DOES MORE HISTORY HELP? UNDERSTANDING CONTEXT LENGTH IN TIME SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series foundation models (TSFMs) for time series forecasting are designed to operate on variable input context lengths. This leads to the question: *how much context is worth using for a given task?* While the current trend favors longer contexts to enable more flexible predictors, their beneficial effect on inference-time accuracy is not guaranteed. In this work, we systematically investigate the influence of context length on forecasting performance across 11 state-of-the-art TSFMs, and show consistent diminishing returns as the context length grows: an effect called *context saturation*. Through experiments in controlled environments, we provide insight into the factors driving context saturation, and show signs of recency bias in TSFMs, where their ability to leverage long-range dependencies varies substantially by model. Furthermore, through oracle context-length selection, we demonstrate the existence of operating points with both lower aggregate forecasting error and lower computational cost than those resulting from maximum context length. Finally, we propose a methodology for *zero-shot* inference-time context length selection, leveraging only synthetic data. Evaluated across all 11 TSFMs on the GiftEval benchmark, our approach achieves substantial efficiency gains, validating the practical impact of our findings.

1 INTRODUCTION

*Time series foundation model*s (TSFMs) make zero-shot forecasting practical: a single model, pre-trained to accept input sequences (i.e., *contexts*) of variable lengths, can serve many domains, predictive horizons, and sampling frequencies without task-specific fine-tuning (Rasul et al., 2023; Das et al., 2024; Ansari et al., 2024; Woo et al., 2024). While in task-specific models the input context length is a fixed hyperparameter, selected at training time, (Wu et al., 2021; Nie et al., 2023), TSFMs shift this selection to inference time. Assuming enough data is available, the natural default is the chosen model's maximum pre-training length. Yet the maximum input length is a capacity limit, not evidence that such amount of input observations is useful for every task.

The problem of searching for an optimum context length exists from before TSFMs. Classical forecasting approaches balance the input length against observations' relevance under time-variance or when structural breaks make older observations not useful for the task (Pesaran & Timmermann, 2007; Inoue et al., 2017; Altansukh & Osborn, 2022). In long-horizon forecasting, deep learning solutions have mostly focused on architectures that could efficiently process longer input sequences: Informer, Autoformer, and FEDformer reduce or restructure long-sequence attention (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022); patching lowers the token cost of longer histories (Nie et al., 2023); inverted attention routes variables across time (Liu et al., 2024); and explicit time representations expose structure beyond a narrow lookback (Young et al., 2022). Recent maximum-effective-window (Zhang et al., 2025) and adaptive-lookback (Tong & Yuan, 2026) methods modify an architecture or its training pipeline to exploit longer contexts or truncate them dynamically. Analyses of long-context forecasting emphasize that long contexts may serve to identify the target process even without presence of direct long-range dependencies (Butera et al., 2026).

The same question appears even outside the time series domain. In *large language model*s (LLMs), prompt compression and selective-context methods filter input tokens before text generation (Jiang et al., 2023; Li, 2023), while associated inference methods prune low-value key-value pairs (Zhang et al., 2023), retain only recent context plus attention sinks (Xiao et al., 2024), or select prompt

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

positions using observed attention patterns (Li et al., 2024). More generally, adaptive-computation and early-exit methods allocate different amounts of work to different inputs (Graves, 2016; Teerapittayanon et al., 2016; Xin et al., 2020; Schuster et al., 2022; Nogales et al., 2025).

For TSFMs, increasing the context length can hurt performance, as shown by Shi et al. (2024); Zhang et al. (2024). In this regard, Nogales et al. (2026) define the *context saturation length* (CSL) as the shortest context for which a TSFM reaches a given fraction of its peak performance, and link this quantity to the temporal scale of seasonality or autoregressive memory in the input sequence. In this work, we extend this perspective, providing empirical evidence, in controlled environments, that CSL coherently varies according to a broader range of target task characteristics, e.g., predictive horizon length, presence of regime changes, *signal-to-noise ratio* (SNR), and may depend on model-specific biases as well. Moreover, we show that, across different TSFMs and real-world tasks in the GiftEval (Aksu et al., 2024) benchmark, there exist hindsight opportunities for efficiency gains. That is, for each considered task and model, there exists a context length shorter than the model's maximum one, for which the prediction error is lower. Hence, both computational cost and prediction performance can be improved by selecting an appropriate context length at inference time. To show that our insights can be practically operationalized, we further demonstrate the feasibility of automatic context length selection, by means of a TSFM-specific learned selector, trained on synthetic data only. Given features extracted from the original input sequence, the selector estimates the shortest length satisfying a calibrated risk threshold compared to maximum context length. This approach allows exploiting the accuracy-cost trade-offs enabled by context saturation, without the need to evaluate the entire span of possible context lengths. Across 11 TSFMs, validated against the GiftEval benchmark, we obtain a median reduction of 50.9% *multiply-accumulate operation*s (MACs) with a cost of only 0.81% median *mean absolute scaled error* (MASE).

Our main novel contributions are as follows:

Empirical Characterization of Context Saturation: We systematically analyze 11 pre-trained TSFMs and reveal consistent diminishing returns as context length increases, linking this effect to model-specific recency biases and task-specific temporal dependency structures.

The Unrealized Efficiency Gains: Through oracle context length selection, we demonstrate that maximizing context length is often suboptimal, establishing the existence of operating points that simultaneously reduce forecast error and computational cost.

Dynamic Context Length Selection: We propose a methodology to automatically select the context length for frozen pre-trained TSFMs, delivering significant inference efficiency gains on the GiftEval benchmark across the 11 evaluated TSFMs.

2 PRELIMINARIES

In this section we introduce the notation used throughout the paper and provide a formal definition of CSL. A *time series* $x = (x_t)_{t \in \mathcal{T}}$ is a sequence of observations where $x_t \in \mathbb{R}^d$ denotes the value recorded at time-step t and $\mathcal{T}$ is a discrete temporal index set. For integers $a < b$, we use $x_{a:b} \coloneqq (x_a, x_{a+1}, \ldots, x_{b-1})$ to denote the contiguous subsequence over the discrete interval $[a, b)$.

A pre-trained TSFM, with fixed learned parameters, is a function f of the following form

$$\hat{x}_{t:t+h} = f\left(x_{t-L:t}, h\right), \qquad \text{with } L \in \mathcal{L}_f, \text{ and } h \in \mathcal{H}_f, \tag{1}$$

where $x_{t-L:t}$ is an input *context* of length L; h is a forecasting *horizon*, i.e., the number of time-steps to forecast; $\mathcal{L}_f \subset \mathbb{N}$ and $\mathcal{H}_f \subset \mathbb{N}$ denote respectively the sets of context and horizon lengths supported by f; and $\hat{x}_{t:t+h}$ denotes the model's forecast. Additionally, we denote by $L_f^+ \coloneqq \max(\mathcal{L}_f)$ the maximum context length supported by model f. A forecasting *task* on series x for horizon h is then defined as a tuple (x, h). Given a loss metric $\ell : \mathbb{R}^{h \times d} \times \mathbb{R}^{h \times d} \to \mathbb{R}^+$, the context-length-dependent error on task (x, h) is defined as:

$$E_f(x, h, L) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \ell(f\left(x_{t-L:t}, h\right), x_{t:t+h}), \qquad \forall L \in \mathcal{L}_f. \tag{2}$$

Following Nogales et al. (2026), *context saturation length* (CSL) with tolerance $\epsilon \geq 0$ is defined as:

$$L_{f,\epsilon}(x, h) = \min\left\{ L \in \mathcal{L}_f : E_f(x, h, L) \leq (1 + \epsilon) \min_{L' \in \mathcal{L}_f} E_f(x, h, L') \right\}. \tag{3}$$

Thus, CSL is the minimum admissible context length for which a pre-trained TSFM achieves an error at task within and $\epsilon\%$ degradation compared to its best error on the task for any supported context length. This quantity allows us to analyze how changing L affects model performance.

3 Understanding Context saturation

In this section, we leverage controlled synthetic data to analyze the behavior of different TSFMs, in terms of CSL (Eq. 3), as the characteristics of the task change. Then, by means of controlled interventions, we provide insights into whether CSL is mainly the result of properties of the target task or biases of the pre-trained models.

3.1 Context saturation and task characteristics

To analyze how CSL varies across TSFMs and across tasks with different characteristics, we design a pool of controlled environments, each consisting of time series generated by known data-generating processes. In particular, we design each environment to probe the effect of varying some specific task parameter q, who's exact meaning is environment-dependent. The environments are as follows:

Seasonal scale Tasks consist of superpositions of multiple sinusoidal signals with additive gaussian noise. The number of components, amplitude, frequency, and phase of each component varies per task. Here, q corresponds to the period of the slowest frequency component.

Horizon length Tasks consist of superpositions of a sinusoidal signal and an *autoregressive* (AR) process. The period of the sinusoid and timescale of the AR process are kept fixed, while the target horizon h changes per task. Here, $q = h$.

Regime-change point Tasks consist of piecewise sinusoidal signals with additive gaussian noise, where changes in signal level, amplitude, and phase occur over time. Here q corresponds to the distance between the forecast origin t and the time-step at which the latest regime change occurred.

Signal-to-noise ratio Tasks consist of sinusoidal signals with additive gaussian noise. Amplitude and period of the signal are fixed, and the standard deviation of the additive noise varies per task. Here, q equals to the standard deviation of each sampled task's noise.

Precise details for the data-generating processes of each environment are in App. A. The rationale here is that each environment allows us to probe how the pre-trained models' error varies on tasks with known temporal characteristics as the context length changes. We evaluate eleven pre-trained TSFMs: Chronos2-Base, Chronos2-Small, and Chronos2-Synth (Ansari et al., 2025); ChronosBolt-Base (Ansari et al., 2024); FlowState-R1 (Graf et al., 2026); Moirai2-Small (Liu et al., 2025a); PatchTST-FM-R1 (Wen et al., 2026); Sundial-Base-128M (Liu et al., 2025b); TimesFM2.5-200M (Das et al., 2024); TiRex2 (Podest et al., 2026); and Toto-2.0-313m (Khwaja et al., 2026). For each environment, we select a range of values for q and, for each value we sample 128 tasks. Then, for each considered model and each task, we produce forecasts varying the context length across a shared set of feasible values. Examples of generated synthetic data and forecasts are in App. A.3

Fig. 1 reports the results of the experiment, where each subplot corresponds to one of the previously mentioned environments. Each curve corresponds to a different TSFM; the x-axis reports the *normalized context length* $\widetilde{L} := {}^{L}/_{q}$, i.e., the context length normalized by the task characteristic parameter; the y-axis reports the MRAE for each $\widetilde{L}$, i.e., the average absolute error ratio between the error for a given $\widetilde{L}$ and the best achievable error for any $\widetilde{L}$. Colored dots denote the CSL (Eq. 3) for $\epsilon = 0.05$. Looking at the MRAE against the normalized context length $\widetilde{L}$, allows us to inspect whether CSL is influenced by task characteristics q, rather than simply the raw observation count L. Below we discuss the results of the experiments for each environment.

Seasonal scale For periodic signals (Fig. 1, top-left), where $\widetilde{L}$ corresponds to the number of periods in the input context, the different TSFMs show a uniform behavior: after the context spans multiple periods the error falls sharply, eventually reaching CSL. While the exact trend is model-dependent, here context saturation is linked to the periodicity of the task, rather than to the raw context length.

Horizon length When the data-generating process is fixed, and we vary the horizon only (Fig. 1, top-right), $\widetilde{L}$ corresponds to the ratio between the context length L and the target horizon h. The

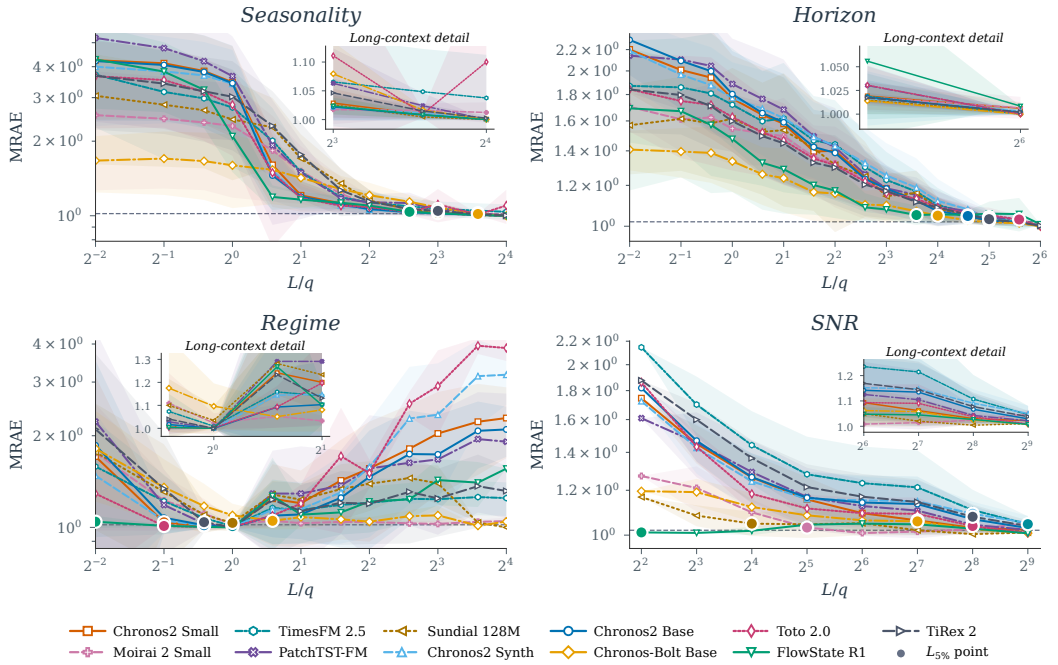

Figure 1: MRAE against normalized context length L/q for 4 synthetic environments and 11 TSFMs. Lines and shading denote $mean \pm 1std$ across tasks; the dotted line marks MRAE $= 1.05$. Large circles mark $L_{f,5\%}$ for each plotted mean curve.

results highlight that context saturation is also affected by this ratio, and longer-horizon tasks may require proportionally longer contexts to reach CSL.

