INVESTIGATING SIMPLE TARGET-COVARIATE RELATIONSHIPS FOR CHRONOS-2 AND TABPFN-TS

Gaspard Berthelier [*], **Mariia Baranova** [†], **Andrei-Tiberiu Pantea,**
Etienne Le Naour, Adrien Petralia, Tahar Nabil
EDF R&D, Palaiseau
{gaspard.berthelier,mariia.baranova,andrei.pantea}@edf.fr
{etienne.le-naour,tahar.nabil,adrien.petralia}@edf.fr

Themis Palpanas
Université Paris Cité
themis@mi.parisdescartes.fr

ABSTRACT

Time Series Foundation Models (TSFMs) have recently achieved state-of-the-art performance, often outperforming supervised models in zero-shot settings. Recent TSFM architectures, such as `Chronos-2` and `TabPFN-TS`, aim to integrate covariates. In this paper, we design controlled experiments based on simple target–covariate relationships to assess this integration capability. Our results show that `TabPFN-TS` captures these relationships more effectively than `Chronos-2`, especially for short horizons, suggesting that the strong benchmark performance of `Chronos-2` does not automatically translate into optimal modeling of simple covariate–target dependencies.

Track: Industry & Application

1 INTRODUCTION

Time Series Foundation Models (TSFMs) enable inference-only forecasting leveraging massive pretraining on large-scale real-world and synthetic datasets. Extensive evaluations in `GiftEval` (Aksu et al., 2024) show that TSFMs outperform supervised deep learning baselines on univariate forecasting tasks from various domains.

However, in real-world applications, univariate forecasting is often insufficient, and effectively leveraging dynamic covariates becomes essential, particularly in domains such as energy or retail. The first generation of TSFMs (Das et al., 2024; Ansari et al., 2024) lacked native covariate handling, with the exception of `MOIRAI` (Woo et al., 2024) which was superseded by `MOIRAI 2.0` for improved univariate performance (Liu et al., 2025). More recently, `Chronos-2` (Ansari et al., 2025) and `TabPFN-TS` (Hoo et al., 2025) have shown impressive results on covariate-based forecasting, as reflected by `fev-bench` (Shchur et al., 2025). To handle covariate integration, `Chronos-2` uses time and group attention layers to facilitate information exchange across multiple time series and is pretrained on synthetic multivariate data. In contrast, `TabPFN` (Hollmann et al., 2022) is a tabular foundation model pretrained on millions of synthetic regression tasks, which can be adapted for time series forecasting with carefully designed features (Hoo et al., 2025). Both models leverage in-context learning to capture feature–target relationships on unseen problems. Despite `Chronos-2`'s strong performance in recent bencharks (e.g. `fev-bench`), these evaluations offer limited insight into actual covariate usage.

In this paper, we investigate how effectively these two models leverage covariate information, focusing on cases where the relationship between the target and covariates follows simple patterns. Our experimental results show that `TabPFN-TS` more reliably captures simple target–covariate relationships particularly for short forecasting horizons. These results suggest that, despite `Chronos-2`'s dominance on large-scale covariate-based forecasting benchmarks, there is still significant potential to design TSFMs that more effectively leverage covariates.

[*] Inria Sophia-Antipolis, Université Côte d'Azur.
[†] Université Paris Cité.

2 EXPERIMENTAL EVALUATION

In the following experiments, we compare `Chronos-2` and `TabPFN-TS` on minimal tasks designed to isolate target and covariate dependency. We compare the two models on a range of real-world datasets and forecasting settings, as well as on synthetically generated time series.

Settings. We consider the usual time series forecasting setting, where models take as input a target's past look-back window of size L, to forecast a future horizon window of size H. Additionally, covariates can be included as inputs, throughout the full context (look-back and horizon). We build four different experiments: (i) Identity: the covariate equals the target to forecast; performed on real-world time series. (ii) Sum: the target is $Z = X + Y$, with X and Y synthetically generated time series. (iii) Aggregate: the target is an aggregate of the form $Z = \sqrt{X + Y - Z^2}$, where X, Y, Z are real-world time series. (iv) Quadratic: the target is of the form $Z = a + bX + cY^2$, with X and Y synthetically generated time series. More details on the experiment protocols are provided in Appendix A.

