fev-bench: A Realistic Benchmark for Time Series Forecasting

Oleksandr Shchur [1 *] **Abdul Fatir Ansari** [1 *] **Caner Turkmen** [2 †] **Lorenzo Stella** [1]

Nick Erickson [3 †] **Pablo Guerron** [1,4 †] **Michael Bohlke–Schneider** [1] **Yuyang Wang** [5 †]

[1] Amazon [2] Keystone AI [3] Prior Labs [4] Boston College [5] NTT DATA AIVista

Abstract

Benchmark quality is critical for meaningful evaluation and sustained progress in time series forecasting, particularly with the rise of time series foundation models (TSFMs). Existing benchmarks often have limited domain coverage or overlook real-world settings such as tasks with covariates. Their aggregation procedures frequently lack statistical rigor, making it unclear whether performance differences reflect true improvements or random variation. Many benchmarks lack consistent evaluation infrastructure or are too rigid for integration into existing pipelines. To address these gaps, we propose `fev-bench`, a benchmark of 100 forecasting tasks across seven domains, including 46 with covariates. Supporting the benchmark, we introduce `fev`, a lightweight Python library for forecasting evaluation emphasizing reproducibility and integration with existing workflows. Using `fev`, `fev-bench` employs principled aggregation with bootstrapped confidence intervals to report performance along two dimensions: win rates and skill scores. We evaluate various models on `fev-bench` and find that existing TSFMs often miss accuracy by ignoring covariates, pointing to a clear direction for future work.

1 Introduction

Pretrained time series forecasting models are transforming forecasting practice. They often outperform traditional methods (Aksu et al., 2024) while enabling zero-shot inference that simplifies production use and lowers the barrier to entry (Cohen et al., 2025).

Advances in pretrained forecasting models are primarily evaluated on benchmarks, making benchmark quality critical for progress. Shortcomings in existing benchmarks directly shape model development; for example, most general benchmarks ignore covariates despite their prevalence in real-world applications, and covariates have so far been evaluated only on individual datasets, such as those from forecasting competitions (Bojer & Meldgaard, 2021), or narrow collections (Arango et al., 2025). Consequently, most pretrained models lack covariate support, limiting their effectiveness in domains such as retail, where promotional and pricing data are essential for accurate demand forecasting (Fildes et al., 2022).

Beyond task coverage, existing benchmarks often lack statistical rigor. Most studies report single-number summaries, making it unclear whether improvements reflect true advances or random variation. Small gains may vanish or even reverse under minor benchmark changes (Roque et al., 2025), undermining the reliability of conclusions about which models perform better.

*Equal contribution

†Work done while at Amazon

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

Benchmark	# datasets	# tasks	# tasks with covariates	# multivariate tasks	Forecast type
Monash (Godahewa et al., 2021)	42	42	0	0	point
LTSF (Zeng et al., 2023)	9	36	0	9	point
TFB (Qiu et al., 2024)	41	116	0	25	point
BasicTS+ (Shao et al., 2024)	20	40	0	20	point
ProbTS (Zhang et al., 2024)	18	18	0	14	point & quantile
TSFM-Bench (Li et al., 2025)	21	21	0	21	point
TIME (Qiao et al., 2026)	50	50	0	36	point & quantile
GIFT-Eval (Aksu et al., 2024)	55	97	0	43	point & quantile
`fev-bench` (this work)	96	100	46	35	point & quantile

Table 1: Overview of general time series forecasting benchmarks. `fev-bench` includes more unique datasets than prior benchmarks and is the first to systematically include tasks with covariates, addressing a major gap in existing evaluation frameworks.

Finally, benchmark infrastructure poses additional barriers to progress and reproducibility. Many benchmarks provide only standalone datasets without evaluation code, leading to inconsistent implementations and incomparable results (Hewamalage et al., 2023). When infrastructure exists, it often consists of monolithic systems that bundle models, datasets, and evaluation logic with extensive dependencies, becoming unmaintainable over time. This rigidity prevents extension to new domains or integration into existing workflows, limiting practical utility and longevity.

To address above challenges, we make three contributions:

- **New benchmark.** We introduce `fev-bench`,[3] a forecast evaluation benchmark containing 100 tasks spanning 7 real-world application domains, built from 96 datasets, 40 of which have rarely or never been used to evaluate pretrained models. Unlike existing general forecasting benchmarks, `fev-bench` includes 46 tasks with covariates alongside univariate and multivariate settings. Our evaluation highlights the importance of covariates for forecasting accuracy, despite most pretrained forecasting models and benchmarks largely ignoring them.
- **Aggregation methods.** In our benchmark, we employ principled aggregation strategies including bootstrap-based confidence intervals that quantify whether performance differences are statistically meaningful. This approach enables more reliable model comparisons and assesses the robustness of conclusions to variations in benchmark composition.
- **Evaluation package.** We introduce `fev`, [4] a lightweight Python package for forecasting evaluation that introduces minimal dependencies while remaining compatible with popular forecasting libraries. The package focuses on reproducibility and extensibility, enabling researchers to easily build and share new benchmarks and the corresponding results.

2 Preliminaries

Problem definition. The multivariate time series forecasting problem can be formally stated as follows. We are given a collection $\{\boldsymbol{y}_{n,1:T}\}_{n=1}^{N}$ of N multivariate time series. For $n = 1, \ldots, N$ and $t = 1, \ldots, T$, let $\boldsymbol{y}_{n,t} = (y_{n,d,t})_{d=1}^{D} \in \mathbb{R}^D$ denote the D-dimensional observation vector for series n at time t, where $D = 1$ corresponds to univariate forecasting. The goal is to predict the future H values $\boldsymbol{y}_{n,T+1:T+H}$ for each series n, where H is the forecast horizon. Each time series may be accompanied by covariates $\mathbf{X}_{n,1:T+H}$, including (i) *static covariates* that do not vary over time (e.g., location), (ii) *past-only dynamic covariates* observed up to time T (e.g., past related series), and (iii) *known dynamic covariates* available for all time steps $1, \ldots, T + H$ (e.g., holiday indicators).

The goal of probabilistic forecasting is to model the distribution $p\big(\boldsymbol{y}_{n,T+1:T+H} \mid \boldsymbol{y}_{n,1:T}, \mathbf{X}_{n,1:T+H}\big)$. While full distributional modeling provides the richest information, it is common in practice to produce *point forecasts* such as conditional means or medians. Alternatively, many applications estimate *predictive quantiles* $p(y_{n,d,t} \mid \boldsymbol{y}_{n,1:T}, \mathbf{X}_{n,1:T+H})$. We denote by $\mathcal{Q} \subset (0, 1)$ the set of quantile levels and produce forecasts $\hat{y}^{(q)}_{n,d,t}$ satisfying $\Pr(y_{n,d,t} \leq \hat{y}^{(q)}_{n,d,t}) = q$ for all $q \in \mathcal{Q}$.

Benchmarks and tasks. A benchmark consists of forecasting tasks together with an evaluation and aggregation procedure. A task specifies a dataset and the parameters defining how forecasts are produced and evaluated, including the forecast horizon H, evaluation cutoffs, target and covariate

[3] `huggingface.co/spaces/autogluon/fev-bench`

[4] `github.com/autogluon/fev`

Benchmark	Energy	Nature	Cloud	Mobility	Econ	Health	Retail
TIME	4	16	3	4	14	3	6
GIFT-Eval	16	9	8	7	6	5	4
fev-bench	26	5	20	7	10	8	20

Table 2: Number of datasets from different domains in GIFT-Eval, TIME and `fev-bench`.

Benchmark	10S	T	5T	10T	15T	30T	H	D	W	M	Q	Y
TIME	0	0	7	1	7	1	8	9	4	9	2	0
GIFT-Eval	2	0	4	2	4	0	13	15	8	5	1	1
fev-bench	0	6	7	2	5	4	22	19	16	7	4	4

Table 3: Number of datasets with different frequencies in GIFT-Eval, TIME, and `fev-bench`.

columns, and evaluation metric. A single dataset can yield multiple tasks by varying these parameters. Since different metrics correspond to different optimal forecasts, combining conflicting metrics within a task creates ambiguity about the intended objective (Kolassa, 2020).

Each task is evaluated using a rolling-origin protocol with W windows (Hyndman & Athanasopoulos, 2018). Let $\tau_1 < \tau_2 < \cdots < \tau_W$ denote the evaluation cutoffs. At each window w, the model receives observations up to τ_w and produces H-step forecasts, yielding W forecast–target pairs per task. This setup mimics real-world deployment and improves robustness, especially for datasets with few series.

Aggregation. While tasks define individual evaluation problems, benchmarks must aggregate results across tasks to answer questions such as "Is model A more accurate than model B overall?" The choice of aggregation directly affects the reliability and interpretability of benchmark results.

3 Task definitions

We introduce `fev-bench` (**F**orecast **EV**aluation Benchmark), a benchmark designed to address limitations of existing forecasting evaluation frameworks. Unlike prior benchmarks with narrow domain coverage or limited evaluation infrastructure, `fev-bench` enables reproducible and statistically sound evaluation across diverse real-world forecasting applications. `fev-bench` consists of 100 forecasting tasks, with full specifications in Appendix A. This section focuses on dataset and task design; aggregation methods and the evaluation package are described in Sec. 4 and Sec. 5.

3.1 Datasets and tasks

Our goal is to construct a *general* benchmark representative of real-world forecasting applications across domains, frequencies, horizons, and time series characteristics. We consider both univariate and multivariate forecasting problems, with covariates covering dynamic (both past-only and known) and static variables. We evaluate both point and probabilistic forecasting performance.

Datasets. We source datasets from established collections including the Monash repository (Godahewa et al., 2021), GIFT-Eval (Aksu et al., 2024), and BOOM (Cohen et al., 2025). Since these collections lack covariates, we additionally include public datasets from Kaggle (Bojer & Meldgaard, 2021) and domain-specific repositories (Wang et al., 2023; Lago et al., 2021; Arango et al., 2025). This process yields 96 unique datasets and 100 forecasting tasks with different target and covariate selections. Among them, 30 tasks include known dynamic covariates, 24 past dynamic covariates, and 19 static covariates. These categories are non-exclusive, and both univariate and multivariate tasks may include covariates. Tables 2 and 3 compare `fev-bench` against GIFT-Eval (Aksu et al., 2024) and TIME (Qiao et al., 2026), showing substantially broader coverage across domains and frequencies. Like existing general benchmarks, `fev-bench` is built by curating public datasets rather than collecting new data, yet 40 of its 96 datasets have rarely or never been used to evaluate pretrained models (Qiao et al., 2026), and popular datasets such as M5 and Rossmann are released together with the covariates discarded by existing collections (Aksu et al., 2024; Woo et al., 2024).

Forecast horizons. Many existing benchmarks reuse the same datasets with different horizons (Zeng et al., 2023; Aksu et al., 2024), creating correlated tasks with limited additional insight. While such setups are useful for studying sensitivity to forecast horizon, `fev-bench` instead prioritizes diversity across datasets and application domains. We therefore avoid horizon duplication within datasets and select horizons that reflect realistic forecasting needs, such as 168 steps for hourly energy demand or 30 steps for daily retail sales.

Rolling evaluation. We use rolling window evaluation to balance computational cost and statistical reliability. Depending on dataset size, we use up to 20 windows for datasets with fewer than 10 series, up to 10 for datasets with 10–2000 series, and 1 for larger datasets, additionally requiring at least $(2 \times H + 1)$ past observations before the first window. Results are averaged across windows.

Representative subset of tasks. In addition to the full 100-task `fev-bench` benchmark, we provide `fev-bench-mini`, a curated subset of 20 tasks (Appendix E). The subset preserves the diversity of covariates, dimensionalities, domains, and horizons in the full benchmark while enabling faster iteration at lower computational cost. `fev-bench-mini` also approximates the relative model rankings of the full benchmark, making it suitable for model development and ablation studies.

3.2 Evaluation metrics

Each of the 100 benchmark tasks evaluates point and probabilistic forecast accuracy using complementary metrics. We evaluate point forecast accuracy using Mean Absolute Scaled Error (MASE), following existing benchmarks (Aksu et al., 2024; Godahewa et al., 2021).

$$\text{MASE} = \frac{1}{NDH} \sum_{n=1}^{N} \sum_{d=1}^{D} \frac{1}{a_{n,d}} \sum_{t=T+1}^{T+H} |y_{n,d,t} - \hat{y}_{n,d,t}|, \tag{1}$$

where the error for each series n and dimension d is normalized by the historical seasonal error $a_{n,d} = \frac{1}{T-m} \sum_{t=m+1}^{T} |y_{n,d,t} - y_{n,d,t-m}|$. Here $\hat{y}_{n,d,t}$ is the point forecast and m is the seasonal period based on the data frequency (e.g., m=12 for monthly data). MASE offers several advantages: it is scale-free, balances contributions across series with different magnitudes, handles trends well, and remains robust when the forecast horizon contains zeros (Hyndman & Koehler, 2006).

