A More Realistic Evaluation of Cross-Frequency Transfer Learning and Foundation Forecasting Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Cross-frequency transfer learning (CFTL) has emerged as a popular framework for curating large-scale time series datasets to pre-train foundation forecasting models (FFMs). Although CFTL has shown promise, current benchmarking practices fall short of accurately assessing its performance. This shortcoming stems from many factors: an over-reliance on small-scale evaluation datasets; inadequate treatment of sample size when computing summary statistics; reporting of suboptimal statistical models; and failing to account for non-negligible risks of overlap between pre-training and test datasets. To address these limitations, we introduce a unified reimplementation of widely-adopted neural forecasting networks, adapting them for the CFTL setup; we pre-train only on proprietary and synthetic data, being careful to prevent test leakage; and we evaluate on 15 large, diverse public forecast competition datasets. Our empirical analysis reveals that statistical models' accuracy is frequently underreported. Notably, we confirm that statistical models and their ensembles consistently outperform existing FFMs by more than 8.2% in sCRPS, and by more than 20% MASE, across datasets. However, we also find that synthetic dataset pre-training does improve the accuracy of a FFM by 7% percent.

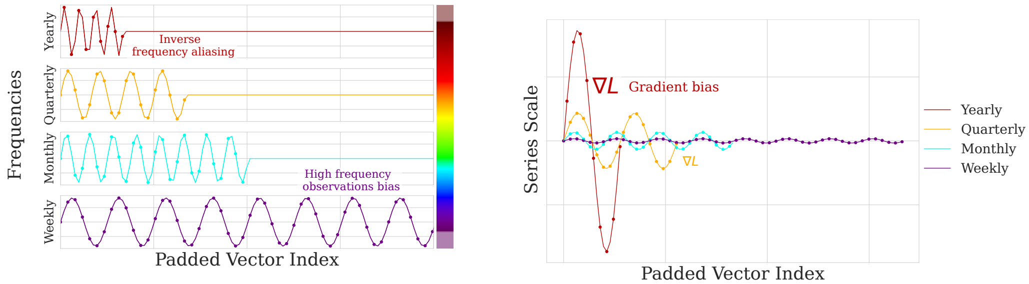
1 Introduction

Access to billions of temporal observations offers exciting opportunities for training foundation forecasting models (FFMs); and yet significant challenges remain. For example, the method known as cross-frequency transfer learning (CFTL) combines series of measurements at different frequencies to train global models [26, 43]; and, as such, it is an intuitive approach to increase time series dataset sizes. As shown in Figure 1, a key challenge in CFTL is the imbalance of observations across series: high-frequency series vastly outnumber lower-frequency ones, causing the model to become saturated and dominated by abundant high-frequency data. Similarly, differences in scale across series bias gradient updates toward larger-scaled series, preventing the model from learning a common representation that performs well across all scales.

Recent work has suggested that zero-shot CFTL can significantly outperform both traditional statistical models as well as full-shot neural forecast models trained on frequency-specific data. `TabPFN` [12], `TimesFM` [5], `Chronos` [3], `Moirai` [43] researchers report improvements of over 35% in probabilistic forecasting accuracy compared to traditional approaches like `ARIMA` [17] and statistical ensembles, and more than 15% relative to smaller deep learning architectures such as `NBEATS` [33]. However, practical adoption of FFMs as out-of-the-box replacements for statistical or frequency-specific neural forecast models remains slow, and forecasting practitioners have questioned the validity of these improvement claims and the experimental conditions under which they were obtained [28].

In this paper we argue that the appropriate criterion for assessing the success of CFTL FFMs is their ability to outperform well-established, frequency-specialized statistical models in zero-shot settings that closely resemble the conditions under which practitioners operate. And explore the question: *are current celebrations of CFTL's superiority over statistical methods premature?*

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.



(a) Unbalanced Observations (b) Heterogeneous Scales

Figure 1: Naively padding and combining series of different frequencies to train global models leads to two challenges: (a) the unbalanced observations of series of different frequencies, saturate learning signals and induce inverse frequency aliasing effects; and (b) heterogeneous time series scales, that bias gradient optimization. These unresolved challenges still prevent FFMs to replace statistical models specialized on each frequency.

Our key contributions include the following.

(i) **Unified CFTL Framework.** We re-implement a collection of well-established neural forecasting models and adapt them to share optimization, forecast outputs and evaluation pipeline. Our framework enables controlled comparisons by standardizing the pre-training data, model estimation strategy, model outputs, and hyperparameter tuning budget.

(ii) **Careful Pre-Train Dataset Curation.** To prevent any test data leakage in our transfer learning task, we pre-train exclusively on proprietary and synthetic datasets, and evaluate on 15 large-scale forecasting competition datasets. Our pre-train corpus comprises over 1.58 billion time series, spanning frequencies from daily to yearly. We further demonstrate that, even with extensive proprietary data, the inclusion of simple synthetic datasets improves CFTL's sCRPS accuracy by 7%. and MASE by 20%.

(iii) **Fair Comparison of CFTL and statistical models.** We benchmark our FFMs against automatic statistical models [15], and ensure their specialization in each series, by properly defining its hyperparameter search space based on their frequency. Furthermore, rather than relying solely on aggregate metrics - which can bias the evaluation toward smaller datasets - we report disaggregated results and use weighted averages to provide a more balanced and representative assessment across datasets. We release the evaluation of our statistical models at `https://anonymous.4open.science/r/neurips_baselines-4BC5`.

The paper is structured as follows: Section 2 introduces the CFTL methodology and reviews relevant literature; Section 3 presents our main experiments and summarizes our main empirical findings; and Section 4 concludes.