Results for the identity experiment. Table 1, reports the errors of each model when the target is also given as a covariate. We observe that (i) while `Chronos-2` is a stronger univariate forecaster, `TabPFN-TS` consistently outperforms it in this experiment; (ii) despite being given access to the ground truth, both models still suffer from very short context lengths.

Table 1: Normalized MSEs for `Chronos-2` and `TabPFN-TS` before and after providing the target as covariate (+id), on real-world datasets and a range of $L - H$ settings.

Dataset	Setting	Chronos-2	Chronos-2(+id)	TabPFN-TS	TabPFN-TS(+id)
ELECTRICITY	1344-336	0.2037	0.0085	0.2367	**0.0056**
	672-168	0.1788	**0.0130**	0.2149	0.0205
	168-24	0.1953	0.0477	0.2662	**0.0415**
	24-24	1.3718	1.0252	2.1728	**0.8121**
SOLAR	1344-336	0.1503	0.0456	0.2421	**0.0000**
	672-168	0.1831	0.0612	0.1925	**0.0000**
	168-24	0.1272	0.0886	0.1434	**0.0000**
	24-24	1.3238	0.8333	1.8876	**0.3565**
TRAFFIC	1344-336	0.2326	0.0276	0.2893	**0.0037**
	672-168	0.2158	0.0251	0.2550	**0.0051**
	168-24	0.1926	0.0561	0.4007	**0.0034**
	24-24	1.7487	1.5570	3.1238	**1.1547**

Results for the sum experiment. Figure 1 reports the relative performance of each model for the sum experiment, across both synthetic datasets. `TabPFN-TS` outperforms `Chronos-2` on short forecasting horizons and achieves comparable performance in several other settings. We hypothesize that `TabPFN-TS` performs better on short horizons because input and output features will be more similarly distributed, resulting in a task that more closely resembles the i.i.d problems encountered during pretraining.

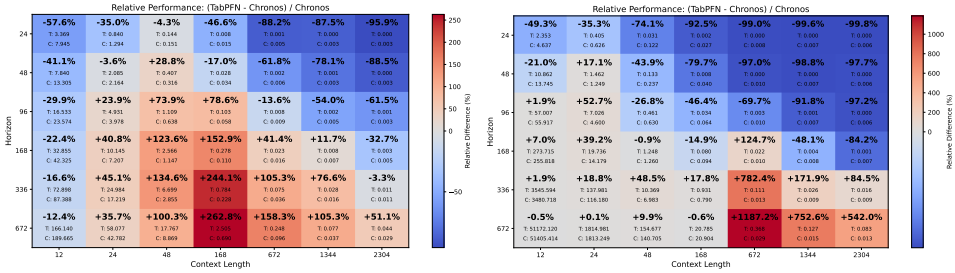


Figure 1: Heatmaps of the relative performance of `Chronos-2` (C) compared to `TabPFN-TS` (T) for the sum experiment, expressed as $100 * (T - C)/C$). Blue indicates `TabPFN-TS` outperforms `Chronos-2`, and red indicates `Chronos-2` outperforms `TabPFN-TS`. Left: Random Walk dataset, Right: KernelSynth dataset.

Results for the aggregate experiment. Table 2, reports the errors of each model before and after providing the covariates. Similarly to the identity experiment, we observe that `Chronos-2` is a stronger univariate forecaster, but `TabPFN-TS` better captures the simple covariate-target relationship. Interestingly, removing the time embeddings and using only covariate features (`TabPFN`(cov) model) does not significantly degrade the performance, and becomes the most efficient model in the short-range setting.

Table 2: Normalized MSEs for `Chronos-2` and `TabPFN-TS` before and after providing covariates (cov). `TabPFN`(+cov) corresponds to `TabPFN-TS`(+cov) without the time embeddings.

