Forecasting with Factor-Augmented Time Series Foundation Models

Annette Jing* Phillip Jang* Davide Pettenuzzo*† Rishab Guha*
Gray Calhoun*

May 31, 2026

Abstract

Despite their strong zero-shot forecasting capabilities, Time Series Foundation Models (TSFMs) lack mechanisms for incorporating the structured domain knowledge that practitioners need for interpretability and forecast control. Dynamic Factor Models (DFMs) provide this structure, decomposing series into interpretable, adjustable factors like trend and seasonality while capturing shared dynamics across related series. However, DFMs are typically linear and require careful specification that demands domain expertise. To bridge this gap, we introduce Factor-Augmented Time Series Foundation Models (FA-TSFMs), a framework that integrates DFM-extracted factors into TSFMs as covariates, while also considering residualization as a complementary pathway. Experiments on a real-world supply chain dataset show that factor augmentation improves forecast accuracy for three of four TSFMs tested, while the fourth benefits when factors are incorporated indirectly via residualization. Through ablation studies and mechanistic analysis, we identify which factor types contribute most and how the TSFM incorporates them. We further find that factors from a small group of core series improve forecasts for series not modeled by the DFM, pointing toward a scalable system where TSFMs leverage DFM factors from core series to forecast related peripheral series without additional model training.

Keywords: Time Series Foundation Models, Factor Models, Time Series Forecasting
JEL Codes: C11, C32, C51

*Amazon
†Brandeis University

1 Introduction

Dynamic Factor Models (DFMs) are a cornerstone of multivariate time series forecasting in economics and operations, decomposing observed series into latent factors that capture shared dynamics and enable interpretable forecast adjustments [Stock and Watson, 2016, Forni et al., 2000]. However, standard DFMs are typically linear, and specifying an appropriate factor structure (number of factors, lag order, prior distributions) often requires substantial domain expertise and careful tuning [Bai and Ng, 2002, Poncela et al., 2022]. In contrast, Time Series Foundation Models (TSFMs), which are transformer-based models pre-trained on large, diverse corpora, have recently demonstrated strong forecasting capabilities on previously unseen series without domain-specific training or tuning ("zero-shot" forecasting) [Ansari et al., 2025, Hoo et al., 2025, Woo et al., 2024, Potapczynski et al., 2026]. Yet TSFMs lack mechanisms for incorporating structured domain knowledge, and their forecasts are neither interpretable nor controllable in the ways practitioners require.

These complementary strengths and limitations motivate a natural question: can the structured outputs of econometric models be integrated into TSFMs to achieve accuracy and scalability without sacrificing interpretability and controllability? We introduce Factor-Augmented Time Series Foundation Models (FA-TSFMs), a framework that integrates DFM-extracted factors into TSFMs via two complementary pathways: as covariates and through residualization. This enables TSFMs to leverage cross-series factor information and learn non-linear factor–target relationships, while preserving the interpretability and adjustment capabilities of the underlying DFM. Our experiments on real-life supply chain data demonstrate two key benefits of the FA-TSFM framework: 1) improved forecast accuracy and 2) controlled interpretability via factor levers, though some series exhibit a trade-off between these objectives. Consistent accuracy improvements are observed across three TSFMs: Chronos-2 [Ansari et al., 2025], TabPFN-TS [Hoo et al., 2025], and ApolloPFN [Potapczynski et al., 2026]. Using Chronos-2, the best-performing TSFM on our dataset, we conduct ablation studies and mechanistic analyses to understand how and when factor augmentation

improves accuracy, and investigate controllability through a factor adjustment experiment.

To our knowledge, this is the first work to actively propagate structured information extracted by econometric models into TSFMs. While recent work by Carriero et al. [2025] uses TSFMs to model residuals of Bayesian vector autoregression (BVAR), their approach relies solely on the TSFM's pre-training to capture econometric model errors. In contrast, our framework considers both residualization and explicitly passing extracted factors as covariates, the latter of which enables the TSFM to leverage cross-series information and learn complex non-linear factor-target relationships.

More broadly, while deep learning forecasters have incorporated interpretable structure or covariate inputs, they do not fill the same role as FA-TSFM. Interpretable models like N-BEATS [Oreshkin et al., 2020] and N-HITS [Challu et al., 2023] produce trend-seasonality decompositions similar to factors, but are inherently univariate and lack a structured mechanism for encoding multivariate relationships. Covariate-capable models such as DeepAR [Salinas et al., 2019] and TFT [Lim et al., 2021] could in principle receive DFM factors as inputs, but require per-dataset training, forgoing the zero-shot scalability that makes TSFMs attractive. Lastly, recent work on covariate-aware TSFM adaptation [Han et al., 2026] addresses *how* to fuse heterogeneous covariates into TSFMs; our work is complementary, addressing *what* to pass and why they are useful.

Our main contributions are summarized below:

1. We propose FA-TSFM, a modular and model-agnostic framework that integrates DFM-extracted factors into TSFMs via two complementary pathways: as covariates and through residualization.

2. We evaluate FA-TSFM with four foundation models on a real-life supply chain dataset, demonstrating consistent accuracy improvements across three of them, with the fourth benefiting from indirect factor information via residualization.

3. We demonstrate the contribution of different factor types through ablation studies and mechanistic interpretability analysis of Chronos-2 activation differences and group atten-

tion weights.

4. We show that DFM factors estimated from a small group of core series can be passed to the TSFM to forecast other series not modeled by the DFM, achieving comparable performance to when every series receives its own relevant factors. This suggests FA-TSFM can scale DFM benefits without exhaustive model tuning for every series of interest.

5. We provide evidence that residualized FA-TSFMs preserve the controllability of the underlying DFM, responding predictably to factor adjustments. Non-residualized variants can yield higher accuracy when the DFM is not well-tuned, but at the cost of reduced controllability.

The remainder of this paper is structured as follows. Section 2 introduces DFMs and TSFMs. Section 3 defines the FA-TSFM framework and experimental setup. Section 4 presents main accuracy results. Section 5 analyzes the contribution of different factor types. Section 6 examines implications for hierarchical forecasting and planning. Section 7 concludes with limitations and future directions.

2 Background

We consider the task of multivariate time series forecasting. The goal is to forecast N target series $\boldsymbol{Y}_t = [Y_{1,t}, \ldots, Y_{N,t}]^\top \in \mathbb{R}^N$ observed over timesteps $t = 1, \ldots, T$ for horizons $h = 1, \ldots, H$.

Dynamic factor model (DFM) A general DFM models $\boldsymbol{Y}_t$ through its observation and transition equations. Following Stock and Watson [2016], we express these in static form, where the observation equation loads only on contemporaneous factors (the equivalent dynamic form uses lagged factors directly):

$$\boldsymbol{Y}_t = \boldsymbol{\Lambda} \boldsymbol{F}_t + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{F}_t = \textstyle\sum_{q=1}^{Q} \boldsymbol{\Psi}_q \boldsymbol{F}_{t-q} + \boldsymbol{\eta}_t. \tag{1}$$

$\boldsymbol{\Lambda} = [\lambda_{i,r}] \in \mathbb{R}^{N \times R}$ is the factor loading matrix, $\boldsymbol{F}_t = [F_{r,t}]_{r=1}^{R} \in \mathbb{R}^R$ the (static) factors, $\{\boldsymbol{\Psi}_q\}_{q=1}^{Q}$ the transition matrices, $\boldsymbol{\varepsilon}_t \in \mathbb{R}^N$ the observation errors, and $\boldsymbol{\eta}_t \in \mathbb{R}^R$ the transition errors. The factors $\boldsymbol{F}_t$ must be estimated from the training data, typically via principal component analysis (PCA) or Bayesian estimation in a state-space framework. The factor loadings $\boldsymbol{\Lambda}$ are either estimated alongside the factors or pre-specified based on known structure. Once estimated, DFM forecasts are obtained as $\hat{\boldsymbol{Y}}_t^{\mathrm{DFM}} = \boldsymbol{\Lambda}\boldsymbol{F}_t$. For an individual series i, the DFM forecasts become

$$\hat{Y}_{i,t}^{\mathrm{DFM}} = \boldsymbol{\Lambda}_i \boldsymbol{F}_t = \textstyle\sum_{r=1}^{R} \lambda_{i,r} F_{r,t}, \tag{2}$$

where $\boldsymbol{\Lambda}_i \in \mathbb{R}^{1 \times R}$ denotes the i-th row of the loading matrix $\boldsymbol{\Lambda}$. In general, $\boldsymbol{\Lambda}_i$ may be dense (most factors affect series i) or sparse (only a subset of factors contribute), depending on the model structure. In the latter case, it is convenient to write $\hat{Y}_{i,t}^{\mathrm{DFM}} = \boldsymbol{\beta}_i \boldsymbol{F}_{i,t}$, with $\boldsymbol{F}_{i,t} = \{F_{r,t} : \lambda_{i,r} \neq 0\} \in \mathbb{R}^{R_i}$ being the subset of $\boldsymbol{F}_t$ that is relevant for series i and $\boldsymbol{\beta}_i \in \mathbb{R}^{1 \times R_i}$ being the corresponding loadings. The notation $\boldsymbol{F}_{i,t}$ proves useful in defining the DFM employed in our experiments (Section 3.2).

Time series foundation model (TSFM) TSFMs represent a paradigm shift in forecasting, leveraging transformer architectures trained on diverse datasets to achieve zero-shot forecasting capabilities across different domains. These models, including Chronos [Ansari et al., 2025, 2024], TabPFN-TS [Hoo et al., 2025], and TimesFM [Das et al., 2024], demonstrate strong performance on previously unseen time series without domain-specific fine-tuning.

TSFM	Max context length W	Max horizon H	Past-only covariates
Chronos-2 [Ansari et al., 2025]	8192	1024	✓
TabPFN-TS [Hoo et al., 2025]	4096	None	×
Moirai 1.0 [Woo et al., 2024]	1120*	128*	✓
ApolloPFN [Potapczynski et al., 2026]	1095	128	×

*Implied by joint token budget constraint; see Appendix C.

Table 1: TSFM constraints.

A key requirement for integrating DFM factors into TSFMs is support for past and future (forward-looking) covariates, as factors are known quantities that can be projected into the forecast horizon. We evaluate four state-of-the-art TSFMs that meet this requirement: Chronos-2 [Ansari et al., 2025], TabPFN-TS [Hoo et al., 2025], ApolloPFN [Potapczynski et al., 2026], and Moirai 1.0 [Woo et al., 2024]. Table 1 summarizes their key constraints. The maximum context length determines how much history each TSFM can observe, as only the most recent W timesteps are passed as input when the full history exceeds this limit. The table also indicates whether each model supports past-only covariates, which is relevant for one of our baselines that passes related series as past-only covariates (see Section 4). We use Chronos-2 [Ansari et al., 2025] as our primary TSFM for detailed analysis beyond accuracy comparisons, as it is the best-performing model in our experiments and supports our full evaluation horizon. Accordingly, we provide a brief overview of its architecture.

Chronos-2 processes input series through robust scaling and patch embedding, then applies a transformer stack that alternates between time attention, which aggregates information across patches within a single series, and group attention, which aggregates information across all series within a group at each patch position. In our case, a group corresponds to a target series with its associated covariates. Hence, *group attention* is the primary mechanism through which covariate information is propagated to the target forecast, and its weights reveal which covariates the model relies on the most. The last layer of the transformer stack

produces the *final forecast embeddings*, from which multi-patch quantile forecasts are generated. As the last intermediate representation before the forecast output, these embeddings provide a window into how the model internally processes different covariate configurations. We revisit the role of group attention and forecast embeddings in Section 5, where we analyze how Chronos-2 incorporates different factor types under the FA-TSFM framework.

3 Methodology

3.1 The Factor-Augmented Time Series Foundation Model (FA-TSFM) framework

Having introduced the two base model components, we now describe the FA-TSFM framework, depicted in Figure 1. Colors in the steps below correspond to the figure (red: fitted values, blue: forecasts, green: factors). In its most general form, an FA-TSFM performs the following steps:

1. Run the DFM on $\boldsymbol{Y}_{1:T}$ to produce fitted values $\hat{\boldsymbol{Y}}^{\text{DFM}}_{1:T}$, forecasts $\hat{\boldsymbol{Y}}^{\text{DFM}}_{T+1:T+H}$, and factors $\boldsymbol{F}_{1:T+H}$.
2. Transform the original targets $\boldsymbol{Y}_{1:T}$ to obtain TSFM targets $\boldsymbol{\xi}_{1:T}$. Transformations may include residualization ($\hat{\varepsilon}_{i,t} := Y_{i,t} - \hat{Y}^{\text{DFM}}_{i,t}$ using the DFM fitted values), normalization, and other preprocessing steps. See Appendix B for more details about transformations.
3. For each series i, obtain TSFM forecasts $\hat{\xi}_{i,T+1:T+H} = \text{TSFM}(\xi_{i,1:T}, \mathcal{F}_{i,1:T+H})$ by passing in a subset $\mathcal{F}_{i,1:T+H}$ of the factors $\boldsymbol{F}_{1:T+H}$ as past-future covariates. Note that $\mathcal{F}_{i,t}$ need not equal $\boldsymbol{F}_{i,t}$ (defined in Section 2): the former is the set of factors passed to the TSFM, while the latter is the subset of factors that contribute to the DFM forecast for series i.
4. Inverse transform $\hat{\boldsymbol{\xi}}_{T+1:T+H}$ (using the DFM forecasts $\hat{\boldsymbol{Y}}^{\text{DFM}}_{T+1:T+H}$ if we choose to residualize) to obtain the final forecasts $\hat{\boldsymbol{Y}}_{T+1:T+H}$.

The choice in Step 2 significantly impacts the TSFM's inference task. Residualization

shifts the TSFM's task from forecasting $Y_{i,t}$ directly to learning the DFM's systematic errors $\hat{\varepsilon}_{i,t}$. This is ideal when the series is under tight control for scenario planning, as it keeps final forecasts closely tied to the DFM's baseline predictions, with the TSFM providing an additive correction. We examine this setup in Section 6.2. If the DFM is not well-tuned for the target series, not residualizing may be preferable, as it allows the TSFM more freedom in utilizing factor information without being anchored to potentially poor predictions. This is particularly relevant in large-scale forecasting, where it is prohibitive to produce properly tuned DFM forecasts for every series. In such settings, a practical alternative is to fit DFMs on a handful of core series and pass their factors as covariates to the TSFM for forecasting the remaining series. We explore this setup in Section 6.1.

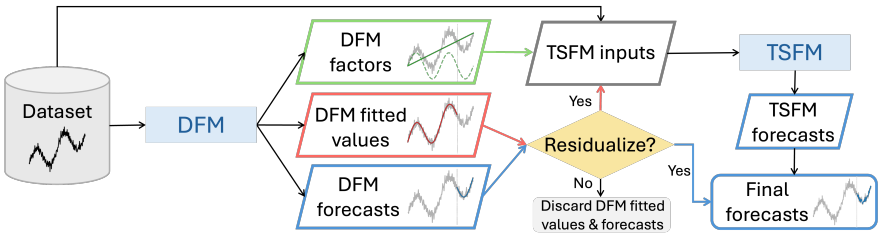

Figure 1: The FA-TSFM pipeline. The DFM produces factors (covariates for the TSFM), fitted values, and forecasts. When residualizing, fitted values transform TSFM inputs into DFM residuals, and forecasts are added back to produce final forecasts. Without residualization, only factors are used.

3.2 Experimental setup

Dataset & evaluation As part of an ongoing effort to develop a production supply chain forecasting system, we evaluate on a proprietary daily dataset organized in a four-level tree hierarchy (Figure 2). The hierarchy contains 155 series observed over 3006 time steps: 1 root node at level 0, 3 at level 1, 12 at level 2, and 139 at level 3. For our analysis, we group levels 0 and 1 as *coarse grain* (4 series) and levels 2 and 3 as *fine grain* (151 series). This distinction is relevant because in our experiments the DFM is well-tuned for coarse-grain series but less so for fine-grain series, which, as we show in Section 4, affects the optimal residualization strategy.

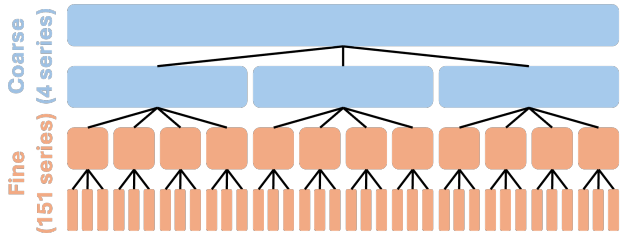

Figure 2: Tree structure of the dataset.

Forecasts are evaluated through backtests across 17 monthly-spaced estimation end dates. Each estimation end date defines a cutoff T up to which data is observed; the DFM and TSFMs are then evaluated on horizons $h = 1, \ldots, H = 298$ beyond this cutoff. We report MAPE (mean absolute percentage error) and RMSE (root mean squared error), averaged across all estimation end dates, horizons, and series within a given grain. To complement accuracy metrics, we also report statistical significance results using a multivariate extension [Jing and Jang, 2025] of the Diebold-Mariano (DM) test [Diebold and Mariano, 1995, Harvey et al., 1997]. For each horizon h, all series within a grain are tested jointly, producing one p-value per horizon and yielding $H = 298$ p-values per grain and model configuration. See Appendix D for additional details.