Regime-change point In the environment where regime-changes occur (Fig. 1, bottom-left), $\widetilde{L}$ reflects the ratio of old-regime to current-regime observations in the input context. For $\widetilde{L} < 1$, the context only contains the current regime; as $\widetilde{L}$ grows greater than 1, a progressively larger portion of the context comes from previous regimes. Here CSL is in most cases near $\widetilde{L} = 1$, i.e., the point where the most context is available before including old regimes information. Depending on the TSFM, the error may plateau or increase when observations from old regimes are introduced. This suggests that properties of the task such as *time-variance* may affect CSL, while the detrimental impact of old regimes may be model-specific.

Signal-to-noise ratio When varying the noise variance for a fixed data-generating process (Fig. 1, bottom-right), $\widetilde{L}$ reflects the context length L adjusted by the task's noise level. Here we observe that CSL is again influenced by the probed task property, rather than raw context length alone: for most TSFMs tasks with higher noise level require proportionally longer contexts to saturate. Note that here the error is computed against the known expected value rather than noisy realizations; the baseline error change induced by higher noise levels would otherwise skew the results.

In App. A, we extend the experiment in Fig. 1 to additional environments. The results are consistent with the previous observations on how task characteristics affect CSL. Given our empirical analysis, we can draw the following takeaways about context saturation in pre-trained TSFMs.

Takeaways: (*i*) increasing the context length yields diminishing returns and may lead to performance degradation; (*ii*) task characteristics influence CSL coherently across TSFMs, yet saturation rates and exact CSL are model-specific; (*iii*) in absence of regime changes, model performance is often stable between CSL and the maximum supported context length for the considered scenarios.

3.2 Context Saturation and Model Biases

The analysis in Sec. 3.1 reveals that context saturation occurs and may be influenced by various task characteristics, however it does not tell us whether systematic model biases are also at play.

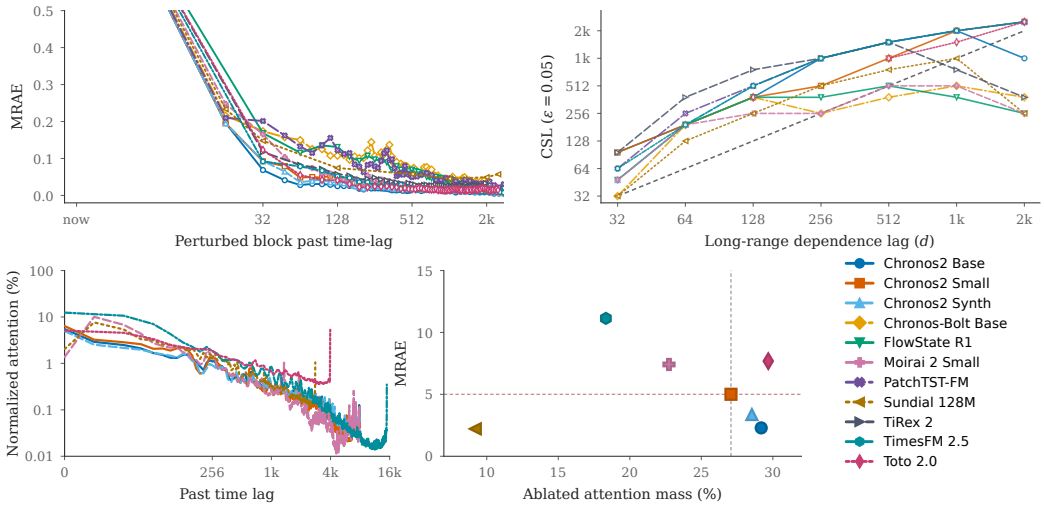

Figure 2: Context usage tests: temporal sensitivity (top-left), long-range dependencies (top-right), attention allocation (bottom-left), and attention ablation effect (bottom-right). Curves show per-model median results.

To gather insight on this matter we ask the following questions: (*a*) are TSFMs' forecasts more sensitive to particular past time-lags; (*b*) can TSFMs exploit predictive signals regardless of where they appear in the context; (*c*) for attention-based TSFMs, does attention allocation to parts of the context reflect their importance for the forecast? We seek to answer these questions empirically, by means of interventions in controlled environments.

Which time-lags are TSFMs most sensitive to? To answer this question, we produce a dataset of time series coming from heterogeneous data-generating processes (details in App. B), where for each process we sample 256 forecasting tasks. For each of the 11 TSFMs introduced in Sec. 3.1 and each task, we compute the MRAE induced by applying perturbations to different parts of the context, compared to leaving the context untouched. In particular, the context is divided into contiguous blocks of size P, where P is set to the target model's patch size for patch-based TSFMs, or to 32 otherwise. Then we iteratively perturb each block while leaving the others untouched. Fig. 2 (top-left), shows the results of the experiment when the perturbation consists of replacing a block's observations with the block's mean value. Each curve corresponds to a different TSFM, and shows the normalized MRAE between clean and perturbed forecast (x-axis), against the perturbed block's position in terms of past time-lags. We use each models maximum context length. Forecasts are more sensitive to observations near the forecast origin, and, in most of the considered TSFMs, sensitivity decays rather than presenting a sharp cutoff. In App. B.1 we repeat the experiment with other perturbation strategies and observe similar results. Overall, this points to a recency bias shared across TSFMs, likely inherited from the pre-training data distribution, as these models do not adopt explicit inductive biases to favor giving importance to recent observations, e.g., Hegazy et al. (2026).

Can TSFMs exploit predictive signals across their entire context? The observed decaying sensitivity profiles might partly be a byproduct of absence of useful predictive signal at distant past time-lags. To directly probe whether TSFMs are able to leverage predictive information across even at distant time-lags, we design the following experiment. We produce a dataset of forecasting tasks, each sampled from an AR process with an isolated long-range dependency at a variable time-lag d

$$x_t = \left(\sum_{r=1}^{8} a_r x_{t-r}\right) + b x_{t-d} + \varepsilon_t, \text{ with } \varepsilon_t \sim \mathcal{N}(0, 0.01) \text{ and } d \in \{2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\}. \quad (4)$$

We set $b = 0.35$ and sample 256 tasks for each value of d. App. B.2 provides details on how coefficients a_r were sampled to ensure stable processes with significant d-lagged predictive dependency. For each of the 11 considered TSFMs and each lag d, we compute the CSL $L_{f,5\%}$, using each models maximum context length, and say that a model f *reaches* the d-lagged dependency if $L_{f,5\%} \geq d$. Fig. 2 (top-right) reports the results of the experiment. Across 11 models: all reach lag 2^7, 10 reach

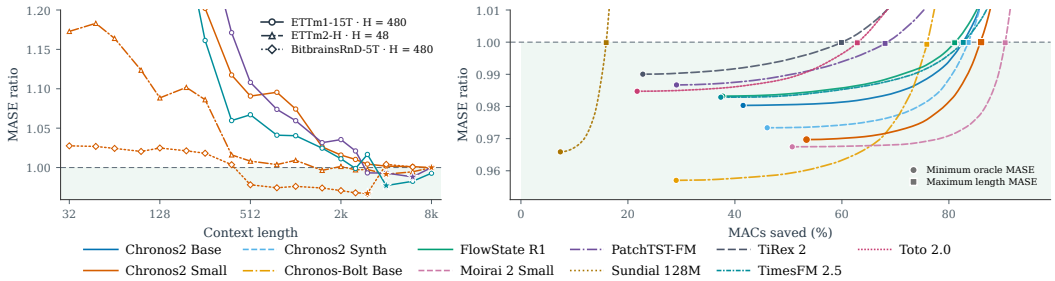

Figure 3: Examples of real-world context length to MASE trade-offs (left). GiftEval oracle Pareto frontiers (right). Moving right saves compute (MACs); moving down improves error (MASE). Markers denote minimum oracle MASE and equal to maximum length MASE points.

2^8, 7 reach 2^9, and 5 reach 2^{10}. Thus decaying sensitivity profiles do not imply TSFMs are unable to exploit dependencies at distant lags, however the extent of this capability is model-dependent and some models may be less capable of picking up predictive signals beyond a certain time-lag.

Is attention allocation a good proxy for past time-lags importance? We observed TSFMs' forecasts exhibit decaying sensitivity to the value of older time-lags (Fig. 2, top-left). To provide additional insight into whether this effect may be the result of systematic model bias, we take the same dataset used in the experiment in Fig. 2 (top-left) and look at the layer-wise average attention weight allocated to each non-overlapping block of P past time-lags (details on how we compute attention allocation are in App. B.3). Results for attention-based TSFM (7 out of the 11 considered ones) are reported in Fig. 2 (bottom-left). Most TSFMs allocate a stable amount of attention weight to distant time-lags, even though, according to our previous analysis, forecasts are not particularly sensitive to such parts of the input. This suggests that despite past-lags' impact on forecasts, attention-based TSFMs are not systematically biased towards ignoring distant portions of the input. Moreover, as already shown in the language and vision domains (Jain & Wallace, 2019; Serrano & Smith, 2019), attention weight should not be treated as a faithful proxy for input importance. To further substantiate this observation, we study the effect on forecasting performance of zeroing-out attention to past time-lags beyond 2048. Details on how we carry out the attention ablation are in App. B.3. Fig. 2 (bottom-right) reports the results of the experiment, where, for each attention-based TSFMs, the x-axis denotes the percentage of attention mass ablated on average, while the y-axis denotes the percentage *mean absolute error* (MAE) penalty resulting from the ablation. We can observe how ablated attention mass does not correlate to error degradation scale across different models. This suggests that attention allocation cannot be considered as a model-agnostic proxy for an input's importance to downstream performance.

Takeaways: (*i*) TSFMs collectively show signs of recency bias which may affect CSL, yet they are not systematically blind to parts of the input context; (*ii*) the ability to recover predictive signals at distant time-lags is a model-specific factor that may impact CSL beyond task-dependent effects.

4 Context Saturation on Real Forecasting Tasks

So far our analysis established context saturation as a model- and task-specific phenomenon by means of experiments in controlled environments. In this section we show how similar effects are present in real-world tasks as well and then demonstrate how context saturation reveals opportunities for Pareto-optimal efficiency gains, compared to using a TSFM's maximum input context length. In the following, we consider real-world forecasting tasks from the GiftEval benchmark (Aksu et al., 2024), and evaluate TSFMs under *channel-independence*, i.e., time series with multiple channels have each channel forecast independently from the others.

Real-world context saturation patterns We start by analyzing representative tasks from the GiftEval benchmark, reporting the ratio between the MASE at varying context lengths L and that at the maximum context length supported by the considered pre-trained model L_f^+. Fig. 3 (left) reports the MASE ratio against context length for different model-task pairs. In particular, the curve

color denotes the considered TSFM, while the marker shape denotes the task. Notably, while curve shape may vary significantly per-setting, we can clearly observe a pattern of diminishing returns as context length increases. Moreover, in some settings, better or equal error can be achieved by using $L < L_f^+$, denoting the opportunity for Pareto-optimal improvements by selecting an appropriate task-dependent context length. However, it also appears clear that such context length depends on individual model-task pairs, and cannot be chosen a-priori.

Hindsight opportunities for Pareto-optimal improvements To quantify these hindsight opportunities to improve both forecasting cost and error at once, we trace accuracy-cost Pareto frontiers for the 11 considered TSFMs across all forecasting tasks in GiftEval. In particular, for each task and model we determine the *oracle context length* $L_\lambda^\star$, i.e., the context length which minimizes $\log(\text{MASE}) + \lambda\text{MACs}$ on average across task samples. To quantify the attainable trade-off for model f at a given λ, we compute the geometric mean of the MASE across tasks and the total MACs at context length $L_\lambda^\star$, both relative to those at maximum context length L_f^+. The ratio of the MASE geometric means reflects the average attainable error improvement, while the percentage reduction in MACs reflects the computational savings. Fig. 3 (right) reports the Pareto frontiers obtained for increasing values of λ, where higher values of λ place a higher penalty on computational cost when looking for the optimal hindsight context length $L_\lambda^\star$. Precise details on the computation of Pareto curves and MACs are in App. C.2 and App. C.3. Notably, we observe that every TSFM admits, on average, operating points that are both computationally cheaper and more accurate than using the maximum context length supported by the model.

5 Zero-shot context selection

In Sec. 4 we demonstrated the existence of opportunities for efficiency gains when an oracle context length is selected. In this section we propose a methodology to turn this insight, which relied on hindsight information, into a practical solution that allows *zero-shot* dynamic context length selection for a pre-trained TSFM at inference time. We first introduce the proposed methodology, then evaluate its performance on the GiftEval benchmark across 11 TSFMs, considering both the univariate and multivariate forecasting setting.