We use Scaled Quantile Loss (SQL) computed on levels $\mathcal{Q} = \{0.1, 0.2, ..., 0.9\}$ for probabilistic forecast accuracy:

$$\text{SQL} = \frac{1}{NDH} \sum_{n=1}^{N} \sum_{d=1}^{D} \frac{1}{a_{n,d}} \sum_{t=T+1}^{T+H} \sum_{q \in \mathcal{Q}} \rho_q(y_{n,d,t}, \hat{y}^{(q)}_{n,d,t}) \tag{2}$$

where $\hat{y}^{(q)}_{n,d,t}$ is the quantile forecast at level q, $a_{n,d}$ is the historical seasonal error as in MASE, and $\rho_q(y, \hat{y}) = 2 \max\{q(y - \hat{y}), (1 - q)(\hat{y} - y)\}$ is the quantile loss.

We adopt Scaled Quantile Loss (SQL) as the primary probabilistic metric because it is the natural extension of MASE and inherits its scale-independence properties. Although forecasting benchmarks more commonly use the scale-dependent Weighted Quantile Loss (WQL) (Ansari et al., 2024; Aksu et al., 2024), both SQL and WQL are related to CRPS (Gneiting & Raftery, 2007), Winkler Score, and Weighted Interval Score (Tibshirani, 2023). The key difference is that SQL normalizes each series by scale, whereas WQL aggregates errors in a scale-dependent manner, analogous to the distinction between MASE and Weighted Absolute Percentage Error (WAPE). Our choice follows the rationale of the M4 and M5 competitions, which also used SQL-equivalent metrics (Makridakis et al., 2020, 2022).

Both MASE and SQL can encounter numerical issues with intermittent time series where the seasonal error $a_{n,d}$ approaches zero (Hewamalage et al., 2023). We verified that this problem does not occur in our benchmark tasks. For completeness, we also report WQL and WAPE scores in the supplement.

4 Aggregating the results

After evaluating M models on R tasks, we obtain the error matrix $E \in \mathbb{R}_{\geq 0}^{R \times M}$, where E_{rj} denotes the error (e.g., MASE) of model j averaged over all evaluation windows of task r. Lower values correspond to more accurate forecasts. For failed task-model pairs (e.g., due to timeouts or crashes), we substitute E_{rj} with $E_{r\beta}$, where β denotes a predefined baseline model (Seasonal Naive).

4.1 Marginal performance

The primary goal of any benchmark is to rank models by average performance. We use two complementary aggregation methods that capture different aspects of model quality.

Average win rate W_j represents the probability that model j achieves lower error than another randomly chosen model $k \neq j$ on a randomly chosen task:

$$W_j = \frac{1}{R(M-1)} \sum_{r=1}^{R} \sum_{k=1, k \neq j}^{M} \Big(\mathbb{1}(E_{rj} < E_{rk}) + \frac{1}{2} \cdot \mathbb{1}(E_{rj} = E_{rk}) \Big). \tag{3}$$

Here $\mathbb{1}(\cdot)$ is the binary indicator function. Ties ($E_{rj} = E_{rk}$) are treated as half-wins for each model. The win rate ranges from 0 (worst) to 1 (best) and provides an intuitive measure of relative model performance. However, win rate has two limitations: it is insensitive to the magnitude of performance differences and changes as new models are added to the benchmark, motivating our second aggregation method.

Skill score (Hyndman & Athanasopoulos, 2018) S_j quantifies how much model j reduces forecasting error compared to the fixed baseline model β on average:

$$S_j = 1 - \exp\left(\frac{1}{R}\sum_{r=1}^{R}\log\left(\text{clip}\left(E_{rj}/E_{r\beta}; \ell, u\right)\right)\right). \tag{4}$$

where $\text{clip}(x; \ell, u) = \max(\ell, \min(x, u))$ clips x to the interval $[\ell, u]$. We aggregate relative errors across tasks using geometric mean, clipping values between $\ell = 10^{-2}$ and $u = 100$ to avoid excessive influence from extreme values. The skill score ranges from 1 (perfect forecasts) to $-\infty$ (arbitrarily poor performance). Positive values indicate that the model outperforms the baseline on average, while negative values indicate underperformance.

Geometric mean aggregation is less sensitive to outliers than the arithmetic mean and ensures that the final ranking remains invariant to the choice of baseline model (as long as all relative errors stay within $[\ell, u]$) (Fleming & Wallace, 1986). The geometric mean appropriately handles the multiplicative nature of relative performance comparisons, averaging ratios in a meaningful way where opposing relative errors like $\frac{1}{2}$ and 2 cancel out.

4.2 Pairwise comparison

While marginal performance provides overall rankings, pairwise comparisons reveal specific model relationships that may be obscured in aggregate statistics. The above aggregation methods can be easily generalized to comparing any two models j and k.

Pairwise win rate W_{jk} represents the fraction of tasks where model j outperforms model k

$$W_{jk} = \frac{1}{R}\sum_{r=1}^{R}\left(\mathbb{1}(E_{rj} < E_{rk}) + \frac{1}{2}\cdot\mathbb{1}(E_{rj} = E_{rk})\right). \tag{5}$$

Pairwise skill score S_{jk} quantifies how much model j reduces error compared to model k on average

$$S_{jk} = 1 - \exp\left(\frac{1}{R}\sum_{r=1}^{R}\log\left(\text{clip}\left(E_{rj}/E_{rk}; \ell, u\right)\right)\right). \tag{6}$$

4.3 Significance of performance differences

A critical concern in benchmarking is the reliability of reported performance differences. State-of-the-art claims often rest on minor improvements that may vanish under small changes to benchmark composition, casting doubt on whether they reflect genuine advances (Roque et al., 2025).

To address this concern, we compute 95% confidence intervals using paired bootstrap over tasks (Efron, 1992). We generate $B = 1000$ bootstrap samples by drawing rows with replacement from E, where each $\tilde{E}^{(b)} \in \mathbb{R}_{\geq 0}^{R\times M}$ contains R tasks sampled from the original benchmark. For each $\tilde{E}^{(b)}$, we compute the aggregate statistics to obtain bootstrap distributions such as $\{\tilde{W}_{jk}^{(b)}\}_b$. The $(1-\alpha)$ confidence interval for the pairwise win rate W_{jk} is then

$$\left[Q_{\alpha/2}\left(\{\tilde{W}_{jk}^{(b)}\}_b\right),\ Q_{1-\alpha/2}\left(\{\tilde{W}_{jk}^{(b)}\}_b\right)\right], \tag{7}$$

where $Q_p(\cdot)$ denotes the empirical p-th quantile of the bootstrap distribution. Analogous intervals are constructed for the pairwise skill scores S_{jk}. These intervals quantify how conclusions about model comparisons vary under alternative benchmark compositions.

We report bootstrap confidence intervals only for the pairwise statistics (W_{jk}, S_{jk}), as these directly answer the question of interest: "Does model j consistently outperform model k under different

benchmark compositions?" Equivalently, the intervals define paired null-hypothesis tests over tasks, with null hypotheses $W_{jk} = 0.5$ (no systematic advantage) and $S_{jk} = 0$ (no average error reduction). Confidence intervals for the marginal statistics (W_j, S_j) instead describe the variability of each model's average score in isolation and ignore correlations between models.

Summary. Benchmark interpretation proceeds in two steps. First, the marginal statistics (W_j, S_j) provide an overall model ranking. Second, the pairwise statistics with confidence intervals (W_{jk}, S_{jk}) refine this picture by showing which performance differences are robust to changes in benchmark composition. For example, if a model j ranks highest by marginal win rate W_j and all of its pairwise win rates W_{jk} against other models $k \neq j$ have lower bounds above 50%, then model j can be regarded as outperforming every competitor with high confidence.

4.4 Data leakage

The goal of `fev-bench` is to assess the *zero-shot* capabilities of pretrained forecasting models. Since all datasets in `fev-bench` come from public sources, parts of them may have been included in the pretraining corpora of the evaluated models, which would undermine this goal. We distinguish two forms of leakage: (i) training on the benchmark training split and (ii) exposure to the test split. While the second form directly inflates the reported accuracy, the first one also violates the zero-shot setting, since the model has already seen the series it is asked to forecast.

Self-reported overlap. To guard against both forms of leakage, contributors must report, for each model–task pair, whether *any* part of the dataset at the same frequency was used during training; resampled variants of a dataset at different frequencies are not considered leakage. This indicator is required only for pretrained models, since task-specific models are expected to train on the task training split and the zero-shot requirement does not apply to them. Self-reporting of training overlap is the standard practice for static benchmarks, where both historical and future values of every series are publicly available. For example, GIFT-Eval relies on authors disclosing their pretraining data to flag in-domain results (Aksu et al., 2024), which is exactly the information required by `fev-bench` and which is already published for every open-weight pretrained model. Similarly, TIME introduces fresh datasets that have not been used for evaluation before (Qiao et al., 2026), but nothing prevents future models from training on them, so it ultimately depends on self-reporting as well. One could instead attempt to detect leakage automatically, but this amounts to membership inference over pretraining corpora, which remains an open research problem even for large language models and is even harder for numerical data (Wahdany et al., 2026). We therefore see self-reporting as the only practical option available to any public benchmark today, and consider automatic detection an interesting direction for future work.

Handling leaked tasks. Where `fev-bench` differs from existing benchmarks is in how reported leakage is handled. Rather than disqualifying the model or merely flagging its results, we discard the submitted results on the affected tasks and impute them with the scores of the strongest fully zero-shot univariate model at release time, Chronos-Bolt (Base), similar to the approach used in TabArena (Erickson et al., 2025). This removes the impact of leaked tasks on the aggregate scores while preserving the evaluation on the remaining tasks. A natural alternative would be to exclude the leaked tasks altogether, but this does not hold up as `fev-bench` accepts new model submissions. If tasks are dropped only for the affected model, each submission is scored on a different set of tasks, whereas win rates and skill scores are comparable only when computed over the same tasks. If tasks are instead dropped for all models, the scores remain comparable, but every leaky submission permanently costs all other models a task, and the benchmark keeps shrinking as submissions accumulate. Imputation avoids both problems: it gives contributors a reason not to train on `fev-bench` data, without making honest disclosure too costly. In practice, this choice does not affect the conclusions of our evaluation. When we drop the 10 tasks that are flagged as leaky for at least one model (and omit Moirai-2.0, whose reported overlap covers too many tasks to be dropped), the ranking of the remaining models stays unchanged (Table 16).

Leakage-free alternatives. Eliminating leakage entirely requires a fundamentally different benchmark design. Two such designs are withheld test data, as in Kaggle-style competitions, and prospective evaluation, where forecasts are submitted before the ground truth exists (Meyer et al., 2026; Garza et al., 2026). Both close the leakage channel, but neither is well suited for model development. Withheld test data requires rate-limiting submissions and exposes no per-series error breakdown, so researchers cannot see where their model fails, while prospective evaluation requires waiting months for the ground truth to become available. We therefore view these approaches as complementary to `fev-bench` rather than as substitutes for it.

5 Software package

Comprehensive task coverage and principled evaluation are essential, but standardized infrastructure is equally important for benchmark relevance and reproducibility. This includes code for task definition, evaluation, and result aggregation.

Motivation. Existing forecasting benchmarks typically fall into two categories: standalone datasets without supporting infrastructure (Godahewa et al., 2021; Zeng et al., 2023), and end-to-end systems that bundle models, datasets, and forecasting tasks (Qiu et al., 2024; Aksu et al., 2024). Dataset-only benchmarks provide no guarantee that results are comparable across users, since evaluation may differ in horizons, cutoffs, metrics, or aggregation strategies even on the same dataset. End-to-end systems reduce this ambiguity but are often rigid and difficult to extend or integrate into existing workflows. They impose many dependencies and assumptions, while model implementations tend to become outdated over time. Version constraints on libraries such as `torch`, `numpy`, and `pandas` can further reduce transparency; for example, changes in `pandas` may alter frequency inference, seasonal periods m, and thus MASE computation.