Hierarchical DFM In many applications, the series in $\boldsymbol{Y}_t$ share known relationships that can be encoded as a directed acyclic graph (DAG) $\mathcal{G}$ with nodes $\{1, \ldots, N\}$, where ancestor nodes represent drivers of their descendants, e.g., total sales as an aggregate of category sales, or macroeconomic indicators as drivers of sector-level metrics. Our dataset exhibits exactly this structure: the series are organized in a tree as depicted in Figure 2, motivating a hierarchical DFM with sparse loadings in which each series loads only on its ancestor factors and its own factors. The forecast for series i is:

$$\hat{Y}_{i,t}^{\text{DFM}} = \boldsymbol{\beta}_i^{\text{anc}} \boldsymbol{F}_{i,t}^{\text{anc}} + \mathcal{T}_{i,t} + \mathcal{S}_{i,1,t} + \mathcal{S}_{i,2,t} + \mathcal{S}_{i,3,t} + c_{i,t}, \tag{3}$$

so $\boldsymbol{\beta}_i = [\boldsymbol{\beta}_i^{\text{anc}}, 1, 1, 1, 1, 1]$ and $\boldsymbol{F}_{i,t} = [\boldsymbol{F}_{i,t}^{\text{anc}}, \mathcal{T}_{i,t}, \mathcal{S}_{i,1,t}, \mathcal{S}_{i,2,t}, \mathcal{S}_{i,3,t}, c_{i,t}]$. The main factor types are

- $\mathcal{T}_{i,t}$: A local linear trend component with mean-reverting slope;
- $\mathcal{S}_{i,1,t}$: A Fourier-based annual seasonality component;
- $\mathcal{S}_{i,2,t}$: A weekly seasonality component estimated from day-of-week dummies;
- $\mathcal{S}_{i,3,t}$: An event seasonality component estimated from (potentially floating) event dummies;
- $c_{i,t}$: An AR(1) cycle component that captures persistent deviations from the trend.

We focus on the first four interpretable factors (trend and seasonality components), as these are most relevant for supply chain planning. The ancestor factors are $\boldsymbol{F}_{i,t}^{\text{anc}} = \{\mathcal{T}_{j,t}, \mathcal{S}_{j,1,t} : j \text{ is an ancestor of } i\}$, where ancestors are determined based on the known hierarchy tree $\mathcal{G}$. Lastly, all factors are estimated via Markov Chain Monte Carlo (MCMC), while the loadings onto ancestor factors $\boldsymbol{\beta}_i^{\text{anc}}$ are obtained via an orthogonalization step where $Y_{i,t}$ is regressed on pre-extracted ancestor factors. Additional details about this hierarchical DFM are deferred to Appendix A.

Returning to the FA-TSFM framework, the factor subset $\mathcal{F}_{i,t}$ passed to the TSFM in Step 3 (Section 3.1) can in principle include any subset of the full factor set $\boldsymbol{F}_t$. In our experiments, we set $\mathcal{F}_{i,t} = \boldsymbol{F}_{i,t} \setminus \{c_{i,t}\}$ for all nodes i at hierarchy levels 0, 1, and 2, that is, all interpretable factors relevant to node i in the DFM observation equation. For nodes at level 3, we set $\mathcal{F}_{i,t} = (\boldsymbol{F}_{i,t} \setminus \{c_{i,t}\}) \cup \{\mathcal{S}_{1,3,t}\}$, additionally including the root node's ($i = 1$) event seasonality factor. This is motivated by the greater volatility of level 3 series resulting in less reliable extracted event seasonality factors.

4 Main results: FA-TSFM forecast accuracy

We now evaluate the benefit of factor inclusion by comparing the following covariate configurations:

- **Raw**: *No covariates*, the TSFM only receives past observations of the target series, $Y_{i,1:T}$;
- **EI**: *Event indicator variables* as past-future covariates, encoding when specific events occur;
- **Anc**: EI plus *ancestor series* as past-only covariates (available only for models supporting past-only covariates; see Table 1). Under this configuration, the TSFM in theory receives the same multivariate information as FA-TSFM, but not in the form of structured factors;
- **FA**: Our FA-TSFM approach, with *DFM-extracted factors* $\mathcal{F}_{i,1:T+H}$ as past-future covariates.

Each configuration is crossed with two residualization strategies, for a total of eight specifications per model. Under residualization, the TSFM forecasts a transformation of the DFM residuals $Y_{i,t} - \hat{Y}_{i,t}^{\text{DFM}}$, and the DFM forecast is added back to produce the final forecast; consequently, all residualized configurations indirectly contain factor information through $\hat{Y}_{i,t}^{\text{DFM}}$.

Table 2 reports backtest metrics organized by grain (coarse, fine), horizon bucket ($\leq$ 128, $>$ 128), TSFM, and metric (MAPE, RMSE). Naive TSFMs (non-residualized Raw, EI, and Anc) receive no factor information, while factor-informed configurations receive it either explicitly through factor covariates (non-residualized FA) or implicitly through residualization (residualized Raw, EI, and Anc), or both (residualized FA). The five rightmost non-DFM columns (**bolded headers**) are therefore all anchored by the DFM and factor-informed. For each row, cells are shaded from green (best) to red (worst), with the best additionally **bolded** and the second best underlined. Results that are also the best across all TSFMs for a given grain-horizon combination are marked in **blue**. ApolloPFN and Moirai 1.0 are reported only for horizons ≤ 128 due to their constraints (Table 1). Analogous results on additional datasets and a different DFM are in Appendices I and J.

We first discuss results for Chronos-2, TabPFN-TS, and ApolloPFN. At coarse grain, non-residualized FA-TSFM consistently outperforms all naive TSFM configurations (non-residualized Raw, EI, Anc) by a notable margin, underscoring the importance of factor

information for forecast accuracy. Residualized FA-TSFM achieves the best results across all TSFMs and metrics. Furthermore, nearly all coarse-grain FA-TSFM configurations outperform the standalone DFM, though the improvements are modest — consistent with the DFM being already well-tuned at this grain.

At fine grain, an anti-residualization pattern emerges: non-residualized FA-TSFM tends to outperform its residualized counterpart, particularly in RMSE, though the pattern is less consistent for TabPFN-TS and ApolloPFN compared to Chronos-2. From inspecting individual series, we found this reversal reflects non-residualized TSFMs' flexibility to correct forecast defects stemming from the DFM being less well-tuned. The stronger effect in RMSE is expected, as it places greater weight on the outlier errors characteristic of such defects. All factor-informed configurations significantly outperform the standalone DFM at fine grain, with TabPFN-TS benefiting the most from factor inclusion.

The coarse versus fine grain pattern connects to a broader insight about residual structure in factor models. Ng and Scanlan [2024] show that modeling the dynamics of idiosyncratic residuals after factor extraction is often more important than the choice of factor estimator itself. Our results are consistent with this when the DFM is well-calibrated: residualization succeeds because the DFM residuals contain exploitable serial structure that the TSFM can learn. However, when the DFM is less well-calibrated (fine grain), residuals also contain systematic forecast errors from the DFM, which propagate into the final output and degrade accuracy. In this regime, the covariate pathway is more robust: the TSFM learns factor–target relationships directly without being anchored to flawed residuals. This finding is confirmed across eight additional proprietary datasets (113–128 fine-grain series each) using Chronos-2, where DFM priors were transferred without further calibration. Despite the suboptimal priors, FA-Chronos-2 achieves the best or near-best accuracy in nearly every dataset, grain, and horizon combination, demonstrating that the covariate pathway retains substantial value even from poorly calibrated DFMs.

Moirai 1.0 exhibits a distinct pattern: adding covariates degrades performance, with

Raw outperforming EI, Anc, and FA within each residualization group, suggesting it does not effectively leverage covariates in our setting. Nevertheless, at coarse grain it benefits from indirect factor information via residualization, with all residualized configurations substantially outperforming non-residualized Raw. At fine grain, the picture is more nuanced: residualized configurations remain superior in MAPE, but non-residualized configurations achieve better RMSE. This divergence suggests a trade-off between the factor information from residualization and the flexibility to correct DFM forecast defects, the latter of which echoes the anti-residualization pattern observed for other TSFMs at fine grain.

Table 2: Forecast accuracy metrics of naive TSFMs ("Not residualized - Raw, EI, Anc"), factor-informed TSFMs ("Not residualized - FA" and all "Residualized" columns), and the base DFM.

				Not residualized				Residualized				DFM
Grain	Horizon	TSFM	Metric	Raw	EI	Anc	**FA**	**Raw**	**EI**	**Anc**	**FA**	
Coarse	≤ 128	Chronos-2	MAPE	0.3382	0.3018	0.3017	0.2443	0.2348	0.2390	0.2343	**0.2313**	0.2439
			RMSE	0.1271	0.0877	0.0874	0.0703	0.0705	0.0708	0.0702	**0.0696**	0.0715
		TabPFN-TS	MAPE	1.2225	1.1265	–	0.2752	0.2486	0.2482	–	**0.2422**	0.2439
			RMSE	0.2834	0.2384	–	0.0750	0.0723	0.0722	–	**0.0712**	0.0715
		Moirai 1.0	MAPE	0.5575	0.6459	0.6513	0.6316	**0.2351**	0.2506	0.2548	0.2392	0.2439
			RMSE	0.1684	0.1835	0.1812	0.1812	**0.0707**	0.1643	0.0715	0.0711	0.0715
		ApolloPFN	MAPE	0.3934	0.3237	–	0.3037	0.2332	0.2291	–	**0.2254**	0.2439
			RMSE	0.1408	0.0881	–	0.0797	0.0701	0.0689	–	**0.0688**	0.0715
	> 128	Chronos-2	MAPE	0.4028	0.3797	0.3986	0.2976	0.2974	0.2999	0.2931	**0.2903**	0.3059
			RMSE	0.1353	0.1055	0.1077	**0.0840**	0.0866	0.0869	0.0860	0.0849	0.0877
		TabPFN-TS	MAPE	1.4531	1.2070	–	0.3416	0.3107	0.3121	–	**0.3078**	0.3059
			RMSE	0.3252	0.2551	–	0.0924	0.0884	0.0885	–	**0.0874**	0.0877
Fine	≤ 128	Chronos-2	MAPE	1.5496	1.5355	1.5206	**1.4644**	1.5372	1.5371	1.5256	1.5036	2.6663
			RMSE	0.2689	0.2571	0.2556	**0.2514**	0.3463	0.3240	0.3295	0.3075	0.4736
		TabPFN-TS	MAPE	3.1023	3.2204	–	**1.7018**	2.0875	2.0753	–	1.7049	2.6663
			RMSE	0.4278	0.4176	–	**0.2677**	0.4181	0.4165	–	0.3576	0.4736
		Moirai 1.0	MAPE	1.8031	2.2632	2.2730	2.3225	**1.6754**	1.7544	1.7620	1.7770	2.6663
			RMSE	**0.3002**	0.3455	0.3437	0.3478	0.3935	0.4040	0.3975	0.3982	0.4736
		ApolloPFN	MAPE	1.7244	1.6918	–	1.5495	1.5776	1.6011	–	**1.5488**	2.6663
			RMSE	0.2882	0.2734	–	**0.2581**	0.3680	0.3694	–	0.3447	0.4736
	> 128	Chronos-2	MAPE	1.8496	1.8049	1.8485	**1.7441**	1.9125	1.8888	1.8746	1.8249	3.0397
			RMSE	0.2843	0.2713	0.2729	**0.2673**	0.3727	0.3688	0.3699	0.3252	0.4972
		TabPFN-TS	MAPE	3.9666	4.1161	–	2.1902	2.5443	2.5279	–	**2.1595**	3.0397
			RMSE	0.4986	0.4960	–	**0.3054**	0.4487	0.4468	–	0.3960	0.4972

To illustrate how factor information improves forecasts, Figure 3 shows non-residualized Chronos-2 forecasts for a level 2 series (temporally aggregated for confidentiality). Raw fails to capture sharp peaks during floating events. EI addresses this but introduces other deficiencies: the model does not anticipate the pre-event dip at time step 6, a pattern

captured by both the DFM and FA-Chronos-2. Both non-factor-augmented configurations also exhibit minor trend-level issues around time steps 10-17 and 28-35.

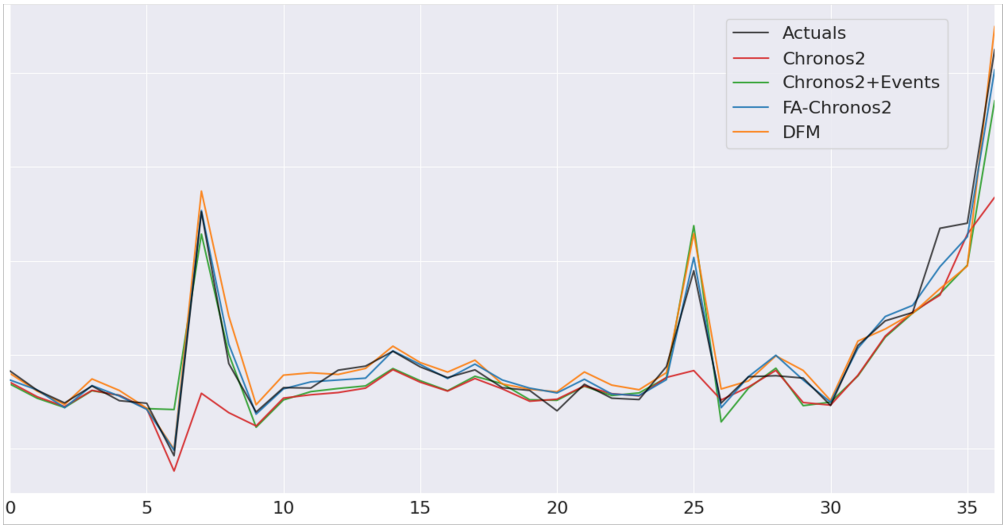

Figure 3: Sample forecasts for a level 2 series.

We further assess the statistical significance of accuracy differences using the multivariate DM test [Jing and Jang, 2025], which tests whether one model significantly improves over another in at least one series within a grain at each horizon. Table 3 reports results for FA-Chronos-2 against seven baselines. For each (horizon, grain, comparison), we run two tests: a *direct test* evaluating whether FA-Chronos-2 is significantly better than the baseline for some series, and a *reverse test* evaluating the opposite. The "% FA-Chronos-2 > Baseline" columns report the percentage of horizons at which the direct test rejects, and "% Baseline > FA-Chronos-2" the percentage for the reverse test. Bolded headers correspond to the recommended residualization strategy per grain. Since the test targets improvement in *any* series, both tests can reject at the same horizon if different series favor different models.

Table 3: Percentage of horizons where the one-sided multivariate DM test rejects at $\alpha = 0.1$.

		% FA-Chronos-2 ¿ Baseline				% Baseline ¿ FA-Chronos-2			
		Coarse		Fine		Coarse		Fine	
Baseline		Non-resid	**Resid**	**Non-resid**	Resid	Non-resid	**Resid**	**Non-resid**	Resid
Non-resid	Raw	22.8%	29.2%	27.2%	32.2%	2.3%	5.7%	2.0%	38.9%
	EI	45.3%	41.3%	60.7%	54.4%	1.7%	5.4%	0.7%	36.6%
	Anc	52.7%	49.3%	65.4%	60.7%	0.7%	5.7%	0.0%	33.2%
Resid	Raw	23.8%	18.1%	74.5%	60.1%	22.1%	4.4%	16.8%	11.7%
	EI	25.2%	53.4%	83.9%	95.3%	25.5%	1.7%	22.1%	0.0%
	Anc	21.5%	16.1%	74.5%	61.7%	26.2%	1.7%	20.8%	1.3%
DFM		30.5%	69.8%	99.7%	99.3%	20.8%	0.3%	1.0%	1.0%

At coarse grain, the direct test rejects at 16.1–69.8% of horizons. Reverse rejections remain low (0.3–5.7%), indicating improvements concentrated at certain horizons but rarely at the cost of degradation. At fine grain, direct rejections are more substantial (27.2–99.7%). Against the DFM and non-residualized counterparts, reverse rejections are minimal (0.0–2.0%). Against residualized baselines, reverse rejections are higher (16.8–22.1%), but offset by substantially larger direct rejection rates (74.5–83.9%). Analogous results for the univariate DM test can be found in Appendix E.

Beyond point forecasts, factor augmentation also narrows prediction intervals (PIs) while maintaining coverage for non-residualized Chronos-2, TabPFN-TS, and ApolloPFN (Appendix F, Table 7). For the residualized case, combining DFM and TSFM PIs into a valid PI remains an open challenge.

5 Contribution of factors to TSFM forecasts

Having established the overall value of factor information, we investigate the contribution of individual factor types to FA-TSFM accuracy. Table 4 presents non-residualized FA-Chronos-2 results when $\mathcal{F}_{i,t}$ is restricted to a single factor type, each including factors of that type from all hierarchy levels. The "Default" column includes all factor types. We use non-

residualized configurations to isolate each factor type's contribution without reintroducing excluded factors through $\hat{Y}_{i,t}^{\text{DFM}}$. Residualized results, which instead measure contributions beyond the DFM's linear effects, are in Appendix G.

Table 4: Impact of factors types on non-residualized FA-Chronos-2 forecast accuracy.

			Baselines			Trend	Seasonality			Default
Grain	Horizon	Metrics	Raw	EI	Anc	$\mathcal{T}_t$	$\mathcal{S}_{1,t}$	$\mathcal{S}_{2,t}$	$\mathcal{S}_{3,t}$	$\mathcal{T}_t$, $\mathcal{S}_{1:3,t}$
Coarse	≤ 128	MAPE	0.3382	0.3018	0.3017	0.3639	0.3584	0.3710	0.2617	**0.2443**
		RMSE	0.1271	0.0877	0.0874	0.1285	0.1282	0.1368	0.0755	**0.0703**
	> 128	MAPE	0.4028	0.3797	0.3986	0.4365	0.4426	0.4600	0.3467	**0.2976**
		RMSE	0.1353	0.1055	0.1077	0.1374	0.1389	0.1424	0.0942	**0.0840**
Fine	≤ 128	MAPE	1.5496	1.5355	1.5206	1.6222	1.5659	1.6170	1.4847	**1.4644**
		RMSE	0.2689	0.2571	0.2556	0.2754	0.2751	0.2769	0.2539	**0.2514**
	> 128	MAPE	1.8496	1.8049	1.8485	1.9382	1.8442	1.8850	1.7670	**1.7441**
		RMSE	0.2843	0.2713	0.2729	0.2939	0.2879	0.2869	0.2678	**0.2673**

The event factor $\mathcal{S}_{3,t}$ stands out as the most impactful, with gains far exceeding those from event dummies (EI). In contrast, $\mathcal{T}_t$, $\mathcal{S}_{1,t}$, and $\mathcal{S}_{2,t}$ individually provide minimal benefit or degrade performance. Interestingly, the default specification with all factor types outperforms $\mathcal{S}_{3,t}$ alone, suggesting that Chronos-2 leverages interaction effects between factor types. Two questions arise: a) How does $\mathcal{S}_{3,t}$ differ from EI; b) How do different factor types interact? To investigate, we derive mechanistic insights [Ferrando et al., 2024, Wiliński et al., 2025] from the *final forecast embeddings* and *group attention weights* of Chronos-2 (Section 2). Key findings are reported here, with setup details and additional analyses in Appendix H.

