2 with covariates (ICL) achieves the lowest RMSE distributions across NEE, LE, and H (Wilcoxon signed-rank test $p < 0.01$) without requiring any domain-specific training. ICL clearly outperforms Chronos-2 in univariate mode (U) implying that the model effectively leverages meteorological forcing rather than relying on only autoregression. Qualitatively, Chronos-2 demonstrates an understanding of physical coupling between meteorological forcing and surface exchange; for instance, it correctly predicts increases in latent heat flux during periods of high shortwave radiation, which is consistent with the fundamental principles of surface energy balance where evapotranspiration is constrained by available energy [47].

Lastly, error stratification by month (Figure 6c) reveals consistent seasonal predictability without requiring training or calibration compared to its task-specific counterpart. Collectively, these results demonstrate that a generalist foundation model can generate physically meaningful and uncertainty-aware ecosystem forecasts that outperform strong supervised regressors.

6.4 Hydrology: Rainfall-runoff simulation

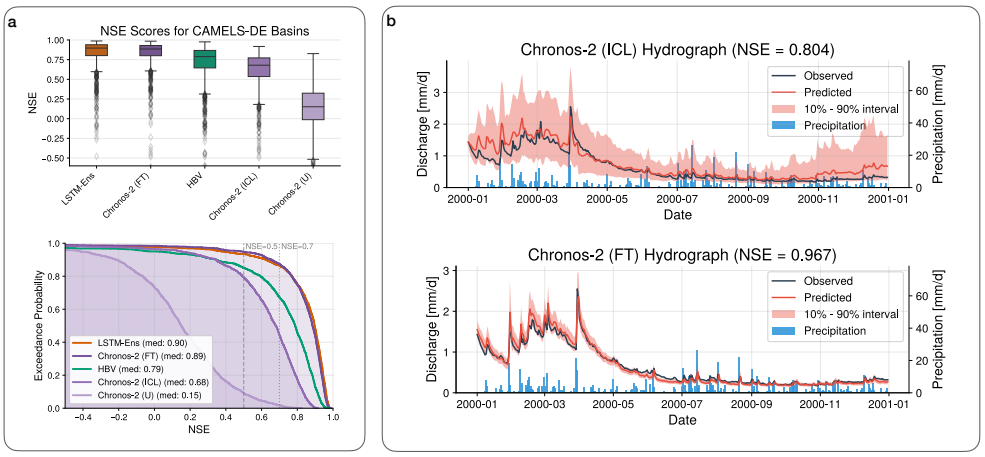


Fig. 7: **Rainfall-runoff simulation using Chronos-2.** Performance is evaluated using the Nash–Sutcliffe Efficiency (NSE), where a score of 1.0 represents perfect correspondence between simulated and observed discharge. (a) Top: Distribution of NSE scores across 1,582 catchments in the CAMELS-DE dataset. The zero-shot Chronos-2 (ICL) model achieves a median NSE of 0.68, significantly outperforming univariate inference (Chronos-2 (U), NSE=0.15) by leveraging meteorological covariates. While the domain-specific LSTM ensemble achieves higher performance (NSE=0.90), fine-tuning Chronos-2 (FT), bridges this gap, matching state-of-the-art performance without ensembling, access to static attributes, or training on the NSE metric. Box plots show the median (center line), interquartile range (box limits), and 1.5x interquartile range (whiskers). Bottom: Exceedance probability curves showing the cumulative proportion of basins (y-axis) achieving an NSE score greater than or equal to a specific threshold (x-axis). The fine-tuned model closely tracks the reliability profile of the LSTM ensemble. (b) Hydrograph for a single catchment (DE110180) over one year. The zero-shot prediction (top) captures the relationship between rainfall and runoff but exhibits high uncertainty (shaded region denotes 10-90% prediction interval). Fine-tuning (bottom) tightens the predictive distribution, accurately modeling the discharge.

Rainfall-runoff simulation is a central problem in hydrology, which is critical for reservoir management, flood forecasting, and climate resilience. It studies how rainfall

over a watershed is transformed into surface runoff (streamflow) at the basin outlet [4]. We evaluated Chronos-2 for this problem on the CAMELS-DE dataset [48], a high-quality benchmark providing hydro-meteorological time series for 1,582 catchments across Germany.

To contextualize performance, we compared Chronos-2 against two domain-specific standards: the HBV model [49], a conceptual physical model calibrated individually for each catchment, and a state-of-the-art LSTM ensemble [50], which represents the current upper bound in hydrological deep learning. The LSTM baseline is trained on data from all catchments to predict the discharge using precipitation, radiation, temperature, and static catchment attributes (e.g., soil texture, elevation) as inputs.

Figure 7(a) presents the distribution of Nash–Sutcliffe Efficiency (NSE) scores [51], a metric commonly used to evaluate rainfall–runoff models. Remarkably, Chronos-2 (with ICL) achieves a median NSE of 0.68. Without ever seeing a hydrograph during training, the model successfully identifies the relationship between precipitation events and streamflow response. This performance is starkly superior to univariate inference (median NSE = 0.15), confirming that the model is actively leveraging meteorological covariates rather than simply autoregressing on historical discharge.

While zero-shot performance lags behind the highly specialized baselines, Chronos-2 provides viable hydrological forecasts instantly, without the training or calibration these baselines require. Lightweight fine-tuning on CAMELS-DE data effectively bridges this gap, with Chronos-2 matching the performance of the state-of-the-art LSTM ensemble — despite using a single model without static catchment attributes or the NSE objective function. Figure 7(b) illustrates this progression: in zero-shot mode, the model captures the general shape of discharge but exhibits high uncertainty; after fine-tuning, the predictive distribution tightens around the ground truth, effectively learning the rainfall-runoff relationship. This demonstrates Chronos-2's flexibility as both an immediately useful generalist and a robust initialization for specialist models.

7 Conclusion

In this work, we introduced Chronos-2, a time series foundation model designed to handle a wide range of forecasting scenarios — both univariate and multivariate — in a zero-shot manner. Chronos-2 marks a shift from fitting parameters to inferring dynamics. Unlike traditional approaches that require training bespoke models for every dataset, Chronos-2 implicitly infers coupled dynamics purely from historical context. By training extensively on multivariate synthetic data, we demonstrated that the fundamental grammar of temporal variation can be learned and transferred to real-world complexity, effectively bridging the gap between domains ranging from hydrology to macroeconomics. Chronos-2 democratizes access to advanced forecasting by eliminating the barrier of training task-specific models, offering a versatile tool for understanding the coupled dynamics of the world.