`fev` **Python library.** To address these limitations, we introduce `fev`, a lightweight library that provides essential benchmarking functionality without unnecessary constraints. Its core features include task definition, data loading and splitting, prediction scoring, and result aggregation. `fev` depends only on Hugging Face `datasets` (Lhoest et al., 2021) and `pydantic` (Colvin et al., 2025) for input validation, and does not fix versions of commonly used packages such as `torch` or `numpy`. This enables seamless integration into existing pipelines. Using `datasets` also simplifies data loading and supports extensions such as multimodal forecasting. The `fev` code is open sourced under the Apache-2.0 license at `https://github.com/autogluon/fev`.

`fev` deliberately excludes model implementations, which are prone to becoming outdated. Instead, it provides adapters that convert data into formats compatible with popular forecasting libraries including GluonTS (Alexandrov et al., 2020), darts (Herzen et al., 2022), StatsForecast (Garza et al., 2022), AutoGluon (Shchur et al., 2023), and sktime (Löning et al., 2019).

Library API. The `fev` library is built around three main constructs. An `EvaluationWindow` is a single train–test split defined by a cutoff. A `Task` specifies a forecasting problem, including the dataset, horizon, covariates, targets, and evaluation metric, and may include multiple evaluation windows. Finally, a `Benchmark` is a collection of tasks. Benchmarks in `fev` can be defined using YAML files. Each task produces an evaluation summary that includes both metric values and the full task specification, ensuring unambiguous and comparable results. `fev` also provides utilities for aggregating results across tasks, as described in Sec. 4.

6 Related work

6.1 Existing benchmarks

Early deep learning approaches for time series forecasting (Lai et al., 2018; Salinas et al., 2020; Rangapuram et al., 2018) evaluated on small collections of datasets that varied across studies. The M4 (Makridakis et al., 2020) and M5 (Makridakis et al., 2022) competitions later became widely used evaluation tasks. The LTSF benchmark (Zhou et al., 2021; Zeng et al., 2023) introduced new datasets with a focus on long-horizon forecasting, but remained narrow in scope and was later criticized for over-representing similar datasets and unrealistic evaluation setups (Hewamalage et al., 2023).

Subsequent benchmarks, including the Monash repository (Godahewa et al., 2021), BasicTS+ (Shao et al., 2024), TFB (Qiu et al., 2024), ProbTS (Zhang et al., 2024), TSFM-bench (Li et al., 2025), Chronos Benchmarks (Ansari et al., 2024), TIME (Qiao et al., 2026), and GIFT-Eval (Aksu et al., 2024), broadened evaluation across domains, frequencies, and forecasting settings, while domain-specific benchmarks such as BOOM (Cohen et al., 2025) focused on particular applications. However, these benchmarks largely ignore covariates despite their practical importance.

In contrast, `fev-bench` provides broad coverage across domains, frequencies, and forecasting settings, including 46 tasks with covariates. Complementing this, the `fev` library provides lightweight evaluation infrastructure with standardized aggregation and confidence intervals for reproducible and statistically robust comparisons.

Model	Avg. win rate (%)	Skill score (%)	Median runtime / 100 series (s)	Leakage (%)	# failures
Chronos-2	91.4	47.3	0.8	0	0
TiRex	83.3	42.6	0.2	1	0
TimesFM-2.5	79.6	42.2	1.9	10	0
FlowState	73.3	39.9	2.3	8	0
Toto-1.0	72.4	40.8	22.3	8	0
TabPFN-TS	69.7	41.5	109.4	0	1
Moirai-2.0	67.4	39.3	0.4	28	0
Chronos-Bolt	67.0	38.9	0.3	0	0
TFT	51.4	32.2	828.8	0	0
Stat. Ensemble	45.5	20.7	146.9	0	4
CatBoost	34.4	23.0	29.5	0	0
Seasonal Naive	19.6	0.0	0.5	0	0

Table 4: Marginal probabilistic forecasting performance of select models (according to the SQL metric) on the full `fev-bench` benchmark. Full results are available in Appendix D.

Model	Avg. win rate (%)	Skill score (%)	Median runtime / 100 series (s)	Leakage (%)	# failures
Chronos-2	87.2	35.5	0.8	0	0
TiRex	77.0	30.0	0.2	1	0
TimesFM-2.5	76.1	30.2	1.9	10	0
FlowState	70.5	27.8	2.3	8	0
Toto-1.0	66.6	28.2	22.3	8	0
TabPFN-TS	63.9	29.8	109.4	0	1
Moirai-2.0	62.4	27.2	0.4	28	0
Chronos-Bolt	61.3	26.5	0.3	0	0
CatBoost	52.4	23.7	29.5	0	0
Stat. Ensemble	47.7	16.4	146.9	0	4
TFT	45.7	20.5	828.8	0	0
Seasonal Naive	20.0	0.0	0.5	0	0

Table 5: Marginal point forecasting performance of select models (according to the MASE metric) on the full `fev-bench` benchmark. Full results are available in Appendix D.

6.2 Aggregation strategies

Several aggregation methods have been proposed for forecasting benchmarks, each with different trade-offs.

Average rank is widely used in forecasting benchmarks (Aksu et al., 2024; Ansari et al., 2024). As shown in Appendix C.1, average rank is mathematically equivalent to the average win rate W_j, inducing the same model ordering. However, ranks scale with the number of models and do not naturally extend to pairwise comparisons. We therefore prefer win rates, which are bounded between 0 and 1 and extend directly to pairwise evaluation.

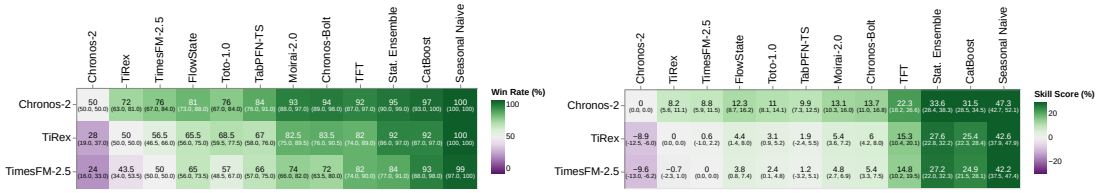
Bradley–Terry (Elo) scores (Bradley & Terry, 1952) have been applied across domains, from large language models (Chiang et al., 2024) to tabular benchmarks (Erickson et al., 2025). As shown in Appendix C.2, when all models are compared on all tasks, Bradley–Terry scores induce the same ranking as average win rates. Thus, average rank, average win rate, and Bradley–Terry scores are equivalent under our setup (Sec. 4), and we report win rates for simplicity.

Nemenyi post-hoc tests with critical difference diagrams (Demšar, 2006) are also rank-based, relying on average ranks. While they control family-wise error rates, they are often conservative and provide only binary significance decisions without effect sizes (Garcia & Herrera, 2008). In contrast, confidence intervals on win rates directly convey the magnitude and uncertainty of performance differences.

Geometric mean relative error (GMRE) (Ansari et al., 2024; Aksu et al., 2024) yields rankings identical to skill scores since $\mathrm{GMRE}_j = 1 - S_j$. We adopt this approach with two modifications: clipping extreme values to reduce outlier influence and reporting its complement (the skill score) to maintain a consistent "higher-is-better" interpretation.

7 Results

Models. Our evaluation focuses primarily on pretrained forecasting models that represent the current frontier in time series forecasting research. We select models based on three criteria: strong performance on existing benchmarks such as GIFT-Eval, publicly available implementations, and computational feasibility on consumer hardware (single NVIDIA A10G GPU with 24GB RAM). More details about the model configuration are provided in Appendix B.



(a) (b)

Figure 1: Pairwise win rates (a) and skill scores (b) of the top-3 models against other models under the SQL metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Full pairwise results are available in Appendix D. Best viewed on screen.

We evaluate nine pretrained models: Chronos-2 (Ansari et al., 2025), TimesFM-2.5 (Das et al., 2024), TiRex (Auer et al., 2025), FlowState (Graf et al., 2025), Chronos-Bolt (Ansari et al., 2024), Toto (Cohen et al., 2025), Moirai-2.0 (Woo et al., 2024), TabPFN-TS (Hoo et al., 2025), and Sundial (Liu et al., 2025). Chronos-2 and Toto-1.0 natively support multivariate forecasting ($D > 1$); all other models forecast each dimension independently. Chronos-2 supports past and future covariates, TabPFN-TS supports known covariates, and the remaining pretrained models ignore covariates.

We additionally evaluate statistical models (AutoETS, AutoARIMA, AutoTheta, and the SCUM ensemble) (Garza et al., 2022; Petropoulos & Svetunkov, 2020), simple baselines (Seasonal Naive, Naive, Drift) (Hyndman & Athanasopoulos, 2018), and task-specific models including CatBoost (Prokhorenkova et al., 2018), LightGBM (Ke et al., 2017), PatchTST (Nie et al., 2022), TFT (Lim et al., 2021), and DeepAR (Salinas et al., 2020).

Evaluation metrics. We follow the evaluation protocol from Sections 3 and 4. Tasks are evaluated using SQL for probabilistic forecasting and MASE for point forecasting, with results aggregated across all 100 tasks using win rates and skill scores for both marginal and pairwise comparisons. For broader context, we also report the following quantities.

Data leakage. For pretrained models, results on tasks with self-reported training overlap are imputed following the policy in Sec. 4.4.

Runtime. We report the median end-to-end runtime (training and inference across all evaluation windows) normalized to 100 time series per task by scaling runtimes by $100/(N \times W)$, where N is the number of series and W the number of evaluation windows. Although runtime depends on hardware and implementation details, it provides a useful measure of computational efficiency across modeling paradigms and training setups.

Model failures. If a model fails to produce a forecast on certain tasks (e.g., due to crashing), we replace its performance with the score of the Seasonal Naive baseline.

7.1 Results

Tables 4 & 5 summarize the marginal performance of selected models on `fev-bench`, showing the strongest models from each category for brevity. Full results for all 21 models are provided in the appendix, and the live leaderboard is available on Hugging Face. Chronos-2 achieves the highest average win rate and skill score values, followed by TiRex and TimesFM-2.5. Older pretrained models such as Chronos-Bolt and Toto rank lower. All pretrained models are substantially more accurate and faster than the best statistical and task-specific models (TFT and SCUM Ensemble).

To assess whether differences among the leading models reflect genuine improvements rather than evaluation noise, we examine pairwise comparisons under the SQL metric in Figures 1a and 1b. The confidence intervals indicate a statistically significant gap between Chronos-2 and the remaining models. In contrast, TiRex and TimesFM-2.5 show heavy overlap, suggesting no clear winner; under different benchmark compositions or task weightings, either could emerge as the second-best performer. At the same time, the overall ranking is not driven by the domains or frequencies that contribute the most tasks to `fev-bench`: when every domain, or alternatively every frequency group, receives equal total weight, the ranking of the models remains nearly identical (Appendix D.2).

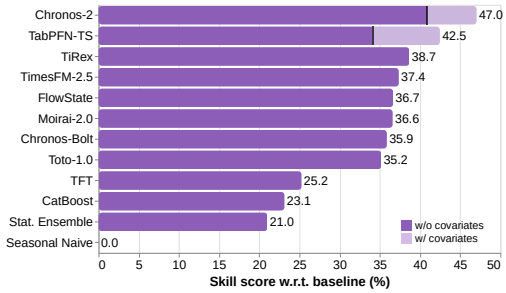


Figure 2: Average skill scores (SQL) on the 42 `fev-bench` tasks with dynamic covariates.

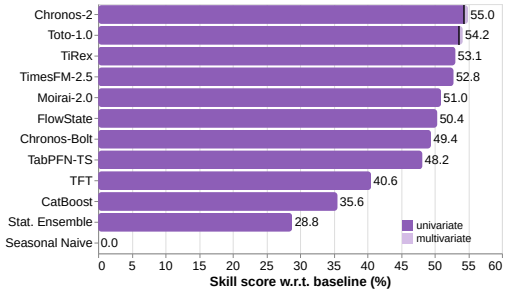


Figure 3: Average skill scores (SQL) on 35 multivariate tasks of `fev-bench`.

7.2 Covariates and multivariate forecasting

Forecasting with covariates. To measure the impact of covariates, we compare models with and without covariate inputs on the 42 `fev-bench` tasks containing dynamic covariates (Figure 2), reporting skill scores under the SQL metric. Covariates substantially improve performance for both Chronos-2 (47.0% vs. 40.9%) and TabPFN-TS (42.5% vs. 34.1%), showing that covariate information is a key driver of their strong results. Yet most pretrained forecasting models still ignore covariates, leaving substantial gains untapped and making covariate support an important direction for future work.