For a), we compare activation differences in the forecast embeddings. Let $\mathbf{D}^{A,B} := \mathbf{E}^A - \mathbf{E}^B$ be the difference in embeddings between configurations A and B (see Appendix H.1 for the exact definition of embeddings $\mathbf{E}$). Figure 4 shows the top 5 principal component (PC) scores of $\mathbf{D}^{\text{EI,Raw}}$ and $\mathbf{D}^{\mathcal{S}_3,\text{Raw}}$. Each heatmap has estimation end dates on the x-axis and output patches on the y-axis. Since both EI and $\mathcal{S}_3$ introduce event information absent in Raw, event-related structure appears as diagonal bands that track event dates

across estimation end dates. The leading PCs and the pair $(\mathrm{PC}_3^{\mathrm{EI}}, \mathrm{PC}_5^{\mathcal{S}_3})$ have scores that capture similar event structure, confirming shared event encoding. However, $\mathcal{S}_3$'s PC2–4 scores overlay additional off-diagonal patterns, indicating that $\mathcal{S}_3$ also modifies embeddings at non-event patches — a potential source of its accuracy advantage over EI.

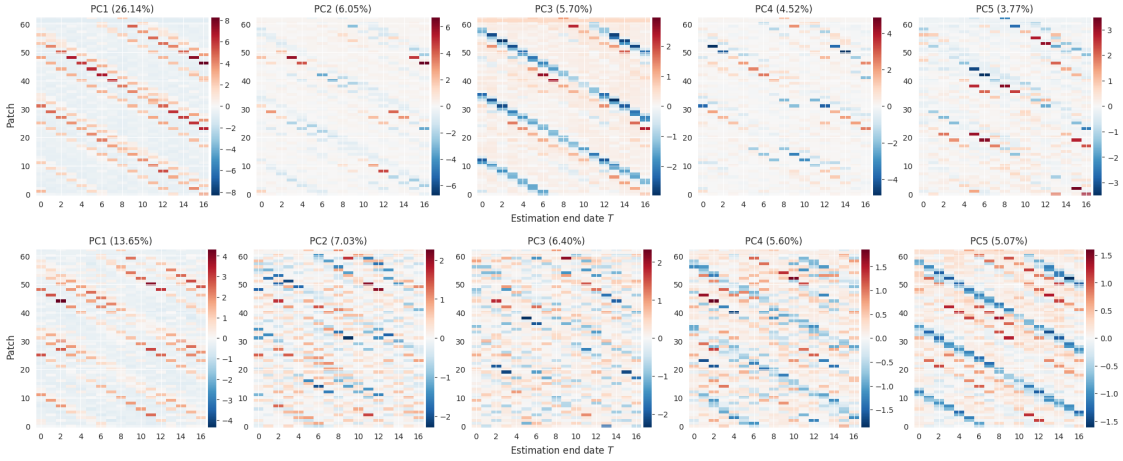

Figure 4: Top 5 PC scores of $\mathbf{D}^{\mathrm{EI,\ Raw}}$ (top) and $\mathbf{D}^{\mathcal{S}_3,\ \mathrm{Raw}}$ (bottom).

To answer b), we examine how group attention weights shift when factor types are added or removed in Figure 5 (with the same axes as Figure 4). We aggregate attention weights by summing over all factors of the same type and average across nodes, layers, and heads (see Appendix H.4 for details). The left panel compares the full FA setup against an $\mathcal{S}_{3,t}$-only run, while the right panel compares FA against a run with all factors except $\mathcal{S}_{3,t}$. In both comparisons, the largest attention shifts concentrate at event patches. The left panel (FA vs $\mathcal{S}_{3,t}$-only) shows that adding $\{\mathcal{T}_t, \mathcal{S}_{1,t}, \mathcal{S}_{2,t}\}$ increases self-attention and decreases attention on $\mathcal{S}_{3,t}$ during events, suggesting that when trend and seasonality are available, the model relies less on $\mathcal{S}_{3,t}$ to contextualize event effects.

In a complementary pattern, the right panel shows that without $\mathcal{S}_{3,t}$, the model increases self-attention during event periods, relying on its own history to explain event-driven fluctuations. When $\mathcal{S}_{3,t}$ is introduced, self-attention during events decreases, with the freed attention redirected toward trend $\mathcal{T}_t$ and weekly seasonality $\mathcal{S}_{2,t}$. This indicates that $\mathcal{S}_{3,t}$ provides a direct signal for event effects, freeing the model to attend to other factors and

learn how events interact with trend and seasonality. Such attention redistribution may help explain the ablation finding that combining factor types outperforms any individual factor. Confirming this causal link via activation patching [Meng et al., 2022] is a natural next step.

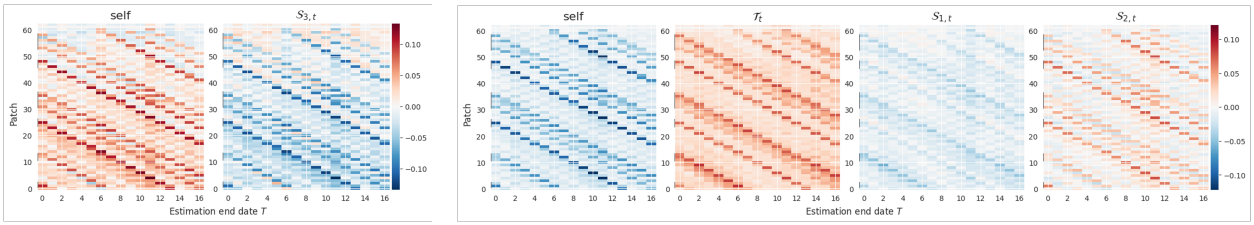

Figure 5: Difference in group attention: FA vs. $\mathcal{S}_{3,t}$ (left) and FA vs. $\{\mathcal{T}_t, \mathcal{S}_{1,t}, \mathcal{S}_{2,t}\}$ (right).

6 Implications for hierarchical forecasting and scenario planning

6.1 DFM-anchored hierarchical forecasting

In this section, we investigate whether factors from a subset of hierarchy levels suffice for accurate forecasting, motivated by the prospect of a DFM-anchored system introduced in Section 3.1: fitting DFMs on a handful of "core series" and passing their factors to a TSFM to forecast the remaining "peripheral series." Table 5 examines non-residualized FA-Chronos-2 accuracy on level 3 series when only factors from specific hierarchy levels are provided. Each column indicates which levels' factors are passed; for example, "0, 1" uses only factors from levels 0 and 1 series to forecast level 3 series. As in Section 5, we use non-residualized configurations to isolate the effect of each factor subset.

Table 5: Impact of hierarchical factors on FA-Chronos-2 forecast accuracy of series in hierarchy level 3.

		Baselines			Top-down			Bottom-up			Default
Horizon	Metrics	Raw	EI	Anc	0	0, 1	0, 1, 2	3	2, 3	1, 2, 3	0, 1, 2, 3
≤ 128	MAPE	1.6571	1.6446	1.6289	1.5729	**1.5721**	1.5750	1.6025	1.5914	1.5833	1.5739
	RMSE	0.2789	0.2678	0.2661	0.2627	0.2628	**0.2553**	0.2626	0.2619	0.2625	0.2623
> 128	MAPE	1.9756	1.9296	1.9755	1.9034	1.8776	1.8751	1.9128	1.8976	1.8814	**1.8702**
	RMSE	0.2946	0.2820	0.2835	0.2804	0.2800	**0.2773**	0.2787	0.2779	0.2787	0.2784

We see that bottom-up strategies relying on node-specific factors from level 3 noticeably outperform the baselines, indicating that factor information retains value even when the underlying DFM is not well-calibrated for these series. More notably, top-down strategies using only ancestor factors perform comparably to the default configuration that includes all levels' factors, with differences that are small relative to the gap over the baselines. This means factors extracted from well-tuned coarse grain series are sufficient to achieve most of the accuracy gains at fine grain, supporting the vision of a DFM-anchored forecasting system in which TSFMs leverage a small number of core series factors to forecast peripheral series without requiring them to be modeled by the DFM.

6.2 Forecast adjustments under factor augmentation

Beyond scalability, a key advantage of DFM-based systems is the ability to perform forecast adjustments through factor manipulation. Consider the DFM described in Section 2. An adjustment to factor $F_{r,t}$ (denoted as $\tilde{F}_{r,t}$ post-adjustment) impacts all series i such that $F_{r,t} \in \boldsymbol{F}_{i,t}$. In general, the adjustment affects forecasts solely through the observation equation: the factor's contribution to series i's forecast is updated from $\lambda_{i,r}F_{r,t}$ to $\lambda_{i,r}\tilde{F}_{r,t}$, with the loadings $\lambda_{i,r}$ held fixed [Waggoner and Zha, 1999, Bańbura et al., 2015]. The same mechanism extends naturally to FA-TSFMs. An adjustment to factor $F_{r,t}$ simply replaces its values within each covariate set $\mathcal{F}_{i,1:T+H}$ passed to the TSFM with the adjusted $\tilde{F}_{r,t}$. When residualization is used, the TSFM target must also be updated: residuals are recomputed against the adjusted

DFM fitted values, and the adjusted DFM forecasts are added back in the final step.

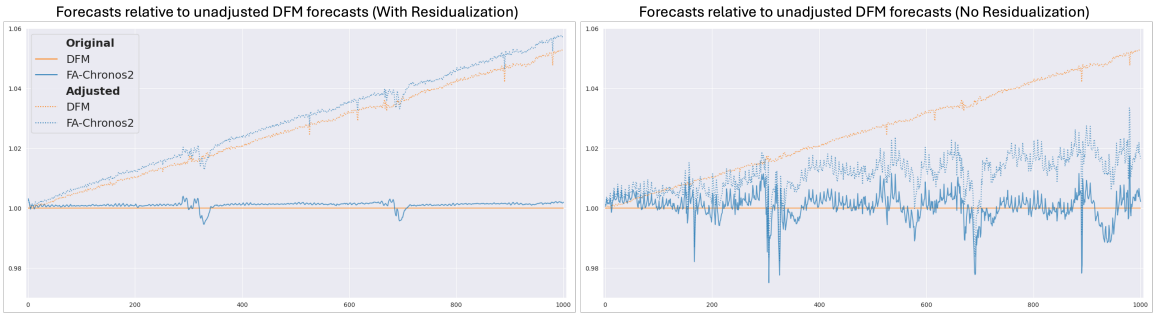

Figure 6: Forecast ratios relative to unadjusted DFM (solid: original; dotted: adjusted).

Figure 6 illustrates how FA-TSFM forecasts respond to a linear trend adjustment. Solid lines denote original forecasts and dotted lines denote adjusted forecasts, with orange corresponding to the DFM and blue to FA-Chronos-2. All forecasts are normalized by the original DFM forecast, so the solid orange baseline is $y = 1$. As expected, the adjusted DFM forecast (dotted orange; both sides) closely tracks the imposed linear slope. For the residualized setup, adjusted FA-Chronos-2 (dotted blue; left) closely follows the adjusted DFM, preserving its linear, predictable adjustment behavior. In contrast, the non-residualized FA-Chronos-2 (dotted blue; right) dampens the positive adjustment to a flatter shape, exhibiting non-linear behavior as Chronos-2 applies its own learned dynamics to the adjusted factors. This highlights a situational trade-off: residualization preserves predictable, controllable responses essential for scenario planning, while non-residualization allows the TSFM greater flexibility, improving forecast accuracy when the DFM is not well-tuned.

7 Conclusion, limitations, and future work

This paper introduced FA-TSFM, a framework that integrates DFM-extracted factors into TSFMs. On a real-life supply chain dataset, FA-TSFM consistently improved forecast accuracy across three of four evaluated TSFMs. Ablation studies and mechanistic interpretability analysis revealed which factor types contribute most and how Chronos-2 incorporates them

through attention redistribution. Further, ancestor factors alone sufficed for forecasting fine-grain series, supporting DFM-anchored systems that scale factor information without per-series tuning. We also characterized an accuracy-controllability trade-off: residualization preserves forecast adjustability at the cost of flexibility.

Our work has several limitations. The framework requires the TSFM to effectively integrate covariates, which did not hold for Moirai 1.0; this may vary across datasets and architectures. Whether residualization improves accuracy is also model- and data-dependent, requiring practitioners to evaluate both strategies. Empirically, the main results are based on one dataset and DFM, though we explore additional datasets and a simpler DFM in Appendices I and J. Results in Sections 5 and 6 are based on Chronos-2 and may not fully transfer to other TSFMs.

Several future directions remain. First, combining DFM and TSFM probabilistic forecasts to produce valid prediction intervals under residualization is an open challenge. Second, preliminary experiments with learned per-factor weights show out-of-sample gains, suggesting that optimizing the factor representation for the TSFM, or domain-specific fine-tuning more broadly, could further improve accuracy. Finally, FA-TSFMs can identify DFM defects, and re-tuning priors for flagged nodes yields improvements without actuals, pointing toward an iterative DFM-TSFM improvement pipeline.

Acknowledgments

We thank Carlos Carvalho, Andrea Tambalotti, Allan Timmermann, and Pedro Lima for their joint work with Davide Pettenuzzo in developing the hierarchical DFM used in this paper. We also thank Abdul Fatir Ansari, Andrea Tambalotti, Andreas Auer, Boris Oreshkin, Boyuan Zhang, Danila Maroz, Dmitry Efimov, Hanyu Zhang, Mengfei Cao, Ramon Huerta, Rob Vigfusson, Steven Klee, Su Jia, and Yuqian Liu for their feedback on this work.

References

D. Andrews. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, 59(3):817–858, 1991.

A. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. Rangapuram, S. Pineda Arango, S. Kapoor, J. Zschiegner, D. Maddix, M. Mahoney, K. Torkkola, A. Gordon Wilson, M. Bohlke-Schneider, and Y. Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research (TMLR)*, 2024.

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

S. Aranguri, J. Drori, and N. Nanda. SAE on activation differences. Available at `https://www.alignmentforum.org/posts/XPNJSa3BxMAN4ZXc7/sae-on-activation-differences`, 2024.

J. Bai and S. Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

M. Bańbura, D. Giannone, and M. Lenza. Conditional forecasts and scenario analysis with vector autoregressions for large cross-sections. *International Journal of Forecasting*, 31: 739–756, 2015.

V. Capdeville, K. Gonçalves, and J. Pereira. Bayesian factor models for multivariate categorical data obtained from questionnaires. *Journal of Applied Statistics*, 48(16):3150–3173, 2019.

A. Carriero, D. Pettenuzzo, and S. Shekhar. Macroeconomic forecasting with large language models. Available at SSRN: `https://dx.doi.org/10.2139/ssrn.4881094`, 2025.

C. Challu, K. Olivares, B. Oreshkin, F. Garza, M. Mergenthaler-Canseco, and A. Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. *Proceedings of the*

AAAI Conference on Artificial Intelligence, 37(6):6989–6997, 2023.

A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. *41st International Conference on Machine Learning (ICML)*, pages 10148–10167, 2024.

F. Diebold and R. Mariano. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):134–144, 1995.

evhub AI Alignment Forum. Chris Olah's views on AGI safety. Available at `https://www.lesswrong.com/posts/X2i9dQQK3gETCyqh2/chris-olah-s-views-on-agi-safety`, 2019.

J. Ferrando, G. Sarti, A. Bisazza, and M. Costa-jussá. A primer on the inner workings of transformer-based language models. Available at `https://arxiv.org/abs/2405.00208`, 2024.

M. Forni, M. Hallin, M. Lippi, and L. Reichlin. The generalized dynamic-factor model: Identification and estimation. *The Review of Economics and Statistics*, 82(4):540–554, 2000.

L. Han, Y. Liu, L. Li, Q. Deng, J. Jiang, Y. Sun, Z. Yu, B. Wang, X. Lu, L. Ma, H. Ye, and D. Zhang. UniCA: Unified covariate adaptation for time series foundation model. *14th International Conference on Learning Representations (ICLR)*, 2026.

D. Harvey, S. Leybourne, and P. Newbold. Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13:281–291, 1997.

J. Hewitt and P. Liang. Designing and interpreting probes with control tasks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, 2019.

S. B. Hoo, S. Müller, D. Salinas, and F. Hutter. From tables to time: Extending TabPFN-v2 to time series forecasting, 2025. URL `https://arxiv.org/abs/2501.02945`.

A. Jing and P. Jang. Multivariate tests for point and probabilistic forecasts, with an applica-

tion to spatio-temporal weather forecast evaluation. *Proceedings of the 19th International Symposium on Spatial and Temporal Data*, pages 17–28, 2025.

S. Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica*, 59(6):1551–1580, 1991.

S. Kaufmann and C. Schumacher. Bayesian estimation of sparse dynamic factor models with order-independent and ex-post mode identification. *Journal of Econometrics*, 210: 116–134, 2019.

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

J. Lindsey, A. Templeton, J. Marcus, T. Conerly, J. Batson, and C. Olah. Sparse crosscoders for cross-layer features and model diffing. `https://transformer-circuits.pub/2024/crosscoders/index.html`, 2024.

E. Makalic and D. Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182, 2016.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The M5 competition: Background, organization, and implementation. *International Journal of Forecasting*, 38(4):1325–1336, 2022.

K. Meng, D. Bau, A. Andonianm, and Y. Belinkov. Locating and editing factual associations in GPT. *36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

J. Murray, D. Dunson, L. Carin, and J. Lucas. Bayesian gaussian copula factor models for mixed data. *J Am Stat Assoc*, 108(502):656–665, 2013.

W. Newey and K. West. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3):703–708, 1987.

S. Ng and S. Scanlan. Constructing high frequency economic indicators by imputation. *The Econometrics Journal*, 27(1):C1–C30, 2024.

B. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. N-BEATS: Neural basis expansion

analysis for interpretable time series forecasting. *8th International Conference on Learning Representations (ICLR)*, 2020.