2 Methodology

We consider the univariate forecasting task. Let's start by introducing its mathematical notation. Let the forecast creation dates be $[t] = [1, ..., T]$ and the forecast horizon be denoted by $[h] = [1, 2, ..., H]$. Given a time series target variable $\mathbf{y} = \mathbf{Y}_{[t][h]}$ and target history $\mathbf{y}_{[:t]}$, the forecasting task estimates the following conditional probability:

$$\mathbb{P}\left(\mathbf{Y}_{[t][h]} \mid \boldsymbol{\theta},\ \mathbf{y}_{[:t]}\right). \tag{1}$$

Model Estimation. Consider a source forecast dataset $D^{(S)}$, defined as the set of realization tuples $D^{(S)} = \{(\mathbf{x}, \mathbf{y}) | \ \mathbf{x} \in \mathcal{Y}_{[t-L:t]},\ \mathbf{y} \in \mathcal{Y}_{[t][h]}\}$, where the $\mathcal{Y}_{[t][h]}$ and $\mathcal{Y}_{[t-L:t]}$ are the target variable and regressor support space. We estimate each forecasting model parameters $\boldsymbol{\theta}$ by minimizing the empirical risk based on Quantile Loss (QL; [19]).

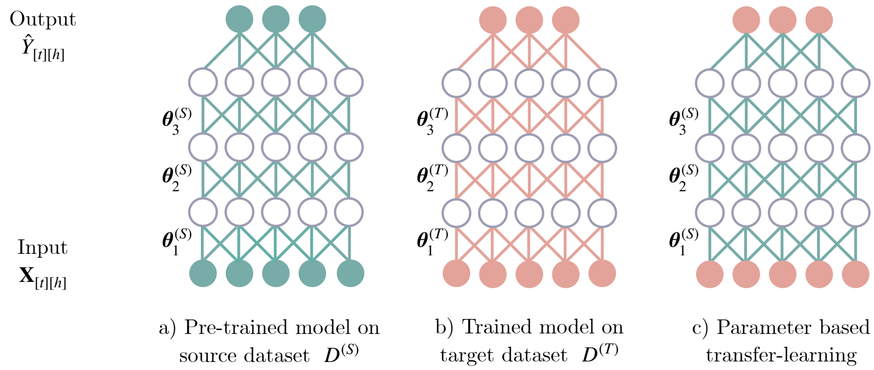


Figure 2: Three-layer fully connected network predictive function. Classic forecasting applications optimize distinct model parameters for source $D^{(S)}$ and target $D^{(T)}$ datasets, a) and b) columns. Parameter-based transfer-learning leverages source dataset knowledge by using a pre-trained model's parameters $\boldsymbol{\theta}_l^{(S)}$, to initialize another model's parameters $\boldsymbol{\theta}_l^{(T)}$ that can specialize on a target dataset.

Transfer Learning Forecasting Task. As shown in Figure 2, the zero-shot forecast task distinguishes the source data sets $D^{(S)}$ and target $D^{(T)}$, the task indirectly uses the information from the source domain by using the transfered parameters [44] as the forecasting function from Equation (1). Literature review for forecasting transfer learning is available in Appendix A.

3 Experiments

Pre-train Datasets. To pretrain our models, we use a diverse collection of 1.58 billion large online retail time series spanning daily, weekly, monthly, quarterly, and yearly frequencies. These datasets include demand data from cashierless convenience stores, grocery delivery services, and physical grocery stores. We augment the large-scale online retail demand data with a synthetic dataset composed using a combination of Fourier harmonic signals to mimic seasonalities, polynomial trends, Gaussian processes that we depict in Figure 3. Dataset details in Appendix B.

Evaluation Datasets. We consider 15 large scale forecast datasets comprising over 100,000 time series, curated from major forecasting competitions: `M1` [22], `M3` [23], `M4` [25], and `Tourism` [4]. These datasets, represent a broad range of domains and temporal frequencies. To ensure comparability with recent neural forecasting literature, we adopt the data handling and pre-processing practices of `Chronos` [2, 3] and `NBEATS` [33]. Importantly, we use the datasets solely for evaluation purposes – excluding them from model optimization – to assess their true zero-shot forecasting capabilities of our models and avoid any potential test leakage.

Forecasting Baselines. We compare FFMs with two statistical models: `AutoARIMA` [9, 17], and the Simple Combination of Univariate Models (`SiCoUM`, [34]), using the `StatsForecast` library [8, 15]. Details of the implementation can be found in Appendix E. In addition, we consider the following neural forecasting baselines `NBEATS` [33, 29] , `MQCNN` [41, 31], `PatchTST` [27], `ChronosBolt` [3], and `Moirai` [43]. Details on the unified CFTL framework implementation and hyperparameters are available in Appendix D.

Although we are unable to control hyperparameters or ensure the zero-shot regime (as external FFMs used M-competitions to train), we still evaluate external FFMs from the original `TabPFN` [12], `Moirai` [43], `TimesFM` [5], and `Chronos` [3] publications. Regarding `Moirai`, reasonable accuracy requires manual selection of patch sizes and context lengths, as the automatic heuristic frequently leads to catastrophic results. For longer daily/weekly series, the memory footprint scaled unfavorably with sequence length, causing out-of-memory failures even with batch size 1, which prevented us from evaluating daily and weekly settings with constant contexts.

Table 1: Empirical evaluation of probabilistic forecasts. Mean *scaled continuous ranked probability score* (sCRPS) averaged over 5 runs. The best united CFTL framework result is highlighted (lower measurements are preferred). The methods without standard deviation have deterministic solutions.

*`Chronos-S` stands for a pretrained `Chronos-Bolt-Small`. Zero-shot predictions correspond to the original Hugging face model published by Fatir et al [3]. *`Chronos` is trained in our unified CFTL framework, we are able to replicate or improve `Chronos-S` accuracy on the majority of datasets.
**Neither `TimesFM` nor `Chronos-S` are zero-shot forecasting models as they are trained on the `M4` dataset [3, 5].