Multivariate forecasting. To measure the impact of native multivariate modeling, we compare models in multivariate and univariate modes on the 35 multivariate tasks in `fev-bench` (Figure 3), again using SQL skill scores. The gains from multivariate modeling are modest for both Chronos-2 (55.0% vs. 54.4%) and Toto-1.0 (54.7% vs. 54.1%). Similar findings were reported by Nie et al. (2022), who observed that univariate models often match multivariate ones. In contrast to the large gains from covariates, most forecasting benchmarks emphasize multivariate forecasting while largely ignoring covariates, highlighting the gap addressed by `fev-bench`.

8 Scope and limitations

`fev-bench` is designed as a broad benchmark spanning many domains, frequencies, and forecasting settings, providing a general view of model capabilities rather than optimizing for any single application domain. As a result, some models may perform differently on specialized tasks or domain-specific benchmarks, and we hope the lightweight design of `fev` will facilitate the creation of complementary benchmarks for specific use cases. Since we selected tasks by application domain following surveys of forecasting practice (Petropoulos et al., 2022), some domains and frequencies are represented by more tasks than others, although the model rankings remain nearly identical when every domain or frequency group receives equal weight (Appendix D.2). We deliberately exclude sub-minute frequencies, since the tasks studied on such data (e.g., EEG or industrial sensors) are typically anomaly detection or classification rather than forecasting. Finally, `fev-bench` prioritizes broad coverage, fast evaluation, and suitability for model development over fully leakage-free evaluation, and therefore relies on self-reported training overlap, as discussed in Sec. 4.4.

9 Conclusion

We introduced `fev-bench`, a benchmark with covariates, multivariate tasks, and principled aggregation methods for statistically robust evaluation across diverse forecasting domains. Our results highlight covariate support as a promising direction for pretrained forecasting models. Complementing this, we presented `fev`, a lightweight package for reproducible and extensible forecasting evaluation. Together, `fev-bench` and `fev` provide a practical foundation for developing and comparing the next generation of forecasting models.

References

Walmart recruiting - store sales forecasting. Kaggle competition, 2014. URL https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting.

Rossmann store sales. Kaggle competition, 2015. URL https://www.kaggle.com/competitions/rossmann-store-sales.

Recruit restaurant visitor forecasting. Kaggle competition, 2017. URL https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting.

Store sales - time series forecasting. Kaggle competition, 2020. URL https://www.kaggle.com/competitions/store-sales-time-series-forecasting.

CO2 emissions by country. Kaggle dataset, 2025a. URL https://www.kaggle.com/datasets/ulrikthygepedersen/co2-emissions-by-country. Country-level CO_2 emissions from 1960 to present.

Global life expectancy data (1950–2023). Kaggle dataset, 2025b. URL https://www.kaggle.com/datasets/nafayunnoor/global-life-expectancy-data-1950-2023. Life expectancy data by country from 1950 to 2023.

Renewable energy and weather conditions. Kaggle dataset, 2025c. URL https://www.kaggle.com/datasets/samanemami/renewable-energy-and-weather-conditions. Hourly weather and renewable energy generation data.

Tourism and economic impact. Kaggle dataset, 2025d. URL https://www.kaggle.com/datasets/bushraqurban/tourism-and-economic-impact. Key tourism and economic indicators for 200+ countries, 1999–2023.

UK COVID-19 dashboard data. Kaggle dataset, 2025e. URL https://www.kaggle.com/datasets/happyadam73/uk-covid19-dashboard-data-sqlite-compressed. UK COVID-19 data from official UK government sources.

Rohlik sales forecasting challenge V2. Kaggle competition, 2025. URL https://www.kaggle.com/competitions/rohlik-sales-forecasting-challenge-v2/data.

Aksu, T., Woo, G., Liu, J., Liu, X., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Gift-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. GluonTS: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda Arango, S., Kapoor, S., Zschiegner, J., Maddix, D. C., Mahoney, M. W., Torkkola, K., Gordon Wilson, A., Bohlke-Schneider, M., and Wang, Y. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

Ansari, A. F., Shchur, O., Küken, J., Auer, A., Han, B., Mercado, P., Rangapuram, S. S., Shen, H., Stella, L., Zhang, X., Goswami, M., Kapoor, S., Maddix, D. C., Guerron, P., Hu, T., Yin, J., Erickson, N., Desai, P. M., Wang, H., Rangwala, H., Karypis, G., Wang, Y., and Bohlke-Schneider, M. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. URL https://arxiv.org/abs/2510.15821.

Arango, S. P., Mercado, P., Kapoor, S., Ansari, A. F., Stella, L., Shen, H., Senetaire, H., Turkmen, C., Shchur, O., Maddix, D. C., et al. ChronosX: Adapting pretrained time series models with exogenous variables. *arXiv preprint arXiv:2503.12107*, 2025.

Athanasopoulos, G., Ahmed, R. A., and Hyndman, R. J. Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166, 2009.

Auer, A., Podest, P., Klotz, D., Böck, S., Klambauer, G., and Hochreiter, S. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *arXiv preprint arXiv:2505.23719*, 2025.

Bojer, C. S. and Meldgaard, J. P. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 37(2):587–603, 2021.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

Christiano, L. J., Eichenbaum, M., and Evans, C. L. Monetary policy shocks: What have we learned and to what end? *Handbook of macroeconomics*, 1:65–148, 1999.

Cohen, B., Khwaja, E., Doubli, Y., Lemaachi, S., Lettieri, C., Masson, C., Miccinilli, H., Ramé, E., Ren, Q., Rostamizadeh, A., du Terrail, J. O., Toon, A.-M., Wang, K., Xie, S., Xu, Z., Zhukova, V., Asker, D., Talwalkar, A., and Abou-Amal, O. This time is different: An observability perspective on time series foundation models, 2025. URL `https://arxiv.org/abs/2505.14766`.

Colvin, S., Jolibois, E., Ramezani, H., Garcia Badaracco, A., Dorsey, T., Montague, D., Matveenko, S., Trylesinski, M., Runkle, S., Hewitt, D., Hall, A., and Plot, V. Pydantic Validation, July 2025. URL `https://github.com/pydantic/pydantic`.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

David, E., Bellot, J., and Corff, S. L. HERMES: Hybrid error-corrector model with inclusion of external signals for nonstationary fashion time series. *arXiv preprint arXiv:2202.03224*, 2022.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

ECDC. Respiratory viruses weekly data. `https://github.com/EU-ECDC/Respiratory_viruses_weekly_data/tree/main`, 2025. Open data repository; weekly respiratory virus surveillance in the EU/EEA.

Efron, B. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pp. 569–593. Springer, 1992.

Erickson, N., Purucker, L., Tschalzev, A., Holzmüller, D., Desai, P. M., Salinas, D., and Hutter, F. TabArena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*, 2025.

et al., J. M. mlforecast: Scalable machine learning for time series forecasting. `https://github.com/Nixtla/mlforecast`, 2021. GitHub repository.

Fildes, R., Ma, S., and Kolassa, S. Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4):1283–1318, 2022.

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

Garcia, S. and Herrera, F. An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of machine learning research*, 9(12), 2008.

Garza, A., Canseco, M. M., Challú, C., and 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`.

Garza, A., Rosillo, R., Mendoza-Smith, R., Salinas, D., Williams, A. R., Ashok, A., Goswami, M., and Juárez, J. M. Impermanent: A live benchmark for temporal generalization in time series forecasting. *arXiv preprint arXiv:2603.08707*, 2026.

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

Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., and Montero-Manso, P. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

Graf, L., Ortner, T., WoŁşniak, S., Pantazi, A., et al. Flowstate: Sampling rate invariant time series forecasting. *arXiv preprint arXiv:2508.05287*, 2025.

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., and Grosch, G. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022. URL `http://jmlr.org/papers/v23/21-1177.html`.

Hewamalage, H., Ackermann, K., and Bergmeir, C. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2):788–832, 2023.

Hong, T., Pinson, P., and Fan, S. Global energy forecasting competition 2012, 2014.

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

Hyndman, R. J. and Athanasopoulos, G. *Forecasting: Principles and Practice*. OTexts, 2018.

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

Jiang, J., Han, C., Jiang, W., Zhao, W. X., and Wang, J. LibCity: A unified library towards efficient and comprehensive urban spatial-temporal prediction. *arXiv preprint arXiv:2304.14343*, 2023.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kolassa, S. Why the "best" point forecast depends on the error or accuracy measure. *International Journal of Forecasting*, 36(1):208–211, 2020.

Lago, J., Marcjasz, G., De Schutter, B., and Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.emnlp-demo.21`.

Li, Z., Qiu, X., Chen, P., Wang, Y., Cheng, H., Shu, Y., Hu, J., Guo, C., Zhou, A., Jensen, C. S., et al. TSFM-Bench: A comprehensive and unified benchmark of foundation models for time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5595–5606, 2025.

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

Liu, Y., Qin, G., Shi, Z., Chen, Z., Yang, C., Huang, X., Wang, J., and Long, M. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., and Király, F. J. sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*, 2019.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020.

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

Mancuso, P., Piccialli, V., and Sudoso, A. M. A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182:115102, 2021.

McCracken, M. and Ng, S. FRED-QD: A quarterly database for macroeconomic research. Technical report, National Bureau of Economic Research, 2020.

McCracken, M. W. and Ng, S. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

Meyer, M., Kaltenpoth, S., Albers, H., Zalipski, K., and Mueller, O. Ts-arena: A live forecast pre-registration platform for leakage-free evaluation of time series foundation models. In *1st ICLR Workshop on Time Series in the Age of Large Models*, 2026.

Mohaddes, K. and Raissi, M. Compilation, revision and updating of the Global VAR (GVAR) Database, 1979Q2-2023Q3. 2024.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

of Health Affairs, G. D. and Ministry of Health, S. A. Riyadh hospital admissions dataset (2020–2024). Kaggle dataset, 2025. URL `https://www.kaggle.com/datasets/datasetengineer/riyadh-hospital-admissions-dataset-20202024`. Hospital admissions in Riyadh over 2020–2024.

Open Power System Data. Data package time series. `https://doi.org/10.25832/time_series/2020-10-06`, 2020. (Primary data from various sources, for a complete list see URL).

Palaskar, S., Ekambaram, V., Jati, A., Gantayat, N., Saha, A., Nagar, S., Nguyen, N. H., Dayama, P., Sindhgatta, R., Mohapatra, P., et al. Automixer for improved multivariate time-series forecasting on business and IT observability data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 22962–22968, 2024.

Petropoulos, F. and Svetunkov, I. A simple combination of univariate models. *International journal of forecasting*, 36(1):110–115, 2020.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., and Babai, e. a. Forecasting: theory and practice. *International Journal of Forecasting*, 38(3):705–871, 2022.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Qiao, Z., Pan, S., Wang, A., Zhukova, V., Liu, Y., Jiang, X., Wen, Q., Long, M., Jin, M., and Liu, C. It's time: Towards the next generation of time series forecasting benchmarks. *arXiv preprint arXiv:2602.12147*, 2026.

Qiu, X., Hu, J., Zhou, L., Wu, X., Du, J., Zhang, B., Guo, C., Zhou, A., Jensen, C. S., Sheng, Z., et al. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*, 2024.

Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems*, 31, 2018.

Roque, L., Cerqueira, V., Soares, C., and Torgo, L. Cherry-picking in time series forecasting: How to select datasets to make your model shine. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20192–20199, 2025.

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

Shao, Z., Wang, F., Xu, Y., Wei, W., Yu, C., Zhang, Z., Yao, D., Sun, T., Jin, G., Cao, X., et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

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

Staffell, I., Pfenninger, S., and Johnson, N. A global model of hourly space heating and cooling demand at multiple spatial scales. *Nature Energy*, 8(12):1328–1344, 2023.

Tibshirani, R. Forecast scoring and calibration, 2023. URL `https://www.stat.berkeley.edu/~ryantibs/statlearn-s23/lectures/calibration.pdf`. Advanced Topics in Statistical Learning, Spring 2023.

van Renen, A., Horn, D., Pfeil, P., Vaidya, K. E., Dong, W., Narayanaswamy, M., Liu, Z., Saxena, G., Kipf, A., and Kraska, T. Why TPC is not enough: An analysis of the Amazon Redshift fleet. In *VLDB 2024*, 2024. URL `https://www.amazon.science/publications/why-tpc-is-not-enough-an-analysis-of-the-amazon-redshift-fleet`.