Setting	`Chronos-2`	`Chronos-2`(+cov)	`TabPFN-TS`	`TabPFN-TS`(+cov)	`TabPFN`(cov)
1344-336	0.1880	0.1940	0.2375	**0.0008**	0.0013
672-168	0.1860	0.1905	0.2106	**0.0013**	0.0020
168-24	0.1861	0.1999	0.3608	**0.0010**	0.0011
24-24	0.5580	0.5625	1.6802	0.1160	**0.0950**

Results for the quadratic experiment. Figure 2 reports the results for the quadratric experiment. Once again, `TabPFN-TS` outperforms `Chronos-2` on the majority of settings, and particularly for short forecasting horizons.

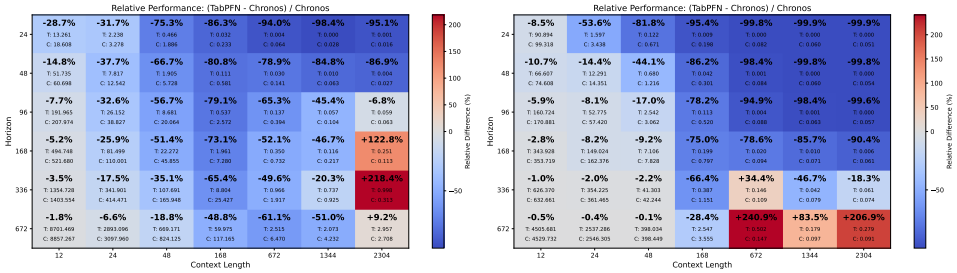


Figure 2: Heatmaps of the relative performance of `Chronos-2` (C) compared to `TabPFN-TS` (T) for the quadratic experiment, expressed as $100 * (T - C)/C$). Blue indicates `TabPFN-TS` outperforms `Chronos-2`, and red indicates `Chronos-2` outperforms `TabPFN-TS`. Left: Random Walk dataset, Right: KernelSynth dataset.

Auto-regressive experiment Additionally to the previous experiments, we ran an experiment with a target $Z = a + bX_t + cY_t^2 + \phi Z_{t-1}$. Results are included in Appendix B. By progressively increasing ϕ, we increase the importance of time-dependence over the target-covariates relationship, which enables `Chronos-2` to better perform against `TabPFN-TS`. This suggests that the relative performance of both models depends on the relative importance of the time-to-target and covariate-to-target relationships for a given task.

3 Conclusions

In this paper, we investigated how `Chronos-2` and `TabPFN-TS` leverage covariate information to improve forecasting performance, by evaluating their performance on simple target-covariates tasks. Although `Chronos-2` achieves stronger performance on large-scale benchmarks such as `fev-bench`, our experiments show that `TabPFN-TS` demonstrates superior results on our simple synthetic experiments, especially for short forecasting horizons. We hypothesize that this advantage stems from its pretraining based on expressive prior and extensive synthetic regression problems. This conclusion highlights a promising direction for future research in the development of TSFMs. In particular, a foundation architecture that combines explicit temporal dependency modeling, as in `Chronos-2`, with in-context regression capabilities inspired by `TabPFN-TS`, could represent an interesting direction for covariate-aware time series foundation models.

ACKNOWLEDGMENTS

Supported by EDF R&D. Mariia Baranova, Adrien Petralia and Themis Palpanas are also supported by EU Horizon project DataGEMS (101188416), and by $Y\Pi AI\Theta A$ & NextGenerationEU project HARSH ($Y\Pi 3TA - 0560901$) that is carried out within the framework of the National Recovery and Resilience Plan "Greece 2.0" with funding from the European Union – NextGenerationEU.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda-Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Trans. Mach. Learn. Res.*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: Mining efficiency. *Transportation Research Record*, 1748(1): 20–28, 2001.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, 2024.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2022.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: How tabpfn-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945*, 2025.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, pp. 95–104, New York, NY, USA, 2018. ACM. doi: 10.1145/3209978.3210006.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

National Renewable Energy Laboratory. Solar power data for integration studies, 2006. URL `https://www.nrel.gov/grid/solar-resource-data.html`. Accessed: 2026-02-13.

Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Yuyang Wang. fev-bench: A realistic benchmark for time series forecasting. *arXiv preprint arXiv:2509.26468*, 2025.