		StatsForecast		Unified CFTL framework					External FFMs			
	Freq	`ARIMA`	`SiCoUM`	Best	`NBEATS`	`MQCNN`	`PatchTST`	`Chronos`*	`Moirai-S`	`TabPFN`	`Chronos-S`*	`TimesFM`**
`M1`	M	0.154 (-)	0.168 (-)	**0.152** (-)	0.152 (0.014)	0.155 (0.001)	0.156 (0.003)	0.156 (0.008)	0.135 (-)	0.168 (0.003)	0.173 (-)	0.130 (-)
	Q	0.088 (-)	0.084 (-)	**0.083** (-)	0.087 (0.015)	0.083 (0.001)	0.107 (0.007)	0.133 (0.024)	0.077 (-)	0.095 (0.0114)	0.084 (-)	0.113 (-)
	Y	0.133 (-)	**0.129** (-)	0.134 (-)	0.151 (0.016)	0.182 (0.022)	0.137 (0.011)	0.163 (0.023)	0.210 (-)	0.143 (0.012)	0.119 (-)	0.145 (-)
`M3`	O	0.034 (-)	**0.034** (-)	0.045 (-)	0.052 (0.021)	0.045 (0.008)	0.073 (0.010)	0.077 (0.03)	0.035 (-)	0.038 (0.008)	0.036 (-)	0.040 (-)
	M	0.098 (-)	**0.095** (-)	0.104 (-)	0.111 (0.0010)	0.117 (0.008)	0.105 (0.002)	0.104 (0.004)	0.093 (-)	0.107 (0.001)	0.113 (-)	0.089 (-)
	Q	0.077 (-)	**0.073** (-)	0.080 (-)	0.083 (0.016)	0.080 (0.009)	0.103 (0.006)	0.121 (0.025)	0.077 (-)	0.077 (0.005)	0.074 (-)	0.075 (-)
	Y	0.156 (-)	0.144 (-)	**0.127** (-)	0.127 (0.012)	0.167 (0.017)	0.129 (0.008)	0.156 (0.020)	0.135 (-)	0.132 (0.007)	0.114 (-)	0.144 (-)
`M4`	D	0.024 (-)	0.024 (-)	**0.023** (-)	0.077 (0.003)	0.023 (0.001)	0.021 (0.001)	0.019 (0.001)	0.033 (-)	0.023 (0.001)	0.028 (-)	0.021 (-)
	W	**0.046** (-)	0.049 (-)	0.047 (-)	0.067 (0.002)	0.047 (0.005)	0.050 (0.002)	0.050 (0.001)	0.071 (-)	0.046 (0.001)	0.053 (-)	0.042 (-)
	M	0.096 (-)	**0.096** (-)	0.101 (-)	0.105 (0.001)	0.108 (0.004)	0.095 (0.002)	0.097 (0.003)	0.117 (-)	0.101 (0.001)	0.108 (-)	0.066 (-)
	Q	0.079 (-)	**0.078** (-)	0.085 (-)	0.090 (0.001)	0.085 (0.005)	0.092 (0.005)	0.091 (0.002)	0.151 (-)	0.084 (0.002)	0.080 (-)	0.062 (-)
	Y	0.125 (-)	**0.115** (-)	0.133 (-)	0.133 (0.010)	0.159 (0.017)	0.121 (0.010)	0.144 (0.019)	0.187 (-)	0.121 (0.008)	0.106 (-)	0.091 (-)
	M	0.0910 (-)	**0.082** (-)	0.122 (-)	0.211 (0.007)	0.122 (0.009)	0.201 (0.005)	0.194 (0.010)	0.275 (-)	0.193 (0.004)	0.155 (-)	0.085 (-)
	Q	0.099 (-)	**0.075** (-)	0.116 (-)	0.140 (0.007)	0.116 (0.012)	0.141 (0.006)	0.141 (0.013)	0.251 (-)	0.162 (0.0034)	0.148 (-)	0.070 (-)
	Y	0.128 (-)	0.1450 (-)	**0.116** (-)	0.116 (0.011)	0.157 (0.002)	0.119 (0.011)	0.156 (0.030)	0.275 (-)	0.141 (0.000)	0.103 (-)	0.167 (-)

Probabilistic Forecast Results. We evaluate the accuracy of the forecasts using *scaled Continuous Ranked Probability Score* (sCRPS, [10]), defined as follows.

$$\text{sCRPS}\left(\mathbf{y},\ \hat{\mathbf{Y}}\right) = \frac{\sum_{i,t,h} \text{CRPS}(y_{i,t,h}, \hat{Y}_{i,t,h})}{\sum_{i,t,h} |y_{i,t,h}|}. \tag{2}$$

Summary of Results. Table 1 shows that `ARIMA` and `SiCoUM` consistently outperform FFMs, achieving the lowest errors across all frequencies for the 11 of 15 datasets. Neural architectures occasionally match or surpass the baselines, but they never achieve the best score for an entire competition. We complement Table 1 with point forecast evaluations using the *mean average scaled error* (MASE), reported in Appendix C.

Overall, confirming observations from the forecasting community [28], and in contrast to recent claims of major advances over statistical models [43, 3, 14, 5], our results show that `ARIMA` and `SiCoUM` outperform CFTL-FFMs in both probabilistic and point forecasting tasks. Excluding `TimesFM` (non-zero-shot), the statistical models and best FFMs performance differs by 8.2% in weighted sCRPS and by 20% in weighted MASE.

Although this study already covers 15 large-scale datasets, we view it as a step toward a more representative assessment. We plan to extend our evaluation with the missing datasets from the GIFT-eval collection [1], as well as new FFMs.

4 Discussion and Conclusion

We have conducted a comprehensive evaluation of CFTL. Overall, our results serve as a surprising reality check for current claims regarding FFMs. However, they also point to promising directions for improvement. As Appendix F shows, augmenting the pretraining datasets with synthetic time series improves `NBEATS`'s sCRPS performance by 7%. Similar gains are observed for `MQCNN`, `PatchTST`, and `Chronos`. Synthetic data generation is a line of research [21] that will likely be able to bridge the gap between statistical models and FFMs in their CFTL zero-shot regime.

References

[1] Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation, 2024.

[2] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Tarkmen, and Yuyang Wang. GluonTS: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

[3] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024.