Building on this framework, future development of Chronos-2 can expand along several critical dimensions. First, scaling the diversity of the synthetic pretraining corpus — e.g., by incorporating outputs from domain-specific simulators — can further enhance zero-shot generalization for highly complex dynamics. Second, while the

current architecture natively handles time-varying covariates, integrating static metadata (e.g., soil composition or demographic profiles) is an important frontier. This capability is vital for extreme *cold-start* scenarios, e.g., the grand challenge of prediction in ungauged basins (PUB) in hydrology, where streamflow must be inferred entirely from meteorological drivers and static basin properties without historical target data. In addition, since critical exogenous signals are often qualitative (e.g., central bank statements or epidemiological reports), extending the model to support multimodal natural language inputs is a promising direction. As these foundation models assume a larger role in science and decision-making, advancing their interpretability is also essential. This includes exploring methods to extract explicit causal graphs or coupling mechanisms directly from the model's internal representations. Lastly, the universal temporal representations learned by Chronos-2 hold immense potential for broader downstream applications beyond forecasting, including anomaly detection, classification, and clustering.

Methods

Problem statement

Time series forecasting aims to predict future values of a temporal sequence given historical observations. Formally, let $\boldsymbol{Y}_{1:T} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$ denote a historical time series of length T, where each observation $\boldsymbol{y}_t \in \mathbb{R}^D$ can either be univariate ($D = 1$) or multivariate ($D > 1$). Given this historical context, the goal is to predict the next H time steps $\boldsymbol{Y}_{T+1:T+H}$, where H defines the forecast horizon. Forecasts may be supported by covariates (also known as *exogenous variables*) $\boldsymbol{X}_{1:T+H} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+H}]$, where $\boldsymbol{x}_t \in \mathbb{R}^M$ 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 $\mathcal{P}(\boldsymbol{Y}_{T+1:T+H} \mid \boldsymbol{Y}_{1:T}, \boldsymbol{X}_{1:T+H})$ 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.

Chronos-2 architecture

Chronos-2 is a 120M parameter encoder-only transformer [13] model which closely follows the design of the T5 encoder [14]. In the following, we discuss the key architectural components of the Chronos-2 pipeline.

Input construction

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

Categorical covariates in $\boldsymbol{X}_{1:T+H}$ are transformed into real-valued representations before being concatenated into $\boldsymbol{V}$ and $\boldsymbol{W}$. For univariate targets, we apply target encoding [52, 53], 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, $\boldsymbol{V}$ and $\boldsymbol{W}$, 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. Formally,

each historical value $v_{t,d}$ and the future value $w_{t,d}$ are normalized as

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

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

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

Meta features

During tokenization, each dimension of $\boldsymbol{U}$ is processed independently by the model. To describe the tokenization procedure, consider a single column $\boldsymbol{u}_d = [u_{1,d}, \ldots, u_{T+H,d}]^\top$ 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 $\boldsymbol{j} = \left[-\frac{T}{C}, -\frac{T-1}{C}, \ldots, 0, \ldots, \frac{H-1}{C}\right]$ 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 $\boldsymbol{m}_d$ 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 known (future) covariates. After construction of the mask, all missing values in $\boldsymbol{u}_d$ are replaced with zeros.

Patching and embedding

The input $\boldsymbol{u}_d$ with the corresponding meta features, $\boldsymbol{j}$ and $\boldsymbol{m}_d$, are split into non-overlapping patches of length $P = 16$ [8]. 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 $\overline{\boldsymbol{u}}_p$, $\overline{\boldsymbol{j}}_p$, and $\overline{\boldsymbol{m}}_p$ 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, $f_\phi^{\text{in}} : \mathbb{R}^{3P} \to \mathbb{R}^{D_{\text{model}}}$,

$$\boldsymbol{h}_p = f_\phi^{\text{in}}\left(\left[\overline{\boldsymbol{u}}_p, \overline{\boldsymbol{j}}_p, \overline{\boldsymbol{m}}_p\right]\right), \tag{3}$$

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 [54].

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) [55] which have become the de-facto standard for position embeddings in modern transformer-based models [18].

Group attention

The *group attention* layer 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.

Within a batch of size B, multiple groups of varying sizes are possible, each identified by group IDs $\boldsymbol{g}$, 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 $\hat{\boldsymbol{Z}} \in \mathbb{R}^{H \times D \times |\mathcal{Q}|}$. 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 $\mathcal{Q} = \{0.01, 0.05, 0.1, \ldots, 0.9, 0.95, 0.99\}$. Any queried quantiles falling outside this set are obtained through linear interpolation.

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 (future) 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 $\boldsymbol{g}$ with whether the future input $\boldsymbol{W}$ is observed allows the model to infer the specific forecasting setup.

The model is trained using the quantile regression objective

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

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 (future) covariates or missing target values excluded from the objective. The number of output patches is randomly sampled for each batch during training.

The model is trained in two stages using the Adam optimizer with weight decay of 0.1 and maximum gradient norm set to 1. First, the model is pretrained with a maximum context length of 2048, 12 maximum output patches and learning rate of 10^{-4}. In the second stage, the context length is extended to 8192, the maximum number of sampled output patches is increased to 64 and the learning rate is reduced to 10^{-5}. 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.

Training data

The training corpus combines real univariate datasets (Supplementary Table 1) with synthetic multivariate data as discussed in Section 3.

Our multivariate synthetic data generation process begins with univariate generators capable of producing a rich diversity of temporal patterns. A randomly chosen number of time series are first sampled from these univariate generators. A function is then applied to these univariate series to introduce correlations between them.

The set of univariate generators includes: (i) **KernelSynth** [9], which generates time series using Gaussian processes (GP) by randomly combining basic kernels (such as linear and periodic) through addition and multiplication to construct complex composite kernels; (ii) **Trend-Seasonality-Irregularity (TSI)** [20], which exploits typical time series decomposition to construct series by sampling trends, seasonalities, and irregularities from a suite of predefined components and combining them through an additive or multiplicative model; (iii) **AutoRegression (AR)** [56], which constructs a random order-p autoregressive model and simulates it to generate time series; and (iv) **Exponential Smoothing (ETS)** [21], which simulates time series from a randomly constructed exponential smoothing model.

With time series sampled from the aforementioned univariate generators, we construct multivariate data by applying suitable random transformation functions which introduce instantaneous or time-delayed relationships between the time series.

Instantaneous transformations applied across the univariate series at the same time-step. We use two types of instantaneous transformation functions: linear and non-linear. The linear function is realized by a simple transformation matrix which may be dense, sparse, or a sign matrix, or may encode other relationships such as aggregation, where the number of output time series is larger than the number of input series and the additional outputs denote summations of the input series. The random non-linear transformation is characterized by a multi-layer perceptron (MLP), where the number of layers, type of nonlinearities, and number of hidden units are randomly sampled.