Artur Trindade. ElectricityLoadDiagrams20112014 Data Set. UCI Machine Learning Repository, 2015. URL `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

A EXPERIMENTAL PROTOCOL

Datasets. We consider three real-world datasets covering domains of energy, transport and climate: ELECTRICITY (Trindade, 2015), TRAFFIC (Chen et al., 2001), (Lai et al., 2018) and SOLAR (National Renewable Energy Laboratory, 2006). SOLAR and ELECTRICITY are aggregated hourly (sum) and a dozen of users with many missing values are removed from ELECTRICITY. For the sum and quadratic experiments, we generate two synthetic datasets: one with the KernelSynth method, which was used to pretrain `Chronos-2` (Ansari et al., 2025), and the latter with a random walk process. For the aggregate experiment, we manually build a CONSO dataset: $\text{Conso} = \sqrt{\text{Traffic}} + \text{Electricity} - \text{Solar}^2$ (one time series for each user). For the quadratic experiment, the target is $Z = 5 + 10X + 20Y^2$.

Setting. We consider the usual look-back window to horizon forecasting setting, with varying look-back L and horizon H window sizes:

$$f : x \in \mathbb{R}^L \mapsto \hat{y} \in \mathbb{R}^H$$

Optionally, covariates $c \in \mathbb{R}^{L+H}$ can be included as inputs.

Our metric for evaluation is the normalized MSE, which corresponds to the MSE computed on predictions and ground-truths, normalized *by instance*. That is:

$$L_f(x, y) = ||\hat{y} - \tilde{y}||_2^2, \quad \text{with} \quad \hat{y} = f(\tilde{x}),\ \tilde{x} = \frac{x - \mu_x}{\sigma_x},\ \tilde{y} = \frac{y - \mu_x}{\sigma_x}.$$

During evaluation, and for all $L - H$ settings, we apply a stride of 512 to obtain non overlapping and non periodic windows. We evaluate deterministically over those windows and for all individuals.

Embeddings of `TabPFN`. Inspired by `TabPFN-TS` (Hoo et al., 2025), we encode time indices for `TabPFN` with periodic time embeddings:

$$\phi(t \in [0, L + H]) = \left(\frac{t}{L}, \cos\left(2\pi \frac{t}{T_i}\right), \sin\left(2\pi \frac{t}{T_i}\right), \dots\right) \quad T_i \in \{\text{Chosen periods}\}$$

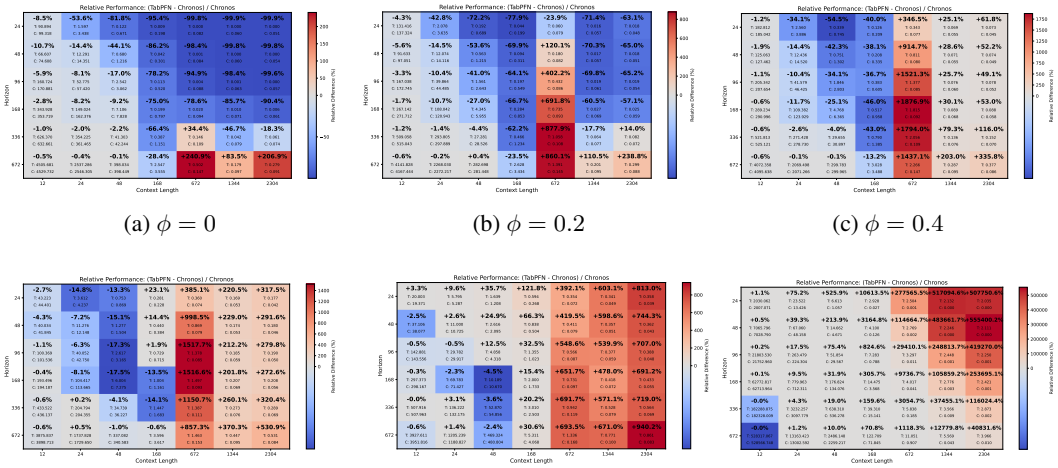
We use daily ($T = 24$) and weekly ($T = 168$) periods for our real-world datasets, since all are hourly and are expected to follow these periods. This corresponds to 5 time index features. When covariates are available as context, we encode them as features as well. When this is the case, we may remove the time embeddings. In our experiments with synthetic covariates, we observed better results without time embeddings, using only covariates used as features. Thus, `TabPFN-TS` from the synthetic experiments (results on Figure 1 and Figure 2) does not include time embeddings, and is consequently identical to the original `TabPFN`.