K. Park, Y. Choe, and V. Veitch. The linear representation hypothesis and the geometry of large language models. *41st International Conference on Machine Learning (ICML)*, pages 39643–39666, 2024.

P. Poncela, E. Ruiz, and K. Miranda. Factor extraction using kalman filter and smoothing: This is not just another survey. *International Journal of Forecasting*, 37(4):1399–1425, 2022.

A. Potapczynski, R. Selvam, T. Konstantinova, S. Ramasubramanian, M. Wolff, K. Olivares, R. Ma, M. Cao, M. Mahoney, A. Wilson, B. Oreshkin, and D. Efimov. Time-aware prior fitted networks for zero-shot forecasting with exogenous variables. Available at `https://arxiv.org/abs/2603.15802`, 2026.

D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2019.

J. Stock. Asymptotic properties of least squares estimators of cointegrating vectors. *Econometrica*, 55(5):1035–1056, 1987.

J. Stock and M. Watson. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In J. Taylor and H. Uhlig, editors, *Handbook of Macroeconomics*, volume 2A, pages 899–960. Elsevier B.V., 2016.

D. Waggoner and T. Zha. Conditional forecasts in dynamic multivariate models. *The Review of Economics and Statistics*, 81(4):639–651, 1999.

M. Wiliński, M. Goswami, W. Potosnak, N. Żukowska, and A. Dubrawski. Exploring representations and interventions in time series foundation models. *42nd International Conference on Machine Learning (ICML)*, 2025.

G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. *41st International Conference on Machine Learning*

(ICML), 2024.

A Additional details about the hierarchical DFM

This appendix covers details about the hierarchical DFM presented in Section 2. For clarity, we restate the hierarchical DFM forecast equation from (3):

$$\hat{Y}_{i,t}^{\text{DFM}} = \boldsymbol{\beta}_i^{\text{anc}} \boldsymbol{F}_{i,t}^{\text{anc}} + \mathcal{T}_{i,t} + \mathcal{S}_{i,1,t} + \mathcal{S}_{i,2,t} + \mathcal{S}_{i,3,t} + c_{i,t}.$$

As mentioned in the main paper, the factor components are estimated via MCMC, while the ancestor loadings $\boldsymbol{\beta}_i^{\text{anc}}$ are obtained through an ordinary least squares (OLS) orthogonalization procedure.

A.1 Factor components

We now describe the mathematical form of each factor component, dropping the series subscript i for clarity. Specific hyperparameter values are omitted due to confidentiality constraints; we report prior families and structural forms.

Trend The local linear trend follows a random walk with mean-reverting slope:

$$\begin{aligned} \mathcal{T}_t &= \delta_{t-1} + \mathcal{T}_{t-1} + u_t^{\tau}, & u_t^{\tau} &\sim \mathcal{N}(0, \sigma_{\tau}^2), \\ \delta_t &= (1-\rho)\mu_{\delta} + \rho\delta_{t-1} + u_t^{\delta}, & u_t^{\delta} &\sim \mathcal{N}(0, \sigma_{\delta}^2), \end{aligned}$$

where $\rho \in (-1, 1)$ is the persistence parameter controlling how quickly the slope reverts to its long-run mean μ_{δ}. The variance parameters σ_{τ}^2 and σ_{δ}^2 have Inverse-Gamma priors, μ_{δ} has a Normal prior, and ρ has a Normal prior truncated to $(-1, 1)$ to ensure stationarity.

Annual seasonality The annual seasonality has a Fourier representation with K harmonics:

$$\mathcal{S}_{1,t} = \sum_{k=1}^{K} \theta_k^{\cos} \cos\left(\frac{2\pi kt}{365.25}\right) + \theta_k^{\sin} \sin\left(\frac{2\pi kt}{365.25}\right)$$

The coefficient vector $\boldsymbol{\theta}^a = [\theta_1^{\cos}, \theta_1^{\sin}, \ldots, \theta_K^{\cos}, \theta_K^{\sin}]^\top$ follows a hierarchical horseshoe prior [Makalic and Schmidt, 2016] along with other time-static seasonality coefficients (see details in A.1).

Weekly seasonality Weekly seasonality captures day-of-week effects via dummy variables:

$$\mathcal{S}_{2,t} = \sum_{d=1}^{6} \mathbf{1}\{t \text{ is day-of-week } d\}\theta_d^w,$$

where a base day-of-week is dropped and its effect absorbed into the trend factor. The coefficient vector $\boldsymbol{\theta}^w = [\theta_1^w, \ldots, \theta_6^w]^\top$ follows the hierarchical horseshoe prior like $\boldsymbol{\theta}^a$. We note that all TSFMs except Chronos-2 receive day-of-week information by default: ApolloPFN receives input start dates, while TabPFN-TS and Moirai 1.0 receive timestamp arrays.

Event seasonality The event seasonality captures a combination of time-static and time-varying event effects through dummy variables:

$$\mathcal{S}_{3,t} = \sum_{d \in \mathcal{D}^{\text{TS}}} \theta_d^e \mathbf{1}\{t \text{ is a time-static event } d\} + \sum_{d \in \mathcal{D}^{\text{TV}}} \gamma_{d,t} \mathbf{1}\{t \text{ is a time-varying event } d\},$$
$$\boldsymbol{\gamma}_t = \boldsymbol{\gamma}_{t-1} + u_t^\gamma, \quad \boldsymbol{\gamma}_t = [\gamma_{d,t}]_{d \in \mathcal{D}^{\text{TV}}}^\top, \quad u_t^\gamma \sim \mathcal{N}\Big(\mathbf{0}, \text{diag}\left(\sigma_{\gamma,1}^2, ..., \sigma_{\gamma,|\mathcal{D}^{\text{TV}}|}^2\right)\Big),$$

where $\mathcal{D}^{\text{TS}}$ and $\mathcal{D}^{\text{TV}}$ denote time-static and time-varying (potentially floating) events, respectively. The time-static coefficients $\boldsymbol{\theta}^e = [\theta_d^e]_{d \in \mathcal{D}^{\text{TS}}}$ follow the hierarchical horseshoe prior like $\boldsymbol{\theta}^a$ and $\boldsymbol{\theta}^w$, and the time-varying variance parameters $\sigma_{\gamma,d}^2$ have i.i.d. Inverse-Gamma

priors.

We now give details about the joint hierarchical horseshoe prior [Makalic and Schmidt, 2016] on $\boldsymbol{\theta} := [\boldsymbol{\theta}^a, \boldsymbol{\theta}^w, \boldsymbol{\theta}^e]$. The k-th element of $\boldsymbol{\theta}$ is sampled as

$$\theta_k \mid \kappa, \psi_k \sim \mathcal{N}\big(0, \kappa^2 \psi_k^2\big), \quad \psi_k \sim \mathcal{C}^+(0,1), \quad \kappa \sim \mathcal{C}^+(0,1),$$

where ψ_k are local shrinkage parameters and κ is a global shrinkage parameter, both with half-Cauchy priors. Sampling is performed via auxiliary variables that exploit the representation of a squared half-Cauchy as an Inverse-Gamma mixture of Inverse-Gammas:

$$\begin{aligned}
\psi_k^2 \mid z_{\psi_k} &\sim \text{Inv-Gamma}\big(0.5, z_{\psi_k}^{-1}\big), \quad z_{\psi_k} \sim \text{Inv-Gamma}(0.5, 1), \\
\kappa^2 \mid z_\kappa &\sim \text{Inv-Gamma}\big(0.5, z_\kappa^{-1}\big), \qquad z_\kappa \sim \text{Inv-Gamma}(0.5, 1).
\end{aligned}$$

This parameterization enables closed-form Gibbs updates for all shrinkage parameters conditional on the current draw of $\boldsymbol{\theta}$. The horseshoe prior encourages sparsity by shrinking small coefficients toward zero while allowing large coefficients to remain unshrunk.

Cycle component The cycle component is an AR(1) process with stochastic volatility that captures persistent deviations from the trend while maintaining long-run mean reversion through the stationary constraint $|\phi| < 1$:

$$\begin{aligned}
c_t &= \phi c_{t-1} + u_t^c, \quad u_t^c \sim \mathcal{N}\big(0, \exp(h_t)\big), \\
h_t &= h_{t-1} + v_t, \qquad v_t \sim \mathcal{N}(0, \sigma_h^2)
\end{aligned}$$

where h_t is a log-volatility process and σ_h^2 is the stochastic volatility parameter. The parameter σ_h^2 has an Inverse-Gamma prior, and ϕ has a Normal prior truncated to $(-1, 1)$ to ensure stationarity, with prior parameters that depend on the volatility path $\{h_t\}$.

A.2 Hierarchical estimation

The DFM is estimated top-down across the hierarchy levels of the tree $\mathcal{G}$ (Figure 2). At the root level $i = 1$, $\boldsymbol{F}_{1,t}^{\text{anc}}$ is empty, so the node-specific components ($\mathcal{T}_{1,t}$, $\mathcal{S}_{1,1,t}$, $\mathcal{S}_{1,2,t}$, $\mathcal{S}_{1,3,t}$, $c_{1,t}$) are estimated via MCMC directly on $Y_{1,1:T}$. For each subsequent level, the estimation proceeds as follows:

1. **Ancestor loading estimation**: The ancestor factors $\boldsymbol{F}_{i,t}^{\text{anc}} = \{\mathcal{T}_{j,t}, \mathcal{S}_{j,1,t} : j \text{ is an ancestor of } i\}$ (trend and annual seasonality) are collected, and the loadings $\boldsymbol{\beta}_i^{\text{anc}}$ are estimated via OLS regression of $Y_{i,t}$ on $\boldsymbol{F}_{i,t}^{\text{anc}}$.

2. **Orthogonalization**: The ancestor contribution is removed:

$$\tilde{Y}_{i,t} = Y_{i,t} - \boldsymbol{\beta}_i^{\text{anc}} \boldsymbol{F}_{i,t}^{\text{anc}}.$$

3. **Series-specific estimation**: The series-specific factors $\mathcal{T}_{i,t}, \mathcal{S}_{i,1,t}, \mathcal{S}_{i,2,t}, \mathcal{S}_{i,3,t}, c_{i,t}$ are estimated via MCMC on the orthogonalized residuals $\tilde{Y}_{i,1:T}$.

The final forecast is reconstructed by combining the ancestor contribution with the node-specific components as in (3). This procedure ensures that node-specific factors capture global variation not already explained by ancestor factors.

We note that while the ancestor factors include non-stationary trend components, the hierarchical structure of our data implies a cointegrating relationship between each series and its ancestor factors, which we have validated empirically via Johansen's trace test [Johansen, 1991]. This ensures the OLS estimates of $\boldsymbol{\beta}_i^{\text{anc}}$ are consistent [Stock, 1987], and the regression is not spurious despite the non-stationarity of both the regressors and the node-specific components absorbed into the error term.

B FA-TSFM input transformations

This appendix formalizes the input transformations referenced in Step 2 of the FA-TSFM pipeline described in Section 3.1. We consider two transformations: residualization, which replaces the TSFM's target with DFM residuals, and normalization, which standardizes the target to zero mean and unit variance. Note that all reported results use normalization, and we only include the unnormalized definitions here for clarity.

Let $\hat{Y}_{i,t}^{\mathrm{DFM}}$ denote the DFM's forecasts for series i and $\hat{\varepsilon}_{i,t} := Y_{i,t} - \hat{Y}_{i,t}^{\mathrm{DFM}}$ denote its residuals.

- **Residualize = False, Normalize = False** (not reported): $\xi_{i,t} = Y_{i,t}$. The TSFM forecasts the original series directly:

$$\hat{Y}_{i,T+1:T+H} = \mathrm{TSFM}(Y_{i,1:T}, \mathcal{F}_{i,1:T+H}).$$

- **Residualize = False, Normalize = True**: $\xi_{i,t} = Z_{i,t}$. The original series is standardized before being passed to the TSFM, and the output is unnormalized to recover forecasts in the original scale:

$$Z_{i,t} := \frac{Y_{i,t} - \mu_Y}{\sigma_Y}, \quad \mu_Y := \mathrm{Mean}(Y_{i,1:T}), \quad \sigma_Y := \mathrm{Std}(Y_{i,1:T}),$$
$$\hat{Y}_{i,T+1:T+H} = \mu_Y + \sigma_Y \cdot \mathrm{TSFM}(Z_{i,1:T}, \mathcal{F}_{i,1:T+H}).$$

- **Residualize = True, Normalize = False** (not reported): $\xi_{i,t} = \hat{\varepsilon}_{i,t}$. The TSFM forecasts the DFM residuals, and the DFM forecast is added back to produce the final forecast:

$$\hat{Y}_{i,T+1:T+H} = \hat{Y}_{i,T+1:T+H}^{\mathrm{DFM}} + \mathrm{TSFM}(\hat{\varepsilon}_{i,1:T}, \mathcal{F}_{i,1:T+H}).$$

- **Residualize = True, Normalize = True**: $\xi_{i,t} = Z_{i,t}^{\varepsilon}$. The DFM residuals are standard-

ized before being passed to the TSFM. The output is first unnormalized, then the DFM forecast is added back:

$$Z^{\varepsilon}_{i,t} := \frac{\hat{\varepsilon}_{i,t} - \mu_{\varepsilon}}{\sigma_{\varepsilon}}, \quad \mu_{\varepsilon} := \mathrm{Mean}(\hat{\varepsilon}_{i,1:T}), \quad \sigma_{\varepsilon} := \mathrm{Std}(\hat{\varepsilon}_{i,1:T}),$$
$$\hat{Y}_{i,T+1:T+H} = \hat{Y}^{\mathrm{DFM}}_{i,T+1:T+H} + \mu_{\varepsilon} + \sigma_{\varepsilon} \cdot \mathrm{TSFM}(Z^{\varepsilon}_{i,1:T}, \mathcal{F}_{i,1:T+H}).$$

All normalization statistics are computed on the training window $t = 1, \ldots, T$.

We do not scale the factor covariates in $\mathcal{F}_{i,1:T+H}$ before passing them to the TSFM, as the relative magnitudes across factor types may carry information about their respective contributions. Preliminary experiments with learned per-factor scaling weights (multiplying each factor $F_{r,t} \in \mathcal{F}_{i,t}$ by a learned scalar w_r before input) suggest that the optimal scaling is node- and factor-dependent. For example, weekly seasonality is consistently amplified while event seasonality is downweighted for some nodes. These learned weights yield modest out-of-sample RMSE improvements (2–8% on early held-out dates), though the gains decay over time, suggesting periodic re-optimization may be needed. It is worth noting that the learned weights in our experiments do not align with naive factor normalization (zero mean and unit variance), though normalizing factors before learning residual weights could improve upon direct weight optimization. We leave systematic exploration of factor scaling and learned factor representations to future work.

C Experimental details

C.1 Compute resources

DFM estimation was performed on AWS Batch using 4 vCPUs and 8 GB RAM per job, with one job per (node, estimation end date) combination. Jobs were run level-by-level (lower levels blocked until ancestor levels completed), with individual jobs taking 10 minutes to 2 hours depending on the series. Across 155 nodes and 17 estimation end dates, this

amounts to 2635 jobs. With AWS Batch parallelization across nodes within each level, the full DFM backtest (all series and estimation end dates) completes in approximately 6 to 8 hours wall-clock time.

TSFM inference was conducted via Amazon SageMaker endpoints on ml.g5.2xlarge instances (NVIDIA A10G GPU, 24 GB VRAM). Each run corresponds to one covariate configuration and submits a payload per (node, estimation end date), with both residualization strategies computed simultaneously. A single run over all nodes and estimation end dates completes within 30 minutes to 2 hours depending on the model and number of covariates. Chronos-2 forward passes for embedding and attention weight extraction for mechanistic interpretability analyses were run on an m6a.12xlarge instance (48 vCPUs, 192 GB RAM).

The full research project, including preliminary experiments, hyperparameter exploration, and runs not reported in the paper (failed runs, debugging, etc.), required approximately 2 to 3 times the compute of the final reported experiments.

C.2 TSFM model versions

All TSFMs used in our experiments are zero-shot with no fine-tuning or hyperparameter modifications beyond those described in Section 2.

- **Chronos-2** [Ansari et al., 2025]: v1.1.0 (base), accessed via Amazon SageMaker JumpStart (Apache 2.0 license).
- **ApolloPFN** [Potapczynski et al., 2026]: Pre-release version made available by the model developers.
- **Moirai 1.0** [Woo et al., 2024]: `Salesforce/moirai-1.0-R-large` from HuggingFace (Apache 2.0 license).
- **TabPFN-TS** [Hoo et al., 2025]: `tabpfn-time-series` v1.0.9 (MIT license).

TSFM constraints are described in Table 1. Specifically, Moirai 1.0 imposes a joint token budget $(\lceil W/P \rceil + \lceil H/P \rceil) \cdot V \leq 512$, where P is the patch size and V the number of variates.

To ensure sufficient context history without sacrificing temporal resolution through larger patch sizes, we set $P = 32$ and $H = 128$. Additionally, we fix $W = 1120$ for Moirai 1.0 across all covariate configurations to avoid accuracy confounding due to differences in context length.

C.3 Backtest setup

Since standard cross-validation is not applicable to time series, we evaluate all models via backtesting. For each estimation end date $T \in \mathcal{E}$, the model receives observations $Y_{i,1:T} := [Y_{i,1}, \ldots, Y_{i,T}]$ and produces forecasts for horizons $h = 1, \ldots, H$ beyond T. Our backtest spans $|\mathcal{E}| = 17$ monthly-spaced estimation end dates, with the earliest at $T = 2222$ and the latest at $T = 2708$ out of 3006 total observed timesteps. For the DFM, Chronos-2, and TabPFN-TS, we generate forecasts up to 1000 steps ahead but evaluate only up to $H = 298$, the maximum number of horizons for which actuals are available beyond the latest estimation end date. For Moirai 1.0 and ApolloPFN, we forecast and evaluate 128 steps ahead due to their horizon constraints (Table 1).

To ensure fair comparison, all TSFM forecasts are based on the same set of DFM backtest results, with any (node, estimation end date, horizon) combination excluded across all model configurations if missing from any one of them. After combining all results into a unified table indexed by (model, residualization, covariate configuration, estimation end date, horizon, node), 1.41% of rows were dropped due to missing actuals and 0.32% due to missing forecasts, out of 27,967,700 total rows prior to cleaning.