Vito, S. D. D., Massera, E., Piga, M., Martinotto, L., and Francia, G. D. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757, 2008. doi: 10.1016/j.snb.2007.09.060. UCI Machine Learning Repository - Air Quality dataset.

Wahdany, D., Cresswell, J., Guan, N., Ghomi, A. A., Boenisch, F., and Dziedzic, A. Dataset inference for data provenance and privacy auditing in tabular foundation models. *arXiv preprint arXiv:2610.24278*, 2026.

Wang, Z., Wen, Q., Zhang, C., Sun, L., Von Krannichfeldt, L., Pan, S., and Wang, Y. Benchmarks and custom package for energy forecasting. *arXiv preprint arXiv:2307.07191*, 2023.

Wilms, I. and Croux, C. Forecasting using sparse cointegration. *International Journal of Forecasting*, 32(4):1256–1267, 2016.

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

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, J., Wen, X., Zhang, Z., Zheng, S., Li, J., and Bian, J. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. *Advances in Neural Information Processing Systems*, 37:48045–48082, 2024.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Zhou, J., Lu, X., Xiao, Y., Su, J., Lyu, J., Ma, Y., and Dou, D. SDWPF: A dataset for spatial dynamic wind power forecasting challenge at KDD Cup 2022. *arXiv preprint arXiv:2208.04360*, 2022.

A Tasks

In total, `fev-bench` contains 100 time series forecasting tasks. In this section we provide the main statistics of these tasks together with citations to the sources of the datasets. For competition datasets we use their fixed forecast horizon H; for all others, H is set by a frequency–horizon mapping, except for a subset of hourly datasets where we use $H = 168$ to enable long-range forecasting. The number of evaluation windows W is then chosen to evenly split the series while ensuring that sufficient historical data is available for each forecast of length H. Dataset frequencies are reported using `pandas` frequency aliases (minu**T**ely, **H**ourly, **D**aily, **W**eekly, **M**onthly, **Q**uarterly, **Y**early).

The precise task definitions in YAML format are available under `https://github.com/autogluon/fev`. The datasets used for evaluation under `https://huggingface.co/datasets/autogluon/fev_datasets`.

A.1 GIFT-Eval

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
BizITObs - L2C	cloud	5T	288	20	31,968	1	7	0	0	0
BizITObs - L2C	cloud	H	24	20	2,664	1	7	0	0	0
ETT	energy	15T	96	20	69,680	2	7	0	0	0
ETT	energy	H	168	20	17,420	2	7	0	0	0
ETT	energy	D	28	20	724	2	7	0	0	0
ETT	energy	W	13	5	103	2	7	0	0	0
Hierarchical Sales	retail	D	28	10	1,825	118	1	0	0	0
Hierarchical Sales	retail	W	13	10	260	118	1	0	0	0
Hospital	healthcare	M	12	4	84	767	1	0	0	0
Jena Weather	nature	10T	144	20	52,704	1	21	0	0	0
Jena Weather	nature	D	28	11	366	1	21	0	0	0
Jena Weather	nature	H	24	20	8,784	1	21	0	0	0
Loop Seattle	mobility	D	28	10	365	323	1	0	0	0
Loop Seattle	mobility	5T	288	10	105,120	323	1	0	0	0
Loop Seattle	mobility	H	168	10	8,760	323	1	0	0	0
M-DENSE	mobility	D	28	10	730	30	1	0	0	0
M-DENSE	mobility	H	168	10	17,520	30	1	0	0	0
SZ Taxi	mobility	15T	96	10	2,976	156	1	0	0	0
SZ Taxi	mobility	H	168	2	744	156	1	0	0	0
Solar	energy	W	13	1	52	137	1	0	0	0
Solar	energy	D	28	10	365	137	1	0	0	0

Table 6: Tasks based on datasets coming from the GIFT-Eval corpus (Aksu et al., 2024).

The GIFT-Eval corpus (Aksu et al., 2024) contains various univariate and multivariate datasets, none of which provide covariates. The original datasets have been collected from sources such as Godahewa et al. (2021); Jiang et al. (2023); Mancuso et al. (2021); Wu et al. (2021); Palaskar et al. (2024).

A.2 Macroeconomic datasets

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
Australian Tourism	econ	Q	8	2	36	89	1	0	0	0
FRED-MD - CEE	econ	M	12	20	798	1	3	4	0	0
FRED-MD - Macro	econ	M	12	20	798	1	51	0	0	0
FRED-QD - CEE	econ	Q	8	20	266	1	3	4	0	0
FRED-QD - Macro	econ	Q	8	20	266	1	51	0	0	0
GVAR	econ	Q	8	10	178	33	6	3	0	0
US Consumption	econ	M	12	10	792	31	1	0	0	0
US Consumption	econ	Q	8	10	262	31	1	0	0	0
US Consumption	econ	Y	5	10	64	31	1	0	0	0
World CO2 Emissions	econ	Y	5	9	60	191	1	0	0	0
World Life Expectancy	econ	Y	5	10	74	237	1	0	0	0
World Tourism	econ	Y	5	2	21	178	1	0	0	0

Table 7: Tasks based on various macroeconomic datasets.

We consider various macroeconomic datasets such as GVAR (Mohaddes & Raissi, 2024), US Consumption (Wilms & Croux, 2016), Australian Tourism (Athanasopoulos et al., 2009), FRED-MD (McCracken & Ng, 2016), FRED-QD (McCracken & Ng, 2020), world CO2 emmissions (Kag, 2025a), life expectancy (Kag, 2025b) and global tourism (Kag, 2025d).

For each of FRED-MD and FRED-QD, we create two forecasting tasks. The first follows the CEE model (Christiano et al., 1999) and focuses on forecasting employment, inflation, and federal funds rate indicators. The second task involves jointly forecasting 51 core macroeconomic indicators. Note that we use the snapshot of FRED-MD corresponding to August 2025, which is different from FRED-MD snapshot used in Godahewa et al. (2021).

A.3 Energy datasets

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
ENTSO-e Load	energy	15T	96	20	175,292	6	1	0	3	0
ENTSO-e Load	energy	30T	96	20	87,645	6	1	0	3	0
ENTSO-e Load	energy	H	168	20	43,822	6	1	0	3	0
EPF-BE	energy	H	24	20	52,416	1	1	0	2	0
EPF-DE	energy	H	24	20	52,416	1	1	0	2	0
EPF-FR	energy	H	24	20	52,416	1	1	0	2	0
EPF-NP	energy	H	24	20	52,416	1	1	0	2	0
EPF-PJM	energy	H	24	20	52,416	1	1	0	2	0
ERCOT	energy	D	28	20	6,452	8	1	0	0	0
ERCOT	energy	H	168	20	154,872	8	1	0	0	0
ERCOT	energy	M	12	15	211	8	1	0	0	0
ERCOT	energy	W	13	20	921	8	1	0	0	0
GFC12	energy	H	168	10	39,414	11	1	0	1	0
GFC14	energy	H	168	20	17,520	1	1	0	1	0
GFC17	energy	H	168	20	17,544	8	1	0	1	0
Solar with Weather	energy	15T	96	20	198,600	1	1	2	7	0
Solar with Weather	energy	H	24	20	49,648	1	1	2	7	0

Table 8: Tasks based on datasets related to energy generation and consumption.

These datasets include electricity price forecasting (EPF) benchmark (Lago et al., 2021), ERCOT generation data (Ansari et al., 2024), ENTSO-e load data (Open Power System Data, 2020) with weather originating from `Renewables.ninja` (Staffell et al., 2023), and solar generation with weather (Kag, 2025c).

A.4 BOOMLET

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
BOOMLET - 1062	cloud	5T	288	20	16,384	1	21	0	0	0
BOOMLET - 1209	cloud	5T	288	20	16,384	1	53	0	0	0
BOOMLET - 1225	cloud	T	60	20	16,384	1	49	0	0	0
BOOMLET - 1230	cloud	5T	288	20	16,384	1	23	0	0	0
BOOMLET - 1282	cloud	T	60	20	16,384	1	35	0	0	0
BOOMLET - 1487	cloud	5T	288	20	16,384	1	54	0	0	0
BOOMLET - 1631	cloud	30T	96	20	10,463	1	40	0	0	0
BOOMLET - 1676	cloud	30T	96	20	10,463	1	100	0	0	0
BOOMLET - 1855	cloud	H	24	20	5,231	1	52	0	0	0
BOOMLET - 1975	cloud	H	24	20	5,231	1	75	0	0	0
BOOMLET - 2187	cloud	H	24	20	5,231	1	100	0	0	0
BOOMLET - 285	cloud	T	60	20	16,384	1	75	0	0	0
BOOMLET - 619	cloud	T	60	20	16,384	1	52	0	0	0
BOOMLET - 772	cloud	T	60	20	16,384	1	67	0	0	0
BOOMLET - 963	cloud	T	60	20	16,384	1	28	0	0	0

Table 9: Tasks based on datasets from BOOMLET (Cohen et al., 2025).

We include the multivariate observability datasets from the BOOMLET benchmark (Cohen et al., 2025). BOOMLET is a subset of the larger BOOM benchmark curated by the original authors. We additionally limit our attention to datasets with frequency of at least 1 minute to avoid including too many datasets from a single source to `fev-bench`.

A.5 Forecasting competitions

We use datasets from forecasting competitions held on `kaggle.com` (Bojer & Meldgaard, 2021). These include Favorita store sales & transactions (Kag, 2020), the M5 competition (Makridakis et al., 2022), restaurant visitor & reservation (Kag, 2017), Rossmann (Kag, 2015), Walmart (Kag, 2014), and Rohlik (Roh, 2025) store sales forecasting competitions. We also consider the KDD Cup 2022 dataset where the goal is to predict wind power generation (Zhou et al., 2022), and the Global Energy Forecasting Competitions held in 2012, 2014 and 2017 (Hong et al., 2014).

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
Favorita Store Sales	retail	M	12	2	54	1,579	1	1	1	6
Favorita Store Sales	retail	W	13	10	240	1,579	1	1	1	6
Favorita Store Sales	retail	D	28	10	1,688	1,579	1	1	2	6
Favorita Transactions	retail	M	12	2	54	51	1	1	0	5
Favorita Transactions	retail	W	13	10	240	51	1	1	0	5
Favorita Transactions	retail	D	28	10	1,688	51	1	1	1	5
KDD Cup 2022	energy	D	14	10	243	134	1	9	0	0
KDD Cup 2022	energy	10T	288	10	35,279	134	1	9	0	0
KDD Cup 2022	energy	30T	96	10	11,758	134	1	9	0	0
M5	retail	M	12	1	58	30,490	1	0	8	5
M5	retail	W	13	1	257	30,490	1	0	8	5
M5	retail	D	28	1	1,810	30,490	1	0	8	5
Restaurant	retail	D	28	8	296	817	1	0	0	4
Rohlik Orders	retail	W	8	5	170	7	1	9	4	0
Rohlik Orders	retail	D	61	5	1,197	7	1	9	4	0
Rohlik Sales	retail	W	8	1	150	5,243	1	1	13	7
Rohlik Sales	retail	D	14	1	1,046	5,390	1	1	13	7
Rossmann	retail	W	13	8	133	1,115	1	1	4	10
Rossmann	retail	D	48	10	942	1,115	1	1	5	10
Walmart	retail	W	39	1	143	2,936	1	0	10	4

Table 10: Tasks based on datasets coming from various forecasting competitions

A.6 Other sources

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
ECDC ILI	healthcare	W	13	10	201	25	1	0	0	0
Hermes	retail	W	52	1	261	10,000	1	0	1	2
Hospital Admissions	healthcare	D	28	20	1,731	8	1	0	0	0
Hospital Admissions	healthcare	W	13	16	246	8	1	0	0	0
Redset	cloud	5T	288	10	25,920	118	1	0	0	1
Redset	cloud	15T	96	10	8,640	126	1	0	0	1
Redset	cloud	H	24	10	2,160	138	1	0	0	1
UCI Air Quality	nature	H	168	20	9,357	1	4	0	3	0
UCI Air Quality	nature	D	28	11	389	1	4	0	3	0
UK COVID - Nation - Cumulative	healthcare	D	28	20	729	4	3	5	0	0
UK COVID - Nation - Cumulative	healthcare	W	8	4	105	4	3	5	0	0
UK COVID - Nation - New	healthcare	D	28	20	729	4	3	5	0	0
UK COVID - Nation - New	healthcare	W	8	4	105	4	3	5	0	0
UK COVID - UTLA - Cumulative	healthcare	W	13	5	104	214	1	0	0	0
UK COVID - UTLA - New	healthcare	D	28	10	721	214	1	0	0	0

Table 11: Tasks based on datasets collected from other sources.