B EXPERIMENT WITH AUTO-REGRESSIVE TERM

In complement to the quadratic experiment (Paragraph 2), the following experiment investigates the effect of introducing temporal dependence into a nonlinear regression task.

Settings. The target variable is generated according to $Z_t = a + bX_t + cY_t^2 + \phi Z_{t-1}$, where X_t and Y_t are synthetically generated time series and a, b, c are fixed coefficients, as in the quadratic experiment (See Appendix A). The auto-regressive coefficient ϕ is taken in the range $\{0, 0.2, 0.4, 0.6, 0.8\}$, allowing a smooth transition from a purely instantaneous nonlinear relationship ($\phi = 0$) to a temporally dependent process.

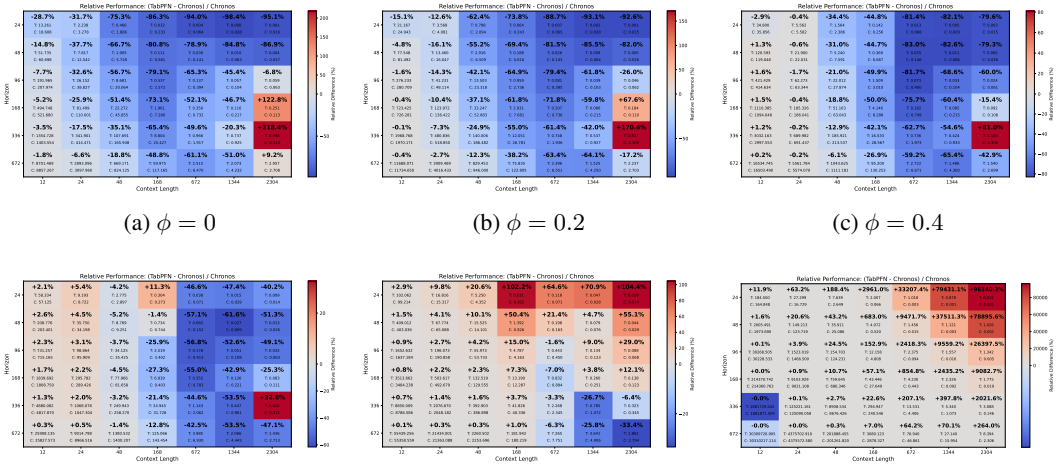
Results. Figures 3 and 4 showcase that as ϕ increases, the task transitions from a purely nonlinear tabular regression setting, where `TabPFN-TS` performs best , to a temporally dependent forecasting problem in which `Chronos-2` increasingly outperforms `TabPFN-TS` by exploiting the growing temporal structure. This suggests that the relative performance of both models depends on the relative importance of the time-to-target and covariate-to-target relationships for a given task.



(a) $\phi = 0$ (b) $\phi = 0.2$ (c) $\phi = 0.4$

(d) $\phi = 0.6$ (e) $\phi = 0.8$ (f) $\phi = 1$

Figure 3: Heatmaps showing the relative performance of `Chronos-2` (C) compared to `TabPFN-TS` (T), expressed as $100 * (T - C)/C$), across context lengths and horizons for different values of ϕ on the KernelSynth dataset. Blue regions indicate cases where `TabPFN-TS` outperforms `Chronos-2`, while red regions indicate the opposite.



(a) $\phi = 0$ (b) $\phi = 0.2$ (c) $\phi = 0.4$

(d) $\phi = 0.6$ (e) $\phi = 0.8$ (f) $\phi = 1$

Figure 4: Heatmaps showing the relative performance of `Chronos-2` (C) compared to `TabPFN-TS` (T), expressed as $100 * (T - C)/C$), across context lengths and horizons for different values of ϕ on the Random Walk dataset. Blue regions indicate cases where `TabPFN-TS` outperforms `Chronos-2`, while red regions indicate the opposite.