[4] George Athanasopoulos, Rob J. Hyndman, Haiyan Song, and Doris C. Wu. The tourism forecasting competition. *International Journal of Forecasting*, 27(3):822–844, 2011. Special Section 1: Forecasting with Artificial Neural Networks and Computational Intelligence. Special Section 2: Tourism Forecasting.

[5] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024.

[6] Jose A. Fiorucci, Tiago R. Pellegrini, Francisco Louzada, Fotios Petropoulos, and Anne B. Koehler. Models for optimising the theta method and their relationship to state space models. *International Journal of Forecasting*, 32(4):1151–1161, 2016.

[7] Azul Garza and Max Mergenthaler-Canseco. TimeGPT 1. *arXiv preprint arXiv:2310.03579*, 2023.

[8] Federico Garza, Max Mergenthaler Canseco, Cristian Challú, and Kin G. Olivares. StatsForecast: Lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, US 2022, 2022.

[9] Box George, Jenkins Gwilym, and Reinsel Gregory. *Time Series Analysis: Forecasting and Control*. Prentice Hall, 1970.

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

[11] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *CoRR*, abs/2105.06643, 2021.

[12] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

[13] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *(O.N.R. Memorandum No. 52)*, 1957.

[14] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. The tabular foundation model tabPFN outperforms specialized time series forecasting models based on simple features. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

[15] Rob J Hyndman, George Athanasopoulos, Azul Garza, Cristian Challu, Max Mergenthaler, and Kin G. Olivares. *Forecasting: Principles and Practice, the Pythonic Way*. OTexts, Melbourne, Australia, 2025. available at https://otexts.com/fpppy/.

[16] Rob J. Hyndman and Baki Billah. Unmasking the theta method. *International Journal of Forecasting*, 19(2):287–290, 2003.

[17] Rob J. Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, 27(3):1–22, 2008.

[18] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679 – 688, 2006.

[19] Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

[20] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[21] Xu Liu, Taha Aksu, Juncheng Liu, Qingsong Wen, Yuxuan Liang, Caiming Xiong, Silvio Savarese, Doyen Sahoo, Junnan Li, and Chenghao Liu. Empowering time series analysis with synthetic data: A survey and outlook in the era of foundation models, 2025.

[22] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153, 1982.

[23] Spyros Makridakis and Michèle Hibon. The M3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000. The M3- Competition.

[24] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS One*, 13(3):e0194889, 2018.

[25] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. M4 Competition.

[26] Mike Van Ness, Huibin Shen, Hao Wang, Xiaoyong Jin, Danielle C. Maddix, and Karthick Gopalswamy. Cross-frequency time series meta-forecasting, 2023.

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

[28] Nixtla. Amazon Chronos is 10% less accurate and 500% slower than training classical statistical models. Technical report, Nixtla, 2024.

[29] Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 2022.

[30] Kin G. Olivares, Cristian Challú, Federico Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022.

[31] Kin G. Olivares, Nganba Meetei, Ruijun Ma, Rohan Reddy, Mengfei Cao, and Lee Dicker. Probabilistic hierarchical forecasting with deep poisson mixtures. *International Journal of Forecasting, accepted*, Preprint version available at arXiv:2110.13179, 2023.

[32] Boris N. Oreshkin, Dimitri Carpov, Nicolas Chapados, and Yoshua Bengio. Meta-learning framework with applications to zero-shot time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9242–9250, May 2021.

[33] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[34] Fotios Petropoulos and Ivan Svetunkov. A simple combination of univariate models. *International Journal of Forecasting*, 36(1):110–115, 2020. M4 Competition.

[35] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting, 2024.

[36] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[37] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, Spyros Makridakis, and Vassilios Assimakopoulos. Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, 37(3):1072–1084, 2021.

[38] Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 07 2019.

[39] Ivan Svetunkov, Nikolaos Kourentzes, and John Keith Ord. Complex exponential smoothing. *Naval Research Logistics (NRL)*, 69(8):1108–1123, 2022.

[40] M Syntetos, John Boylan, and JD Croston. On the categorization of demand patterns. *Journal of the Operational Research Society*, 56, 05 2005.

[41] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A Multi-horizon Quantile Recurrent Forecaster. In *31st Conference on Neural Information Processing Systems NIPS 2017, Time Series Workshop*, 2017.

[42] Malcolm Wolff, Kin G. Olivares, Boris Oreshkin, Sunny Ruan, Sitan Yang, Abhinav Katoch, Shankar Ramasubramanian, Youxin Zhang, Michael W. Mahoney, Dmitry Efimov, and Vincent Quenneville-Bélair. ♠ SPADE ♠: Split Peak Attention DEcomposition. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems NeurIPS 2024*, volume Time Series in the Age of Large Models Workshop, Vancouver, Canada, 2024. NeurIPS 2024.

[43] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024.

[44] Jason Yosinski, J eff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks?, 2014.

[45] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *The Association for the Advancement of Artificial Intelligence Conference 2021 (AAAI 2021).*, abs/2012.07436, 2020.

A Forecasting Transfer Learning

In this section, we summarize the large body of related work on transfer learning for time series forecasting.

A.1 Single-Frequency Transfer Learning

Recent advancements in neural forecasting have addressed earlier concerns around computational cost and predictive accuracy, enabling models to consistently outperform traditional statistical approaches [24]. A key driver of this progress is the adoption of cross-learning strategies [37], where global models are trained on large collections of related time series to extract shared patterns. The cross-learning paradigm underpinned the success of top-performing models in competitions like M4 and M5 [38, 33], as well as industry models such as DeepAR, MQCNN, TFT and SPADE [36, 41, 20, 42].

Transfer learning offers two key practical advantages. First, it enables accurate forecasting in scenarios with limited data. Second, it streamlines forecasting workflows by reducing the need for extensive model design and hyperparameter tuning, allowing practitioners to obtain strong performance with minimal customization. In this sense, transfer learning extends forecasting research agenda initiated by the automation of the Box-Jenkins methodology, which led to models such as AutoARIMA [17, 15].