5.1 A synthetic-data-only context length risk predictor

We cast the problem of inference-time context length selection as that of predicting a calibrated relative risk of using context length $L < L_f^+$, instead of L_f^+, for a given sample (x, t, h) and pre-trained model f. The context length can be then selected by picking the smallest L that satisfies a chosen risk threshold. For a sample (x, t, h), we define the relative risk of using context length L as

$$R_f\left(x, t, h, L\right) = {}^{E_f(x,t,h,L)}/_{E_f\left(x,t,h,L_f^+\right)}, \tag{5}$$

where E_f is the error made by model f for sample (x, t, h) at length L. Then define the *log-relative-risk* r_L and *exceedance indicator* b_L for a given sample and pre-trained model f as

$$r_L \coloneqq \log R_f\left(x, t, h, L\right) \qquad \text{and} \qquad b_L \coloneqq \mathbb{I}\left(R_f\left(x, t, h, L\right) > \tau_b\right) \tag{6}$$

respectively. Here $\mathbb{I}$ is the indicator function, evaluating to 1 if the condition is true or to 0 otherwise; τ_b is a relative risk threshold. Given these definitions, the proposed methodology involves a learnable *companion model* g that evaluates candidate context lengths L for target samples (x, t, h):

$$(\hat{r}_L, \hat{\sigma}_L, \hat{b}_L) = g(x_{t-L_f^+:t}, h, L), \tag{7}$$

where $\hat{r}_L \in \mathbb{R}$ denotes an estimate of the expected log-relative-risk $\mathbb{E}[r_L \mid x_{t-L_f^+:t}, h, L]$; $\hat{\sigma}_L \in \mathbb{R}$ denotes an epistemic uncertainty estimate capturing estimator variance; and $\hat{b}_L \in [0, 1]$ denotes an estimate of error exceedance probability $P(b_L = 1 \mid x_{t-L_f^+:t}, h, L)$. Finally, we define the calibrated risk score q_L and associated context length selection policy as

$$q_L = \hat{r}_L + u\hat{\sigma}_L + v\hat{b}_L, \quad \text{and} \quad \hat{L} = \min\left(\{L \in \mathcal{L}_f : q_L \leq \tau_q\} \cup \{L_f^+\}\right), \tag{8}$$

respectively. Here $u \geq 0$ and $v \geq 0$ are calibration parameters weighting predictive uncertainty and the probability of exceeding the relative risk threshold τ_b, while τ_q is the calibrated risk threshold for the context length selection policy. $\hat{L}$ denotes the selected context length, and the selection defaults to the maximum supported context length L_f^+ if no candidate satisfies the risk threshold.

Companion Model Training. The companion model g is trained entirely on a corpus of synthetic time series $\mathcal{D}_{train}$. For each synthetic sample and candidate context length $L \in \mathcal{L}_f$, the pre-trained TSFM f is evaluated to compute the ground-truth relative error ratio R_f, yielding supervision targets for both the log-relative-risk r_L and the exceedance indicator b_L. Crucially, training relies strictly on synthetic data, with the goal of using the companion model *zero-shot* on downstream tasks.

Policy Calibration. Once fitted, the parameters of the companion model g are frozen and the decision policy in Eq. 8 is governed by the hyperparameter triplet (u, v, τ_q), which dictates the operational trade-off between compute reduction (i.e., favoring smaller $\hat{L}$) and risk mitigation. The triplet (u, v, τ_q) is calibrated post hoc on an independent synthetic calibration set $\mathcal{D}_{cal}$, disjoint from $\mathcal{D}_{train}$. This provides a key practical advantage: a single trained companion can be calibrated into multiple distinct operating policies, ranging from highly conservative ones that prioritize accuracy preservation to aggressive policies that maximize computational savings.

5.2 Policy implementation and evaluation

To evaluate the proposed approach we implement the companion model g by means of ExtraTrees models (Geurts et al., 2006), which are given as input the target horizon h, the candidate context length L, and a vector of handcrafted features extracted from the input sample $x_{t-L_f^+:t}$ (features' details are in App. D.2). In particular, we learn an ExtraTrees regressor to compute $\hat{r}_L$ and $\hat{\sigma}_L$, and an ExtraTrees classifier for $\hat{b}_L$. These choices have practical motivations: (i) ExtraTrees are an ensemble of multiple regressors, which allows us to compute $\hat{\sigma}_L$ as the standard deviation across individual base regressors' predicted risk (i.e., the ensemble disagreement); (ii) at inference time, g runs for every call of the downstream TSFM, hence it needs to be comparatively inexpensive to avoid eating away the computational advantage it sets to provide; ExtraTrees paired with handcrafted features satisfy this requirement. We train this proposed implementation on a dataset $\mathcal{D}_{train}$ of 12000 synthetic samples (details on data generation are in App. D.1), using 192 trees for g, and setting $\tau_b = 1.05$ (i.e., we flag a sample as exceeding the risk threshold at length L if the error degrades by more than 5%). Finally, using a synthetic calibration set $\mathcal{D}_{cal}$ of 3000 samples, we instantiate 3 different risk policies: *Balanced*, *Very aggressive*, and *Max efficiency*, each increasingly favoring computational savings at the expense of higher error risk. App. D.1 provides details on the policy calibration procedure, as well as empirical results for additional risk policies.

We evaluate the proposed implementation across 11 pre-trained TSFMs over the entire GiftEval benchmark, considering forecasting tasks as univariate, i.e., time series with n channels are split into n single-channel tasks. Tab. 1 (A) compares the 3 aforementioned policies to a *naive* length cap policy which always sets $\hat{L} = 4096$. Each row reports the percentage difference in MASE (error) and theoretical MACs reduction (cost) for each policy, relative to the values obtained by using the maximum supported length L_f^+. The results show that, in most of the considered scenarios, the 3 adaptive selection policies yield better efficiency gains in terms of percentage MACs saved per point of percentage MASE drop, compared to the naive length cap baseline. Moreover, while this baseline yields competitive results, it lacks the flexibility of calibration to specific risk profiles, as well as the ability to dynamically save compute in an input-dependent manner. Note that, as the results show, the achievable efficiency gains vary per-model; this is also affected by their specific architecture and how their computational cost intrinsically scales with input length.

Additionally, we evaluate the efficacy of our selection policies in multivariate forecasting settings, re-using the frozen policies from the previous experiment. In this case, we use the policy to separately propose a context length for each channel, and then pick the longest proposed one and use it across all channels of the target time series. In Tab. 1 (B) we report the results of the experiment on GiftEval's multivariate time series, where the considered TSFMs are those which can handle multiple input channels. For the Chronos2 family we also consider the *batch learning* (BL) inference mode. The results show that, as in the univariate case, the dynamic selection policies are often superior to the fixed length cap baseline. Additionally, in the multivariate settings we can often achieve better efficiency trade-offs compared to the univariate case. Overall, our empirical results establish dynamic context length selection policies as a practical companion for *zero-shot* TSFMs, allowing substantial efficiency gains through calibrated risk profiles that can accommodate the needs of diverse applications.

Table 1: Accuracy–compute trade-offs relative to max context length on GiftEval (5 seeds avg.). Univariate (**A**) and multivariate (**B**) evaluation. ΔMASE is the percentage change in MASE (lower is better), while MAC save is the theoretical percentage compute reduction (higher is better). Green cells have a better accuracy–compute trade-off than the fixed 4K policy. Hatched cells indicate cases where the trade-off comparison is undefined.

	Balanced		Very aggressive		Max efficiency		Fixed 4K	
Model / mode	ΔMASE ↓	ΔMAC ↑	ΔMASE ↓	ΔMAC ↑	ΔMASE ↓	ΔMAC ↑	ΔMASE ↓	ΔMAC ↑
A. Univariate								
Chronos2-Base	0.00	14.7	+0.15	51.0	+1.02	75.9	+0.66	46.0
Chronos2-Small	−0.01	5.0	−0.10	33.7	+0.44	61.0	+0.23	46.8
Chronos2-Synth	−0.07	10.1	+0.02	43.2	+1.08	69.4	+0.60	46.0
ChronosBolt-Base	−0.01	4.7	+0.05	15.0	+0.69	32.1	0.00	0.0
FlowState-R1	+0.05	7.0	+0.09	26.8	+0.37	58.8	+0.86	54.7
Moirai2-Small	−0.04	2.4	+0.14	16.0	+0.77	50.9	−0.06	47.7
PatchTST-FM-R1	+0.09	7.6	+0.27	26.9	+0.75	49.5	+1.76	46.3
Sundial-Base-128M	+0.51	3.0	+0.97	6.2	+1.84	9.2	0.00	0.0
TimesFM2.5-200M	+0.01	6.1	+0.14	32.3	+0.81	64.7	+1.70	60.7
TiRex2	+0.02	4.1	+0.22	23.0	+0.92	47.6	+0.88	36.3
Toto-2.0-313m	0.00	0.1	+0.36	27.3	+1.40	46.8	0.00	0.0
Mean	+0.05	5.9	+0.21	27.4	+0.92	51.5	+0.60	35.0
B. Multivariate								
Chronos2-Base	−0.004	18.99	+0.159	58.02	+1.541	83.17	+1.212	49.44
Joint	−0.029	18.69	+0.001	56.94	+0.632	82.13	+0.817	49.44
Batch Learning	−0.062	0.005	−0.063	6.76	+0.552	55.97	+0.762	49.43
Chronos2-Small	−0.001	4.80	−0.061	37.61	+0.991	68.64	+0.712	50.20
Joint	−0.022	4.56	−0.404	37.09	+0.171	68.16	+0.297	50.20
Batch Learning	−0.022	0.34	−0.494	18.07	+0.156	51.41	+0.254	50.18
Chronos2-Synth	−0.157	13.46	−0.354	52.86	+0.245	77.33	+0.611	49.44
Joint	−0.034	13.26	−0.143	51.97	+0.003	76.14	+0.366	49.44
Batch Learning	−0.041	0.002	−0.158	5.49	−0.077	35.69	+0.336	49.43
TiRex2	+0.025	7.72	+0.109	30.61	+0.681	60.30	+1.435	42.53
Joint	+0.019	7.40	+0.033	28.95	+0.467	57.10	+1.797	40.64
Toto-2.0-313m	0.000	0.28	+0.474	41.34	+1.737	61.14	0.000	0.00
Joint	0.000	0.28	+0.175	40.80	+1.146	60.66	0.000	0.00

6 CONCLUSIONS

In this work, we showed that using the maximum context length a pre-trained TSFM can support is often inefficient at inference time. We connect this to *context saturation*, an effect by which increasing the context length yields diminishing returns. We show, through evidence in controlled environments, that this effect is driven by forecasting task characteristics such as seasonality, target horizon, and regime shifts, yet model-specific effects are also at play. Furthermore, we show context saturation presents clear hindsight opportunities for Pareto-optimal efficiency gains at inference time. To exploit this trade-off, we introduce a lightweight companion model for *zero-shot* context length selection, and demonstrate its practical efficacy across 11 TSFMs on the GiftEval benchmark. Overall, our work establishes a new perspective on how TSFMs can be efficiently operationalized, as well as providing insight into the factors that may govern context saturation.

Limitations and future work Our empirical analysis provides insights on the factors driving context saturation, however it does not identify a causal mechanism for it. In this regard, future works might explore whether this effect can be connected to systematic model behaviors e.g., activation or attention patterns, or measures of information content in the input data, e.g., conditional mutual information. Our proposed context selection policy demonstrates the practical applicability of our insights; nonetheless, we hope future research may further develop efficient adaptive context selection policies, e.g., by directly considering multivariate series during policy training/calibration, or by tailoring the design to specific TSFM architectures.

AI USE STATEMENT

In this work, we used generative AI tools, including OpenAI Codex, Anthropic Claude and Google Gemini, to assist with refining research methodology, hypotheses, and experimental designs; implementing, reviewing, and testing software; developing workflows; analyzing experimental results; searching and summarizing relevant literature; preparing scientific figures; and drafting, restructuring, and editing the manuscript. We did not use generative AI to formulate or prove formal mathematical claims. The authors reviewed all AI-assisted text, code, analyses, and figures, checked reported claims against experimental outputs and cited sources, and take responsibility for the correctness, originality, and final content of this work.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024. URL https://arxiv.org/abs/2410.10393.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. GluonTS: Probabilistic Time Series Models in Python. *arXiv*, June 2019. doi: 10.48550/arXiv.1906.05264.

Gantungalag Altansukh and Denise R. Osborn. Using structural break inference for forecasting time series. *Empirical Economics*, 63:1–41, 2022. doi: 10.1007/s00181-021-02137-w. URL https://doi.org/10.1007/s00181-021-02137-w.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From Univariate to Universal Forecasting. *arXiv*, October 2025. doi: 10.48550/arXiv.2510.15821. URL https://arxiv.org/abs/2510.15821.

Luca Butera, Giovanni De Felice, Andrea Cini, and Cesare Alippi. Why do time series models need long context windows? *arXiv preprint arXiv:2606.01999*, 2026. doi: 10.48550/arXiv.2606.01999. URL https://arxiv.org/abs/2606.01999.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1. URL https://doi.org/10.1007/s10994-006-6226-1.

Lars Graf, Thomas Bohnstingl, Stanisław Woźniak, and Angeliki Pantazi. FlowState: Sampling-Rate-Equivariant Time-Series Forecasting. In *ICML 2026 conference paper*, 2026. URL https://icml.cc/virtual/2026/poster/65805.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. URL https://arxiv.org/abs/1603.08983.

Kareem Hegazy, Michael W. Mahoney, and N. Benjamin Erichson. Recency biased causal attention for time-series forecasting. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 2233–2241. PMLR, 2026. URL https://proceedings.mlr.press/v300/hegazy26b.html.

Atsushi Inoue, Lu Jin, and Barbara Rossi. Rolling window selection for out-of-sample forecasting with time-varying parameters. *Journal of Econometrics*, 196(1):55–67, 2017. doi: 10.1016/j.jeconom.2016.03.006. URL https://doi.org/10.1016/j.jeconom.2016.03.006.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, Chenghao Liu, Ameet Talwalkar, and David Asker. Toto 2.0: Time series forecasting enters the scaling era. *arXiv preprint arXiv:2605.20119*, 2026. URL https://arxiv.org/abs/2605.20119.

Yucheng Li. Unlocking context constraints of LLMs: Enhancing context efficiency of LLMs with self-information-based content filtering. *arXiv preprint arXiv:2304.12102*, 2023. URL https://arxiv.org/abs/2304.12102.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2404.14469.

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

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9J6aG4e548.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A Family of Highly Capable Time Series Foundation Models. *arXiv*, February 2025b. doi: 10.48550/arXiv.2502.00816. URL https://arxiv.org/abs/2502.00816.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Miguel Nogales, Matteo Gambella, and Manuel Roveri. DYNAMAX: Dynamic computing for transformers and Mamba based architectures, 2025. URL https://doi.org/10.1109/IJCNN64981.2025.11228828.