Time-delayed transformations are functions which take a window of the last k time-steps from the univariate time series as inputs and are recursively applied to yield multivariate time series with time-varying relationships. As before, we use linear and non-linear transformation functions. The linear function is characterized by a temporal causal model (TCM). The algorithm first selects a subset of the univariate series as exogenous variables — they mimic external factors which affect the system but are not affected by it. A set of edges across time and variables are then sampled to

denote relationships, with edge weights sampled from a truncated normal distribution. This yields the parameters of a vector autoregressive (VAR) process [22] which is then applied to the univariate series. For non-linear time-delayed relationships, we apply a recurrent neural network (RNN) to the univariate time series, where the number and type of layers are randomly sampled.

Inference

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

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

to obtain the prediction $\hat{y}^q_{t,d}$ 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 $\boldsymbol{W}$ corresponding to known covariates contain their known future values. The predictions generated by the model for covariates are ignored.

Evaluation protocol for standard benchmarks

In this section, we detail our evaluation protocol including the benchmark datasets, baselines and metrics used in the experiments.

Benchmarks

We evaluated Chronos-2 on three established forecasting benchmarks: `fev-bench` [25], `Chronos-Bench-II` [9], and `GIFT-Eval` [26]. We strictly adhered to the data splits and evaluation protocols defined in the respective publications. `fev-bench` provides comprehensive coverage of diverse real-world scenarios, comprising 100 forecasting tasks (68 of which involve multivariate dependencies). `Chronos-Bench-II`, originally proposed to evaluate early Chronos models [9], consists of 27 tasks characterized predominantly by short historical contexts (averaging fewer than 300 time steps). `GIFT-Eval` focuses on high-frequency time series and long-horizon forecasting, containing 97 tasks derived from 55 distinct datasets.

Baselines

We compared Chronos-2 against a range of leading task-specific models (trained individually on each benchmark task) and pretrained foundation models. Task-specific baselines included statistical methods (AutoETS, AutoARIMA, AutoTheta, and their ensemble [5, 27]), tree-based methods (LightGBM [28], CatBoost [29]), and deep learning architectures (DeepAR [6], TFT [7], PatchTST [8]). Foundation model baselines included TiRex [30], TimesFM-2.5 [10], Toto-1.0 [12], Chronos-Bolt [9], TabPFN-TS [19, 32], and Moirai-2.0 [31].

Where available, we utilized reported baseline results from the original benchmarking papers; otherwise, we reproduced results using standard libraries. Statistical models were implemented via StatsForecast [57] (max. runtime 12 hours/task) on AWS `m6i.4xlarge` instances (16 vCPUs, 64 GB RAM). Tree-based models were implemented via MLForecast; to address sensitivity to preprocessing, we tuned differencing, scaling, and lag transformations for each task, selecting lag/time features via frequency-based heuristics. Deep learning models were implemented via AutoGluon [58] on AWS `g5.2xlarge` instances (A10G GPU, 24 GB VRAM). To prevent under-fitting, we extended the training budget to 1,000 epochs (with early stopping) and a 24-hour maximum runtime. Task-specific models were re-trained for each evaluation window to account for distribution shifts. Foundation models were evaluated using their respective official libraries on `g5.2xlarge` instances.

Metrics and aggregation strategy

We assessed both point and probabilistic forecasting performance. Point forecast accuracy was evaluated using the Mean Absolute Scaled Error (MASE), which normalizes the mean absolute error by the historical seasonal error. For a task with N time series and forecast horizon H, MASE is defined as:

$$\text{MASE} = \frac{1}{N}\frac{1}{H}\sum_{i=1}^{N}\frac{1}{a_i}\sum_{t=T+1}^{T+H}|y_{i,t} - f_{i,t}|, \tag{6}$$

where a_i is the historical absolute seasonal error, given by:

$$a_i = \frac{1}{T-m}\sum_{t=m+1}^{T}|y_{i,t} - y_{i,t-m}|, \tag{7}$$

and m is the seasonal period.

Probabilistic forecast quality was measured using the Scaled Quantile Loss (SQL), which aggregates the pinball loss across quantile levels, normalized by the same seasonal error a_i:

$$\text{SQL} = \frac{1}{N}\frac{1}{H}\sum_{i=1}^{N}\frac{1}{a_i}\sum_{t=T+1}^{T+H}\sum_{q}\rho_q(y_{i,t}, f_{i,t}^{q}), \tag{8}$$

where $f_{i,t}^q$ denotes the predicted quantile q for time series i at time t, and the quantile loss ρ_q is defined as:

$$\rho_q(y_{i,t}, f_{i,t}^q) = \begin{cases} 2(1-q)(f_{i,t}^q - y_{i,t}), & \text{if } y_{i,t} < f_{i,t}^q \\ 2q(y_{i,t} - f_{i,t}^q), & \text{if } y_{i,t} \geq f_{i,t}^q. \end{cases} \tag{9}$$

Consistent with the original benchmarking protocols, we used the Weighted Quantile Loss (WQL) for `Chronos-Bench-II` and `GIFT-Eval`. Both SQL and WQL approximate the Continuous Ranked Probability Score (CRPS); however, unlike SQL, WQL depends on the scale of the time series.

To aggregate results across tasks, we report average win rates (W) and skill scores (S) following the methodology in Ref. [25]. These metrics are mathematically equivalent to the average rank (R) and geometric mean relative error (G) used in prior work [9, 26], where $R = 1 + (1 - \frac{W}{100})(N-1)$ and $G = 1 - \frac{S}{100}$. We prefer win rates and skill scores for their interpretability: the win rate indicates the proportion of pairwise comparisons where a model is superior, while the skill score represents the average percentage improvement over a baseline (here, the Seasonal Naive model). Statistical significance of Chronos-2's performance against baselines was assessed using the Wilcoxon signed-rank test.

Details on case studies

In this section, we provide the details for the various real-world case studies ranging from epidemiology and macro-economics to earth sciences and hydrology.

Infectious disease dynamics

We utilized incidence data for Acute Respiratory Infection (ARI) and Influenza-Like Illness (ILI) spanning October 2024 to January 2026, sourced from the European Centre for Disease Prevention and Control (ECDC) via the RespiCast repository [38]. To preclude look-ahead bias from data revisions, Chronos-2 was evaluated on historical data snapshots, strictly adhering to the RespiCast protocol. We evaluated two configurations: Chronos-2 (U), a univariate mode where each country's incidence is forecast in isolation; and Chronos-2 (ICL), an in-context learning mode where forecasts are generated jointly across countries. Baseline forecasts for the quantile model and the RespiCast Hub-Ensemble were obtained directly from the RespiCast snapshots.