D Evaluation details

D.1 Forecast accuracy metrics

The metrics MAPE and RMSE are calculated by aggregating across the 17 estimation end dates $\mathcal{E}$, horizons in a given bucket $\mathcal{H} \subseteq \{1, ..., H = 298\}$, and all nodes in a given granularity

$\mathcal{I}$:

$$\mathrm{MAPE}_{\mathcal{I}} := \frac{100}{17|\mathcal{I}||\mathcal{H}|} \sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{E}} \sum_{h\in\mathcal{H}} \left| \frac{Y_{i,T+h} - \hat{Y}_{i,T+h|T}}{Y_{i,T+h}} \right|,$$
$$\mathrm{RMSE}_{\mathcal{I}} := \sqrt{\frac{1}{17|\mathcal{I}||\mathcal{H}|} \sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{E}} \sum_{h\in\mathcal{H}} \left(Y_{i,T+h} - \hat{Y}_{i,T+h|T} \right)^2},$$

where the conditional notation "$|T$" in $\hat{Y}_{i,T+h|T}$ indicates that the last date of actuals the model receives is T. Some summands may be excluded due to missing actuals or invalid forecasts, in which case the normalization constant $17|\mathcal{I}||\mathcal{H}|$ is adjusted accordingly.

Note that the MAPE values reported in our tables are percentages (as indicated by the factor of 100 in the formula); the relatively small magnitudes reflect the scale of the underlying data. In our application, even small percentage errors can have significant downstream impact on planning decisions.

D.2 Diebold-Mariano tests

We provide details on the statistical significance tests reported in Table 3. Let $\hat{Y}^{(1)}_{i,T+h|T}$ and $\hat{Y}^{(2)}_{i,T+h|T}$ denote h-step-ahead forecasts from two models for series i, and define the loss differential

$$\Delta_{i,T} := L\big(\hat{Y}^{(1)}_{i,T+h|T}, Y_{i,T+h}\big) - L\big(\hat{Y}^{(2)}_{i,T+h|T}, Y_{i,T+h}\big),$$

where L is the squared error loss. For a fixed horizon h, let $\mathcal{I}_h \subseteq \mathcal{I}$ denote the subset of series in grain $\mathcal{I}$ without missing or invalid values at horizon h. Collecting loss differentials across these series yields a $|\mathcal{I}_h|$-dimensional vector $\Delta_T = [\Delta_{i,T}]^\top_{i\in\mathcal{I}_h}$ observed across $|\mathcal{E}| = 17$ backtest origins.

One-sided multivariate DM test In a multivariate setting, "one model is better than another" requires careful definition. Following Jing and Jang [2025], we test whether Model

2 (e.g., FA-Chronos-2) is superior to Model 1 (e.g., the base DFM) in at least one series without being worse in any, using a two-step procedure. The first step tests $H_0 : \mu \leq \mathbf{0}_{|\mathcal{I}_h|}$ against $H_a : \exists i$ such that $\mu_i > 0$ (Model 1 is significantly worse for some series $i \in \mathcal{I}_h$), where $\mu = \mathbb{E}[\Delta_T]$. The second step reverses the model assignment to test whether Model 2 is significantly worse for any series. If the first test rejects but the second does not, we have confidence that Model 2 is superior in the sense that it significantly improves in at least one series while not significantly degrading any other series. The "%FA-Chronos-2 ¿ Baseline" columns in Table 3 correspond to tests from the first step, while the "%Baseline ¿ FA-Chronos-2" columns correspond to tests from the second step.

The test statistic is

$$\pi_T := a_T(T-1) \sum_{i \in \mathcal{I}_h} w_i \max(\bar{\Delta}_{i,T},\, 0)^2,$$

where $\bar{\Delta}_{i,T}$ is the sample mean of the i-th component, $\{w_i\}_{i \in \mathcal{I}_h}$ are the per-series weights, and $a_T = (T-1-2q+q(q+1)/T)/T$ is the Harvey-Leybourne-Newbold (HLN) finite-sample correction factor [Harvey et al., 1997] for order-q vector moving average loss differentials.

We report results using inverse marginal variance weights $w_i = 1/\hat{\Sigma}_{T,ii}$, where $\hat{\Sigma}_T$ is the heteroskedasticity and autocorrelation consistent (HAC) covariance estimator [Newey and West, 1987, Andrews, 1991]. The p-value is computed via Monte Carlo sampling: B samples are drawn from the multivariate t-distribution $t_{|\mathcal{I}_h|}(T-1, \mathbf{0}, \hat{\Sigma}_T)$[1], the test statistic is computed for each sample, and the p-value is the fraction of Monte Carlo statistics exceeding π_T.

Implementation details Standard DM theory prescribes $q = h - 1$ for h-step-ahead forecasts to account for the MA structure of overlapping forecast errors. However, our 17 backtest origins are spaced approximately one month apart, so for short horizons ($h \leq 30$),

[1]Note that the scale matrix of the reference t-distribution is $\hat{\Sigma}_T$ rather than $\hat{\Sigma}_T/T$: the Monte Carlo draws represent the approximate distribution of $\sqrt{T-1}\,\hat{\Sigma}_T^{-1/2}\bar{\Delta}_T$ under the null, and the factor of T is accounted for by the HLN correction a_T in the observed test statistic.

consecutive forecast errors have limited temporal overlap. For longer horizons ($h > 30$), forecast windows from consecutive origins do overlap, introducing autocorrelation in the loss differentials that $q = 0$ does not account for. Nevertheless, the small temporal sample size ($T = 17$) makes higher-lag HAC estimators unreliable, so we set $q = 0$ throughout.

We acknowledge that this choice may underestimate the variance of loss differentials at longer horizons, which could inflate rejection rates. Since the test statistic, reference distribution, and HAC estimator are constructed identically for both the direct test (FA-Chronos-2 better in some series) and the reverse test (baseline better in some series), any such inflation applies equally to both directions. This means that while the absolute rejection percentages at longer horizons should be interpreted with caution, the relative comparison between direct and reverse rejection rates (which is the basis for our conclusions) is unlikely to be distorted by this choice.

The multivariate DM test p-values are computed via Monte Carlo with $B = 1000$ samples drawn from $t_D(T-1, \mathbf{0}, \hat{\Sigma}_T)$. For each comparison (FA-Chronos-2 versus a baseline) and grain, the test produces one p-value p_h per horizon $h \in \{1, \ldots, H = 298\}$. The percentages reported in Table 3 are

$$\frac{100}{H} \sum_{h=1}^{H} \mathbf{1}\{p_h < \alpha\},$$

where p_h denotes the first-step p-value (FA-Chronos-2 significantly better in some series) for columns under "% FA-Chronos-2 > Baseline," and the second-step p-value (baseline significantly better in some series) for columns under "% Baseline > FA-Chronos-2."

For completeness, we also report per-series univariate DM-HLN test Diebold and Mariano [1995], Harvey et al. [1997] results in Appendix E, where the test statistic

$$t_i = \frac{\bar{\Delta}_{i,T}}{\sqrt{\hat{\sigma}_i^2/(T \cdot a_T)}}$$

is compared against the Student's t distribution with $T - 1$ degrees of freedom under the

one-sided null $H_0 : \mu_i \leq 0$.

For each comparison, grain $\mathcal{I}$, and horizon h, let $\mathcal{I}_h \subseteq \mathcal{I}$ denote the subset of series in grain $\mathcal{I}$ without missing or invalid values. The univariate test produces one p-value $p_{i,h}$ per series $i \in \mathcal{I}_h$. The percentages reported in Appendix E Table 6 are

$$\frac{100}{\sum_{h=1}^{H} |\mathcal{I}_h|} \sum_{h=1}^{H} \sum_{i \in \mathcal{I}_h} \mathbf{1}\{p_{i,h} < \alpha\},$$

i.e., the fraction of (series, horizon) pairs that reject at $\alpha = 0.1$. As with the multivariate test, $p_{i,h}$ denotes the first-step p-value for columns under "% FA-Chronos-2 > Baseline" and the second-step p-value for columns under "% Baseline > FA-Chronos-2."

E Additional Diebold-Mariano test results

Table 6 reports the analogous results using per-series univariate DM-HLN tests, aggregated as the percentage of (series, horizon) pairs that reject at $\alpha = 0.1$. The overall story is consistent with the multivariate results in Table 3: under the recommended residualization strategy (bolded columns), FA-Chronos-2 significantly improves a considerable fraction of (series, horizon) pairs while rarely degrading them. However, the univariate rejection rates are generally lower for the direct test and sometimes higher for the reverse test. This is expected for two reasons. First, the multivariate test rejects for an entire grain if *any* series shows significant improvement, whereas the univariate test requires each series to be individually significant. Second, the multivariate test statistic is a chi-square type statistic that accumulates small contributions across series and can reject when many series show modest improvements that are individually non-significant.

Table 6: Percentage of (series, horizon) pairs where the univariate DM test rejects at $\alpha = 0.1$.

		% FA-Chronos-2 ¿ Baseline				% Baseline ¿ FA-Chronos-2			
		Coarse		Fine		Coarse		Fine	
Baseline		Non-resid	**Resid**	**Non-resid**	Resid	Non-resid	**Resid**	**Non-resid**	Resid
Non-resid	Raw	23.2%	27.0%	21.3%	18.6%	2.4%	3.9%	8.0%	15.0%
	EI	38.8%	36.6%	28.3%	21.3%	2.1%	4.5%	6.6%	14.4%
	Anc	42.7%	40.2%	30.4%	21.5%	1.3%	4.1%	7.2%	14.1%
Resid	Raw	13.1%	19.7%	24.9%	19.0%	20.0%	5.0%	12.9%	15.0%
	EI	13.8%	32.4%	28.6%	34.3%	20.3%	3.9%	13.3%	8.8%
	Anc	12.3%	18.1%	25.2%	21.2%	21.1%	5.5%	13.8%	12.6%
DFM		15.3%	38.5%	43.2%	49.1%	18.4%	1.6%	6.9%	5.0%

F Predictive interval coverage and width of non-residualized TSFMs

Most TSFMs produce quantile forecasts, which naturally yield prediction intervals, though the underlying mechanisms vary. Chronos-2 [Ansari et al., 2025] is trained on aggregates of the quantile regression loss

$$q \cdot \max\left(Y - \hat{Y}^{(q)}, 0\right) + (1 - q) \cdot \max\left(\hat{Y}^{(q)} - Y, 0\right),$$

where Y is the observed value and $\hat{Y}^{(q)}$ the predicted q-th quantile. TabPFN-TS [Hoo et al., 2025] and ApolloPFN [Potapczynski et al., 2026] output approximate posterior predictive distributions from which quantiles are derived, while Moirai 1.0 [Woo et al., 2024] generates forecast trajectory samples and quantiles are taken empirically. DFMs also produce natural prediction intervals as quantiles of posterior draws. However, for residualized TSFMs, combining the DFM's posterior intervals with the TSFM's quantile forecasts of residuals is not straightforward, as the DFM forecasts and residuals are generally not independent. For non-residualized TSFMs, this issue does not arise: the TSFM's quantile forecasts can be

used directly as prediction intervals. We therefore evaluate prediction interval quality for non-residualized configurations only.

Table 7: Coverage and average width of prediction intervals of non-residualized TSFMs.

		Coarse grain		Fine grain	
TSFM	Config	Coverage	Width	Coverage	Width
Chronos-2	Raw	83.32%	0.1871	78.97%	0.3804
	EI	87.70%	0.2106	83.35%	0.4303
	Anc	85.95%	0.2066	81.40%	0.4049
	FA	81.23%	0.1493	82.91%	0.3984
TabPFN-TS	Raw	97.20%	2.3877	92.29%	2.2828
	EI	98.58%	1.6254	93.01%	1.9066
	FA	98.83%	0.4418	92.88%	0.9936
Moirai 1.0	Raw	90.94%	0.3436	85.25%	0.4863
	EI	84.70%	0.3287	74.54%	0.5198
	Anc	86.03%	0.3573	74.73%	0.5400
	FA	88.06%	0.3839	76.32%	0.5694
ApolloPFN	Raw	94.86%	0.3307	89.06%	0.5037
	EI	96.09%	0.3504	89.12%	0.5384
	FA	87.16%	0.1996	82.17%	0.3969

Table 7 reports the coverage and average width of 80% prediction intervals (10th and 90th percentiles) for non-residualized TSFM configurations. For Chronos-2 at coarse grain, TabPFN-TS, and ApolloPFN, FA-TSFM substantially narrows prediction intervals while maintaining or exceeding the target 80% coverage, indicating that factor information reduces forecast uncertainty. For example, FA-TabPFN-TS achieves 98.83% coverage at coarse grain with an interval width of 0.44, compared to 97.20% and 2.39 for Raw TabPFN-TS — a fivefold reduction in width. At fine grain, FA-Chronos-2 slightly improves coverage over Raw Chronos-2 (82.91% vs 78.97%) with a comparable interval width. Moirai 1.0 again shows degradation when covariates are added, consistent with the accuracy results in Section 4 (Table 2).

G Factor contribution under residualization

In this section, we present Table 8, the residualized counterpart to Table 4. The overall pattern is broadly consistent: the default specification combining all factor types remains

the best or near-best configuration, and $\mathcal{S}_{3,t}$ is generally the strongest individual factor. However, several differences are worth noting.

Table 8: Impact of factors types on residualized FA-Chronos-2 forecast accuracy.

			Baselines			Trend	Seasonality			Default
Grain	Horizon	Metrics	Raw	EI	Anc	$\mathcal{T}_t$	$\mathcal{S}_{1,t}$	$\mathcal{S}_{2,t}$	$\mathcal{S}_{3,t}$	$\mathcal{T}_t$, $\mathcal{S}_{1:3,t}$
Coarse	≤ 128	MAPE	0.2348	0.2390	0.2343	0.2385	0.2337	0.2350	0.2318	**0.2313**
		RMSE	0.0705	0.0708	0.0702	0.0709	0.0703	0.0699	0.0698	**0.0696**
	> 128	MAPE	0.2974	0.2999	0.2931	0.2995	0.2965	0.2939	0.2985	**0.2903**
		RMSE	0.0866	0.0869	0.0860	0.0869	0.0864	**0.0848**	0.0862	0.0849
Fine	≤ 128	MAPE	1.5372	1.5371	1.5256	1.5680	1.5152	1.5332	1.5286	**1.5036**
		RMSE	0.3463	0.3240	0.3295	0.3510	0.3462	0.3503	**0.3033**	0.3075
	> 128	MAPE	1.9125	1.8888	1.8746	1.9174	1.8479	1.8555	1.8682	**1.8249**
		RMSE	0.3727	0.3688	0.3699	0.3758	0.3709	0.3694	**0.3195**	0.3252

First, the margins between configurations are substantially compressed. Under residualization, the TSFM forecasts residual variation after the linear effect of all factors has been subtracted via the DFM's observation equation (3). If a factor's primary contribution is well-captured by this linear structure, its additional value as a covariate is diminished. This is particularly notable for $\mathcal{S}_{3,t}$, whose large non-residualized gains are substantially diminished, suggesting that much of its contribution is already captured by the DFM's linear structure.

Second, the default specification still outperforms individual factors under residualization, though by a smaller margin and less consistently. Since the linear effects of individual factors are already accounted for through residualization, the remaining gains from combining all factors likely reflect interaction effects not captured by the DFM's linear additive structure.

Third, $\mathcal{S}_{3,t}$ alone achieves the best fine-grain RMSE, slightly outperforming the default. This is attributable to a forecast defect correction mechanism similar to that discussed in Section 4: event information helps the TSFM correct outlier errors from the less well-tuned fine-grain DFM, and RMSE is particularly sensitive to such corrections. Adding non-event factors on top may introduce slight noise that offsets this benefit in RMSE, even as it improves MAPE.

H Mechanistic interpretability analysis details and additional results

As described in Section 2, we focus on two internal quantities of Chronos-2 for our mechanistic interpretability analyses:

- **Final forecast embeddings**: Output of the transformer stack that gets passed through a residual block to produce multi-step quantile forecasts;
- **Group attention weights**: Softmax-normalized attention scores from the group attention layers, which determine how much each variate (target or covariate in our setting) attends to every other variate at each patch position.

H.1 Activation differencing

Activation differencing refers to the technique of comparing a model's internal representations under two different conditions by taking the difference of their activations at corresponding layers. This idea is closely related to *model diffing* [evhub AI Alignment Forum, 2019, Lindsey et al., 2024], which aims to understand how models differ mechanistically — analogous to reviewing software in terms of incremental diffs. Model diffing was originally conceived for comparing finetuned models against their base counterparts to isolate changes introduced during training [evhub AI Alignment Forum, 2019], and has since been operationalized through techniques such as crosscoders, which extract shared and model-specific features from concatenated activations [Lindsey et al., 2024], and diff-SAEs, which train sparse autoencoders directly on activation differences to more precisely isolate the relevant changes [Aranguri et al., 2024].

Setup In our setting, we apply activation differencing as a special case: rather than diffing across two different models, we compare activations from the same Chronos-2 model under different covariate input configurations to understand how each configuration alters the

model's internal representations. Our target of comparisons is the *final forecast embeddings* produced by the last layer of Chronos-2's transformer stack. We stack the embeddings across all series and estimation end dates, which results in an array $\mathbf{E}$ with shape `(num_series =` N`, est_end_dates =` $|\mathcal{E}|$ `= 17, output_patches = 63, embed_dim = 768)`. The activation difference between two configurations is then

$$\mathbf{D}^{\text{config1,config2}} := \mathbf{E}^{\text{config1}} - \mathbf{E}^{\text{config2}},$$

which is an array with the same shape as the embeddings. We choose final forecast embeddings $\mathbf{E}$ as our target for two reasons. First, they are the terminal output of the transformer stack from which each series' forecast is directly generated, meaning all covariate information is summarized therein. Second, unlike attention weights, they share the same shape regardless of input length or number of covariates, enabling direct comparison across configurations.