Miguel Nogales, Luca Butera, Alberto Ferrante, and Cesare Alippi. Context saturation in zero-shot time-series foundation models. In *1st Workshop on Combining Theory and Benchmarks at ICML*, 2026. URL https://openreview.net/forum?id=oHM0cg2YkF.

M. Hashem Pesaran and Allan Timmermann. Selection of estimation window in the presence of breaks. *Journal of Econometrics*, 137(1):134–161, 2007. doi: 10.1016/j.jeconom.2006.03.010. URL https://doi.org/10.1016/j.jeconom.2006.03.010.

Patrick Podest, Marco Pichler, Elias Bürger, Levente Zólyomi, Bernhard Voggenberger, Wilhelm Berghammer, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex-2: Generalizing TiRex to Multivariate Data and Streaming. *arXiv*, July 2026. doi: 10.48550/arXiv.2607.01204. URL https://arxiv.org/abs/2607.01204.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023. doi: 10.48550/arXiv.2310.08278. URL https://arxiv.org/abs/2310.08278.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17456–17472, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f6f966116da1ef4554832be264560ea-Abstract-Conference.html.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2931–2951. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1282. URL https://aclanthology.org/P19-1282/.

Jingzhe Shi, Qinwei Ma, Huan Ma, and Lei Li. Scaling law for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2650. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97c2f0fac182353062d304d0322ae285-Abstract-Conference.html.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *23rd International Conference on Pattern Recognition*, 2016. doi: 10.1109/ICPR.2016.7900006. URL https://doi.org/10.1109/ICPR.2016.7900006.

Suxin Tong and Jingling Yuan. Efficiently enhancing long-term series forecasting via adaptive look-back with wavelets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25966–25974, 2026. doi: 10.1609/aaai.v40i31.39797. URL https://ojs.aaai.org/index.php/AAAI/article/view/39797.

Yunshi Wen, Wesley M. Gifford, Chandra Reddy, Lam M. Nguyen, Jayant Kalagnanam, and Anak Agung Julius. Revisiting the generic transformer: Deconstructing a strong baseline for time series foundation models. *arXiv preprint arXiv:2602.06909*, 2026. URL https://arxiv.org/abs/2602.06909.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2246–2251. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.204. URL https://aclanthology.org/2020.acl-main.204/.

Julong Young, Junhui Chen, Feihu Huang, and Jian Peng. Dateformer: Time-modeling transformer for longer-term series forecasting. *arXiv preprint arXiv:2207.05397*, 2022. URL https://arxiv.org/abs/2207.05397.

Jiahui Zhang, Zhengyang Zhou, Wenjie Du, and Yang Wang. Enhancing the maximum effective window for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2817. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/79345e916c5611bf39138892d6d58a86-Abstract-Conference.html.

Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1523. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/55f2a27b1ac39dbfdd0fc83742dc87d7-Abstract-Datasets_and_Benchmarks_Track.html.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1506. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6ceefa7b15572587b78ecfcebb2827f8-Abstract-Conference.html.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325. URL https://ojs.aaai.org/index.php/AAAI/article/view/17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A SYNTHETIC EXPERIMENTS AND DATA-GENERATING PROCESSES

This appendix defines the controlled processes used to isolate how required context changes with the data-generating scale. It first specifies the data-generating processes, then explains curve alignment and normalization, and finally provides representative forecasts for qualitative verification.

A.1 DATA-GENERATING PROCESSES

Each controlled experiment varies one structural parameter while holding the remaining process fixed. We generate 128 independent series per parameter value and evaluate every model on the same series. Context lengths follow the ratio grid

$$L = rq, \qquad r \in \{0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12, 16\},$$

where q is the process scale. Standardization uses the available context pool only and never uses the forecast region.

Periodicity and composite seasonality. The periodic process is a noisy sinusoid with period T. The closely related composite-seasonality process first draws between three and six distinct harmonics from frequencies $1/S, \ldots, 8/S$. Harmonic j receives a random phase and amplitude proportional to $j^{-1/2}$, after which the resulting length-S profile is standardized and tiled. Each repetition is multiplied by an independent $\mathcal{N}(1, 0.1^2)$ amplitude factor before observation noise is added. This produces a structured repeating shape rather than a single smooth wave. Their scale variables are $q = T$ and $q = S$, respectively.

Autoregressive memory and order. The memory sweep varies the persistence timescale $\tau = (1-\phi)^{-1}$ of an AR(1) process. The related order sweep uses stable AR(p) processes for $p \in \{1, \ldots, 4\}$ while matching their dominant pole, and hence their decay timescale. This separates the effect of coefficient count from the duration of predictive memory. Explicitly,

$$x_t = \sum_{j=1}^{p} a_j x_{t-j} + \varepsilon_t, \qquad 1 - \sum_{j=1}^{p} a_j z^j = \prod_{k=1}^{p} (1 - \lambda_k z), \tag{9}$$

with dominant pole $\lambda_1 = 1 - 1/\tau$. Any additional real or complex-conjugate poles have magnitude at most 0.7, so they change order but not the long memory scale. The AR(1) case reduces to $x_t = (1 - 1/\tau)x_{t-1} + \varepsilon_t$. Both plots use $q = \tau$.

Delayed dependence. Sparse trigger events are followed by predictable responses after lag d, with the remaining local process held fixed. Here $q = d$, so a curve aligned in L/d tests whether the required window follows the trigger location.

Regime duration and break age. The regime experiment uses a piecewise sinusoidal process with changes in level, amplitude, and phase; the active regime begins D steps before the forecast origin. The closely related break-age sweep varies the age A of one change point. Their scales are $q = D$ and $q = A$. We expect error to improve as context first covers the current regime, then flatten or worsen once $L/D > 1$ or $L/A > 1$ introduces observations generated under stale level, amplitude, and phase parameters. The break-age experiment isolates this crossing because it contains only one change point.

Period drift. Let u_t be the log-period displacement from a base period $T_0 = 64$:

$$u_t = (1 - 1/M)u_{t-1} + \eta_t, \qquad x_t = \sin\left(\sum_{s \le t} \frac{2\pi}{T_0 e^{u_s}} + \varphi\right) + \varepsilon_t. \tag{10}$$

Thus M is the correlation time of the instantaneous log period. Recent cycles estimate the current frequency, whereas cycles much older than M were generated at increasingly stale frequencies and accumulate phase error. This experiment therefore tests whether too much old context can hurt even without an abrupt change point; its normalized scale is $q = M$.

Missing gap. A constant-filled interval of length G occupies lookbacks from $G/2$ to $3G/2$ before the forecast origin, leaving a valid recent tail of length $G/2$. We set the interval to zero in raw signal space before context-pool standardization; its standardized value is therefore $-\mu_{\text{pool}}/\sigma_{\text{pool}}$, rather than an imputed sample mean. With $q = G$, windows shorter than $G/2$ contain only the valid tail, windows between $G/2$ and $3G/2$ extend into the constant span, and windows longer than $3G/2$ recover valid pre-gap observations. This tests access across an uninformative constant interval, not the wrappers' native handling of missing-value masks.

Forecast horizon. The horizon experiment varies h for a fixed process combining a sinusoid and an autoregressive component. It uses $q = h$, directly testing whether context demand expands in proportion to forecast distance. Separately, every non-horizon sweep listed above—periodicity, seasonality, AR order and memory, delay, regime duration, break age, multiscale periodicity, period drift, missing gap, and signal-to-noise ratio—is repeated at $h \in \{16, 64, 256\}$ while holding its own structural parameter fixed.

Signal-to-noise ratio. Observation noise is varied while the underlying period is fixed. The x axis coordinate is $L/T\sigma$. We multiply by T, a constant in this particular process, due to the small values that σ takes, which would result in less readable plots. Note that here the prediction target is the true expected value of future realizations, instead of the noisy signal. This is needed to avoid baseline changes in error skewing the results.

Multiscale periodicity. The process combines a period-T local oscillation with a slower amplitude envelope of length kT. We use $q = kT$ to test whether the slower modulation, rather than the fast carrier, controls context demand.

In Fig. 4 we extend the results from Fig. 1, considering the additional synthetic environments described above.

A.2 Curve alignment and normalization

The ratio $\widetilde{L} = L/q$ tests whether context demand scales with the controlled process parameter rather than merely providing a plotting convenience. For example, suppose a period-100 process saturates near $L = 200$ while a period-1,000 process saturates near $L = 2{,}000$. The raw knees differ by a factor of ten, but both occur at $L/T = 2$, supporting cycle coverage as the relevant scale. We therefore sweep several values of each structural parameter on a common grid of L/q ratios. For every parameter value, we divide errors by that value's attainable minimum, interpolate the resulting relative-error curve onto the common ratio grid, and then compute the cross-task mean and standard deviation. Collapse of the aligned curves supports scaling with q; failure to collapse indicates that raw length or another factor governs saturation.

A.3 Representative forecasts

For qualitative checks, we compare forecasts from short context ($L/q = 1$) and longer context ($L/q = 8$) on each process. The forecast timesteps are never used for standardization. The examples are useful for interpreting the error curves, but all quantitative conclusions in the paper use the complete parameter sweeps rather than selected examples.

The figures below cover all 12 controlled process families. In every process, the upper and lower rows use the same forecast origin and target; only the supplied history changes from $L/q = 1$ to $L/q = 8$. This makes the examples a direct qualitative counterpart to the normalized context sweeps in Figure 4. In each figure, the left column shows the final 2,048 pre-forecast observations and the right column enlarges the final 256 observations and the withheld horizon. The dark target is drawn beneath the colored model forecasts.

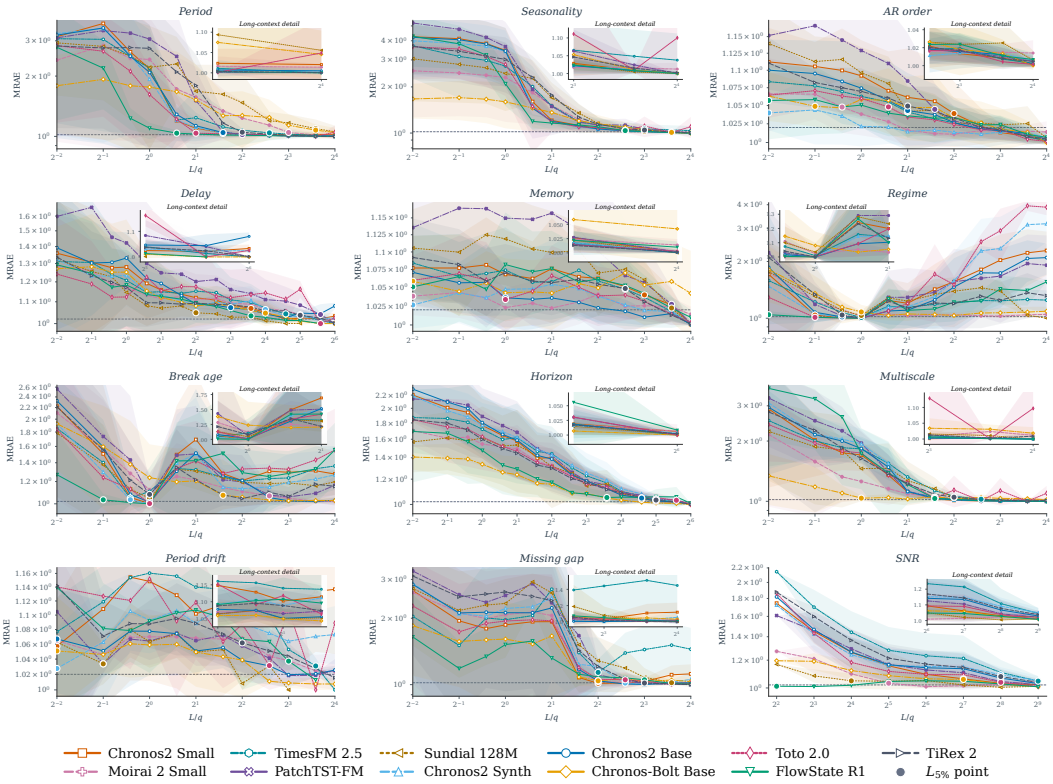

Figure 4: MRAE against normalized context length L/q for 12 environments and 11 TSFMs. Lines and shading denote $mean \pm 1std$; the dotted line marks MRAE $= 1.05$. Large circles mark $L_{f,5\%}$ for each plotted mean curve.

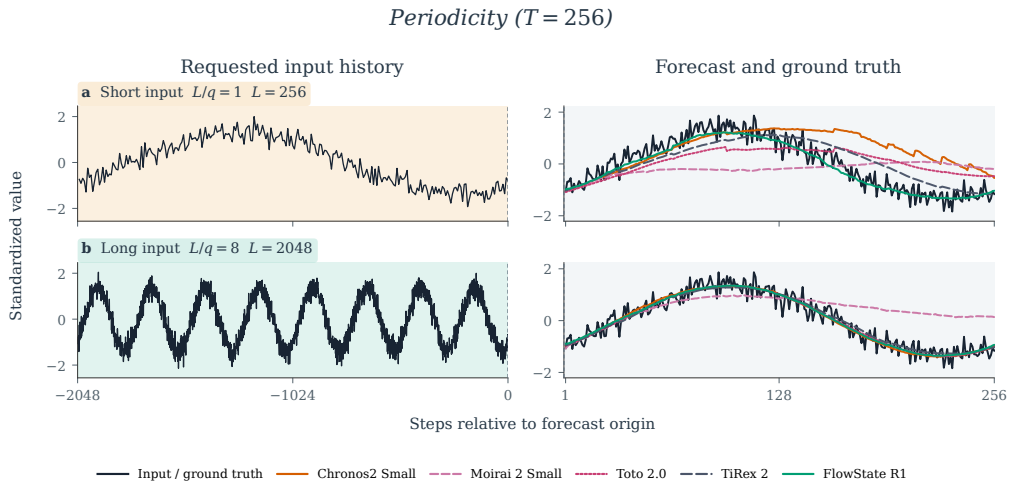

Figure 5: Periodicity. Several models recover the repeating target after seeing eight cycles rather than one.