The early approaches to transfer learning in time series forecasting focused on one global model per frequency, where success was measured by the model's ability to outperform traditional statistical baselines—such as ARIMA, ETS, and Theta—in zero-shot settings [17, 13, 6, 15]. In deep learning forecasting literature, this line of research was pioneered by the introduction of meta-learning approach and zero-shot experiments with NBEATS [32], which laid the groundwork for transfer learning in forecasting. Since then, a series of pre-trained models have emerged, including TimeGPT [7], TimesFM [5], LagLlama [35], and Chronos [3].

A.2 Cross-Frequency Transfer Learning

The first attempt to relax the same-frequency constraint in transfer learning was conducted by Van Ness et al. [26], testing the generalization capabilities of neural forecasting models when the source and target datasets differ in frequency. However, their primary results only compared their proposed meta-learning approach, Cross-Frequency Adapter (CFA), and other neural forecasting models such as LSTM and NBEATS. Their evaluation left unanswered the critical question of whether CFTL outperforms traditional statistical baselines.

Woo et al. [43], introduced Moirai, a Universal Time Series Forecasting model capable of cross-frequency transferability. By pretraining on their LOTSA dataset, Moirai claims that CFTL improved upon fully trained neural forecasting models and statistical baselines. While the paper's primary focus is on long-horizon forecasting tasks, they report aggregated results from the Monash Time Series Forecasting Benchmark [11], using the normalized Mean Absolute Error (nMAE) as the evaluation metric. In these evaluations, Moirai claimed to achieve relative improvements over Theta, ARIMA, and ETS, by an average of 38%, 36%, and 35%, and 15% upon fully trained NBEATS. A revision of Moirai's Table 20 on disaggregated evaluation on the Monash repository revealed suspiciously volatile measurements where they improve performance by 94% upon ETS on M4-hourly, while degrade performance by 77.24% on Tourism-Quarterly. This raises questions on the execution of their statistical baselines.

In a parallel line of work, Fatir et al. introduced Chronos, a model also designed to perform CFTL. In their experiments, they evaluate Chronos's zero-shot accuracy across 27 datasets, including the M-forecasting competitions, Tourism and Dominick datasets, as well as long-horizon datasets [45]. With sCRPS measures, Chronos asserts improved average performance upon Theta, ARIMA, and ETS by 47%, 35%, and 47%. A potential issue with the statement of their performance gains lies in the uniformly averaged performance calculation across datasets; such a reporting is convenient and common, but it disproportionately skews the measurements towards the smaller datasets like long-horizon [45].

B Dataset Details

In this section, we provide a summary of the data we used in our evaluation.

B.1 Pre-Training Datasets

Here, we describe the datasets we used in our pre-training. See Table 2 and Figure 3 for a summary.

Real-world data. The primary source of data for our pre-training consisted of several real-world datasets, which we summarize here.

Table 2: Summary of forecasting datasets used for pre-training.

	Frequency	Horizon	Series	Min Length	Max-Length
`Dataset A`	Daily	24	65K	1	1857
	Daily	24	28K	1	1857
`Dataset B`	Weekly	24	4MM	1	262
`Dataset C`	Daily	24	900K	1	1857
`Dataset D`	Daily	24	350K	1	1857
	Daily	24	320K	1	1857
	Daily	24	1.5MM	1	1857
`Dataset F`	Daily	24	290MM	1	1834
	Weekly	24	290MM	1	262
	Monthly	24	290MM	1	58
`Dataset E`	Weekly	24	700MM	1	314

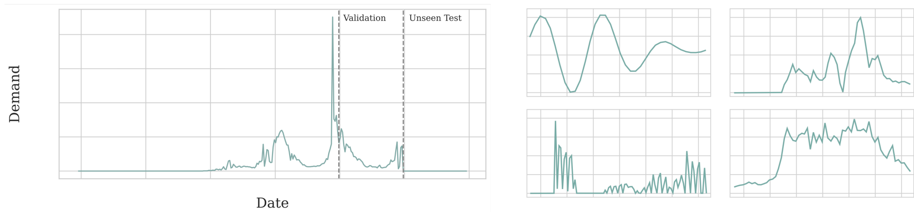
`Dataset A` comes from a chain of convenience stores operating in multiple countries. The dataset contains demand data for various consumer products including food items.

`Dataset D` represents daily demand from a grocery delivery service operating in multiple regions globally. The service offers various food and household products to subscribers.

`Dataset C` originates from a hybrid retail format that combines multiple fulfillment methods. It includes daily demand data from stores in North America, supporting both in-store shopping and delivery options.

`Dataset B` contains weekly demand data from a third-party fulfillment service operating across six developed countries. The service handles all aspects of product storage and delivery for external sellers.

`Dataset F` comprises national-level demand data from a major retail platform, including information from multiple countries around the world. `Dataset E` is a more granular version of `Dataset F`'s data for one country, broken down by postal code prefixes. It shows more irregular demand patterns than the national-level data.



(a) Sample from Pretraining Dataset (b) Synthetic Dataset

Figure 3: For our CFTL task, we use two datasets: (a) a set of real-world datasets composed of large-scale online retail demand; and (b) a set of synthetic dataset composed of Gaussian processes, Fourier harmonic signals, and polynomial trends.

Synthetic Datasets. We also used carefully-constructed synthetic data for pre-training.

Dataset G was artificially generated to supplement the training data, incorporating various time series patterns. These include basic constants, sinusoidal and cosinusoidal seasonalities, linear trends, polynomial trends, frequency drift curves, gaussian waves, exponential trend, and logistic growth curves across different time granularities as seen in Figure 3(b). The patterns were combined and modified with random noise to create realistic variations. The basic component signal equations are provided below:

$$\mathbf{y}_t = k \qquad \mathbf{y}_t = \sin(\pi * a * t + b) \qquad \mathbf{y}_t = \cos(\pi * a * t + b)$$

$$\mathbf{y}_t = a * t + b \qquad \mathbf{y}_t = \sin(\frac{\pi * a}{t} + b) \qquad \mathbf{y}_t = a * \exp\left(-\frac{(t-b)}{c}^2\right)$$

$$\mathbf{y}_t = a * \exp\left(b * t\right) \qquad \mathbf{y}_t = a * t^2 + b * t + c \qquad \mathbf{y}_t = \frac{a}{1 + \exp\left(-b * (t - c)\right)}$$

B.2 Evaluation Datasets

Here, we describe the datasets we used in our evaluation. See Table 3 for a summary.