Visualization The high dimensionality of the activation differences poses a challenge for interpretability. To simplify the analysis, we first average across nodes, reducing the array to shape `(est_end_dates, output_patches, embed_dim)`. Figure 7 visualizes the activation differences $\mathbf{D}^{\text{config, Raw}}$ for non-residualized configurations EI (event indicators), $\mathcal{S}_3$ (event factor), $\mathcal{S}_{1:3}$ (seasonality factors), and FA ($\mathcal{T}$, $\mathcal{S}_{1:3}$). For each fixed estimation end date T, the embeddings are averaged across all series. We retain a fixed estimation end date rather than averaging across dates because the vertical banding structure is related to "event patches" that track calendar event positions, which shift relative to the forecast horizon as the estimation end date changes.

As can be seen, both EI (1st from left) and $\mathcal{S}_3$ (2nd) primarily introduce vertical bands at event-coincident patches, consistent with their role as event encodings. The seasonality (3rd) and FA (4th) configurations additionally show differences at non-event patches and across embedding features (horizontal bands), suggesting broader representational changes

beyond event localization. Beyond these qualitative observations, the high dimensionality of the activation differences limits what can be inferred from direct visualization alone.

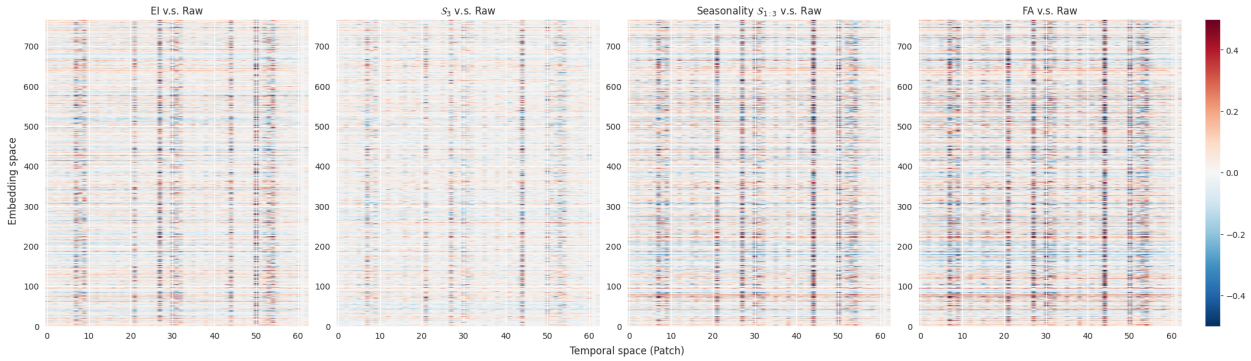

Figure 7: Activation differences against "Raw" (fixed estimation end date T).

PCA As the embedding dimension of 768 remains prohibitively large for direct inspection, we apply PCA along this axis to extract the dominant directions of variation. Each PC score has shape `(est_end_dates, output_patches)`, with samples indexed by the embedding dimension. Figure 4 (main paper) shows the top 5 PC scores of $\mathbf{D}^{\text{EI,Raw}}$ and $\mathbf{D}^{\mathcal{S}_3,\text{Raw}}$, addressing question a) from Section 5: how does $\mathcal{S}_{3,t}$ differ from EI? The main-paper discussion in Section 5 is not repeated here.

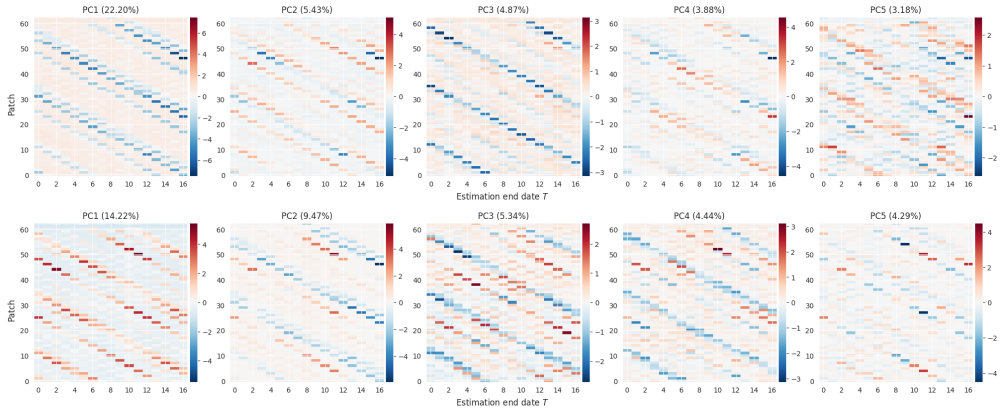

Figure 8: Top 5 PC scores of $\mathbf{D}^{\mathcal{S}_3,\text{ EI}}$ (top) and $\mathbf{D}^{\text{FA, EI}}$ (bottom).

Figure 8 extends this analysis to address question b): how do different factor types interact? We compare $\mathbf{D}^{\mathcal{S}_3,\text{EI}}$ and $\mathbf{D}^{\text{FA,EI}}$, isolating the effect of adding $\mathcal{S}_{3,t}$ alone versus

all factors on top of the EI baseline. The overlap pattern is sparser than in Figure 4: the strongest alignments are $(\mathrm{PC}_1^{\mathcal{S}_3}, \mathrm{PC}_2^{\mathrm{FA}})$ at 0.78 cosine similarity and $(\mathrm{PC}_2^{\mathcal{S}_3}, \mathrm{PC}_1^{\mathrm{FA}})$ at 0.61, with weaker matches among the remaining PCs. Notably, the top-loading feature identified in the analysis for question a) persists across multiple PCs in both subspaces, suggesting that the dominant event-encoding direction carries over from the $\mathcal{S}_3$-vs-EI comparison into the FA-vs-$\mathcal{S}_3$ comparison.

H.2 Variance decomposition via PCA projection

The use of PCA also provides a natural framework for comparing covariate configurations. The questions a) how $\mathcal{S}_{3,t}$ differs from EI, and b) how factor types interact can both be framed as evaluating the change in embeddings when switching from one configuration (old) to another (new), relative to a baseline that removes patterns common to both. By performing PCA on the activation differences $\mathbf{D}^{\text{old, base}}$ and $\mathbf{D}^{\text{new, base}}$ separately, we obtain subspaces that summarize how each configuration differs from baseline. We can then decompose the change $\mathbf{D}^{\text{new, old}}$ into components that align with the principal structure of one, both, or neither subspace.

Let $\mathcal{O} \equiv \mathcal{O}_k$ and $\mathcal{N} \equiv \mathcal{N}_k \subseteq \mathbb{R}^D$ be the subspaces spanned by the top k principal components of $\mathbf{D}^{\text{old, base}}$ and $\mathbf{D}^{\text{new, base}}$, respectively, where $D := \texttt{embed\_dim} = 768$. Intuitively, $\mathcal{O}$ and $\mathcal{N}$ are k-dimensional subspaces that capture the most variance in how the configurations (old and new) differ from the baseline. We are interested in decomposing the *total variance* of $\mathbf{D}^{\text{new, old}} = \mathbf{D}^{\text{new, base}} - \mathbf{D}^{\text{old, base}}$, defined as the trace of its covariance matrix $\Sigma := \mathbb{V}(\mathbf{D}^{\text{new, old}})$, into four components: 1) variance aligned with directions shared by both $\mathcal{O}$ and $\mathcal{N}$, 2) variance aligned with directions unique to $\mathcal{O}$, 3) variance aligned with directions unique to $\mathcal{N}$, and 4) variance outside both subspaces. To this end, we define the *explained variance ratio* of $\Sigma \in \mathbb{R}^{D \times D}$ by a subspace $\mathcal{S} \subseteq \mathbb{R}^D$ as the share of total variance retained

after orthogonal projection onto $\mathcal{S}$:

$$R(\mathcal{S}) := \frac{\mathrm{tr}(\mathbf{P}_{\mathcal{S}}\Sigma)}{\mathrm{tr}(\Sigma)}, \quad \text{where} \quad \mathbf{P}_{\mathcal{S}} \text{ is the orthogonal projection matrix onto } \mathcal{S},$$

and consider the four-way decomposition

$$\begin{aligned} R_{\text{shared}} &:= R(\mathcal{O}) + R(\mathcal{N}) - R(\mathcal{O}+\mathcal{N}), & R_{\text{outside}} &:= 1 - R(\mathcal{O}+\mathcal{N}), \\ R_{\text{old-unique}} &:= R(\mathcal{O}) - R_{\text{shared}}, & R_{\text{new-unique}} &:= R(\mathcal{N}) - R_{\text{shared}} \end{aligned} \tag{4}$$

based on projections into $\mathcal{O}$, $\mathcal{N}$, and $\mathcal{O}+\mathcal{N}$ (the span of the principal components of both $\mathbf{D}^{\text{old, base}}$ and $\mathbf{D}^{\text{new, base}}$). The decomposition in (4) sums up to 1 by construction and can be interpreted as follows:

- R_{shared}: The fraction of variance in $\mathbf{D}^{\text{new, old}}$ that is redundantly explained by the principal structure of both $\mathbf{D}^{\text{old, base}}$ and $\mathbf{D}^{\text{new, base}}$. This is the portion consistent with the *two configurations encoding similar structure relative to baseline*, but with different strengths across samples.[2]

- $R_{\text{old-unique}}$: The fraction of variance in $\mathbf{D}^{\text{new, old}}$ explained by the principal structure of $\mathbf{D}^{\text{old, base}}$ beyond what is shared with that of $\mathbf{D}^{\text{new, base}}$. This is the portion consistent with the *old configuration's representational structure* relative to baseline *being displaced* when switching to the new configuration.

- $R_{\text{new-unique}}$: The fraction of variance in $\mathbf{D}^{\text{new, old}}$ explained by the principal structure of $\mathbf{D}^{\text{new, base}}$ beyond what is shared with that of $\mathbf{D}^{\text{old, base}}$. This is the portion consistent with the *new configuration introducing representational structure* relative to baseline that is *absent in the old configuration.*

- R_{outside}: The fraction of variance in $\mathbf{D}^{\text{new, old}}$ not explained by the principal structure of

[2]We emphasize that R_{shared} measures variance that both subspaces can account for rather than variance in their geometric intersection $\mathcal{O} \cap \mathcal{N}$. In high-dimensional settings, two k-dimensional subspaces generically intersect only at the origin, yet can still exhibit substantial redundancy when they contain nearly-parallel directions. This redundancy is what R_{shared} captures.

either $\mathbf{D}^{\text{old, base}}$ or $\mathbf{D}^{\text{new, base}}$. This captures structure in the old-to-new change that is *not explained by* the principal structure of *either* configuration's difference from baseline.

We note that the decomposition depends on the truncation parameter k, which controls how many principal directions define each PC subspace. For small k's, $\mathcal{O}_k$ and $\mathcal{N}_k$ capture only the most dominant modes of $\mathbf{D}^{\text{old, base}}$ and $\mathbf{D}^{\text{new, base}}$, so $R_{\text{outside},k}$ is often large. As k increases, both subspaces expand toward the full column spaces of their respective activation difference matrices. Since $\mathbf{D}^{\text{new, old}} = \mathbf{D}^{\text{new, base}} - \mathbf{D}^{\text{old,base}}$, the support of Σ is contained in $\mathcal{O}_k + \mathcal{N}_k$ in the limit, and $R_{\text{outside},k} \to 0$. If each subspace individually also captures the full support of Σ (which occurs when the column spaces of $\mathbf{D}^{\text{old, base}}$ and $\mathbf{D}^{\text{new, base}}$ each contain the column space of $\mathbf{D}^{\text{new,old}}$), then $R(\mathcal{O}_k), R(\mathcal{N}_k) \to 1$, giving $R_{\text{shared},k} \to 1$ and $R_{\text{old-unique},k}, R_{\text{new-unique},k} \to 0$. The decomposition is therefore most informative at intermediate k.

We now apply this framework to try answer the two questions posed in Section 5, using Raw as the baseline for question a) and EI for question b). Figure 9 shows the four-way variance decomposition of $\mathbf{D}^{\mathcal{S}_3, \text{EI}}$ as a function of k (number of PCs), with R_{shared} (blue), $R_{\text{EI-unique}}$ (orange), $R_{\mathcal{S}_3\text{-unique}}$ (green), and R_{outside} (red). Across $k = 10$ to 150, $R_{\text{EI-unique}}$ is consistently roughly twice that of $R_{\mathcal{S}_3\text{-unique}}$ — a ratio that remains stable even as R_{shared} grows and R_{outside} tends to zero (both reference PCAs exceed 80% variance explained by $k = 50$ and 90% by $k = 90$). This asymmetry indicates that switching from EI to $\mathcal{S}_3$ displaces more EI-specific encoding than it introduces $\mathcal{S}_3$-specific encoding, suggesting that $\mathcal{S}_3$ may be achieving its advantage primarily by re-encoding the same event information rather than by adding substantial new representational structure.

Figure 10 shows the variance decomposition of $\mathbf{D}^{\text{FA},\mathcal{S}_3}$ with EI as baseline. As before, R_{shared} increases and $R_{\text{outside}} \to 0$ as k grows. However, $R_{\mathcal{S}_3\text{-unique}}$ and $R_{\text{FA-unique}}$ are now approximately equal across all k. This symmetric pattern indicates that adding $\mathcal{T}_t$, $\mathcal{S}_{1,t}$, and $\mathcal{S}_{2,t}$ displaces about as much $\mathcal{S}_3$-specific encoding as it introduces FA-specific encoding, a balanced substitution rather than the asymmetric displacement seen in Figure 9. This

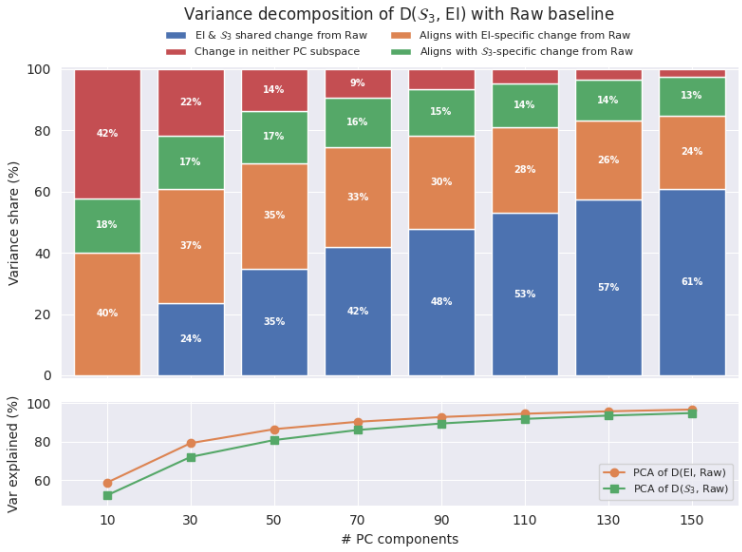

Figure 9: Variance decomposition with new = $\mathcal{S}_3$, old = EI, and baseline = Raw.

suggests the extra factors are reorganizing the embedding space rather than simply layering new information on top of $\mathcal{S}_3$.

For comparison, we repeat the same decomposition with Raw as baseline instead of EI in Figure 11. Under this setup, $R_{\text{FA-unique}}$ is roughly three times $R_{\mathcal{S}_3\text{-unique}}$. The asymmetry arises because the Raw baseline removes no event information: $\mathbf{D}^{\text{FA, Raw}}$ captures everything factor adds (trend, seasonality, *and* events), while $\mathbf{D}^{\mathcal{S}_3, \text{Raw}}$ captures only the event component. The FA-specific directions therefore include the trend and seasonality structure that $\mathcal{S}_3$ never encoded, inflating $R_{\text{FA-unique}}$.

H.3 Linear probing

Linear probing refers to training a linear classifier on a model's internal representations to predict some property of the input, with classification accuracy measuring how readily that property is encoded. Under the linear representation hypothesis [Park et al., 2024], which posits that neural networks encode meaningful features as linear directions or subspaces of their representation space, high probe accuracy indicates the model represents the property in linearly separable form.

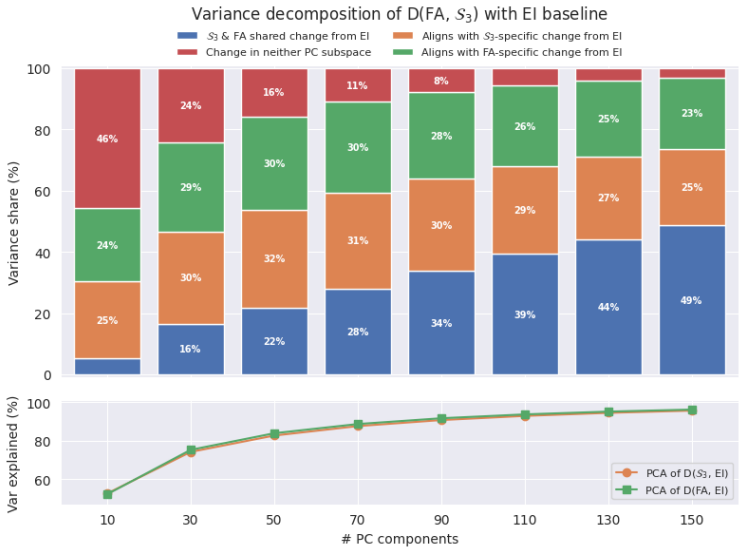

Figure 10: Variance decomposition with new = FA, old = $\mathcal{S}_3$, and baseline = EI.

The variance decomposition of activation differences characterizes where in embedding space the covariate configurations differ, but does not address whether these differences are structured enough to be decoded by a simple classifier. To investigate whether the final forecast embeddings $\mathbf{E}^1$ and $\mathbf{E}^2$ lie in linearly separable subspaces, we train per-patch probes to classify covariate configuration from forecast embeddings, with shuffled-label controls following [Hewitt and Liang, 2019] to separate genuine encoding from probe memorization. Our implementation is summarized in Algorithm 1.

Figure 12 shows that a simple logistic regression achieves near-perfect accuracy at every output patch for both $\mathcal{S}_3$ vs. EI and FA vs. $\mathcal{S}_3$, while the shuffled-label controls remain at chance ($\approx 50\%$). This indicates that covariate configuration identity is encoded in linearly accessible directions of the forecast embeddings, and that this information is accessible from the earliest forecast patch rather than being gradually built up over the horizon.