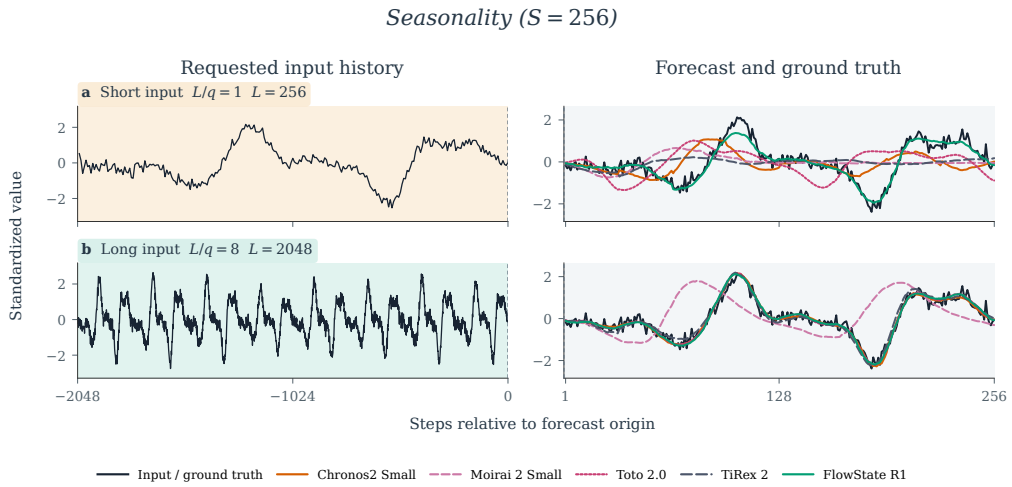

Figure 6: Composite seasonality. Longer context reveals the repeated structure across the superposed seasonal components.

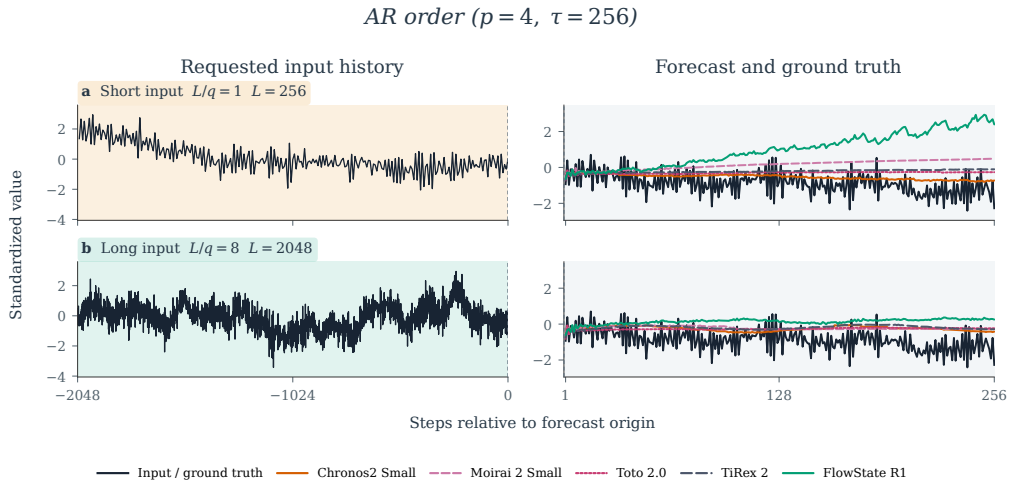

Figure 7: Fourth-order autoregression. Additional history stabilizes the forecasts.

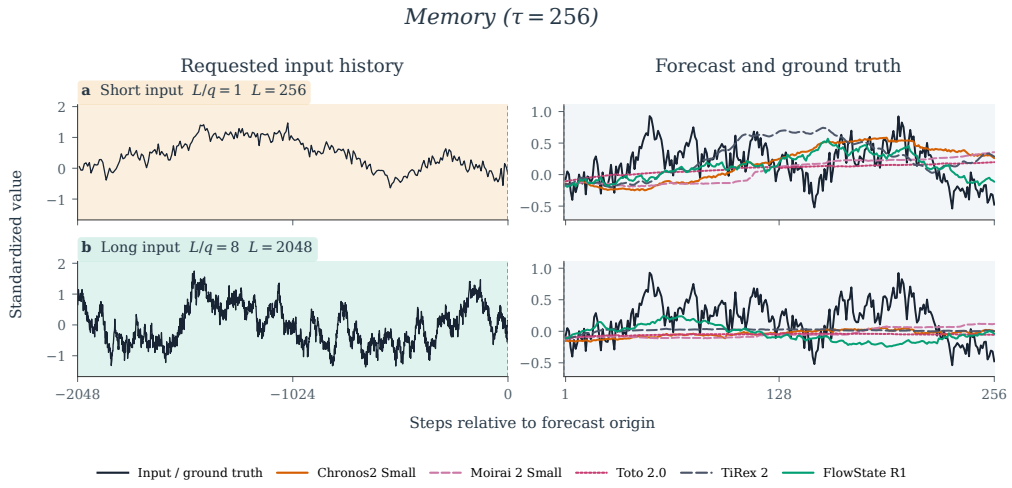

Figure 8: Autoregressive memory. The two inputs show one and eight times the process memory scale, respectively.

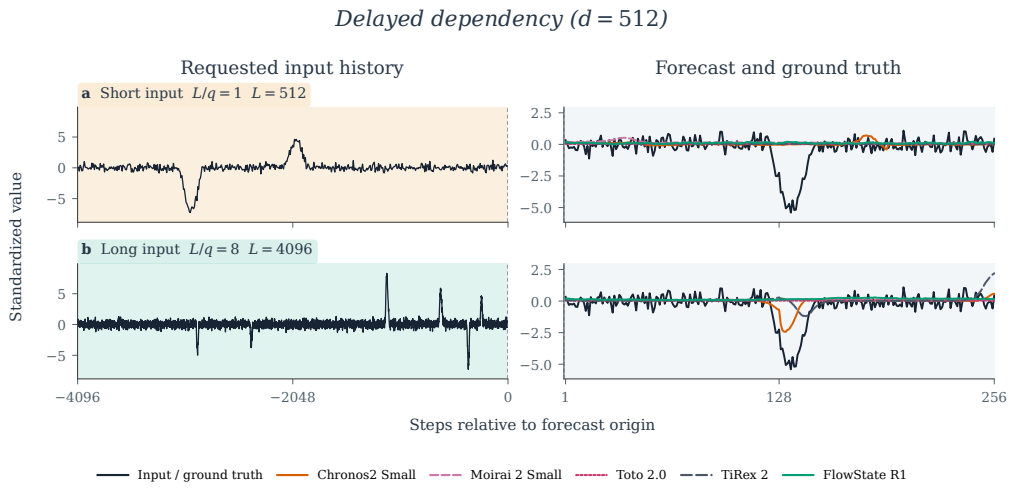

Figure 9: Delayed dependency. The long input reaches trigger events beyond the 512-step lag.

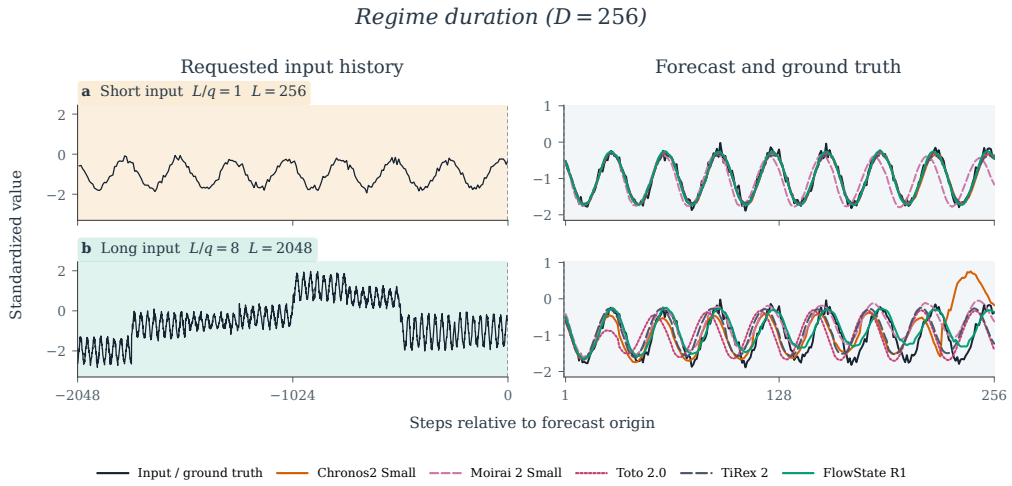

Figure 10: Regime duration. The long input shows a clear degradation.

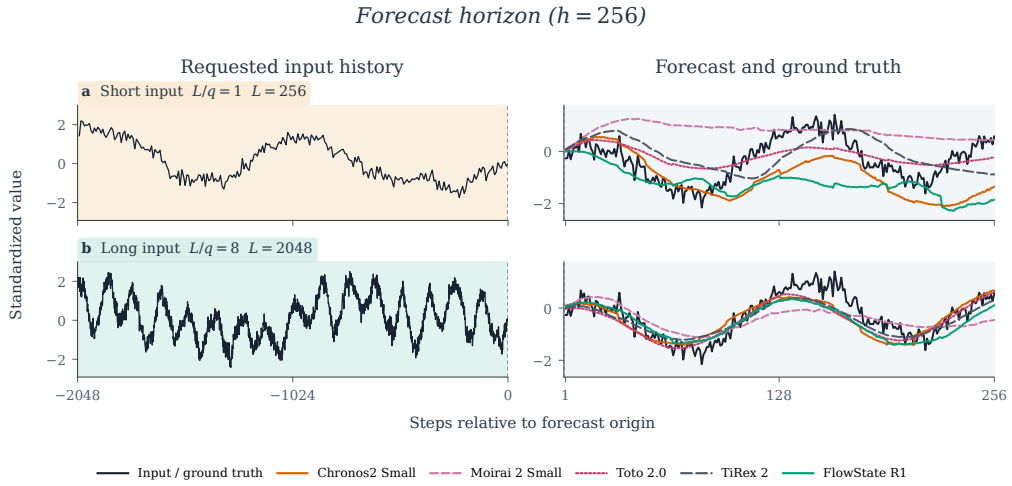

Figure 11: Forecast horizon. Short and long histories are compared against the same 256-step target.

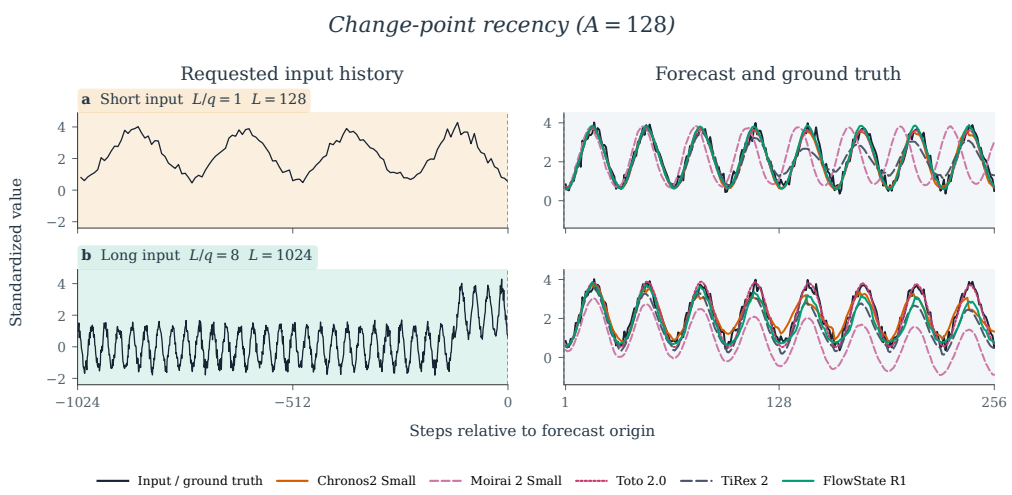

Figure 12: Change-point recency. Longer context crosses the recent break and includes pre-change observations.

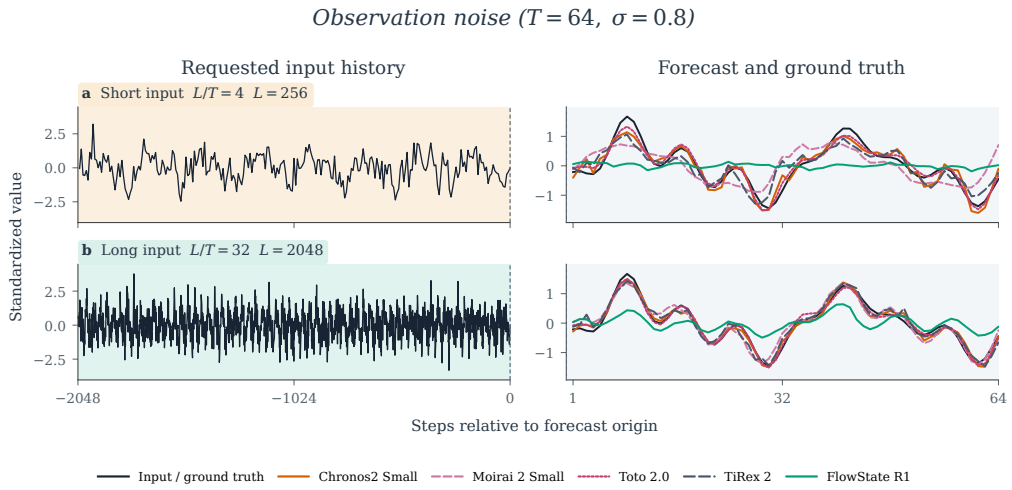

Figure 13: Signal-to-noise ratio. Repeated cycles help separate the latent periodic signal from observation noise.