Consumer price index forecasting

Data for German Consumer Price Index (CPI) components were retrieved from GENESIS-Online, the central database of the German Federal Statistical Office, on January 19, 2026. We selected 110 components at Level 4 of the Classification of Individual Consumption According to Purpose (COICOP) for evaluation. A rolling window evaluation was conducted annually from 2021 to 2025, capturing the post-COVID-19 economic recovery period. Statistical baselines were implemented using the StatsForecast library. For Chronos-2 (U), items were forecast independently; for Chronos-2 (ICL), items within the same Level 2 COICOP group were predicted jointly.

Carbon flux prediction

We utilized daily eddy-covariance measurements of carbon and energy fluxes (NEE, LE, H) alongside meteorological drivers (TA, SW_IN, VPD) from the FLUXNET dataset [43]. Analysis was restricted to sites with high-quality (>80% quality control flag), continuous observation coverage across the context and evaluation periods. Evaluations included a static summer window (June–August 2020) and a monthly rolling forecast for the entire year 2020, utilizing a 12-month historical context to predict the subsequent month. We evaluated two zero-shot configurations: Chronos-2 (U), utilizing only the isolated target flux history; and Chronos-2 (ICL), conditioned jointly on the historical fluxes and future meteorological covariates. Supervised baselines consisted of pooled and site-specific XGBoost regressors [46] trained on available historical data. Forecast accuracy was evaluated using RMSE and normalized RMSE.

Rainfall-runoff simulation

We utilized the CAMELS-DE dataset for hydrological modeling. Pre-computed simulations for the HBV and LSTM-Ensemble baselines were obtained directly from the dataset. These baselines were calibrated on hydrometeorological data spanning 1970 to 1999. For this study, we selected the subsequent year (2000) as the evaluation period. Chronos-2 was conditioned on historical context from 1970 to 1999 to predict streamflow for the year 2000, utilizing precipitation, radiation, and temperature as future covariates. To ensure fair comparison, Chronos-2 was fine-tuned on the same 1970–1999 data interval used for the baselines. Fine-tuning was performed using the Adam optimizer with a learning rate of 10^{-4} for 10,000 gradient update steps.

Code and data availability. The model weights are available on HuggingFace at https://huggingface.co/amazon/chronos-2 under the Apache-2.0 license. A python package to run the model is available at https://github.com/amazon-science/chronos-forecasting. The evaluation datasets are publicly available at: `fev-bench` (https://huggingface.co/datasets/autogluon/fev_datasets), `Chronos-Bench-II` (https://huggingface.co/datasets/autogluon/chronos_datasets) and `GIFT-Eval` (https://huggingface.co/datasets/Salesforce/GiftEval). Evaluation scripts are available at https://github.com/autogluon/fev and https://github.com/SalesforceAIResearch/gift-eval.

Supplementary information. None.

Acknowledgements. We thank the developers of open-source libraries used in the development of Chronos-2, including but not limited to `torch` [59], `numpy` [60], `pandas` [61, 62], `statsmodels` [63], `transformers` [64], `gluonts` [65], `autogluon` [58], `statsforecast` [57], `einops` [66] and `scikit-learn` [52]. We thank Caner Turkmen, Tam Nguyen, Suprabhath Kalahasti, Matteo Chinazzi, Imry Kissos, Devamanyu Hazarika, Tim Januschowski, Jan Gasthaus, William Gilpin, Annan Yu, Andrew Gordon Wilson and Michael W. Mahoney for their feedback.

Declarations

- **Competing interests**: The authors are affiliated with Amazon Web Services and declare no competing interests.
- **Author contributions**: A.F.A. conceived the idea, developed the training infrastructure, designed the model architecture, contributed to the synthetic data generation, conducted experiments, and wrote the paper. O.S. developed the benchmarking infrastructure, contributed to the model architecture, conducted experiments, and wrote the paper. J.K. contributed to the synthetic data generation and training protocol, conducted experiments, and wrote the paper. A.A. contributed to improving the training data augmentation, conducted qualitative analysis, and wrote the paper. B.H. conducted the case study on carbon flux estimation and wrote the paper. P.M. contributed to the experiment design and conducted initial experiments on tasks with covariates. S.S.R. contributed to the synthetic data generator based on Gaussian processes. H.S. conducted initial experiments on synthetic data generation recipes and provided feedback on the experiment design. L.S. prepared the corpus of real univariate datasets, designed the user-facing API, and refactored the code for external release. X.Z. contributed fine-tuning support and provided feedback on the training protocol. M.G. contributed to the case study on epidemiology and provided feedback on the model architecture. S.K. contributed to improving the efficiency of the model through multi-patch prediction. D.M. provided technical advice on the scientific case studies and wrote the paper. P.G. provided feedback on the design of multivariate synthetic data and contributed to the macroeconomics case study. F.S. provided technical advice and wrote the paper. M.R. helped design the carbon flux estimation case study and provided technical feedback. T.H. and P.M.D. prepared the training infrastructure for large-scale training. J.Y. contributed to the development of the training recipe. N.E. contributed to supporting categorical covariates and designing the evaluation protocol. H.W. contributed to the design and verification of the case studies. H.R., G.K, B.W, and M.B-S. supervised the project, contributed ideas and technical advice, and wrote the paper.

References

[1] Vespignani, A. *et al.* Modelling covid-19. *Nature Reviews Physics* **2**, 279–281 (2020).

[2] Sims, C. A. Macroeconomics and reality. *Econometrica: journal of the Econometric Society* 1–48 (1980).

[3] Reichstein, M. *et al.* Deep learning and process understanding for data-driven earth system science. *Nature* **566**, 195–204 (2019).

[4] Beven, K. J. *Rainfall-runoff modelling: the primer* (John Wiley & Sons, 2012).

[5] Hyndman, R. J. & Athanasopoulos, G. *Forecasting: principles and practice* (OTexts, 2018).

[6] Salinas, D., Flunkert, V., Gasthaus, J. & Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting* **36**, 1181–1191 (2020).

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

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

[9] Ansari, A. F. *et al.* Chronos: Learning the language of time series. *Transactions on Machine Learning Research* (2024).

[10] Das, A., Kong, W., Sen, R. & Zhou, Y. *A decoder-only foundation model for time-series forecasting*. Forty-first international conference on machine learning (2024).

[11] Woo, G. *et al.* Unified training of universal time series forecasting transformers. *arXiv:2402.02592* (2024).

[12] Cohen, B. *et al. This time is different: An observability perspective on time series foundation models*. Advances in Neural Information Processing Systems (2025).

[13] Vaswani, A. *et al. Attention Is All You Need*. Advances in Neural Information Processing Systems (2017).

[14] Raffel, C. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* **21**, 5485–5551 (2020).