Table 3: Summary of forecasting datasets we used in our evaluation.

	Frequency	Seasonality	Horizon	Series	Min Length	Max Length	% Erratic
M1	Monthly	12	18	617	48	150	0
	Quarterly	4	8	203	18	114	0
	Yearly	1	6	181	15	58	0
M3	Other	4	8	174	71	104	0
	Monthly	12	18	1428	66	144	2
	Quarterly	4	8	756	24	72	1
	Yearly	1	6	645	20	47	10
M4	Hourly	24	48	414	748	1008	17
	Daily	1	14	4,227	107	9933	2
	Weekly	1	13	359	93	2610	16
	Monthly	12	18	48,000	60	2812	6
	Quarterly	4	8	24,000	24	874	11
	Yearly	1	6	23,000	19	841	18
Tourism	Monthly	12	24	366	91	333	51
	Quarterly	4	8	427	30	130	39
	Yearly	1	4	518	11	47	23

M1 Dataset Details. The early `M1` competition [22], organized by Makridakis et al., focused on 1,001 time series drawn from demography, industry, and economics, with lengths ranging from 9 to 132 observations and varying in frequency (monthly, quarterly, and yearly). A key empirical finding of this competition was that simple forecasting methods, such as ETS [13], often outperformed more complex approaches. These results had a lasting impact on the field, initiating a research legacy that emphasized accurate forecasting, model automation, and caution against overfitting. The competition also marked a conceptual shift, helping to distinguish time-series forecasting from traditional time series analysis.

M3 Dataset Details. The `M3` competition [23], held two decades after the `M1` competition, featured a dataset of 3,003 time series spanning business, demography, finance and economics. These series ranged from 14 to 126 observations and included monthly, quarterly, and yearly frequencies. All series had positive values, with only a small proportion displaying erratic behavior and none exhibiting intermittency [40]. The `M3` competition reinforced the trend of simple forecasting methods outperforming more complex alternatives, with the `Theta` method [16] emerging as the best performing approach.

M4 Dataset Details. The `M4` competition marked a substantial increase in both the size and diversity of the M competition datasets, comprising 100,000 time series across six frequencies: hourly, daily, weekly, monthly, quarterly, and annual. These series covered a wide range of domains, including demography, finance, industry, and both micro- and macroeconomic indicators. The competition also introduced the evaluation of prediction intervals in addition to point forecasts, broadening the assessment criteria. `M4`'s proportion of non-smooth or erratic time series increased to 18 percent [40]. For the first time, a neural forecasting model - ESRNN[38] - outperformed traditional methods. The competition also helped popularize cross-learning [37] in global models.

Tourism Dataset Details. The Tourism dataset [4] was designed to evaluate forecasting methods applied to tourism demand data across multiple temporal frequencies. It comprises 1,311 time series at monthly, quarterly, and yearly frequencies. This competition introduced the Mean Absolute Scaled Error (MASE) as an alternative metric to evaluate scaled point forecasts, alongside the evaluation of forecast intervals. Notably, 36% of the series were classified as erratic or intermittent. Due to this high proportion of irregular data, the Naïve1 method proved particularly difficult to outperform at the yearly frequency.

Table 4: Empirical evaluation of point forecasts. Mean *absolute scaled error* (MASE) averaged over 5 runs. The best united CFTL framework result is highlighted (lower measurements are preferred). The methods without standard deviation have deterministic solutions.

*`Chronos-S` *stands for a pretrained `Chronos-Bolt-Small`. Zero-shot predictions correspond to the original Hugging face model published by Fatir et al [3].

*`Chronos` was, trained in our unified CFTL framework, with the exception of `M1` we are able to replicate or improve accuracy `Chronos-S` accuracy.