We also include datasets from the following miscellaneous sources.

- Influenza-like-illness cases collected by the European Centre for Disease Prevention and Control (ECDC, 2025).
- Fashion trend data from Hermes (David et al., 2022).
- Hospital admissions data from Riyadh (of Health Affairs & Ministry of Health, 2025).
- Query counts for Amazon Redshift database servers (van Renen et al., 2024).
- Solar energy generation with corresponding weather covariates (Kag, 2025c).
- Air quality measurements in an Italian city with accompanying weather data (Vito et al., 2008).
- COVID-19 cases, hospital admissions, and deaths in the United Kingdom at different administrative levels (Kag, 2025e).

B Models

We evaluate nine pretrained models on `fev-bench`, whose key properties are summarized in Table 12. Most are decoder-only transformers, except TiRex (decoder-only xLSTM), Chronos-Bolt Base (encoder-decoder transformer), Chronos-2 (encoder-only transformer), and FlowState (SSM encoder with functional basis decoder). All models except TabPFN-TS and FlowState process non-overlapping patches of time series rather than individual observations. Toto, TabPFN-TS, and Sundial produce sample forecasts, while the others produce direct multi-step-ahead quantile forecasts. Toto and Chronos-2 are the only models that natively support multivariate targets. We provided only known

covariates to TabPFN-TS, excluding past covariates as recommended by the authors. Chronos-2 supports both past and known covariates.

For all pretrained models, we keep hyperparameters at their default values unless specified in the table. For Toto, we reduced the samples per batch and batch size to avoid out-of-memory errors on large multivariate datasets. We run all pretrained models on a g5.2xlarge AWS instance with a single A10G GPU (24GB GPU RAM, 32GB RAM), using PyTorch 2.6 with CUDA 12.6 via AWS Deep Learning Containers. The version in the table below refers to either the PyPI package version, or the date on which the official repository was cloned if no PyPI package is provided by the authors.

Model Name	Hugging Face ID	Batch size	Version	Max Context	Hyperparameters
Chronos-2	`amazon/chronos-2`	100	2.2.2	8192	`{cross_learning: True}`
TiRex	`NX-AI/TiRex`	512	2025-09-01	2048	-
Toto	`Datadog/Toto-Open-Base-1.0`	24	2025-08-01	4096	`{samples_per_batch: 8}`
Moirai 2.0	`Salesforce/moirai-2.0-R-small`	128	2025-08-10	4000	-
Chronos-Bolt	`amazon/chronos-bolt-base`	256	1.5.3	2048	-
TimesFM 2.5	`google/timesfm-2.5-200m-pytorch`	256	2025-09-28	16000	-
TabPFN-TS	`Prior-Labs/TabPFN-v2-reg`	-	1.0.10	4096	`{checkpoint: '2noar4o2'}`
FlowState	`ibm-granite/granite-timeseries-flowstate-r1`	16	2026-05-06	auto	-
Sundial	`thuml/sundial-base-128m`	512	2025-09-01	2880	-

Table 12: Properties of different pretrained time series forecasting models.

We include statistical baselines from the StatsForecast `v2.0.1` (Garza et al., 2022), such as AutoETS, AutoARIMA, AutoTheta, as well as the SCUM ensemble (Petropoulos & Svetunkov, 2020). To avoid long runtimes, we truncate the context length of statistical models to 1000 steps and set the maximum season length to 200 for AutoETS, AutoTheta, AutoARIMA and SCUM Ensemble.

We train LightGBM (Ke et al., 2017) and CatBoost (Prokhorenkova et al., 2018) in recursive mode using MLForecast `v0.14.0` (et al., 2021), with preprocessing hyperparameters (target transforms, lag transforms) selected via grid search. We also train DeepAR (Salinas et al., 2020), PatchTST (Nie et al., 2022), and TFT (Lim et al., 2021) using AutoGluon-TimeSeries `v1.5.0` (Shchur et al., 2023), refitting on each evaluation window. We use seasonal naive as a fallback when there is insufficient training data for deep learning models in some windows.

Statistical and global models were run on `m6i.4xlarge` AWS instances with 16 vCPU cores and 64GB RAM. All model wrappers are available at `github.com/autogluon/fev/models`.

C Extended discussion of aggregation methods

This appendix clarifies how average win rates W_j (Eq. (3)) relate to other aggregation methods commonly used in benchmarking, such as average ranks (Aksu et al., 2024; Ansari et al., 2024) and Bradley–Terry ("Elo") scores (Erickson et al., 2025). We show that all three induce the same ordering of models.

C.1 Average win rate and average rank are equivalent

For a given task r and model j, let

$$M_{\text{lower}} = \sum_{\substack{k=1 \\ k\neq j}}^{M} \mathbb{1}(E_{rk} < E_{rj}), \qquad M_{\text{tied}} = \sum_{\substack{k=1 \\ k\neq j}}^{M} \mathbb{1}(E_{rk} = E_{rj}).$$

The midrank of j on task r is

$$\text{rank}_{rj} \;=\; 1 + M_{\text{lower}} + \tfrac{1}{2} M_{\text{tied}}.$$

Its contribution to W_j equals

$$\frac{1}{M-1} \sum_{k\neq j} \Big(\mathbb{1}(E_{rj} < E_{rk}) + \tfrac{1}{2}\, \mathbb{1}(E_{rj} = E_{rk}) \Big) = 1 - \frac{\text{rank}_{rj} - 1}{M-1}.$$

Averaging over tasks gives

$$W_j = 1 - \frac{\overline{\text{rank}}_j - 1}{M-1}, \qquad \overline{\text{rank}}_j = \tfrac{1}{R} \sum_{r=1}^{R} \text{rank}_{rj}\,.$$

Thus, W_j and $\overline{\mathrm{rank}}_j$ are affinely equivalent and induce the same ordering of models (with lower rank $\leftrightarrow$ higher win rate).

C.2 Average win rate and Bradley–Terry (Elo) scores result in the same ranking

The Bradley–Terry (BT) model, also known as Elo rating (Chiang et al., 2024), provides a parametric way to convert pairwise win rates into latent skill scores. In contrast to the nonparametric average win rate W_j, the BT model assumes that each model j has an underlying skill parameter $\theta_j \in \mathbb{R}$, and that the probability of j outperforming k follows a logistic link:

$$\Pr(E_{rj} < E_{rk}) \;=\; \sigma\big(\lambda(\theta_j - \theta_k)\big), \qquad \sigma(x) = \tfrac{1}{1+e^{-x}}, \;\; \lambda > 0.$$

Here λ is a scaling constant (in Elo, $\lambda = \ln 10/400$). The parameters $\theta = (\theta_1, \ldots, \theta_M)$ are estimated by maximum likelihood:

$$\hat{\theta} \;\in\; \arg\max_{\theta \in \mathbb{R}^M} \sum_{j<m} \Big[W_{jm} \log \sigma(\lambda(\theta_j - \theta_m)) + (1 - W_{jm}) \log \sigma(\lambda(\theta_m - \theta_j)) \Big].$$

Typically some identifiability constraint is added, such as fixing $\theta_\beta = 1000$ for a chosen baseline β.

Proposition C.1. *Suppose all M models are compared on the same R tasks, with pairwise win rates W_{jk} (Eq. (5)) and average win rates $W_j = \frac{1}{M-1} \sum_{k \neq j} W_{jk}$ (Eq. (3)). At the BT MLE with scale $\lambda > 0$,*

$$\theta_j > \theta_k \iff W_j > W_k, \qquad \theta_j = \theta_k \iff W_j = W_k.$$

Proof. Differentiating the log-likelihood gives the score equations

$$\frac{\partial \ell}{\partial \theta_j} = \lambda \sum_{m \neq j} \Big(W_{jm} - \sigma(\lambda(\theta_j - \theta_m)) \Big) = 0.$$

Subtracting the equations for j and k yields

$$(M-1)(W_j - W_k) = \sum_{m \neq j,k} \big[\sigma(\lambda(\theta_j - \theta_m)) - \sigma(\lambda(\theta_k - \theta_m)) \big] + \big[\sigma(\lambda\Delta) - \sigma(-\lambda\Delta) \big],$$

where $\Delta = \theta_j - \theta_k$. Each term on the right is strictly increasing in Δ, so the whole expression has the same sign as Δ. Thus $\mathrm{sign}(W_j - W_k) = \mathrm{sign}(\theta_j - \theta_k)$, with equality iff $\Delta = 0$. Strict concavity of the BT log-likelihood ensures uniqueness of the solution up to translation. □

Conclusion. Average win rate W_j and BT/Elo scores θ_j induce the same ordering of models, with higher win rate corresponding to higher Elo score.

D Extended results

Tabular results for each dataset-task combination, updated marginal and pairwise results in an interactive format are available at `huggingface.co/spaces/autogluon/fev-bench`. The raw results in CSV format are available at `github.com/autogluon/fev`.

The failures for Stat. Ensemble, AutoARIMA, and AutoETS models in the tables below correspond to the models exceeding the 6 hour time limit for a single task. TabPFN-TS failed on 1 task due to out of memory errors, and DeepAR failed due to encountering NaNs during training.

D.1 Marginal performance

Model	Avg. win rate (%)	Skill score (%)	Median runtime / 100 series (s)	Leakage (%)	# failures
Chronos-2	91.4	47.3	0.8	0	0
TiRex	83.3	42.6	0.2	1	0
TimesFM-2.5	79.6	42.2	1.9	10	0
FlowState	73.3	39.9	2.3	8	0
Toto-1.0	72.4	40.8	22.3	8	0
TabPFN-TS	69.7	41.5	109.4	0	1
Moirai-2.0	67.4	39.3	0.4	28	0
Chronos-Bolt	67.0	38.9	0.3	0	0
TFT	51.4	32.2	828.8	0	0
Sundial-Base	48.0	33.4	8.0	1	0
PatchTST	45.9	30.1	663.6	0	0
Stat. Ensemble	45.5	20.7	146.9	0	4
DeepAR	44.2	29.1	1113.5	0	3
AutoARIMA	41.4	21.3	20.1	0	4
CatBoost	34.4	23.0	29.5	0	0
AutoETS	33.0	-26.8	3.5	0	3
LightGBM	31.8	21.0	3.0	0	0
AutoTheta	27.9	5.5	3.3	0	0
Seasonal Naive	19.6	0.0	0.5	0	0
Naive	12.7	-45.4	0.5	0	0
Drift	10.1	-45.8	0.5	0	0

Table 13: Marginal probabilistic forecasting performance of all models (according to the SQL metric) on the full `fev-bench` benchmark. The reported metrics are defined in Sections 4.1 and 7.

Model	Avg. win rate (%)	Skill score (%)	Median runtime / 100 series (s)	Leakage (%)	# failures
Chronos-2	87.2	35.5	0.8	0	0
TiRex	77.0	30.0	0.2	1	0
TimesFM-2.5	76.1	30.2	1.9	10	0
FlowState	70.5	27.8	2.3	8	0
Toto-1.0	66.6	28.2	22.3	8	0
TabPFN-TS	63.9	29.8	109.4	0	1
Moirai-2.0	62.4	27.2	0.4	28	0
Chronos-Bolt	61.3	26.5	0.3	0	0
Sundial-Base	53.6	24.7	8.0	1	0
CatBoost	52.4	23.7	29.5	0	0
LightGBM	49.4	21.7	3.0	0	0
Stat. Ensemble	47.7	16.4	146.9	0	4
TFT	45.7	20.5	828.8	0	0
PatchTST	41.3	18.4	663.6	0	0
DeepAR	38.6	17.5	1113.5	0	3
AutoARIMA	36.2	11.6	20.1	0	4
AutoTheta	33.3	11.0	3.3	0	0
AutoETS	33.1	2.3	3.5	0	3
Seasonal Naive	20.0	0.0	0.5	0	0
Naive	18.2	-16.7	0.5	0	0
Drift	15.4	-18.1	0.5	0	0

Table 14: Marginal point forecasting performance of all models (according to the MASE metric) on the full `fev-bench` benchmark. The reported metrics are defined in Sections 4.1 and 7.