[15] Koenker, R. & Bassett, G., Jr. Regression quantiles (1978).

[16] Wen, R., Torkkola, K., Narayanaswamy, B. & Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053* (2017).

[17] Camps-Valls, G. *et al.* Artificial intelligence for modeling and understanding extreme weather and climate events. *Nature Communications* **16**, 1919 (2025).

[18] Touvron, H. *et al.* Llama 2: Open Foundation and Fine-Tuned Chat Models (2023). arXiv:2307.09288.

[19] Hollmann, N., Müller, S. *et al.* Accurate predictions on small data with a tabular foundation model. *Nature* (2025).

[20] Bahrpeyma, F., Roantree, M., Cappellari, P., Scriney, M. & McCarren, A. A methodology for validating diversity in synthetic time series generation. *MethodsX* **8**, 101459 (2021).

[21] Hyndman, R., Koehler, A., Ord, K. & Snyder, R. *Forecasting with exponential smoothing: the state space approach* (Springer, 2008).

[22] Lütkepohl, H. *Introduction to multiple time series analysis* (Springer Science & Business Media, 2013).

[23] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).

[24] Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. *Deep learning* Vol. 1 (MIT press Cambridge, 2016).

[25] Shchur, O. *et al.* fev-bench: A realistic benchmark for time series forecasting (2025).

[26] Aksu, T. *et al.* GIFT-eval: A benchmark for general time series forecasting model evaluation (2024).

[27] Petropoulos, F. & Svetunkov, I. A simple combination of univariate models. *International journal of forecasting* **36**, 110–115 (2020).

[28] Ke, G. *et al.* Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **30** (2017).

[29] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V. & Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems* **31** (2018).

[30] Auer, A. *et al. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning.* Advances in Neural Information Processing Systems (2025).

[31] Liu, C. *et al.* Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698* (2025).

[32] Hoo, S. B., Müller, S., Salinas, D. & Hutter, F. From tables to time: How TabPFN-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945* (2025).

[33] Quandt, R. E. The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the American Statistical Association* 873–880 (1958).

[34] Hamilton, J. D. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the econometric society* 357–384 (1989).

[35] Filardo, A. J. Business-cycle phases and their transitional dynamics. *Journal of Business & Economic Statistics* **12**, 299–308 (1994).

[36] Auer, A., Gauch, M., Klotz, D. & Hochreiter, S. Conformal prediction for time series with modern hopfield networks. *Advances in neural information processing systems* **36**, 56027–56074 (2023).

[37] Viboud, C. & Vespignani, A. The future of influenza forecasts. *Proceedings of the National Academy of Sciences* **116**, 2802–2804 (2019).

[38] European Centre for Disease Prevention and Control. RespiCast - ECDC respiratory diseases forecasting hub (2026). URL https://respicast.ecdc.europa.eu/. Consulted on 4 March 2026.

[39] Bracher, J., Ray, E. L., Gneiting, T. & Reich, N. G. Evaluating epidemic forecasts in an interval format. *PLoS computational biology* **17**, e1008618 (2021).

[40] Reich, N. G. *et al.* A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the united states. *Proceedings of the National Academy of Sciences* **116**, 3146–3154 (2019).

[41] Stock, J. & Mark, W. Core inflation and trend inflation. *Review of Economics and Statistics* **98**, 770–784 (2016).

[42] Carriero, A., Pettenuzzo, D. & Shekhar, S. Macroeconomic forecasting with large language models (2025). Available at SSRN: https://ssrn.com/abstract=4881094.

[43] Pastorello, G., Trotta, C., Canfora, E. *et al.* The fluxnet2015 dataset and the oneflux processing pipeline for eddy covariance data. *Scientific Data* (2020).

[44] Fu, Z., Stoy, P. C. *et al.* A comparison of gap-filling algorithms for eddy covariance fluxes. *Geoscientific Instrumentation, Methods and Data Systems* (2021).

[45] Yuan, W. *et al.* Increased atmospheric vapor pressure deficit reduces global vegetation growth. *Science Advances* **5**, eaax1396 (2019). URL https://www.science.org/doi/abs/10.1126/sciadv.aax1396.

[46] Chen, T. & Guestrin, C. *Xgboost: A scalable tree boosting system.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).

[47] Moorhead, J. E. *et al.* Evaluation of sensible heat flux and evapotranspiration estimates using a surface layer scintillometer and a large weighing lysimeter. *Sensors (Basel)* **17**, 2350 (2017). URL https://pmc.ncbi.nlm.nih.gov/articles/

PMC5676675/.

[48] Loritz, R. *et al.* Camels-de: hydro-meteorological time series and attributes for 1555 catchments in germany. *Earth System Science Data Discussions* **2024**, 1–30 (2024).

[49] Bergström, S. *Development and application of a conceptual runoff model for Scandinavian catchments* (1976).

[50] Acuña Espinoza, E., Loritz, R., Álvarez Chaves, M., Bäuerle, N. & Ehret, U. To bucket or not to bucket? analyzing the performance and interpretability of hybrid hydrological models with dynamic parameterization. *Hydrology and Earth System Sciences* **28**, 2705–2719 (2024).

[51] Nash, J. E. & Sutcliffe, J. V. River flow forecasting through conceptual models part i—a discussion of principles. *Journal of hydrology* **10**, 282–290 (1970).

[52] Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *the Journal of machine Learning research* **12**, 2825–2830 (2011).

[53] Micci-Barreca, D. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD explorations newsletter* **3**, 27–32 (2001).

[54] Xiao, G., Tian, Y., Chen, B., Han, S. & Lewis, M. *Efficient streaming language models with attention sinks.* International Conference on Learning Representations (2024).

[55] Su, J. *et al.* Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024).

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

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

[58] Shchur, O. *et al. Autogluon–timeseries: Automl for probabilistic time series forecasting.* International Conference on Automated Machine Learning, 9–1 (PMLR, 2023).

[59] Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019).

[60] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020). URL https://doi.org/10.1038/s41586-020-2649-2.

[61] pandas development team, T. pandas-dev/pandas: Pandas (2020). URL https://doi.org/10.5281/zenodo.3509134.

[62] Wes McKinney. Stéfan van der Walt & Jarrod Millman (eds) Proceedings of the 9th Python in Science Conference. (eds Stéfan van der Walt & Jarrod Millman) *Proceedings of the 9th Python in Science Conference*, 56 – 61 (2010).

[63] Seabold, S. & Perktold, J. *statsmodels: Econometric and statistical modeling with python*. 9th Python in Science Conference (2010).

[64] Wolf, T. *et al. Transformers: State-of-the-art natural language processing*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 38–45 (Association for Computational Linguistics, 2020).