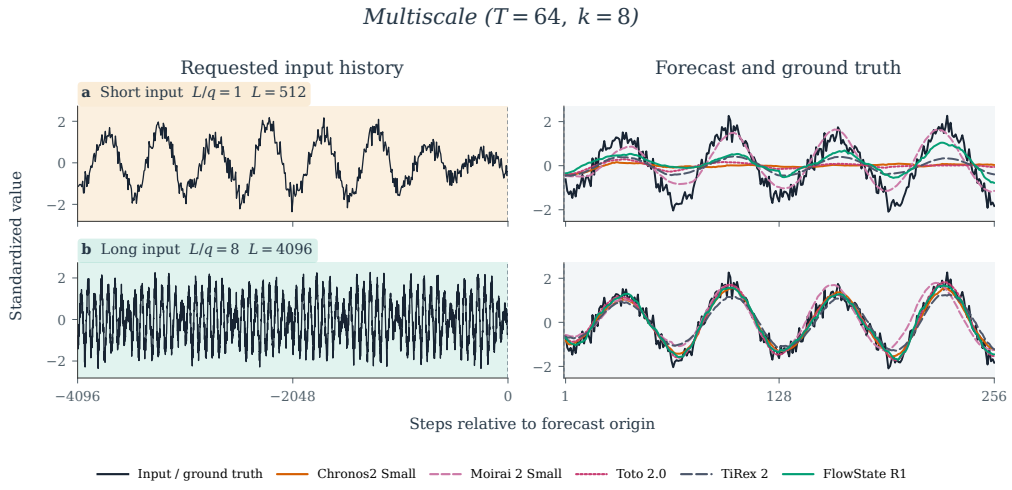

Figure 14: Multiscale periodicity. The long input spans the slow envelope beyond the 64-step carrier signal.

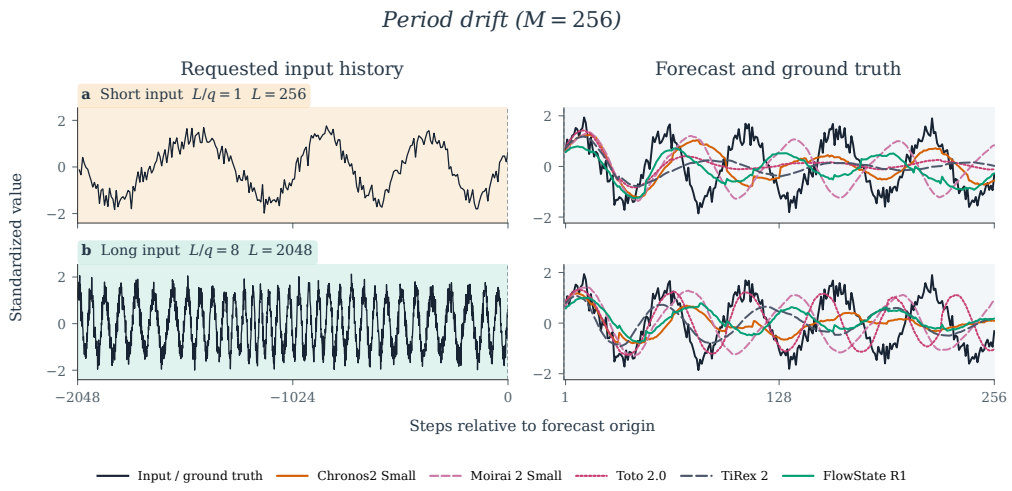

Figure 15: Period drift. Older cycles reflect increasingly stale local frequencies.

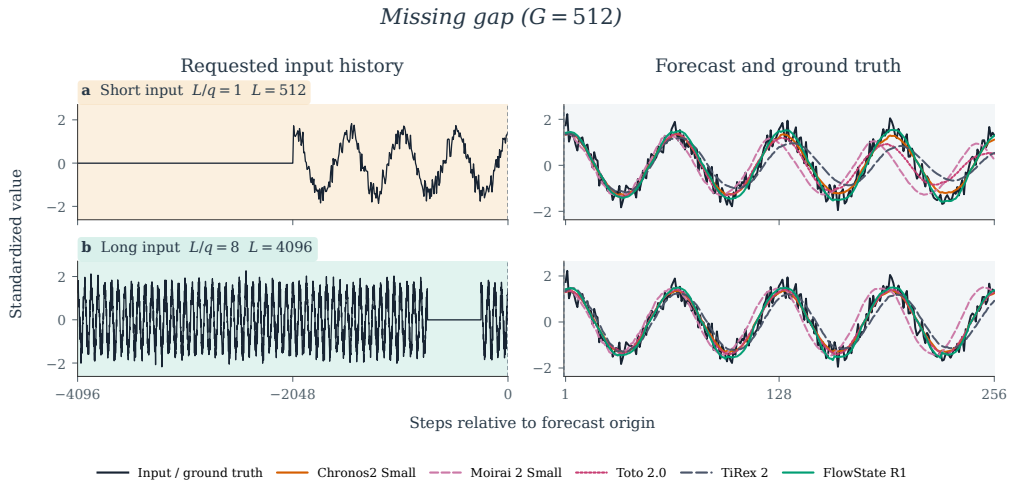

Figure 16: Missing gap. Longer context reaches valid observations beyond the 512-step raw-zero-filled interval.

B INTERVENTIONS AND CONTEXT-USE DIAGNOSTICS

This appendix gives the complete protocols and full-size results from the plots of Figure 2. It starts going through the common setup, temporal-block perturbations, oracle-verified planted-lag reach, and the attention ablation.

B.1 TEMPORAL-BLOCK PERTURBATIONS

We split the observed context into patches, using the original patch length in patch-based models and 32 observations otherwise. Patches are indexed by their position in terms of past-lags from the forecast origin. One patch is intervened on at a time, while total context length and patch position remain fixed. We use four interventions: replacement by the patch mean, within-patch permutation, substitution by a *matching* patch, and additive Gaussian noise at scales 0.1, 0.5, and 1.0. Mean replacement sets every value in the patch the mean of that patch. Permutation shuffles the values in the patch, preserving their distribution while destroying temporal order. Matching substitution uses an equal-length patch from the same series with similar mean and standard deviation, falling back to the same time slot from another series if none qualifies. Gaussian noise adds independent zero-mean draws with standard deviation equal to the chosen scale times the block's standard deviation (floored at 10^{-3}).

For model f and patch p, let $\Delta_{f,x}(p)$ be the absolute difference between perturbed and clean loss for series x. The displayed points on the curve report

$$S_m(p) = \frac{\frac{1}{N}\sum_i \Delta_{f,x}(p)}{\max_{p'} \frac{1}{N}\sum_i \Delta_{f,x}(p')}. \tag{11}$$

The normalization allows to compare the relative curve shape, rather than absolute sensitivity across architectures.

B.2 PLANTED-LAG REACH

Each univariate series follows

$$x_t = \left(\sum_{r=1}^{8} a_r x_{t-r}\right) + b x_{t-d} + \varepsilon_t, \text{ with } \varepsilon_t \sim \mathcal{N}(0, 0.01) \text{ and } d \in \{2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\}. \tag{12}$$

where f is the shared local component and d is the planted lag. The local coefficients have total absolute mass 0.6. We set $b = 0.95 - 0.6 = 0.35$, keeping the combined linear mass below the stability cap. To account for the possibility of distant lags d being not significantly relevant for the target forecast horizon, we retained only synthetic series that passed a control. Specifically, we compared the forecasting error made by two different Ridge regression models, one using only lags up to $t - r$, while the other also included $\{d - 1, d, d + 1\}$. If the error distribution was not significantly different we dropped the corresponding time series from our evaluation pool.

In Tab. 2 we report, for ease of readability, the same results as in Fig. 2 (top-right), showing how CSL changes as the lag of the long-range dependency d varies.

B.3 ATTENTION-MASS AUDIT

This diagnostic measures where native attention assigns probability without intervening in the forward pass. For full width W, retained suffix L, layer ℓ, head h, and token (patch) i, define the old-prefix mass as

$$M_{\ell hi}(L) = \sum_{j < W - L} A_{\ell hij}, \tag{13}$$

where A is the full length post-softmax attention matrix. We average this quantity over the heads and layers and over the token positions used to produce the forecast. These positions depend on the architecture: for example, the intervention uses the last observed-token query for Moirai 2 and Sundial, and the dedicated forecasting-token queries for Chronos-2. It covers the seven architectures whose native attention paths can be instrumented without intervening on the model computation.

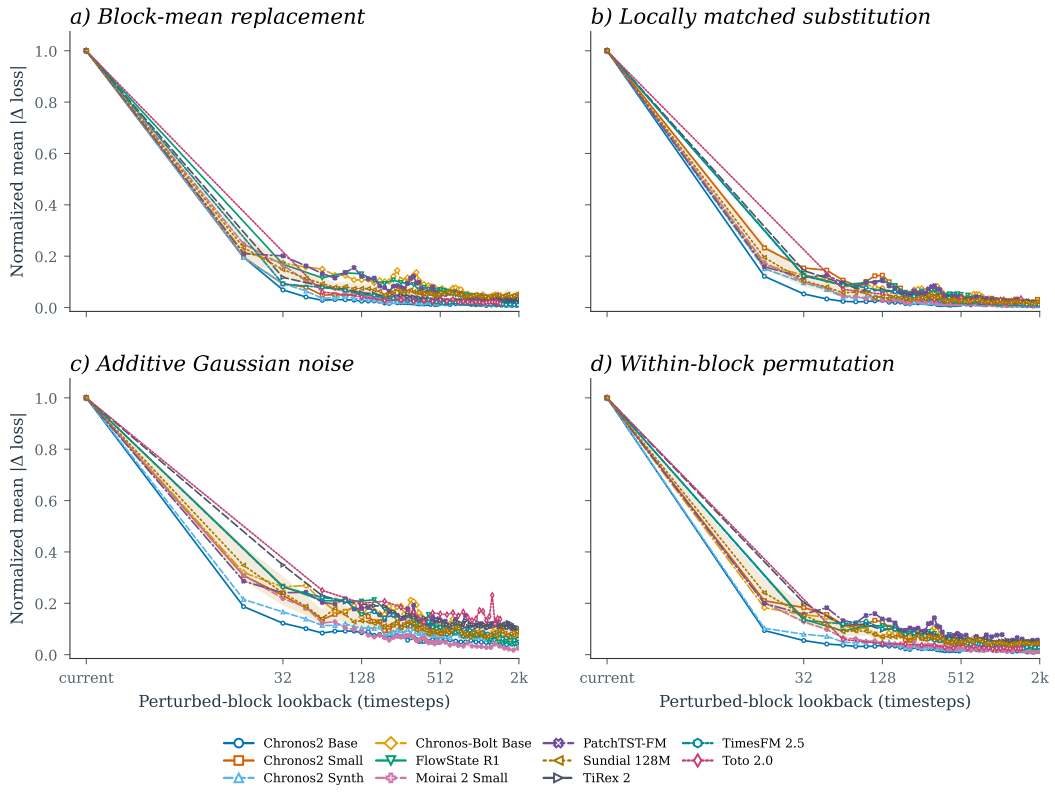

Figure 17: Temporal-perturbation profiles for all eleven TSFMs, separated by intervention family. Mean replacement, locally matched substitution, additive noise, and within-block permutation all concentrate their strongest effects near the forecast origin, showing that the recency ordering is not specific to mean replacement.

Table 2: Sufficient context $L_{5\%}$ for every model in the planted-lag sweep. Columns give planted dependency locations d. Bold entries reach the planted lag ($L_{5\%} \geq d$).

TSFM	$d = 128$	$d = 512$	$d = 1{,}024$	$d = 2{,}048$
Chronos2-Base	**384**	**1,536**	**2,048**	1,024
Chronos2-Small	**384**	**1,024**	**2,048**	**2,560**
Chronos2-Synth	**512**	**1,536**	**2,048**	**2,560**
ChronosBolt-Base	**384**	384	512	384
FlowState-R1	**384**	**512**	384	256
Moirai2-Small	**256**	**512**	512	256
PatchTST-FM-R1	**512**	**1,536**	**2,048**	**2,560**
Sundial-Base-128M	**256**	**768**	**1,024**	256
TiRex2	**768**	**1,536**	768	384
TimesFM2.5-200M	**512**	**1,536**	**2,048**	**2,560**
Toto-2.0-313m	**384**	**1,024**	**1,536**	**2,560**

For each series, the MAE penalty is measured relative to the full input forecast. This penalty is computed by zeroing the attentions weights after softmax normalization.

Table 3: Completed all-model cell-oracle audit. All eleven models use 97 cells. "Oracle save is the theoretical MACs reduction at the minimum-MASE oracle; "no-worse save" is the largest supported reduction whose normalized MASE does not exceed native. TiRex2 uses its state-cached execution cost.

Model	Full nMASE	Oracle nMASE	Δ MASE	Oracle save	No-worse save
Chronos2-Base	0.7047	0.6909	−1.97%	41.54%	82.46%
Chronos2-Small	0.7267	0.7047	−3.03%	53.39%	84.93%
Chronos2-Synth	0.7352	0.7157	−2.66%	46.04%	83.61%
ChronosBolt-Base	0.8096	0.7748	−4.30%	29.02%	75.87%
FlowState-R1	0.7014	0.6896	−1.68%	37.74%	81.08%
Moirai2-Small	0.7378	0.7138	−3.25%	50.72%	90.52%
PatchTST-FM-R1	0.7045	0.6952	−1.33%	29.08%	68.07%
Sundial-Base-128M	0.7507	0.7251	−3.41%	7.37%	15.94%
TiRex2	0.6967	0.6897	−1.00%	22.79%	59.92%
TimesFM2.5-200M	0.7050	0.6930	−1.71%	37.36%	82.68%
Toto-2.0-313m	0.7040	0.6932	−1.53%	21.73%	62.87%

C GIFT-EVAL ORACLE ANALYSIS AND COMPUTE ACCOUNTING

This appendix quantifies the hindsight opportunity available on GiftEval and defines the compute proxy used throughout the paper. It presents the detailed oracle ceiling, the cross-model oracle audit, and the architecture-specific theoretical MAC accounting in that order.