D.2 Robustness of the rankings

In this section we check whether the rankings reported in Table 13 depend on two design choices of `fev-bench`: the uniform weighting of tasks, which may over-represent some domains and frequencies, and the imputation of tasks with reported leakage. For brevity, we report results only for the SQL metric and for the subset of models shown in the main text.

Frequency- and domain-balanced weighting. `fev-bench` contains more tasks from some domains (e.g., energy, retail and cloud) than from others (e.g., mobility and nature), and more tasks with minutely, hourly and daily frequency than with monthly, quarterly or yearly frequency. One may therefore ask whether the rankings are driven by the dominant domains or frequencies. To answer this question, we generalize the average win rate (Eq. (3)) and skill score (Eq. (4)) to weighted averages over tasks, where task r receives weight w_r with $\sum_{r=1}^{R} w_r = 1$, so that uniform weighting corresponds to the special case $w_r = 1/R$. Let the tasks be partitioned into G groups, and let $g(r)$ denote the group of task r and n_g the number of tasks in group g. We then set $w_r = 1/(G \cdot n_{g(r)})$, such that each group receives the same total weight $1/G$. We consider two such groupings: by the $G = 7$ domains from Table 2, and by the $G = 6$ frequency groups T, H, D, W, M, and Q+Y, where all minutely frequencies (e.g., 5T, 10T, 15T) form a single group T, and quarterly and yearly tasks are merged into a single group because of their small number. As shown in Table 15, the ranking of the models remains nearly identical under all three weighting schemes. We nevertheless retain uniform weighting for the main results, as this is the standard practice in existing benchmarks.

Excluding leaked tasks. As discussed in Sec. 4.4, `fev-bench` imputes the scores of pretrained models on tasks with reported training overlap with the scores of Chronos-Bolt (Base). To verify that our conclusions do not depend on this imputation, we instead drop the 10 tasks that are flagged as leaky for at least one model, and recompute the scores for all models on the remaining 90 tasks. We omit Moirai-2.0 from this comparison, since its reported overlap covers too many tasks to be dropped without losing a large fraction of the benchmark. As shown in Table 16, the ranking of the models on the leakage-free tasks is unchanged compared to the full set of tasks.

	Uniform		Frequency-balanced		Domain-balanced	
Model	**Win rate (%)**	**Skill score (%)**	**Win rate (%)**	**Skill score (%)**	**Win rate (%)**	**Skill score (%)**
Chronos-2	91.4	47.3	90.8	44.1	91.2	45.3
TiRex	83.3	42.6	84.1	40.1	84.1	41.7
TimesFM-2.5	79.6	42.2	77.4	39.0	80.2	41.3
FlowState	73.3	39.9	74.2	37.6	74.5	39.2
Toto-1.0	72.5	40.8	69.1	37.2	71.8	39.9
TabPFN-TS	69.7	41.5	70.6	38.9	68.8	39.7
Moirai-2.0	67.5	39.3	64.1	35.7	67.3	38.4
Chronos-Bolt	67.0	38.9	65.8	36.1	67.5	38.1
TFT	51.4	32.2	45.5	26.9	48.7	28.8
Stat. Ensemble	45.5	20.7	52.7	23.2	47.6	21.5
CatBoost	34.4	23.0	31.8	20.5	33.0	20.6
Seasonal Naive	19.7	0.0	19.4	0.0	19.8	0.0

Table 15: Marginal probabilistic forecasting performance (according to the SQL metric) on `fev-bench` with uniform task weights, frequency-balanced task weights (each of the 6 frequency groups receives the same total weight), and domain-balanced task weights (each of the 7 domains receives the same total weight).

	All tasks		Leakage-free tasks	
Model	**Win rate (%)**	**Skill score (%)**	**Win rate (%)**	**Skill score (%)**
Chronos-2	91.3	47.3	90.8	47.0
TiRex	83.3	42.6	83.2	43.1
TimesFM-2.5	79.9	42.2	81.2	43.0
FlowState	74.2	39.9	74.5	40.4
Toto-1.0	73.2	40.8	73.7	41.4
TabPFN-TS	70.6	41.5	71.6	42.2
Chronos-Bolt	68.1	38.9	68.1	39.3
TFT	52.4	32.2	52.3	32.6
Stat. Ensemble	46.7	20.7	46.0	20.1
CatBoost	35.7	23.0	34.7	22.4
Seasonal Naive	20.4	0.0	20.7	0.0

Table 16: Marginal probabilistic forecasting performance (according to the SQL metric) on all 100 tasks of `fev-bench` and on the 90 tasks without reported leakage for any model. Moirai-2.0 is omitted from both columns, so the scores on all tasks differ slightly from Table 13.

D.3 Pairwise comparison

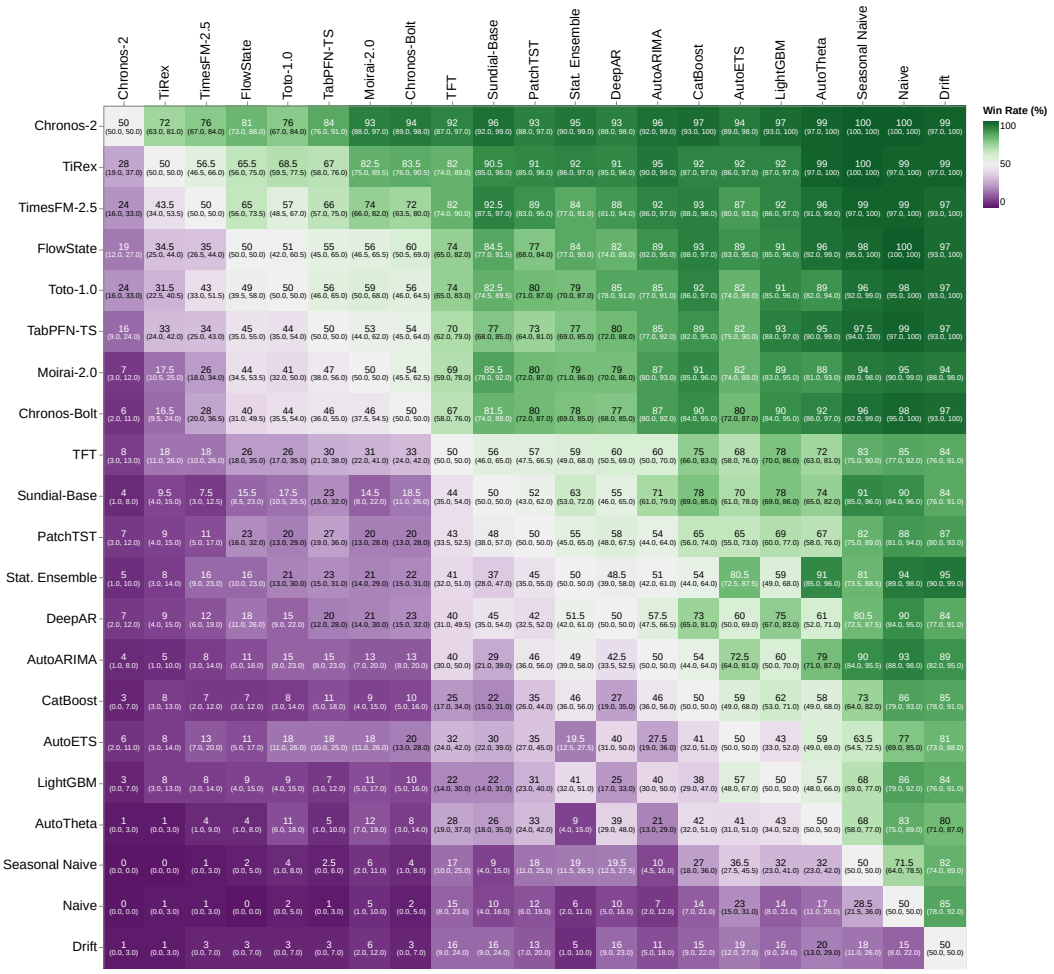


Figure 4: Pairwise win rates W_{jk} (Eq. (5)) of all models against each other under the scaled quantile loss (SQL) metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Best viewed on screen.

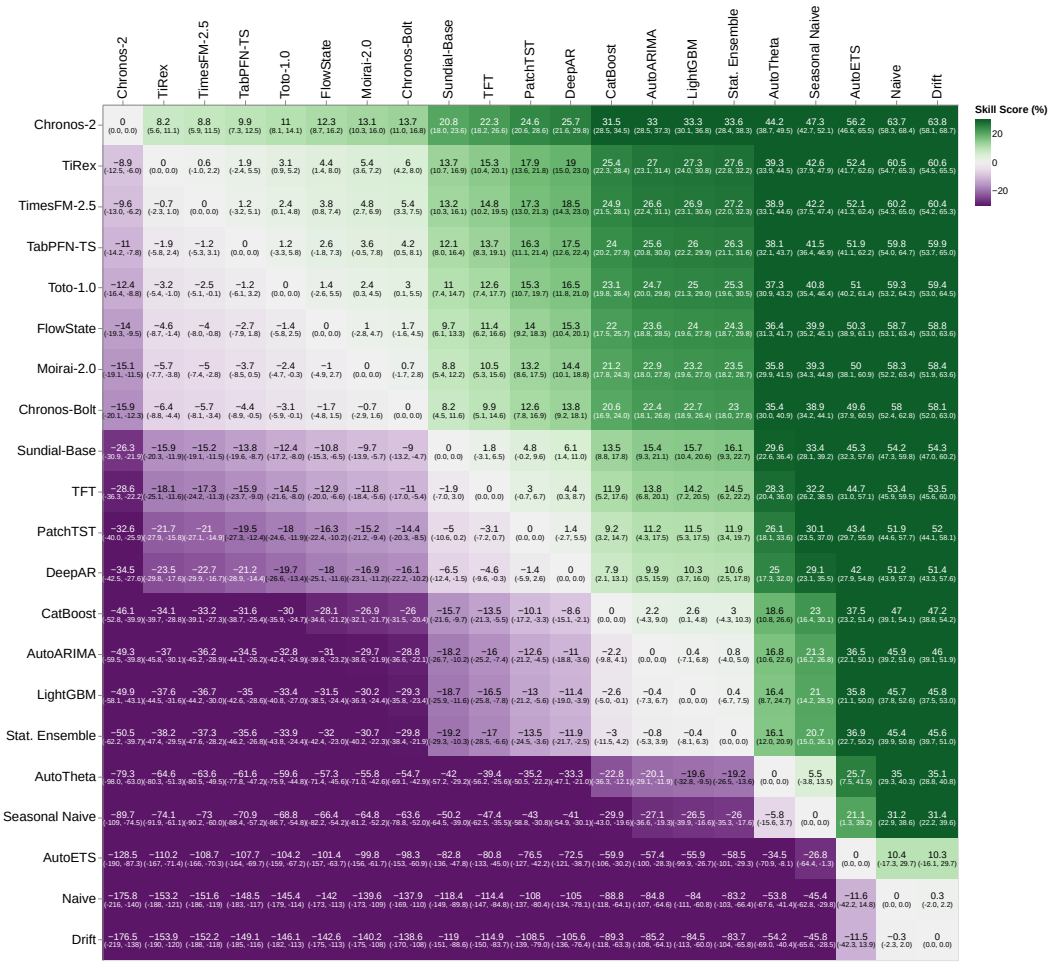


Figure 5: Pairwise skill scores S_{jk} (Eq. (6)) of all models against each other under the scaled quantile loss (SQL) metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Note that pairwise skill score is not symmetric, $S_{jk} \neq S_{kj}$. Best viewed on screen.