[65] Alexandrov, A. *et al.* GluonTS: Probabilistic and Neural Time Series Modeling in Python. *The Journal of Machine Learning Research* **21**, 4629–4634 (2020).

[66] Rogozhnikov, A. *Einops: Clear and reliable tensor manipulations with einstein-like notation*. International Conference on Learning Representations (2022).

[67] Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J. & Montero-Manso, P. *Monash Time Series Forecasting Archive*. Neural Information Processing Systems Track on Datasets and Benchmarks (2021).

[68] Makridakis, S., Spiliotis, E. & Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting* **36**, 54–74 (2020).

[69] Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R. & Gasthaus, J. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems* **32** (2019).

[70] 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.

[71] Rasp, S. *et al.* Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems* **12**, e2020MS002203 (2020).

[72] Woo, G., Liu, C., Kumar, A. & Sahoo, D. Pushing the limits of pre-training for time series forecasting in the cloudops domain. *arXiv preprint arXiv:2310.05063* (2023).

[73] Liu, X. *et al.* Largest: A benchmark dataset for large-scale traffic forecasting. *arXiv:2306.08259* (2023).

[74] Jiang, J., Han, C., Jiang, W., Zhao, W. X. & Wang, J. Libcity: A unified library towards efficient and comprehensive urban spatial-temporal prediction. *arXiv preprint arXiv:2304.14343* (2023).

[75] Emami, P., Sahu, A. & Graf, P. BuildingsBench: A Large-Scale Dataset of 900K Buildings and Benchmark for Short-Term Load Forecasting. *arXiv:2307.00142* (2023).

Dataset Name	Frequencies	# Time Series	Domain	Source
Electricity	15min, 1H, 1W, 1D	370	Energy	[67]
KDD Cup (2018)	1H, 1D	270	Nature	[67]
M4 (Daily)	1D	4227	Various	[68]
M4 (Hourly)	1H	414	Various	[68]
M4 (Monthly)	1M	48000	Various	[68]
M4 (Weekly)	1W	359	Various	[68]
Mexico City Bikes	1H, 1D, 1W	494	Transport	[9]
Pedestrian Counts	1H, 1D, 1W	66	Transport	[67]
Solar	5min, 10min, 1H	5166	Energy	[9]
Taxi	30min, 1H	2428	Transport	[69]
Uber TLC	1H, 1D	262	Transport	[70]
USHCN	1D, 1W	225280	Nature	[9]
Weatherbench	1H, 1D, 1W	225280	Nature	[71]
Wiki	1H, 1D, 1W	100000	Web	[9]
Wind Farms	1H, 1D	337	Energy	[67]
Temperature-Rain	1D	32072	Nature	[67]
London Smart Meters	30min, 1D	5560	Energy	[67]
Alibaba Cluster Trace (2018)	5min, 1H	100000	Cloud Ops	[72]
Azure VM Traces (2017)	5min, 1H	100000	Cloud Ops	[72]
Borg Cluster Data (2011)	5min, 1H	100000	Cloud Ops	[72]
LargeST (2017)	1H, 1D	8196	Transport	[73]
Q-Traffic	15min, 1H	45148	Transport	[74]
Buildings 900K	1H, 1D	100000	Energy	[75]

Table 1: Real univariate datasets used for pretraining Chronos-2.

Model	Skill Score (%)	Skill Score 95% CI	Avg. Win Rate (%)	Win Rate 95% CI	Median runtime (s)	#Failures
Chronos-2	**35.5**	(31.6, 39.5)	**87.5**	(84.5, 90.1)	0.8	0
TiRex	30.0	(26.4, 33.8)	77.5	(74.1, 80.7)	0.2	0
TimesFM-2.5	30.3	(26.7, 34.2)	77.0	(72.8, 80.9)	1.9	0
Toto-1.0	28.2	(23.9, 32.7)	67.0	(61.9, 71.8)	22.1	0
Moirai-2.0	27.3	(23.3, 31.3)	61.4	(56.8, 65.4)	0.3	0
Chronos-Bolt	26.5	(22.5, 30.5)	61.1	(57.1, 65.1)	0.2	0
TabPFN-TS	27.6	(23.0, 32.0)	59.1	(53.8, 64.4)	88.9	2
CatBoost	23.7	(19.0, 28.2)	53.1	(48.1, 58.4)	46.6	0
LightGBM	22.4	(17.3, 27.0)	49.7	(44.4, 55.0)	4.8	0
Stat-Ens	15.7	(11.4, 19.8)	45.9	(40.6, 51.6)	148.6	11
TFT	20.5	(15.7, 25.5)	45.7	(40.0, 51.3)	827.2	0
PatchTST	18.4	(13.4, 23.6)	40.5	(35.4, 45.5)	662.1	0
DeepAR	17.5	(13.6, 21.6)	38.2	(33.9, 42.8)	1113.5	3
AutoARIMA	11.2	(6.9, 15.4)	35.6	(31.3, 40.3)	19.5	10
AutoTheta	11.0	(7.1, 14.5)	33.1	(29.2, 37.0)	3.2	0
AutoETS	2.3	(-6.8, 9.4)	32.9	(27.0, 38.4)	3.5	3
SeasonalNaive	0.0	(0.0, 0.0)	18.5	(14.7, 22.3)	0.5	0
Naive	-16.7	(-29.2, -7.5)	16.3	(12.7, 20.3)	0.5	0

Table 2: **Point forecasting results on `fev-bench`.** The average win rate and skill score are computed with respect to the mean absolute scaled error (MASE) metric. Higher values are better for both. Scores for models with failures on specific tasks were imputed with the baseline (Seasonal Naive) score, as described in ref. [25].

Model	Skill Score (%)	Skill Score 95% CI	Avg. Win Rate (%)	Win Rate 95% CI	Median runtime (s)	#Failures
Chronos-2	**47.3**	(42.9, 52.2)	**91.5**	(89.5, 93.4)	0.8	0
TiRex	42.6	(38.1, 47.5)	83.2	(80.2, 85.8)	0.2	0
TimesFM-2.5	42.3	(37.9, 47.4)	79.3	(75.7, 82.6)	1.9	0
Toto-1.0	40.7	(35.8, 46.3)	72.1	(67.7, 76.1)	22.1	0
Moirai-2.0	39.3	(34.4, 44.5)	66.4	(62.5, 69.6)	0.3	0
Chronos-Bolt	38.9	(34.2, 44.1)	66.1	(63.0, 69.1)	0.2	0
TabPFN-TS	39.6	(34.6, 44.7)	64.4	(59.9, 69.0)	88.9	2
TFT	32.2	(26.1, 38.7)	50.8	(45.4, 55.8)	827.2	0
PatchTST	30.1	(23.7, 36.9)	44.8	(40.2, 49.7)	662.1	0
DeepAR	29.1	(23.2, 35.7)	43.7	(39.5, 48.3)	1113.5	3
Stat-Ens	20.2	(14.3, 24.8)	43.7	(39.1, 48.4)	148.6	11
AutoARIMA	20.6	(14.8, 25.9)	40.5	(36.6, 44.5)	19.5	10
CatBoost	23.0	(17.0, 29.8)	34.8	(30.8, 38.6)	46.6	0
AutoETS	-26.8	(-64.8, -0.6)	32.2	(26.7, 37.8)	3.5	3
LightGBM	21.7	(15.0, 28.5)	32.1	(27.9, 36.1)	4.8	0
AutoTheta	5.5	(-3.2, 12.3)	26.9	(23.0, 30.9)	3.2	0
SeasonalNaive	0.0	(0.0, 0.0)	17.7	(15.1, 20.4)	0.5	0
Naive	-45.4	(-62.9, -30.3)	9.7	(7.0, 12.3)	0.5	0