C.1 INPUT PROTOCOL AND LEADERBOARD COMPARABILITY

The primary oracle and selector results are univariate forecasts for every model. Channels of multivariate source datasets are flattened into separate forecasting instances; neither other target channels nor cross-series learning are supplied to the forecaster. All reported aggregate MASE scores use the geometric mean of cell MASE divided by the corresponding seasonal-naive reference.

Published GIFT-Eval submissions do not necessarily share this inference recipe. The official Chronos2 notebook enables both native multivariate inputs and cross-series learning (`predict_batches_jointly=True`), whereas TiRex2's evaluation documentation identifies independent univariate evaluation as the mode matching its published leaderboard results.[1] Consequently, agreement with a published score for one model does not imply that another model's leaderboard recipe is reproduced by our common protocol.

C.2 HINDSIGHT ORACLE ACCURACY-COMPUTE FRONTIER

For each model, we construct an accuracy-compute frontier over the GiftEval dataset-horizon cells. Let $c \in \{1, \ldots, N\}$ index a cell, where the current full-catalog evaluation has $N = 97$ cells. An action $a \in \mathcal{A}_c$ is either the full context or one of the feasible evaluated context lengths for cell c. A policy $\mathbf{a} = (a_1, \ldots, a_N)$ therefore chooses one context action per cell. The chosen action is shared by all forecast instances in that cell; the oracle does not choose a separate context for each instance.

Task-level accuracy. For action a, the task MASE is computed following the GluonTS seasonal MASE implementation (Alexandrov et al., 2019).

Task-level compute. Let $f_{c,a}$ be the mean theoretical model-forward compute for action a in cell c, evaluated using the action's effective served context widths and the cell's forecast horizon. In the cell-balanced analysis, these per-cell means are summed with equal weight across cells. Define the full length compute reference and the selected policy's compute as

$$F_{\text{full}} = \sum_{c=1}^{N} f_{c,\text{full}}, \qquad F(\mathbf{a}) = \sum_{c=1}^{N} f_{c,a_c}.$$

[1] Evaluation recipes: `https://github.com/SalesforceAIResearch/gift-eval/blob/main/notebooks/chronos-2.ipynb` and `https://github.com/NX-AI/tirex-2/blob/main/examples/gifteval/README.md`.

The reported compute saving is

$$S(\mathbf{a}) = 100 \left(1 - \frac{F(\mathbf{a})}{F_{\text{full}}}\right) \%.$$

These are theoretical forecast-compute estimates, rather than wall-clock timings. Under cumulative serving, a forecast that is unavailable at a requested grid length carries forward the largest evaluated smaller context for that instance; full context is the initial fallback. The same effective served widths are used in the compute accounting, including forecasts whose MASE is undefined.

Scalarization and frontier construction. For a nonnegative global trade-off parameter λ, the oracle minimizes

$$\mathcal{J}_\lambda(\mathbf{a}) = \frac{1}{N} \sum_{c=1}^{N} \log R_{c,a_c} + \lambda \frac{F(\mathbf{a})}{F_{\text{full}}}.$$

Because this objective is additive across cells, each cell's action can be selected independently for a given λ:

$$a_c^*(\lambda) \in \arg\min_{a \in \mathcal{A}_c} \left\{ \frac{1}{N} \log R_{c,a} + \lambda \frac{f_{c,a}}{F_{\text{full}}} \right\}.$$

Before this step, actions dominated within their cell—an action with both higher or equal MASE and higher or equal compute than another action, with at least one strict inequality—are removed.

Each remaining action defines a line in λ, with intercept $\frac{1}{N} \log R_{c,a}$ and slope $f_{c,a}/F_{\text{full}}$. The selected action can change only where two such lines in the same cell intersect. We compute the positive pairwise intersection values, sort the resulting breakpoints, and evaluate one λ in each interval, together with $\lambda = 0$ and a value beyond the last breakpoint. The minimizing action in every cell is constant within each interval, so this traces the distinct policies exposed by the global linear scalarization without relying on a manually chosen coarse grid of λ values. Repeated policies are deduplicated, and the resulting aggregate points are ordered by compute and filtered to retain points whose accuracy improves as compute increases.

The plotted frontier contains the supported points: policies that minimize this scalarized objective for some $\lambda \geq 0$. A discrete nondominated policy that is never optimal for any such λ is unsupported and is not included. The implementation samples the open intervals between breakpoints; it does not separately enumerate every possible combination of tied actions exactly at a breakpoint.

At $\lambda = 0$, the oracle selects the minimum-MASE action in each cell, without a compute penalty. Increasing λ places more weight on compute, so cells can switch to cheaper actions. The plotted coordinates are $Q(\mathbf{a})$ and $S(\mathbf{a})$.

Interpretation. This is a hindsight oracle: it uses the benchmark outcomes for the evaluated actions to choose the per-task action. It therefore measures diagnostic oracle headroom for this action set and task-sharing rule. It does not describe a deployable selector, which would need to choose contexts without access to the evaluation outcomes.

C.3 Theoretical MAC accounting

Convention and scope. The reported quantity is an inference multiply–accumulate (MAC) proxy, and all tables and figures label it as MACs. We count one real multiply–accumulate as one unit. Counting a multiplication and addition separately would approximately multiply all dense terms by two and would therefore cancel in the within-model saving ratios used here. The proxy is intended to compare context choices for the *same checkpoint and forecast horizon*; it is not a claim about hardware throughput.

For a dense Transformer block with sequence length L, hidden width d, and feed-forward width f, the three dominant terms in our estimator are

$$\mathcal{T}(L; d, f) = \underbrace{4Ld^2}_{Q,K,V,O} + \underbrace{2L^2 d}_{QK^\top \text{ and } AV} + \underbrace{2Ldf}_{\text{feed-forward network}}, \tag{14}$$

Table 4: Values encoded in the theoretical MAC estimator for the 11 evaluated checkpoints. $C_{\max}$ is the experiment context limit, except that Moirai's effective cap also depends on H and FlowState reports its model ceiling, with an effective cap depending on cadence and H. PatchTST's active TransformerBlock source sets `mlp_ratio=4.0`, giving $f = 4d$.

Checkpoint	d	f	P	Blocks	$C_{\max}$	Special output/path
Chronos2-Small	512	2,048	16	6	8,192	$q = 13$, REG token
Chronos2-Base	768	3,072	16	12	8,192	$q = 21$, REG token
Chronos2-Synth	768	3,072	16	12	8,192	$q = 13$, REG token
ChronosBolt-Base	768	3,072	16	12+12	2,048	$q = 9$, 64-step recursion
FlowState-R1	512	–	–	6	$\leq 16{,}384$	cadence-scaled S5/CPM, $q = 9$, $r = 256$
Moirai2-Small	384	1,024	16	6	$\leq 8,192$	$q = 9$, four prediction tokens
PatchTST-FM-R1	1,024	4,096	16	20	8,192	unified Transformer
Sundial-Base-128M	768	3,072	16	12	2,880	100-sample TimeFlow
TiRex2	512	custom	32	6+6	8,192	$q = 9$, 928-step future grid
TimesFM2.5-200M	1,280	1,280	32	20	15,360	128-step cached decode
Toto-2.0-313m	1,024	2,736	32	24	4,096	23 time $+1$ variate, $q = 9$

Thus context shortening is not generally quadratic: when $L \ll d$, projections and feed-forward layers can dominate the L^2 attention term. For a decoder of length L_D attending to encoder memory of length L_M, we additionally count

$$\mathcal{X}(L_D, L_M; d) = 2(L_D + L_M)d^2 + 2L_D L_M d, \tag{15}$$

covering decoder query/output projections, memory key/value projections, and the two cross-attention matrix products.

For raw context C, horizon H, and patch size P, we use $n_C = \max(1, \lceil C/P \rceil)$ and $n_H = \max(1, \lceil H/P \rceil)$. Ceilings are important: reducing C without crossing a patch boundary produces no modeled saving. The effective *native context* is the shorter of the series history and the checkpoint's supported maximum. Exact checkpoint identifiers are resolved before family defaults so, for example, Chronos2-Small is not priced as Chronos2-Base.

Table 4 records the parameters that determine the costs of the 11 principal checkpoints. A dash in the patch column means that the encoder consumes raw time steps. "6+6" for TiRex2 denotes six mLSTM and six sLSTM blocks rather than an encoder–decoder split.

Architecture-specific execution paths. The generic expression in Eq. 14 is used only where it matches the audited cost model. The following cases describe the accounting used for the results; measured wrapper behavior is treated separately below.

- **PatchTST-FM.** With $N = 20$ blocks and $L = n_C + n_H$, the architecture-level curve uses $N\mathcal{T}(L; d, 4d)$. It does not add input or output heads. The active source specifies the $4d$ MLP width. The released wrapper pads to a fixed 8,192-context grid, so shortening its input alone does not reduce the executed backbone. Our primary timed path instead compacts the batch-common, fully masked patch prefix while preserving learned positional indices and the wrapper's forecast slicing. The variable-L expression remains a backbone MAC proxy; actual execution also depends on patch alignment and the longest retained history in each batch.
- **TimesFM 2.5.** The first 128 forecast values are emitted directly from the last context token. For longer horizons, each additional 128-value block feeds four 32-value patches through all 20 layers with a KV cache. Let

$$K = n_C + 4 \left\lfloor \frac{\max(H, 1) - 1}{128} \right\rfloor.$$

  The cache is allocated at length K; summing query work across prefill and cached decode calls yields $\mathcal{T}(K; d, f)$ per layer. Unlike the earlier backbone-only approximation, the estimator also counts every executed residual tokenizer and output head:

$$\mathcal{E}_{\mathrm{TFM}}(K) = K\{(2P)d + d^2 + (2P)d\},$$
$$\mathcal{O}_{\mathrm{point}}(K) = K\{d^2 + 2d(128 \cdot 10)\},$$
$$\mathcal{O}_{\mathrm{quant}}(K) = K\{d^2 + 2d(1024 \cdot 10)\}.$$

  Thus its total proxy is $20\mathcal{T}(K; 1280, 1280) + \mathcal{E}_{\mathrm{TFM}} + \mathcal{O}_{\mathrm{point}} + \mathcal{O}_{\mathrm{quant}}$.

- **Toto 2.0.** With $L = n_C + n_H$, all 24 blocks execute Q/K/V/O projections and a three-matrix SwiGLU. Twenty-three blocks attend over the time sequence; the remaining variate block attends over a singleton sequence for each time token in this univariate evaluation. With $q = 9$, its proxy is

$$\begin{aligned}\mathcal{C}_{\text{Toto}}(C, H) = 23\{4Ld^2 + 2L^2d + 3Ldf\} + \{4Ld^2 + 2Ld + 3Ldf\} \\ + L\{(2P)(4d) + (4d)d + (2P)d\} \\ + n_H\{d(4d) + (4d)(qP) + d(qP)\}.\end{aligned}$$

  The final two lines count the residual patch embedding and nine-quantile output head. This preserves the actual distinction between temporal and singleton-variate attention rather than treating all 24 blocks as identical.

- **Chronos2.** The shipped model is an encoder over context patches, one regression token, and masked future patches, so $L = n_C + 1 + n_H$. Each of its N blocks contains time attention, group attention, and one feed-forward sublayer. GiftEval evaluates target series independently; group size is therefore one, but its projections are still executed. We count

$$N\big[\mathcal{T}(L; d, f) + 4Ld^2 + 2Ld\big] + \mathcal{E}_{\text{C2}}(L) + \mathcal{O}_{\text{C2}}(n_H), \tag{16}$$

  where $\mathcal{E}_{\text{C2}}$ is the residual input embedding from $3P$ patch features and $\mathcal{O}_{\text{C2}}$ is the checkpoint's quantile output head. More explicitly, with q output quantiles,

$$\begin{aligned}\mathcal{E}_{\text{C2}}(L) &= L(3Pf + fd + 3Pd), \\ \mathcal{O}_{\text{C2}}(n_H) &= n_H\{df + f(qP) + d(qP)\}.\end{aligned}$$

  Width, depth, and q are separately resolved for Small, Base, and Synth checkpoints.

- **Chronos-Bolt.** One decoder query directly emits a 64-step block. For horizons beyond 64, later blocks are evaluated along nine quantile histories. Let $B = \max(1, \lceil H/64 \rceil)$, $m_0 = 1$, $m_b = 9$ for $b > 0$, and $n_b = \lceil \min(C + 64b, C_{\max})/P \rceil$. Its cost is

$$\sum_{b=0}^{B-1} m_b \left[N_E\mathcal{T}(n_b; d, f) + N_D\{\mathcal{T}(1; d, f) + \mathcal{X}(1, n_b; d)\} + \mathcal{E}_{\text{Bolt}}(n_b) + \mathcal{O}_{\text{Bolt}}\right]. \tag{17}$$

  Here $\mathcal{E}_{\text{Bolt}}(n) = n(2Pf + fd + 2Pd)$ and $\mathcal{O}_{\text{Bolt}} = df + f(64q) + d(64q)$ for $q = 9$. This recursion is a major special case: a horizon of 65 is much more costly than a horizon of 64, rather than only $65/64$ as costly.