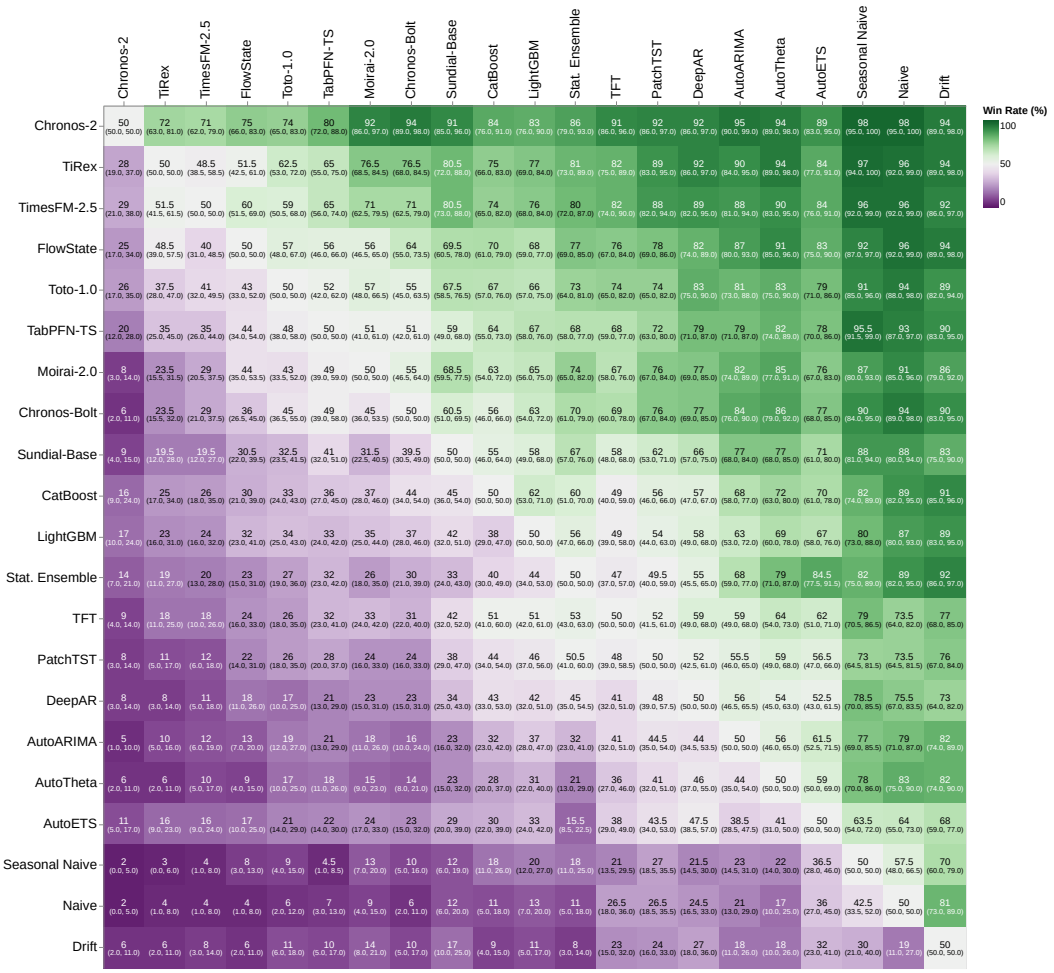


Figure 6: Pairwise win rates W_{jk} (Eq. (5)) of all models against each other under the mean absolute scaled error (MASE) metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Best viewed on screen.

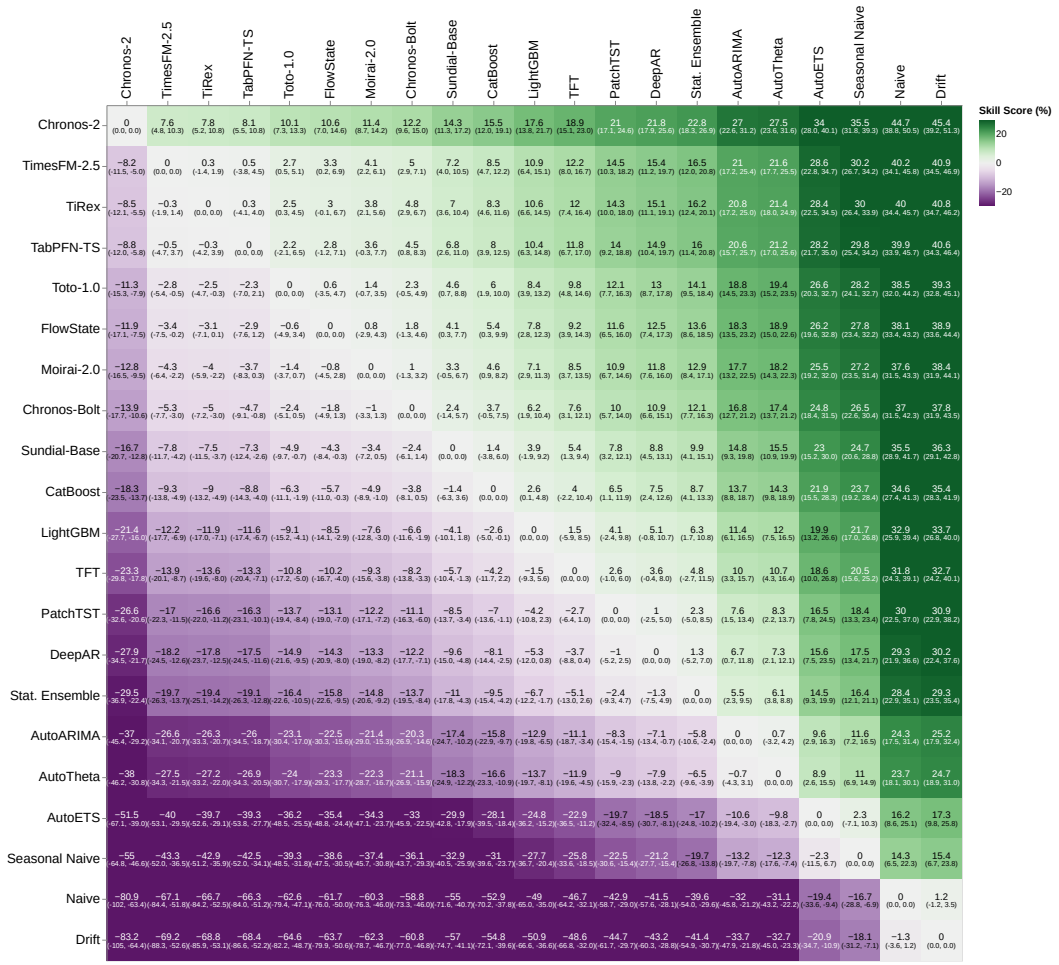


Figure 7: Pairwise skill scores S_{jk} (Eq. (6)) of all models against each other under the mean absolute scaled error (MASE) metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Note that pairwise skill score is not symmetric, $S_{jk} \neq S_{kj}$. Best viewed on screen.

E `fev-bench-mini`: A representative subset of `fev-bench`

E.1 Tasks

`fev-bench-mini` consists of 20 tasks that are representative of the full 100 tasks comprising `fev-bench` (Appendix A).

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
BOOMLET - 1282	cloud	T	60	20	16,384	1	35	0	0	0
BOOMLET - 1676	cloud	30T	96	20	10,463	1	100	0	0	0
BOOMLET - 619	cloud	T	60	20	16,384	1	52	0	0	0
BizITObs - L2C	cloud	5T	288	20	31,968	1	7	0	0	0
EPF-NP	energy	H	24	20	52,416	1	1	0	2	0
ETT	energy	15T	96	20	69,680	2	7	0	0	0
ETT	energy	H	168	20	17,420	2	7	0	0	0
GFC14	energy	H	168	20	17,520	1	1	0	1	0
Hospital Admissions	healthcare	W	13	16	246	8	1	0	0	0
Hospital Admissions	healthcare	D	28	20	1,731	8	1	0	0	0
Jena Weather	nature	H	24	20	8,784	1	21	0	0	0
M-DENSE	mobility	D	28	10	730	30	1	0	0	0
Rohlik Orders	retail	D	61	5	1,197	7	1	9	4	0
Rossmann	retail	W	13	8	133	1,115	1	1	4	10
Rossmann	retail	D	48	10	942	1,115	1	1	5	10
Solar with Weather	energy	H	24	20	49,648	1	1	2	7	0
UCI Air Quality	nature	H	168	20	9,357	1	4	0	3	0
UK COVID - Nation - Cumulative	healthcare	D	28	20	729	4	3	5	0	0
US Consumption	econ	Y	5	10	64	31	1	0	0	0
World CO2 Emissions	econ	Y	5	9	60	191	1	0	0	0

Table 17: Tasks included in `fev-bench-mini`.

E.2 Evaluation results

For completeness, we provide the evaluation results on `fev-bench-mini`. The overall ranking of the models, win rates and skill scores align with the scores on the full benchmark reported in Appendix D.

Model	Avg. win rate (%)	Skill score (%)	Median runtime / 100 series (s)	Leakage (%)	# failures
Chronos-2	90.3	51.1	1.2	0	0
TiRex	79.5	43.4	0.4	0	0
TimesFM-2.5	76.2	44.0	5.2	5	0
Toto-1.0	73.2	43.0	44.1	5	0
TabPFN-TS	70.2	47.6	270.3	0	0
FlowState	70.2	41.8	3.2	5	0
Moirai-2.0	66.4	41.7	0.4	30	0
Chronos-Bolt	63.6	40.4	0.5	0	0
TFT	53.9	35.3	1889.7	0	0
PatchTST	51.2	35.0	1517.7	0	0
Sundial-Base	49.2	37.6	8.3	0	0
DeepAR	46.4	30.9	2863.3	0	0
AutoARIMA	46.0	31.4	95.1	0	1
Stat. Ensemble	46.0	28.1	252.3	0	1
CatBoost	39.2	28.4	95.6	0	0
LightGBM	35.5	24.0	7.7	0	0
AutoETS	32.2	-9.4	6.2	0	0
AutoTheta	25.0	8.5	5.1	0	0
Seasonal Naive	15.5	0.0	0.8	0	0
Naive	11.8	-36.9	0.8	0	0
Drift	8.2	-36.0	0.8	0	0

Table 18: Marginal probabilistic forecasting performance of all models (according to the SQL metric) on the `fev-bench-mini` benchmark. The reported metrics are defined in Sections 4.1 and 7.

Model	Avg. win rate (%)	Skill score (%)	Median runtime / 100 series (s)	Leakage (%)	# failures
Chronos-2	85.5	40.4	1.2	0	0
TimesFM-2.5	73.0	32.3	5.2	5	0
TiRex	72.8	31.4	0.4	0	0
FlowState	71.0	29.9	3.2	5	0
Toto-1.0	68.8	31.0	44.1	5	0
TabPFN-TS	65.2	37.3	270.3	0	0
Moirai-2.0	62.4	30.1	0.4	30	0
Chronos-Bolt	59.4	28.4	0.5	0	0
Sundial-Base	58.5	29.8	8.3	0	0
CatBoost	56.2	30.5	95.6	0	0
LightGBM	50.7	26.2	7.7	0	0
TFT	50.0	25.4	1889.7	0	0
Stat. Ensemble	48.7	22.0	252.3	0	1
PatchTST	43.0	23.9	1517.7	0	0
AutoARIMA	42.5	21.3	95.1	0	1
DeepAR	37.8	20.2	2863.3	0	0
AutoTheta	32.8	14.8	5.1	0	0
AutoETS	32.0	2.3	6.2	0	0
Seasonal Naive	14.0	0.0	0.8	0	0
Naive	13.2	-17.4	0.8	0	0
Drift	12.5	-17.2	0.8	0	0

Table 19: Marginal point forecasting performance of all models (according to the MASE metric) on the `fev-bench-mini` benchmark. The reported metrics are defined in Sections 4.1 and 7.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Main contributions are clearly described in the abstract and in the introduction section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Sec. 8.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: See Appendix C for the proofs of propositions regarding equivalence of various ranking methods.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The accompanying software package `fev` as well as the code to reproduce all experimental results is available under `https://github.com/autogluon/fev`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [Yes]

Justification: Code & evaluation results are available at `https://github.com/autogluon/fev`, and the data is available at `https://huggingface.co/datasets/autogluon/fev_datasets`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Model wrappers include the exact configuration used to run all experiments reported in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We discuss the framework for testing statistical significance based on paired bootstrap in Sec. 6.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are described in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We made sure that our research aligns with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [No]

    Justification: This paper presents a benchmarking framework for time series forecasting that is application-agnostic. After careful deliberation, we could not identify specific negative societal impacts, as our work advances general-purpose evaluation methodology without targeting any particular application domain that could be problematic.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our benchmark consists of publicly available time series datasets (e.g., retail sales, energy, economics) that do not contain personal or sensitive information. We do not release pretrained models. The datasets were reviewed to confirm they pose no risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets used in our benchmark are properly credited with citations to the original papers. We provide links to the original dataset licenses and terms of use at `https://huggingface.co/datasets/autogluon/fev_datasets`. All pretrained models and software libraries are cited with version numbers.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Documentation for the software package is provided at `https://autogluon.github.io/fev`. Documentation for the datasets is provided at the respective Hugging Face page.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [N/A]

Justification: This paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of our methodology. Our work focuses on time series forecasting models and evaluation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.