Table 3: **Probabilistic forecasting results on `fev-bench`.** The average win rate and skill score are computed with respect to the scaled quantile loss (SQL) metric. Higher values are better for both. Scores for models with failures on specific tasks were imputed with the baseline (Seasonal Naive) score, as described in ref. [25].

Model	p-value	Sig.
TimesFM-2.5	1.68×10^{-7}	***
TiRex	1.31×10^{-7}	***
Toto-1.0	1.60×10^{-9}	***
TabPFN-TS	1.05×10^{-10}	***
CatBoost	3.62×10^{-12}	***
LightGBM	1.27×10^{-12}	***
Moirai-2.0	1.36×10^{-13}	***
Stat-Ens	4.44×10^{-14}	***
Chronos-Bolt	9.63×10^{-15}	***
TFT	9.63×10^{-15}	***
AutoETS	4.85×10^{-15}	***
DeepAR	1.91×10^{-15}	***
PatchTST	1.47×10^{-15}	***
AutoARIMA	1.26×10^{-15}	***
AutoTheta	4.36×10^{-16}	***
Naive	4.61×10^{-17}	***
SeasonalNaive	4.22×10^{-17}	***

(a) Point forecasting (MASE)

Model	p-value	Sig.
TiRex	1.65×10^{-7}	***
TimesFM-2.5	1.24×10^{-8}	***
Toto-1.0	1.27×10^{-10}	***
TabPFN-TS	3.27×10^{-12}	***
Moirai-2.0	1.30×10^{-14}	***
Chronos-Bolt	2.43×10^{-15}	***
TFT	1.02×10^{-15}	***
Stat-Ens	5.55×10^{-16}	***
AutoARIMA	2.47×10^{-16}	***
PatchTST	2.44×10^{-16}	***
DeepAR	2.39×10^{-16}	***
AutoETS	1.94×10^{-16}	***
AutoTheta	5.71×10^{-17}	***
LightGBM	4.59×10^{-17}	***
CatBoost	4.59×10^{-17}	***
SeasonalNaive	3.31×10^{-17}	***
Naive	3.31×10^{-17}	***

(b) Probabilistic forecasting (SQL)

Table 4: Holm-Šidák (HS) adjusted p values of the one-sided Wilcoxon signed rank test comparing Chronos-2 against all baselines on `fev-bench`. ***$p < 0.001$, **$p < 0.01$, *$p < 0.05$.

Model	Skill Score (%)	Skill Score 95% CI	Avg. Win Rate (%)	Win Rate 95% CI
Chronos-2	**26.5**	(19.9, 33.7)	**76.2**	(66.4, 85.2)
TimesFM-2.5	23.3	(17.4, 29.8)	67.3	(58.6, 75.3)
TiRex	22.2	(15.7, 28.9)	62.0	(54.0, 68.2)
Stat-Ens	19.6	(12.4, 27.4)	60.2	(47.2, 72.5)
Toto-1.0	22.3	(16.2, 28.5)	54.0	(44.1, 65.4)
DeepAR	18.4	(12.0, 24.9)	52.5	(41.0, 63.0)
TFT	19.2	(12.2, 26.1)	47.7	(35.2, 59.7)
Moirai-2.0	19.8	(13.2, 26.9)	46.6	(35.5, 59.3)
PatchTST	17.7	(10.9, 24.5)	46.1	(34.9, 57.3)
Chronos-Bolt	20.4	(14.7, 26.7)	45.7	(38.3, 52.8)
AutoARIMA	16.0	(8.5, 24.2)	41.4	(29.3, 53.1)
TabPFN-TS	10.5	(0.7, 19.2)	38.4	(28.9, 47.8)
Seasonal Naive	0.0	(0.0, 0.0)	11.9	(7.4, 17.3)

Table 5: **Point forecasting results on `Chronos-Bench-II`.** The average win rate and skill score are computed with respect to the mean absolute scaled error (MASE) metric. Higher values are better for both. Scores for models with failures on specific tasks were imputed with the baseline (Seasonal Naive) score, as described in ref. [9].

Model	Skill Score (%)	Skill Score 95% CI	Avg. Win Rate (%)	Win Rate 95% CI
Chronos-2	**46.6**	(36.3, 56.4)	**75.6**	(64.8, 84.9)
TiRex	41.7	(31.0, 52.1)	69.1	(60.5, 76.9)
TimesFM-2.5	42.4	(31.7, 52.8)	67.6	(59.6, 75.3)
Toto-1.0	41.9	(30.6, 52.4)	58.3	(47.5, 68.8)
Moirai-2.0	40.9	(28.9, 51.7)	55.2	(44.1, 66.7)
TFT	36.2	(24.9, 46.9)	51.1	(38.7, 63.9)
DeepAR	29.9	(9.9, 44.5)	49.4	(38.3, 60.6)
Chronos-Bolt	39.3	(29.2, 49.3)	49.1	(42.6, 55.9)
PatchTST	32.5	(15.8, 45.0)	48.0	(36.3, 59.9)
Stat-Ens	33.0	(22.0, 44.0)	44.4	(34.0, 54.6)
TabPFN-TS	32.6	(21.6, 44.2)	41.8	(32.2, 51.9)
AutoARIMA	30.4	(18.2, 42.6)	31.2	(21.9, 42.6)
Seasonal Naive	0.0	(0.0, 0.0)	9.1	(4.8, 13.9)

Table 6: **Probabilistic forecasting results on `Chronos-Bench-II`.** The average win rate and skill score are computed with respect to the weighted quantile loss (WQL) metric. Higher values are better for both. Scores for models with failures on specific tasks were imputed with the baseline (Seasonal Naive) score, as described in ref. [9].