We note a caveat regarding the cross-validation setup. The probing samples are (series, estimation end date) pairs, and observations sharing the same series or date are not independent. Standard 5-fold CV randomly assigns pairs to folds, allowing the same series or date to appear in both train and test sets. However, because each (series, date) pair appears exactly

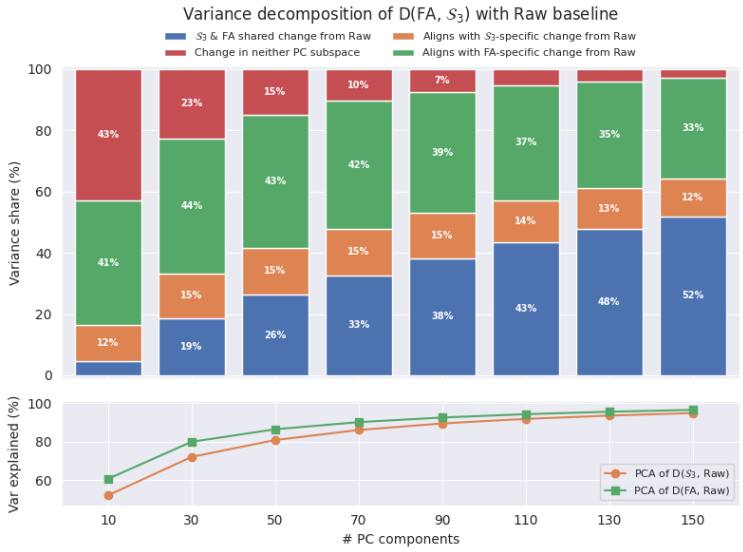

Figure 11: Variance decomposition with new = FA, old = $\mathcal{S}_3$, and baseline = Raw.

once under each configuration, the classification target is perfectly balanced at the pair level: memorizing the baseline representation for a specific (series, date) would predict the same class for both configurations, yielding 50% accuracy for that pair. To achieve near-perfect accuracy, the probe must learn directions that separate the configurations themselves, not exploit identity-level shortcuts. This, combined with the chance-level shuffled-label controls, suggests the high accuracy reflects genuine linear separability rather than information leakage. A grouped CV strategy (e.g., leave-one-date-out or leave-one-series-out) would nonetheless provide additional assurance of generalization.

Algorithm 1: Per-patch linear probing for covariate configuration

Input: Embedding matrices $\mathbf{E}^a, \mathbf{E}^b \in \mathbb{R}^{N\times|\mathcal{E}|\times P\times D}$ under configurations a and b, where $N = 155$ is the number of series, $|\mathcal{E}| = 17$ the number of estimation end dates, $P = 63$ the number of patches, and $D = 768$ the embedded feature dimension.

Output: Per-patch classification accuracy $\text{acc}(p)$ for $p = 1, \ldots, P$

$\mathbf{X} \leftarrow \text{Concatenate}(\mathbf{E}^a, \mathbf{E}^b) \in \mathbb{R}^{2N|\mathcal{E}|\times P\times D}$ along the node & estimation end date axes

Construct labels $\mathbf{Y} \leftarrow [0, ..., 0, 1, ..., 1]$ (each repeated $N|\mathcal{E}|$ times);

for $p = 1, \ldots, P$ **do**

  Fit logistic regression on samples $\{(\mathbf{X}_{i,p,:},\ Y_i)\}_{i=1}^{2N|\mathcal{E}|}$ with 5-fold CV;

  $\text{Accuracy}(p) \leftarrow$ mean CV accuracy;

Control: Repeat with $\mathbf{Y}$ randomly permuted;

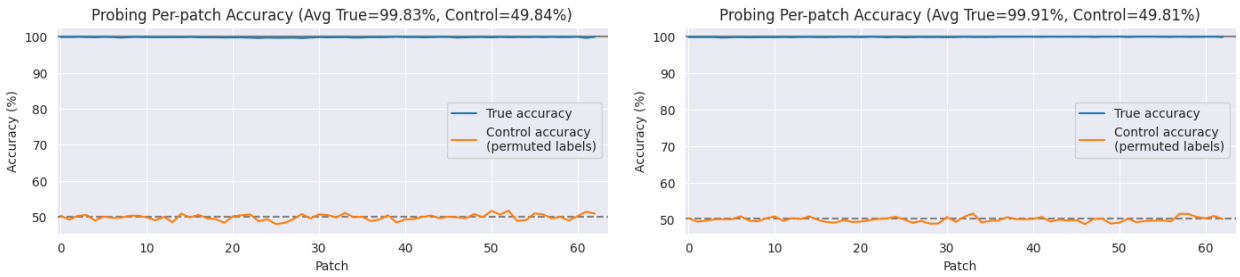

Figure 12: Linear probe accuracy by output patch: EI vs. $\mathcal{S}_3$ (left) & $\mathcal{S}_3$ vs. FA (right).

H.4 Group attention comparisons

In this subsection, we provide details about the group attention comparisons presented in Figure 5. At each group attention layer ℓ, head m, and patch position p, Chronos-2 computes attention across the V variates in a group. For each head, the attention weight of variate u on variate v is

$$w_{uv}^{(\ell)}(m,p) = \frac{\exp\left(\mathbf{q}_u^{(\ell)}(p)^\top \mathbf{k}_v^{(\ell)}(p)\right)}{\sum_{v'=1}^{V} \exp\left(\mathbf{q}_u^{(\ell)}(p)^\top \mathbf{k}_{v'}^{(\ell)}(p)\right)},$$

where $\mathbf{q}_u^{(\ell)}(p)$ and $\mathbf{k}_v^{(\ell)}(p)$ are query and key projections of the layer-normalized hidden states. Under our setting, each series i is forecast in an independent run, where it is placed at variate index 0 with its covariates at indices $1, \ldots, V-1$. Hence, the attention placed on covariate v for forecasting the target series at layer ℓ, head m, and patch p is $w_{0v}^{(\ell)}(m,p)$.

For each series i and estimation end date T, we extract the target's attention weights $w_{0v}^{(\ell)}(m,p)$ from the corresponding run. While these weights depend on i and T, we omit them from the notation until needed. The number of covariates (and thus variates) depends on the hierarchy level of the target series i. For instance, the level 0 root series $i=1$ would only get four covariates from factor augmentation $(\mathcal{T}_{1,t}, \mathcal{S}_{1,1,t}, \mathcal{S}_{1,2,t}, \mathcal{S}_{1,3,t})$, while a level 3 series would receive eleven (six from $\boldsymbol{F}_{i,t}^{\text{anc}} = \{\mathcal{T}_{j,t}, \mathcal{S}_{j,1,t} : j \text{ is an ancestor of } i\}$, $\mathcal{S}_{1,3,t}$, and the series i-specific $\mathcal{T}_{i,t}, \mathcal{S}_{i,1,t}, \mathcal{S}_{i,2,t}, \mathcal{S}_{i,3,t}$). To facilitate cross-level and cleaner comparisons, we aggregate attention weights by factor type. Let $\mathcal{V}_k$ denote the set of variate indices corresponding to

factor type $k \in \{\mathcal{T}, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3\}$ (e.g., $\mathcal{V}_\mathcal{T}$ denotes all trend factor covariates). Then,

$$\tilde{w}_k^{(\ell)}(m,p) := \sum_{v \in \mathcal{V}_k} w_{0v}^{(\ell)}(m,p)$$

is the total attention on factor type k for forecasting the target series. The series' self attention is preserved through $\tilde{w}_{\text{self}}^{(\ell)}(m,p) := w_{00}^{(\ell)}(m,p)$. Under full factor-augmentation, this yields five categories across which Chronos-2 allocates its group attention: $\mathcal{K} = \{\text{self}, \mathcal{T}, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3\}$. Two other set ups considered in Figure 5 have $\mathcal{K} = \{\text{self}, \mathcal{S}_3\}$ (left) and $\mathcal{K} = \{\text{self}, \mathcal{T}, \mathcal{S}_1, \mathcal{S}_2\}$ (right), respectively.

To produce the visualizations in Figure 5, we average across series i, layers ℓ, and heads m:

$$\mathbf{W}_\mathcal{K}[T,p,k] = \frac{1}{N \cdot 12 \cdot 12} \sum_{i=1}^{N} \sum_{\ell=1}^{12} \sum_{m=1}^{12} \tilde{w}_k^{(\ell,i,T)}(m,p),$$

where $\tilde{w}_k^{(\ell,i,T)}(m,p)$ denotes the total attention on $k \in \mathcal{K}$ for series i at estimation end date T. The resulting array $\mathbf{W}_\mathcal{K}$ has shape $(|\mathcal{E}| = 17, P = 63, |\mathcal{K}|)$. Figure 13 shows two examples of $\mathbf{W}_\mathcal{K}$, illustrating how attention is distributed across factor types under different covariate configurations.

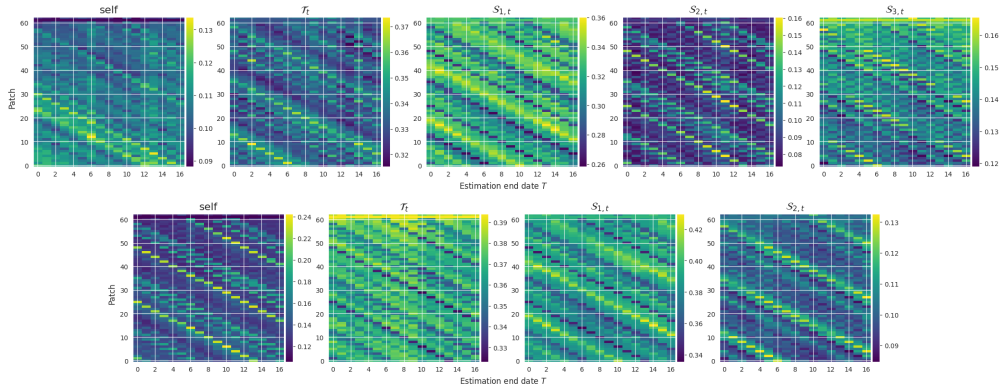

Figure 13: Aggregated group attention weights $\mathbf{W}_\mathcal{K}$ under full factor-augmentation (top; $\mathcal{K} = \{\text{self}, \mathcal{T}, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3\}$) and partial factor-augmentation without $\mathcal{S}_{3,t}$ (bottom; $\mathcal{K} = \{\text{self}, \mathcal{T}, \mathcal{S}_1, \mathcal{S}_2\}$).

Since the softmax-normalized attention weights sum to one across variates, we have by construction $\sum_{k\in\mathcal{K}} \mathbf{W}_{\mathcal{K}}[T,p,k] = 1$ for each estimation end date T and patch p. As a result, configurations with more covariates naturally allocate less attention per category, making it difficult to compare raw attention values across setups with different $\mathcal{K}$ or to visually identify how attention has shifted between configurations from the $\mathbf{W}_{\mathcal{K}}$ plots alone.

Figure 5 compares the full FA configuration $\mathcal{K} = \{\text{self}, \mathcal{T}, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3\}$ against a subset $\mathcal{K}' \subsetneq \mathcal{K}$: the left panel uses $\mathcal{K}' = \{\text{self}, \mathcal{S}_3\}$ and the right panel $\mathcal{K}' = \{\text{self}, \mathcal{T}, \mathcal{S}_1, \mathcal{S}_2\}$. To address the comparability issue, we normalize $\mathbf{W}_{\mathcal{K}}$ to sum to one over the shared categories $\mathcal{K}'$ before taking the difference:

$$\Delta_{\mathcal{K},\mathcal{K}'}[T,p,k] := \frac{\mathbf{W}_{\mathcal{K}}[T,p,k]}{\sum_{k'\in\mathcal{K}'} \mathbf{W}_{\mathcal{K}}[T,p,k']} - \mathbf{W}_{\mathcal{K}'}[T,p,k], \quad k \in \mathcal{K}'.$$

Each category $k \in \mathcal{K}'$ corresponds to a separate heatmap, with estimation end dates T on the x-axis and patches p on the y-axis. A positive value indicates that the full configuration allocates relatively more attention to category k, after controlling for the attention absorbed by the additional categories in $\mathcal{K} \setminus \mathcal{K}'$. Since both terms sum to one over $k \in \mathcal{K}'$, we have $\sum_{k\in\mathcal{K}'} \Delta_{\mathcal{K},\mathcal{K}'}[T,p,k] = 0$, meaning any relative increase in attention to one category must come at the expense of another within $\mathcal{K}'$.

I Additional dataset results

We evaluate FA-Chronos-2 on eight additional proprietary supply chain datasets to assess generalization beyond the main dataset. Each dataset shares the same four-level tree hierarchy structure, with 4 coarse-grain series (levels 0 and 1) across all datasets and between 113 and 128 fine-grain series (levels 2 and 3), as detailed in Table 9. The DFM priors were tuned on the main dataset (where coarse-grain series received careful tuning based on domain knowledge) and applied as-is to these additional datasets without further adjustment. As with the main dataset, coarse-grain series are generally more well-behaved, while fine-grain

series exhibit greater variability.

Table 9: Number of series per grain in each additional dataset.

Grain	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8
Coarse	4	4	4	4	4	4	4	4
Fine	128	124	120	120	115	118	113	123

At coarse grain (Table 10), FA-Chronos-2 (in either residualization strategy) achieves the best or near-best result in nearly every dataset, horizon, and metric, with the preferred strategy varying by dataset. Non-residualized Raw is the worst or near-worst configuration in almost all datasets. EI and Anc improve over Raw but are consistently outperformed by FA, reinforcing the finding that factor covariates are more informative than naive covariates. The optimal residualization strategy is less clear-cut than in the main dataset: residualized FA tends to perform better at short horizons, while non-residualized FA is more often best at long horizons, likely reflecting the lack of dataset-specific DFM tuning.

At fine grain (Table 11), the results are unambiguous: non-residualized FA-Chronos-2 achieves the best performance in nearly every dataset, horizon, and metric. Among non-residualized baselines, EI and Anc improve over Raw, though all are consistently behind FA. This mirrors the main dataset finding in Section 4 that factor covariates provide more value than naive event dummies or raw ancestor series. Residualized configurations generally suffer from large RMSE, again consistent with the forecast defect propagation discussed in Section 4: when the DFM is poorly tuned, its errors carry through residualization and are difficult for Chronos-2 to correct. Nevertheless, residualized FA-Chronos-2 noticeably reduces RMSE compared to residualized Raw for all eight datasets, indicating that factor covariates help mitigate defects even under residualization. This is relevant for practitioners who accept some accuracy loss in exchange for the controllability benefits of residualization (Section 6.2): in such cases, passing factors is preferable to not. The untuned DFM is worst across the board, yet its factors still yield substantial gains when passed to Chronos-2, demonstrating that factor information retains value even from a suboptimal DFM.

Table 10: Forecast accuracy metrics of naive Chronos-2 ("Not residualized - Raw, EI, Anc"), factor-informed Chronos-2 ("Not residualized - FA" and all "Residualized" columns), and the base DFM for coarse-grain series in additional datasets.

				Not residualized				Residualized				DFM
Grain	Horizon	Dataset	Metric	Raw	EI	Anc	**FA**	**Raw**	**EI**	**Anc**	**FA**	
Coarse	≤ 128	Dataset 1	MAPE	0.6456	0.6634	0.6396	**0.5365**	0.5642	0.5661	0.5585	0.5450	0.6000
			RMSE	0.1607	0.1331	0.1288	**0.1120**	0.1157	0.1156	0.1144	0.1125	0.1206
		Dataset 2	MAPE	0.5341	0.4926	0.4884	0.3930	0.3991	0.3926	0.3933	**0.3807**	0.4001
			RMSE	0.1454	0.1150	0.1140	0.0914	0.0932	0.0918	0.0918	**0.0903**	0.0935
		Dataset 3	MAPE	0.6257	0.5936	0.5939	0.4999	0.4789	0.5146	0.4965	**0.4746**	0.5793
			RMSE	0.1470	0.1208	0.1211	0.0998	0.1004	0.1034	0.1013	**0.0984**	0.1116
		Dataset 4	MAPE	0.4625	0.4832	0.4877	0.4028	**0.3834**	0.4456	0.4343	0.4022	0.5719
			RMSE	0.0929	0.0868	0.0876	0.0719	**0.0709**	0.0808	0.0786	0.0729	0.1012
		Dataset 5	MAPE	0.5639	0.5551	0.5468	**0.4309**	0.4981	0.5142	0.4951	0.4559	0.5301
			RMSE	0.1578	0.1489	0.1479	**0.0961**	0.1130	0.1141	0.1113	0.0978	0.1172
		Dataset 6	MAPE	0.5733	0.5573	0.5514	0.4306	0.4074	0.4015	0.3979	**0.3863**	0.4228
			RMSE	0.1379	0.1144	0.1128	0.0901	0.0887	0.0881	0.0878	**0.0860**	0.0938
		Dataset 7	MAPE	0.8249	0.7645	0.7589	**0.5851**	0.6368	0.6169	0.6110	0.6088	0.6527
			RMSE	0.2321	0.1980	0.1970	**0.1570**	0.1709	0.1606	0.1602	0.1641	0.1731
		Dataset 8	MAPE	0.4230	0.3946	0.3853	0.3455	0.3466	0.3619	0.3509	**0.3423**	0.3920
			RMSE	0.1151	0.0891	0.0872	0.0759	0.0755	0.0778	0.0762	**0.0747**	0.0832
	> 128	Dataset 1	MAPE	0.9274	0.9223	0.8917	**0.7652**	0.9145	0.9033	0.8990	0.8836	0.9158
			RMSE	0.1983	0.1794	0.1755	**0.1513**	0.1645	0.1631	0.1625	0.1608	0.1645
		Dataset 2	MAPE	0.5791	0.5640	0.5628	**0.4478**	0.5093	0.5024	0.5049	0.4951	0.5048
			RMSE	0.1558	0.1345	0.1345	**0.1045**	0.1129	0.1125	0.1125	0.1115	0.1145
		Dataset 3	MAPE	0.8130	0.7950	0.8153	0.6046	0.5717	0.6109	0.5924	**0.5653**	0.6679
			RMSE	0.1646	0.1500	0.1521	0.1200	0.1186	0.1227	0.1201	**0.1165**	0.1310
		Dataset 4	MAPE	0.5041	0.5566	0.5644	**0.4235**	0.4262	0.4989	0.4837	0.4596	0.5763
			RMSE	0.0998	0.1014	0.1028	**0.0786**	0.0789	0.0895	0.0873	0.0833	0.1030
		Dataset 5	MAPE	0.6213	0.6409	0.6293	**0.5248**	0.5534	0.5823	0.5630	0.5445	0.5857
			RMSE	0.1643	0.1630	0.1609	**0.1117**	0.1197	0.1239	0.1214	0.1161	0.1253
		Dataset 6	MAPE	0.6412	0.6464	0.6426	0.4779	0.4980	0.4902	0.4867	**0.4733**	0.5224
			RMSE	0.1500	0.1364	0.1352	0.1070	0.1093	0.1082	0.1077	**0.1054**	0.1156
		Dataset 7	MAPE	0.9285	0.9007	0.9057	**0.7070**	0.7395	0.7316	0.7303	0.7281	0.7512
			RMSE	0.2513	0.2308	0.2317	**0.1880**	0.1987	0.1943	0.1948	0.1984	0.2014
		Dataset 8	MAPE	0.5578	0.5198	0.5224	**0.4374**	0.4520	0.4754	0.4635	0.4566	0.5010
			RMSE	0.1348	0.1154	0.1156	**0.0945**	0.0948	0.0982	0.0966	0.0954	0.1026

Table 11: Forecast accuracy metrics of naive Chronos-2 ("Not residualized - Raw, EI, Anc"), factor-informed Chronos-2 ("Not residualized - FA" and all "Residualized" columns), and the base DFM for fine-grain series in additional datasets.