**Neither `TimesFM` nor `Chronos-S` are zero-shot forecasting models as they are trained on the `M4` dataset [3, 5].

		StatsForecast		NF (unified CFTL framework)					NF (external train)			
	Freq	`ARIMA`	`SiCoUM`	Best	`NBEATS`	`MQCNN`	`PatchTST`	`Chronos`*	`Moirai-S`	`TabPFN`	`Chronos-S`*	`TimesFM`**
`M1`	M	0.759 (-)	0.765 (-)	**0.715** (-)	0.896 (0.039)	0.745 (0.002)	0.715 (0.007)	1.048 (0.018)	0.659 (-)	0.838 (-)	0.834 (-)	0.655 (-)
	Q	0.889 (-)	0.801 (-)	**0.699** (-)	1.026 (0.176)	0.791 (0.007)	0.707 (0.028)	1.078 (0.125)	0.778 (-)	0.972 (-)	0.818 (-)	1.039 (-)
	Y	0.718 (-)	0.686 (-)	**0.632** (-)	0.672 (0.092)	0.977 (0.012)	0.629 (0.009)	0.993 (0.072)	1.289 (-)	0.830 (-)	0.723 (-)	0.803 (-)
`M3`	O	0.738 (-)	**0.693** (-)	0.784 (-)	1.040 (0.592)	0.968 (0.021)	0.822 (0.165)	0.784 (0.143)	0.725 (-)	0.866 (-)	0.729 (-)	0.853 (-)
	M	0.775 (-)	**0.721** (-)	0.795 (-)	0.861 (0.150)	0.888 (0.002)	0.795 (0.002)	0.860 (0.006)	0.936 (-)	0.838 (-)	0.880 (-)	0.709 (-)
	Q	0.905 (-)	**0.821** (-)	0.856 (-)	0.959 (0.252)	0.937 (0.005)	0.852 (0.032)	1.394 (0.083)	1.008 (-)	0.941 (-)	0.879 (-)	0.882 (-)
	Y	1.104 (-)	0.998 (-)	**0.736** (-)	0.887 (0.106)	1.212 (0.019)	0.743 (0.038)	1.141 (0.070)	1.045 (-)	0.957 (-)	0.841 (-)	1.023 (-)
`M4`	D	0.977 (-)	**0.962** (-)	1.041 (-)	3.007 (0.196)	1.041 (0.192)	0.847 (0.008)	0.974 (0.030)	1.323 (-)	1.055 (-)	1.087 (-)	0.965 (-)
	W	0.886 (-)	0.931 (-)	**0.861** (-)	1.281 (0.061)	0.861 (0.063)	0.890 (0.009)	1.128 (0.014)	1.378 (-)	0.903 (-)	1.004 (-)	0.814 (-)
	M	0.839 (-)	**0.811** (-)	0.864 (-)	0.898 (0.016)	0.911 (0.005)	0.800 (0.007)	0.864 (0.005)	1.102 (-)	0.895 (-)	0.948 (-)	0.605 (-)
	Q	0.874 (-)	**0.838** (-)	0.936 (-)	0.936 (0.019)	0.970 (0.072)	0.795 (0.015)	1.082 (0.114)	1.234 (-)	0.953 (-)	0.887 (-)	0.695 (-)
	Y	0.921 (-)	**0.814** (-)	0.944 (-)	0.944 (0.093)	1.043 (0.118)	0.700 (0.046)	0.988 (0.128)	1.464 (-)	0.924 (-)	0.789 (-)	0.667 (-)
	M	0.368 (-)	**0.333** (-)	0.509 (-)	0.881 (0.018)	0.509 (0.007)	0.865 (0.019)	0.842 (0.044)	1.148 (-)	0.860 (-)	0.636 (-)	0.357 (-)
	Q	0.727 (-)	**0.539** (-)	0.843 (-)	1.026 (0.059)	0.843 (0.007)	0.912 (0.032)	1.105 (0.105)	1.840 (-)	1.142 (-)	1.028 (-)	0.539 (-)
	Y	0.744 (-)	0.791 (-)	**0.577** (-)	0.672 (0.092)	0.880 (0.012)	0.588 (0.052)	0.773 (0.159)	1.558 (-)	0.842 (-)	0.562 (-)	0.924 (-)

C Point Forecast Results

In this section, we complement our evaluation of probabilistic forecasts (from the main text) with a set of point forecast results. We consider the *Mean Absolute Scaled Error* (MASE, [18]), that considers the ratio between mean absolute error of forecasts over mean absolute error of the `Naive` forecast $\tilde{y}_{i,t,h}$ (i.e., a point forecast using the last observation on the previous season), as described by

$$\mathrm{MASE}\left(\mathbf{y},\ \hat{\mathbf{y}},\ \tilde{\mathbf{y}}\right) = \frac{\sum_{i,t,h} |y_{i,t,h} - \hat{y}_{i,t,h}|}{\sum_{i,t,h} |y_{i,t,h} - \tilde{y}_{i,t,h}|}. \tag{3}$$

Table 4 reports mean point forecast performances of our statistical baselines and neural forecast models using the MASE across the five best checkpoints during the training process. The lowest value in every dataset-frequency cell again belongs to a statistical baseline method. On `M3`-Other, `SiCoUM` reaches a MASE of 0.515 against `Chronos`-P's 0.637; on `M4`-Daily, the exponential-smoothing family (`ARIMA`, `Theta`, `ETS`) MASE ranges from 0.963-0.977 while the best neural forecasts range from 3.000-3.330.

While gaps are smaller for lower frequency data, classical models still lead: `CES` reports a MASE of 0.636 on `M1`-Quarterly compared to the zero-shot `Chronos`-P network's MASE of 0.766, and `Theta` has a MASE of 0.625 on `Tourism`-Quarterly versus the 1.332 of `MQCNN`; neural forecasters never obtain the minimum MASE in any dataset or frequency, and exceed 1.0 in most rows. On the other hand, `SiCoUM`, (Ensemble of `CES`, `ETS`, `Theta`, and `ARIMA`) stay below the 1.0 threshold in all but a few yearly series.

These results mirror our results for probabilistic scores; and they confirm that, also for point forecasts, traditional statistical like `ARIMA`, and `SiCoUM` methods remain the most accurate choice on the four benchmark suites.

D Training Methodology and Hyperparameters

Table 5: `NBEATS`

Hyperparameter	Values
Single GPU SGD Batch Size*.	32 (32*8)
Initial learning rate.	0.001
Maximum Training steps S_{max}.	60,000
Learning rate decay.	0.1
Learning rate steps.	40,000; 50,000
Input size.	48
Main Activation Function.	ReLU
Number of Stacks	4
Number of Blocks within Stacks.	3
MLP layers within Blocks.	2
Coefficients hidden size.	512
Degree of Trend Polynomials (interpretable).	N/A
Number of Fourier Basis (interpretable).	N/A

Table 6: `MQCNN`

Hyperparameter	Values
Single GPU SGD Batch Size*.	32 (32*8)
Initial learning rate.	0.001
Maximum Training steps S_{max}.	400,000
Learning rate decay.	0.1
Learning rate steps.	400,000 / 2
Main Activation Function	ReLU
Temporal Convolution Kernel Size	2
Temporal Convolution Dilations.	$[1, 2, 4, 8, 16, 32]$
Historic Encoder Dimension.	30
Future Encoder Dimension (hf_1).	50
Static Encoder D.Multip. $(\alpha \times \|\sqrt{x}^{(s)}\|)$	30
H-Agnostic Decoder Dimension.	100
H-Specific Decoder Dimension.	20

Table 7: `PatchTST`

Hyperparameter	Values
Single GPU SGD Batch Size*.	32 (32*8)
Initial learning rate.	0.001
Maximum Training steps S_{max}.	100,000
Learning rate decay.	0.1
Learning rate steps.	100,000 / 5
Input Size.	128
Main Activation Function	ReLU
Patching Length.	16
Patching Stride.	8
Number of Attention Heads.	16
Encoder Hidden Size.	128
Decoder Hidden Size.	256
Apply Revin.	True
Residualized Attention.	True