Model	p-value	Sig.
DeepAR	0.045	*
Stat-Ens	0.045	*
TFT	0.045	*
TimesFM-2.5	0.045	*
TiRex	0.035	*
Moirai-2.0	0.031	*
Toto-1.0	0.028	*
PatchTST	0.026	*
Chronos-Bolt	0.0027	**
AutoARIMA	0.0021	**
TabPFN-TS	4.60×10^{-4}	***
Seasonal Naive	2.24×10^{-6}	***

(a) Point forecasting (MASE)

Model	p-value	Sig.
DeepAR	0.046	*
PatchTST	0.046	*
TFT	0.046	*
TimesFM-2.5	0.046	*
Toto-1.0	0.046	*
Moirai-2.0	0.033	*
TiRex	0.028	*
Stat-Ens	0.0050	**
Chronos-Bolt	0.0019	**
TabPFN-TS	0.0017	**
AutoARIMA	4.01×10^{-4}	***
Seasonal Naive	4.47×10^{-7}	***

(b) Probabilistic forecasting (WQL)

Table 7: Holm-Šidák (HS) adjusted p values of the one-sided Wilcoxon signed rank test comparing Chronos-2 against all baselines on `Chronos-Bench-II`. ***$p < 0.001$, **$p < 0.01$, *$p < 0.05$.

Model	Skill Score (%)	Skill Score 95% CI	Avg. Win Rate (%)	Win Rate 95% CI
Chronos-2	**30.2**	(26.5, 34.5)	**86.8**	(83.4, 90.0)
TimesFM-2.5	29.5	(25.5, 34.2)	82.3	(78.9, 85.6)
TiRex	27.6	(23.8, 31.8)	76.1	(71.9, 79.7)
Moirai-2.0	27.2	(23.3, 31.5)	71.1	(67.5, 74.5)
Toto-1.0	25.2	(21.7, 29.1)	69.1	(65.4, 72.6)
Chronos-Bolt	19.2	(11.2, 25.9)	64.4	(59.4, 69.3)
TabPFN-TS	16.6	(10.0, 22.1)	55.8	(52.1, 59.8)
PatchTST	15.1	(9.4, 20.9)	46.8	(42.9, 51.0)
TFT	8.5	(-0.4, 16.1)	38.5	(34.1, 43.3)
Stat-Ens	-52.6	(-111.6, -12.2)	33.9	(29.5, 38.5)
AutoARIMA	-7.4	(-42.3, 4.8)	31.4	(27.0, 35.7)
Seasonal Naive	0.0	(0.0, 0.0)	26.2	(22.4, 30.3)
AutoETS	-21.2	(-34.8, -10.3)	23.7	(19.0, 28.5)
AutoTheta	-9.0	(-17.4, -1.0)	23.1	(18.8, 27.8)
DeepAR	-34.3	(-58.1, -16.8)	20.8	(16.6, 25.2)

Table 8: **Point forecasting results on `GIFT-Eval`.** The average win rate and skill score are computed with respect to the mean absolute scaled error (MASE) metric. Higher values are better for both. Scores for models with failures on specific tasks were imputed with the baseline (Seasonal Naive) score, as described in ref. [26].

Model	Skill Score (%)	Skill Score 95% CI	Avg. Win Rate (%)	Win Rate 95% CI
Chronos-2	**51.4**	(47.0, 55.3)	**84.5**	(81.1, 87.7)
TimesFM-2.5	51.0	(46.4, 55.3)	81.4	(77.2, 84.8)
TiRex	50.2	(45.7, 54.4)	79.0	(75.0, 82.4)
Toto-1.0	48.6	(43.9, 52.9)	73.0	(69.2, 76.4)
Moirai-2.0	48.4	(43.7, 52.8)	69.1	(65.2, 72.8)
TabPFN-TS	43.1	(37.9, 48.0)	60.7	(56.6, 64.4)
Chronos-Bolt	42.6	(36.5, 48.1)	59.6	(54.6, 64.7)
PatchTST	41.3	(36.3, 46.2)	52.2	(48.8, 56.0)
TFT	39.5	(34.0, 45.1)	49.5	(45.2, 54.4)
DeepAR	14.7	(1.9, 24.9)	31.2	(26.5, 35.8)
Stat-Ens	10.3	(1.9, 18.7)	29.4	(25.8, 32.9)
AutoARIMA	8.8	(-1.1, 15.8)	25.7	(22.3, 29.3)
Seasonal Naive	0.0	(0.0, 0.0)	19.9	(17.2, 23.1)
AutoTheta	-24.4	(-41.8, -7.5)	17.7	(13.6, 22.2)
AutoETS	-648.9	(-3747.6, -94.4)	17.0	(13.1, 21.1)

Table 9: **Probabilistic forecasting results on `GIFT-Eval`.** The average win rate and skill score are computed with respect to the weighted quantile loss (WQL) metric. Higher values are better for both. Scores for models with failures on specific tasks were imputed with the baseline (Seasonal Naive) score, as described in ref. [26].

Model	p-value	Sig.
TimesFM-2.5	0.097	
TiRex	3.76×10^{-5}	***
Moirai-2.0	6.38×10^{-7}	***
Chronos-Bolt	4.04×10^{-7}	***
Toto-1.0	1.09×10^{-8}	***
TabPFN-TS	5.22×10^{-13}	***
TFT	8.96×10^{-15}	***
AutoARIMA	2.24×10^{-15}	***
PatchTST	2.24×10^{-15}	***
Stat-Ens	1.25×10^{-15}	***
AutoETS	3.76×10^{-16}	***
Seasonal Naive	1.85×10^{-16}	***
DeepAR	1.72×10^{-16}	***
AutoTheta	1.64×10^{-16}	***

(a) Point forecasting (MASE)

Model	p-value	Sig.
TimesFM-2.5	0.22	
TiRex	0.0090	**
Toto-1.0	3.80×10^{-5}	***
Moirai-2.0	3.50×10^{-6}	***
Chronos-Bolt	1.43×10^{-7}	***
TabPFN-TS	1.57×10^{-9}	***
TFT	4.45×10^{-12}	***
PatchTST	9.36×10^{-13}	***
DeepAR	3.01×10^{-15}	***
AutoTheta	1.54×10^{-16}	***
AutoARIMA	1.33×10^{-16}	***
Stat-Ens	1.28×10^{-16}	***
AutoETS	1.26×10^{-16}	***
Seasonal Naive	8.53×10^{-17}	***

(b) Probabilistic forecasting (WQL)

Table 10: Holm-Šidák (HS) adjusted p values of the one-sided Wilcoxon signed rank test comparing Chronos-2 against all baselines on `GIFT-Eval`. ***$p < 0.001$, **$p < 0.01$, *$p < 0.05$.