Grain	Horizon	Dataset	Metric	Not residualized				Residualized				DFM
				Raw	EI	Anc	**FA**	**Raw**	**EI**	**Anc**	**FA**	
Fine	≤ 128	Dataset 1	MAPE	2.3341	2.3657	2.2821	**2.2015**	2.5495	2.5804	2.5369	2.4797	3.9921
			RMSE	0.2678	0.2553	0.2478	**0.2400**	1.3358	1.3282	1.3280	1.2249	1.4180
		Dataset 2	MAPE	1.3982	1.3696	1.3450	**1.2511**	1.4518	1.4991	1.4532	1.4184	1.9827
			RMSE	0.2169	0.1987	0.1956	**0.1831**	0.8681	0.8612	0.8595	0.6308	0.9502
		Dataset 3	MAPE	2.3350	2.3714	2.3563	**2.2201**	2.4434	2.5358	2.4650	2.4258	3.2033
			RMSE	0.2783	0.2758	0.2739	**0.2637**	2.0219	2.0229	2.0218	1.7439	2.0324
		Dataset 4	MAPE	1.8354	1.8607	1.8796	**1.7450**	1.8705	1.9459	1.8913	1.8656	2.5407
			RMSE	0.1842	0.1823	0.1825	**0.1713**	0.4099	0.4130	0.4104	0.3895	0.4822
		Dataset 5	MAPE	2.1916	2.1789	2.1106	**2.0430**	2.3996	2.4354	2.3252	2.3004	4.0503
			RMSE	0.2788	0.2726	0.2662	**0.2456**	0.3021	0.3063	0.2961	0.2797	0.4971
		Dataset 6	MAPE	2.0314	1.9539	1.9452	**1.8023**	1.9592	1.9619	1.9312	1.8662	3.1329
			RMSE	0.2230	0.2038	0.2007	**0.1866**	0.4332	0.4317	0.4319	0.3261	0.5198
		Dataset 7	MAPE	1.7835	1.7340	1.7126	**1.5385**	1.7169	1.7429	1.7300	1.7278	3.1021
			RMSE	0.2673	0.2428	0.2412	**0.2147**	0.8235	0.8240	0.8243	0.6499	0.8975
		Dataset 8	MAPE	1.6048	1.5761	1.5736	**1.5100**	1.6371	1.6821	1.6278	1.5835	2.1703
			RMSE	0.2941	**0.2818**	0.2873	0.2838	0.3891	0.3429	0.3434	0.3010	0.4406
	> 128	Dataset 1	MAPE	3.0136	3.0389	2.9491	**2.8319**	3.4203	3.4733	3.4100	3.3269	4.2659
			RMSE	0.3181	0.3140	0.3045	**0.2913**	1.1974	1.1973	1.1961	1.1221	1.2384
		Dataset 2	MAPE	1.6445	1.6338	1.6373	**1.4995**	1.7599	1.8472	1.7985	1.7478	2.1174
			RMSE	0.2454	0.2287	0.2281	**0.2133**	0.4376	0.4365	0.4303	0.4017	0.5163
		Dataset 3	MAPE	2.9431	3.0195	3.0694	**2.7801**	2.9247	3.0348	2.9690	2.9215	3.4657
			RMSE	0.3214	0.3290	0.3302	**0.3126**	1.6559	1.6560	1.6536	1.4957	1.6639
		Dataset 4	MAPE	2.1928	2.2132	2.2758	**2.0144**	2.1467	2.2348	2.1839	2.1433	2.6098
			RMSE	0.2154	0.2143	0.2181	**0.1954**	0.4191	0.4224	0.4199	0.4176	0.4563
		Dataset 5	MAPE	2.8095	2.7558	2.6485	**2.6473**	3.1534	3.2461	3.0996	3.0861	4.7487
			RMSE	0.3178	0.3133	0.3037	**0.2922**	0.4768	0.4830	0.4697	0.3536	0.6149
		Dataset 6	MAPE	2.6715	2.5365	2.6212	**2.3446**	2.5545	2.5023	2.4958	2.3772	3.6373
			RMSE	0.2759	0.2543	0.2600	**0.2364**	0.3343	0.3312	0.3297	0.2523	0.4369
		Dataset 7	MAPE	2.0547	2.0543	2.0384	**1.8383**	2.0962	2.1145	2.1319	2.0476	3.5982
			RMSE	0.3024	0.2896	0.2886	**0.2586**	1.5623	1.5611	1.5612	1.4975	1.6181
		Dataset 8	MAPE	2.1597	2.0769	2.1372	**1.9970**	2.1540	2.2466	2.1706	2.1103	2.6518
			RMSE	0.3315	**0.3152**	0.3250	0.3166	0.3526	0.3480	0.3432	0.3273	0.4254

J M5 dataset experiments

Dataset & evaluation To supplement main paper findings, we evaluate FA-TSFMs on the public M5 dataset [Makridakis et al., 2022], which contains daily unit sales from 10 Walmart stores across 3 US states over 1941 days. Table 12 summarizes its hierarchical structure. The M5 competition focuses on forecasting 30,490 item-store series, but these are highly sparse (54% zeros at item level) and neither standard DFMs nor the TSFMs evaluated in this paper are well-suited for such zero-inflated data. Adapting DFMs for count data is possible [Kaufmann and Schumacher, 2019, Murray et al., 2013, Capdeville et al., 2019] but requires non-trivial configuration and domain expertise, which we leave to future work. Instead, we aggregate item-level sales to construct a 124-series hierarchy, which eliminates sparsity through aggregation (0% zeros at all aggregated levels).

Table 12: Hierarchical structure of the M5 dataset.

Series type	Level	Description	Num series	Sample series	% zeros
Core	0	Total	1	`TOTAL`	0%
	1a	State	3	`CA`, `TX`, `WI`	0%
	1b	Category	3	`FOODS`, `HOBBIES`, `HOUSEHOLD`	0%
	2a	Store	10	`CA_1`, ..., `WI_3`	0%
	2b	Department	7	`FOODS_1`, ..., `HOUSEHOLD_2`	0%
Peripheral	3	Store × Category	30	`CA_1_FOODS`, ..., `WI_3_HOUSEHOLD`	0%
	4	Store × Department	70	`CA_1_FOODS_1`, ..., `WI_3_HOUSEHOLD_2`	0%
	5	Store × Item	30490	`CA_1_FOODS_1_001`, ..., `WI_3_HOUSEHOLD_2_123`	54%

We fit the DFM on the 24 core series (levels 0–2b) and evaluate FA-TSFMs on both core and peripheral series, with peripheral series (levels 3, 4) receiving only non-residualized FA-TSFM forecasts as in Section 6.1. All models are trained on days 1 to 1913 and evaluated on days 1914 to 1941 ($H = 28$ days), matching the M5 competition's test window. We report MAE, RMSE, RMSSE (Root Mean Squared Scaled Error), and WRMSSE (Weighted RMSSE), the last of which is the official M5 competition metric. For a single estimation end date T (with $T = 1913$), the metrics for series belonging to type $\mathcal{I}$ (core, peripheral) are calculated as:

$$\mathrm{MAE}_{\mathcal{I}} := \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{1}{H} \sum_{h=1}^{H} \left| Y_{i,T+h} - \hat{Y}_{i,T+h|T} \right|,$$

$$\text{RMSE}_{\mathcal{I}} := \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sqrt{\frac{1}{H}\sum_{h=1}^{H}\left(Y_{i,T+h}-\hat{Y}_{i,T+h|T}\right)^2},$$

$$\text{RMSSE}_{\mathcal{I}} := \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sqrt{\frac{\frac{1}{H}\sum_{h=1}^{H}\left(Y_{i,T+h}-\hat{Y}_{i,T+h|T}\right)^2}{\frac{1}{T-1}\sum_{t=2}^{T}\left(Y_{i,t}-Y_{i,t-1}\right)^2}},$$

$$\text{WRMSSE}_{\mathcal{I}} := \frac{1}{L}\sum_{\ell=1}^{L}\sum_{i\in\mathcal{I}_\ell}\frac{\text{Sales}_i}{\sum_{j\in\mathcal{I}_\ell}\text{Sales}_j}\sqrt{\frac{\frac{1}{H}\sum_{h=1}^{H}\left(Y_{i,T+h}-\hat{Y}_{i,T+h|T}\right)^2}{\frac{1}{T-1}\sum_{t=2}^{T}\left(Y_{i,t}-Y_{i,t-1}\right)^2}}$$

where Sales_i denotes the dollar sales of series i over the last $H = 28$ days of the training window, L is the number of aggregation levels, and $\mathcal{I}_\ell$ is the set of series at level ℓ. The RMSSE denominator is the mean squared error of the naive lag-1 forecast computed on the training window, scaling each series' error by its intrinsic variability. Further, RMSSE uses an unweighted arithmetic mean across series, while WRMSSE weights each series by its historical sales volume, giving more importance to high-volume series. Unlike the main paper's RMSE (which takes the square root after averaging across all series and horizons, see Appendix D.1), here we compute per-series RMSE before averaging across series to stay consistent with the per-series WRMSSE definition used in the M5 competition.

DFM To investigate whether FA-TSFM works with simple, minimally-tuned factor models, we fit a lightweight DFM on the 24 core series in log scale. The DFM extracts the same four factor types as the main paper's DFM (Appendix A.1): annual seasonality $\mathcal{S}_{1,t}$ (Fourier, $K = 3$), weekly seasonality $\mathcal{S}_{2,t}$ (day-of-week dummies), holiday seasonality $\mathcal{S}_{3,t}$ (event dummies from the M5 calendar), and trend $\mathcal{T}_t$. The key differences are in estimation:

- Seasonality factors are estimated via ridge regression rather than MCMC with Bayesian priors.
- The trend is extracted from ridge residuals using a state-space model (level, AR(1) cycle, random-walk drift; 5 MLE parameters via Kalman filter + L-BFGS-B), which is structurally similar to the main paper's local linear trend but replaces the mean-reverting slope with a random-walk drift and a separate AR(1) cycle.

- PCA extracts a single factor per component type ($\mathcal{T}_t$, $\mathcal{S}_{1,t}$, $\mathcal{S}_{2,t}$, $\mathcal{S}_{3,t}$) across the 24 core series, with OLS loadings for reconstruction.
- All 24 core series are estimated jointly: each series is independently decomposed, then PCA pools components across all series regardless of hierarchy level. There is no top-down orthogonalization as in the main paper's DFM (see Appendix A.2).

This DFM is deliberately simple: it uses a single factor per component, a basic SSM, and was developed with AI coding assistance with minimal manual tuning beyond a small grid search over the Fourier order, number of factors, and ridge penalty. The goal is not to build the best possible DFM for M5, but to test whether even a rudimentary factor model can improve TSFM forecasts through the FA-TSFM framework.

Results Table 13 reports evaluation metrics for the same four TSFMs evaluated in Section 4 on both core and peripheral series. In addition to the Raw, EI, and FA configurations from the main paper, we include an EI+FA configuration that combines event indicators with DFM factors as covariates. The Anc baseline is omitted because the M5 DFM extracts factors jointly across all core series rather than exploiting the DAG structure, so passing raw ancestor series would not reflect the same hierarchical information flow as in the main paper.

For Chronos-2, TabPFN-TS, and ApolloPFN, EI provides substantial improvements over Raw, and FA alone offers more modest gains — in some cases performing comparably to or worse than EI. However, EI+FA consistently outperforms EI alone on both core and peripheral series, indicating that even factors from a simple, minimally-tuned DFM capture complementary structure beyond event dummies. On core series, residualized EI+FA achieves the best RMSSE for Chronos-2 (0.539) and TabPFN-TS (0.535), while residualized EI is best for ApolloPFN (0.553) with residualized EI+FA a close second (0.559). On peripheral series, non-residualized EI+FA is best for all three models.

The relatively weaker performance of FA alone likely reflects the simplicity of the M5 DFM compared to the main paper's more carefully designed model. This is evidenced by

Table 13: Forecast accuracy metrics on the M5 dataset for naive TSFMs ("Not residualized - Raw, EI"), factor-informed TSFMs ("Not residualized - FA, EI+FA" and all "Residualized" columns), and the base DFM. Peripheral series have no DFM or residualized results.

Series type	TSFM	Metric	Not residualized				Residualized				DFM
			Raw	EI	**FA**	**EI+FA**	**Raw**	**EI**	**FA**	**EI+FA**	
Core	Chronos-2	MAE	564.61	544.82	520.95	516.55	486.12	478.66	507.58	**470.96**	792.20
		RMSE	749.90	702.57	668.76	655.90	640.16	595.78	658.04	**583.10**	976.95
		RMSSE	0.6597	0.6244	0.6116	0.5973	0.5956	0.5407	0.6023	**0.5393**	0.8661
		WRMSSE	0.5663	0.5291	0.5057	0.4939	0.4838	0.4497	0.4989	**0.4399**	0.7429
	TabPFN-TS	MAE	524.42	485.41	486.81	**441.29**	587.07	541.48	536.29	532.38	792.20
		RMSE	697.21	620.73	632.87	**566.18**	698.77	642.00	643.64	639.75	976.95
		RMSSE	0.6238	0.5665	0.5819	0.5416	0.5748	0.5494	0.5501	**0.5353**	0.8661
		WRMSSE	0.5262	0.4673	0.4772	**0.4257**	0.5280	0.4843	0.4867	0.4844	0.7429
	Moirai 1.0	MAE	798.77	1446.16	1393.00	1439.01	**636.39**	754.63	743.29	761.10	792.20
		RMSE	1007.62	1919.23	1837.97	1913.20	**813.18**	948.37	929.75	948.28	976.95
		RMSSE	0.8760	1.4574	1.3974	1.4596	**0.7053**	0.7854	0.7730	0.7832	0.8661
		WRMSSE	0.7653	1.4530	1.3929	1.4480	**0.6178**	0.7228	0.7076	0.7227	0.7429
	ApolloPFN	MAE	638.77	525.84	720.63	487.79	552.83	**458.97**	543.09	464.60	792.20
		RMSE	823.92	665.52	889.28	625.73	695.70	**579.14**	676.25	584.44	976.95
		RMSSE	0.7072	0.6403	0.7483	0.6033	0.6057	**0.5531**	0.5965	0.5592	0.8661
		WRMSSE	0.6222	0.4995	0.6714	0.4698	0.5250	**0.4361**	0.5111	0.4405	0.7429
Peripheral	Chronos-2	MAE	81.92	77.41	78.54	**74.95**	–	–	–	–	–
		RMSE	104.64	98.14	99.51	**94.76**	–	–	–	–	–
		RMSSE	0.7843	0.7468	0.7532	**0.7343**	–	–	–	–	–
		WRMSSE	0.6552	0.6102	0.6259	**0.5902**	–	–	–	–	–
	TabPFN-TS	MAE	78.98	72.74	74.76	**71.59**	–	–	–	–	–
		RMSE	102.18	93.54	95.48	**91.67**	–	–	–	–	–
		RMSSE	0.7713	0.7302	0.7457	**0.7216**	–	–	–	–	–
		WRMSSE	0.6397	0.5816	0.5917	**0.5675**	–	–	–	–	–
	Moirai 1.0	MAE	**101.88**	156.18	149.81	156.10	–	–	–	–	–
		RMSE	**130.45**	206.68	196.22	206.50	–	–	–	–	–
		RMSSE	**0.9220**	1.2840	1.2121	1.2881	–	–	–	–	–
		WRMSSE	**0.8282**	1.3106	1.2477	1.3085	–	–	–	–	–
	ApolloPFN	MAE	86.95	85.63	92.85	**79.39**	–	–	–	–	–
		RMSE	109.74	107.42	115.79	**100.58**	–	–	–	–	–
		RMSSE	0.8206	0.8068	0.8500	**0.7764**	–	–	–	–	–
		WRMSSE	0.6852	0.6716	0.7253	**0.6266**	–	–	–	–	–

the standalone DFM being the worst configuration across all but one TSFM on core series, with Chronos-2, TabPFN-TS, and ApolloPFN all substantially outperforming it without any covariates. Nevertheless, even factors from this simple model provide consistent incremental value when combined with EI.

Moirai 1.0 replicates the covariate degradation observed in the main paper: all non-residualized covariate configurations perform substantially worse than Raw. On core series, residualized Raw is the best Moirai configuration, benefiting from indirect factor information through the DFM forecast without the covariate pathway that harms performance. On peripheral series where residualization is unavailable, Raw is best.