Table 8: `ChronosBolt`

Hyperparameter	Values
Single GPU SGD Batch Size*.	4 (96)
Initial learning rate.	0.0005
Maximum Training steps S_{max}.	50,000
Learning rate decay.	0.1
Learning rate steps.	50,000 / 5
Input Size.	2048
Main Activation Function	ReLU
Encoder/Decoder Hidden Size.	256
Encoder Type.	`T5Stack`
Decoder Type.	`T5Stack`
Patch Size	16
Patch Stride	16
Encoder Number of Layers.	4
Decoder Number of Layers.	4
Number of Attention Heads.	4
Attention Dropout Rate.	0.1

In this section, we provide details on the training methodology, outlined in Section 3. The optimization of all models is based on the definition of training, validation, and test datasets, depicted in Figure 3. For all our pre-training datasets, we keep the 24 observations immediately following the training data as validation. Given the scale of our evaluation, we focused our hyperparameter optimization solely on the selection of training steps and learning rate, and we rely principally on the default hyperparameters implementation for each baseline. See Tables 5, 6, 7, 8. Hyperparameters not specified in these tables are set to the defaults of the original implementations in the NeuralForecast library [30], or the Chronos repository [3].

We conducted all neural network experiments using a single AWS p4d.24xlarge with 1152 GiB of RAM and 96 vCPUs. Training times mostly depend on the architecture, however we restrict the SGD training steps to 100K per architectures.

E Implementation Details of the Simple Combination of Univariate Models

In this section, we provide details on the implementation of the statistical ensemble used to generate the point and probabilistic forecasts evaluated in Table 1 and Table 4.

As discussed in Section 3, we employ the Simple Combination of Univariate Models (`SiCoUM`; [34]) framework. This ensemble method aggregates forecasts from four classical statistical models. Complex Exponential Smoothing (`CES`; [39]), Dynamic Optimized Theta (`Theta`; [6]), Automatic Autoregressive Integrated Moving Average (`ARIMA`; [17]), and Exponential Smoothing (`ETS`; [13]). For all the models, we use the implementations of the StatsForecast library [8, 15].

Each model is independently fitted to the time series, producing Gaussian-distributed forecasts. Assuming Normality and independence among model forecast distributions, we construct the ensemble by aggregating the means and variances of the individual forecasts. Let `CES`, `Theta`, `ARIMA`, and `ETS` denote the constituent models, the ensemble forecast is computed as:

$$\hat{\mu} = \frac{1}{4}\left(\hat{\mu}_{\mathrm{CES}} + \hat{\mu}_{\mathrm{Theta}} + \hat{\mu}_{\mathrm{ARIMA}} + \hat{\mu}_{\mathrm{ETS}}\right) \tag{4}$$

$$\hat{\sigma}^2 = \frac{1}{4}\left(\hat{\sigma}^2_{\mathrm{CES}} + \hat{\sigma}^2_{\mathrm{Theta}} + \hat{\sigma}^2_{\mathrm{ARIMA}} + \hat{\sigma}^2_{\mathrm{ETS}}\right) \tag{5}$$

We generate the final quantile predictions using the percent point function:

$$\begin{aligned} &\hat{y}^{(q)} = \hat{\mu} + \hat{\sigma} z^{(q)} \\ \text{with} \qquad &z^{(q)} = \inf\{y \in \mathbb{R} \,:\, q \leq \Phi(y)\} \end{aligned} \tag{6}$$

To run the statistical baselines we used a single AWS c5.18xlarge instance with 72 vCPUs and 137 GiB of RAM. To ensure the reproducibility of our experimental results, we provide the implementation of the statistical baselines at the following link: `https://anonymous.4open.science/r/neurips_baselines-4BC5`.

F Ablation Study on Synthetic Data in our Pre-training Datasets

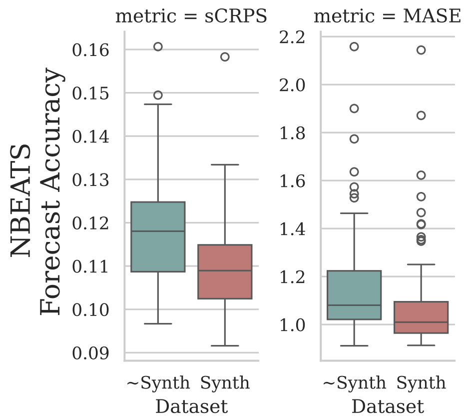


Figure 4: Pre-training datasets ablation, with and without the use of synthetic data. Shown are metrics with (red) and without (green) synthetic data for pre-training, for the NBEATS model.

Our re-implementation of well-established univariate forecasting algorithms, adapted for the CFTL task, enabled us to isolate a primary driver of accuracy improvements across architectures: dataset quality. As shown in Figure 4, our CFTL-adapted NBEATS model improved its sCRPS score from 0.116 to 0.108 - a notable 7% gain - when synthetic data was added to the pre-training set. Similar improvements were observed across other architectures. For this and other models, our results demonstrated that dataset composition, rather than architectural choices, was the primary driver of sCRPS improvements.

Importantly, even in the presence of huge pre-training datasets, of 1.58 billion series, synthetic data are still capable of improving the zero-shot performance of NBEATS, MQCNN, Chronos, and PatchTST (as shown in Table 1 and Table 4), reinforcing the central role of training data in model performance even at large scales. This suggests that better synthetic data generation methodologies will be important to the future advancements of CFTL and FFMs.

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: [NA]

2. **Limitations**

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

   Answer: [Yes]

   Justification: [NA]

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: [NA]

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: [NA]

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: [NA]

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: [NA]

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: [NA]

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: [NA]

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: [NA]

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: [NA]

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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

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: [NA]

    Justification: [NA]

13. **New Assets**

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

    Answer: [NA]

    Justification: [NA]

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: [NA]

    Justification: [NA]

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: [NA]

    Justification: [NA]