- **Moirai 2.0.** Each call appends four prediction tokens and emits $4P$ samples. Its full-native context cap is $\min\{8{,}192,\ P(512 - \lceil H/P \rceil)\}$, so context plus prediction tokens fit the 512-token limit. For call b, we use $L_b = \min\{512, \lceil (C + 4Pb)/P \rceil + 4\}$ and multiplicity one for the first call and nine thereafter. Its gated feed-forward network has three rather than two dense matrices, giving the per-block term $4L_bd^2 + 2L_b^2d + 3L_bdf$. The implemented cost is the sum of this term over all $N = 6$ blocks and $\max(1, \lceil H/(4P) \rceil)$ recursive calls; no separate Moirai embedding or output-head term is added.

- **Sundial.** Only the context patches enter the 12-block Transformer. Its source uses a three-matrix gated feed-forward layer, so define $\mathcal{T}_{\text{gate}}(L; d, f) = 4Ld^2 + 2L^2d + 3Ldf$ rather than using Equation 14's two-matrix FFN. The forecast is produced by a TimeFlow sampler with 100 samples, 50 flow steps, three residual blocks, and a fixed 720-value output. We count

$$N\mathcal{T}_{\text{gate}}(n_C; d, f) + \mathcal{E}_{\text{Sun}}(n_C) + \max(1, \lceil H/720 \rceil)\,\mathcal{F}_{720}, \tag{18}$$

  where $\mathcal{E}_{\text{Sun}}(n) = n(2Pf + fd + 2Pd)$. The sampler term used in code is

$$\mathcal{F}_{720} = 100 \cdot 50\left[720d + (256d + d^2) + d^2 + 3(5d^2) + 2d^2 + 720d\right].$$

  This large, context-independent term explains why aggressive shortening can yield little total MACs saving. Within one 720-step output block, changing H does not change the proxy.

- **FlowState.** FlowState consumes raw observations rather than $P = 6$ context patches; six is the decoder span at the neutral cadence. Each S5 block has complex B and C projections and one real $d \to d$ output gate. There is no $d \to 2d$ gated FFN. To express the complex

kernels in the same real-MAC units as the dense models, one executed complex matrix multiply–accumulate is counted as four real MAC-equivalent units.

The released GiftEval wrapper is cadence-conditioned, so the executed path is not a function of (C, H) alone. For the cell-specific scale factor s, define

$$\begin{aligned} D_s &= \lfloor 6/s + 10^{-6} \rfloor, & M &= \max\{0, H - D_s\}, \\ C_s &= \min\{16384, \lfloor 4096/s \rfloor\}, & C_{\text{obs}} &= \min\{C, C_s - M\}, \\ L &= C_{\text{obs}} + M, & S &= M + 1. \end{aligned}$$

The 4,096-step pretraining budget is cadence-scaled before the 16,384-step model ceiling is applied; it is not a fixed raw-input cap. The validated preparation trims leading missing values, selects the requested suffix, and repeats the leading-missing trim, so costs use the actually consumed width. Here M is the number of masked future values appended for contiguous-patch prediction, L is the sequence actually processed by S5, and S is the number of final-layer states decoded. We use the same per-dataset s as the cached forecasts, reconstructed from the GiftEval sampling frequency and domain by IBM's public recipe.

Only scales and horizons with $D_s \geq 1$ and $C_s - M \geq 1$ are valid; unsupported combinations are rejected. IBM's seasonal scaling rule is $s = 24/P_s$, where P_s is the seasonal period in observations, so the uncapped budget spans $4096/24$ seasonal cycles rather than a fixed number of elapsed hours. For daily data with a weekly cycle, $s = 24/7$ gives 1,194 observations before the horizon reservation; the annual-cycle setting $s = 24/365$ reaches the 16,384 ceiling. For the refreshed five-minute evaluation, $s = 1/12$ gives $D_s = 72$ and $C_{\text{obs}} = \min\{C, 16384 - \max(0, H - 72)\}$.

The first five layers emit all L positions, giving $4Ld^2$ for each of B and C plus Ld^2 for the gate. The final layer still applies B and the FFT to all L positions, but applies C and the output gate to only the S CPM origins. With $F(L, d) = 6Ld \log_2(\max\{2, 2L\})$ for two forward FFTs and one inverse FFT, $N = 6$, nine quantiles, and functional-basis width $r = 256$, we use

$$\begin{aligned} \mathcal{C}_{\text{FS}}(C, H) = {} & 2Ld + (N-1)\{9Ld^2 + F(L, d)\} \\ & + \{4Ld^2 + 5Sd^2 + F(L, d)\} \\ & + S\{dqr + qD_s r\}. \end{aligned} \tag{19}$$

The FFT term remains an explicit approximation over a zero-padded sequence; the dense projection, cadence-dependent truncation, CPM extension, and final-state slicing terms follow the executed source. At fixed (H, s) its context scaling is $O(C \log C)$ until the internal cadence cap is reached, rather than quadratic attention scaling.

- **TiRex2.** The released wrapper pads or truncates every input to 8,192 context values, appends a fixed 928-step future region, and evaluates both the input and its sign flip. Its public-wrapper backbone cost is therefore invariant to selected context and is multiplied by $2\max(1, \lceil H/928 \rceil)$. To state the implemented proxy explicitly, let $L_T = \lceil 8192/P \rceil + \lceil 928/P \rceil$, $q = 9$, and

$$\begin{aligned} I &= L_T\{(2P)2048 + 2048d + (2P)d\}, \\ F &= 3d(1408), \qquad V = 4d^2 + 2d(2048), \\ M &= 2(256)d + 3d^2 + 2(4)d + d(1024) + F, \\ S &= 524288 + d(1024) + F, \\ O &= L_T\{d(2048) + 2048(qP) + d(qP)\}. \end{aligned}$$

The public-wrapper estimate is

$$\mathcal{C}_{\text{TiRex}}(H) = 2\max(1, \lceil H/928 \rceil)\,[I + 6L_T\{M + S + 2V\} + O]\,. \tag{20}$$

Here M and S are the respective mLSTM and sLSTM block terms and V is one variate stack. Elementwise recurrent operations are not counted. This function gives zero saving when only the selected context changes.

The TiRex2 results labelled *state-cached* do not come from Eq. 20. They use a separate executed-grid calculation after validating cached continuation of the repeated missing prefix. For each cell, the selected and native grids are respectively $\lceil (C^\pi + 928)/32 \rceil$ and

$\lceil (C^{\text{native}} + 928)/32 \rceil$ tokens, multiplied by the instance count and $2\lceil H/928 \rceil$ calls before summing. Thus these percentages are recurrent-grid savings, used as a linear MAC proxy for the state-cached path; they are not a full recount of the fixed overheads in Eq. 20. The unpadded wrapper used the same variable-grid cost expression only as a sensitivity, because removing rather than caching the missing prefix can change short-context forecasts.

TiRex2 state preloading and PatchTST state compaction. For these two models, official wrappers add left-padding to bring the input sequence to a fixed maximum length. We bypass this implementation by dropping the padding directly in the case of PatchTST, while with Tirex2 we precompute the states of the padding inputs and just load the model with those states before inference. The error introduced by this methods is negligible.

D ZERO-SHOT CONTEXT PREDICTOR

This appendix specifies and audits the zero-shot candidate selector. It first gives the complete synthetic selector protocol, the PatchTST-FM and TiRex2 execution techniques, and additional robustness and negative results.

D.1 COMPLETE SYNTHETIC SELECTOR PROTOCOL

This subsection specifies the frozen ExtraTrees policies used for the aggregate profile headline. The five-refit robustness table uses the separately split protocol described in Section 5.2. Each series has 15,360 observed positions followed by up to 1,024 targets and contains one to three periodic components (sine, cosine, sawtooth, or square wave) with amplitudes sampled uniformly from $[0.5, 2]$. Periods use a two-component log-uniform mixture: 75% on $[16, 2048]$ and 25% on $[2048, 15360]$. A segment has an AR(1–3) component with probability 0.5, a degree-1–3 polynomial trend with probability 0.5, Gaussian noise with standard deviation in $[0.05, 0.30]$, and impulse spikes with probability 0.3. Series contain one to four regimes according to weights $(1, 2, 2, 1)$, with minimum length 512, cumulative level-shift standard deviation 1.5, amplitude multiplier in $[0.6, 1.6]$, seasonal-change probability 0.5, and transition half-width in $[0, 64]$. Ten percent are shortened to a uniformly sampled genuine length in $[500, 15359]$ and left padded. Standardization uses context positions only (including the prescribed left padding); no target value is used in preprocessing or selector features.

For these TSFMs, one seed permutation chooses 12,000 series for fitting and a disjoint 3,000 for validation from this pool. Six horizons $\{16, 32, 64, 128, 512, 1024\}$ are labeled. From the training split we sample 600,000 series–horizon–window triples uniformly. Triples with non-finite or non-positive candidate or full length MAE are discarded. We clip the value of risk targets r_L to the interval $[-3, 3]$ to avoid large magnitude values dominating the regression.

We define six different policies for the models, from the one which does not tolerate error at all, i.e., *Conservative*, to the one which prioritizes computational savings the most, i.e., *Max efficiency*. The error tolerance profiles used to calibrate each policy are shown in Tab. 5.

Policy calibration For each of the policies, on the held-out synthetic set we search $u \in \{0, 0.5, 1, 1.645\}$ and $v \in \{0, 0.1, 0.25, 0.5\}$. The values of u and v are calibrated for the Balanced policy and re-used across all policies, then τ_q is adjusted on a per policy basis. To calibrate the policy parameters we compute the relative error R_f over a grid of possible (u, v, τ_q) values and select the combination that minimizes the average relative error, while satisfying the constraints in Tab. 5. We evaluate the following possible lengths to be selected:

$$\mathcal{G} = \{32, 48, 64, 96, 128, 192, 256, 384, 512, 768, 1024, 1536,$$
$$2048, 2560, 3072, 4096, 6144, 8192, 12288, 15360\}.$$

We truncate the set of possible lengths if it exceeds the maximum supported by the considered model. The selected score weights are shown in the Tab. 6.

Aggregated results for additional policies Tab. 7 reports the computational savings and corresponding error change for all of the policies we instantiated, including those not reported in the main text. Results are aggregated across the GiftEval benchmark and the 11 considered TSFMs, considering the univariate setting.

Table 5: Synthetic calibration limits for the six selector profiles.

Profile	Maximum error exceedance probability ↓	Maximum avg. error degradation cap ↓
Conservative	0.5%	0%
Balanced	1%	0.1%
Aggressive	3%	0.5%
Very aggressive	7.5%	2%
Efficiency	15%	4%
Max efficiency	20%	7%

Table 6: Frozen score weights and synthetic-calibrated cutoffs in $q = \widehat{\mu} + u\widehat{\sigma} + v\widehat{p}^{(5)}$ (cutoffs are in score units).

TSFM	u	v	τ_{C}	τ_{B}	τ_{A}	τ_{VA}	τ_{E}	τ_{ME}
Chronos2-Base	0	0.50	0.047664	0.058990	0.111308	0.172395	0.283600	0.348552
Chronos2-Small	0	0.25		0.007291	0.017047	0.040669	0.077955	0.104572
Chronos2-Synth	0	0.25	0.014813	0.024272	0.043145	0.065463	0.101177	0.137315
ChronosBolt-Base	1.645	0		0.064077	0.093839	0.128969	0.177930	0.214149
FlowState-R1	0	0.50	-0.190851	0.024855	0.057199	0.107990	0.154217	0.188976
Moirai2-Small	0.5	0.25	0.035626	0.046911	0.074859	0.109102	0.153389	0.184767
PatchTST-FM-R1	1.0	0	0.031581	0.040285	0.057257	0.090302	0.139273	0.167332
Sundial-Base-128M	1.645	0.50	0.013685	0.088360	0.120439	0.169445	0.226470	0.262281
TimesFM2.5-200M	0	0.25	0.000362	0.007527	0.018838	0.052346	0.089686	0.125124
TiRex2	0	0.50	0.047306	0.063062	0.098532	0.152738	0.209187	0.251411
Toto-2.0-313m	0	0		-0.208158	-0.051433	-0.022991	0.012534	0.044273

D.2 Selector feature schema

The following highlights the handcrafted features used by our proposed selector implementation, grouping them conceptually:

- **Valid-history coverage (2):** the linear and logarithmic fractions of the policy's feature cap that are observed.
- **Multiscale suffix summaries (55):** eleven features at each active scale in $\{32, 128, 512, 2048, 8192\}$. They record suffix coverage; standardized mean, standard deviation, last value, and endpoint change; mean absolute and standard deviation of first differences; the 5th–95th percentile range; availability of the preceding block; and standardized mean and log-scale shifts relative to that block.
- **Autocorrelation (22):** correlation and a separate availability indicator at each lag in $\{1, 2, 4, 8, 16, 24, 48, 96, 168, 336, 720\}$, avoiding the ambiguity between an unavailable lag and a measured zero correlation.
- **Spectrum (5):** normalized spectral entropy, dominant-frequency power share, log peak period, low-frequency power share, and spectral centroid, computed from the final at most 4,096 observations.
- **Candidate geometry (10):** linear and logarithmic requested- and effective-window fractions, normalized log horizon, three horizon/window ratios, a policy-cap indicator, and an indicator that the candidate truncates valid history.

Table 7: Selector profiles on GiftEval (%), reported as mean ± sample standard deviation over five distinct-seed refits.

Method	MAC save ↑	ΔMASE ↓
Synthetic-calibrated selector		
Conservative	2.59 ± 0.72	$+0.006 \pm 0.017$
Balanced	5.89 ± 0.66	$+0.050 \pm 0.012$
Aggressive	15.13 ± 1.41	$+0.088 \pm 0.019$
Very aggressive	27.41 ± 1.99	$+0.210 \pm 0.045$
Efficiency	43.15 ± 2.08	$+0.541 \pm 0.092$
Max efficiency	51.45 ± 1.92	$+0.916 \pm 0.109$