THE PRICE OF GENERALITY IN TIME SERIES FOUNDATION MODELS: GRAIN FOR YEARLY FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Foundation models are typically trained as generalists, optimised for average predictive performance across broad and heterogeneous tasks. Motivated by the classical James-Stein phenomenon, we develop a theory that explains why such models can remain suboptimal on specific domains despite strong aggregate performance. We interpret pretraining as empirical Bayes-risk minimisation under a corpus-induced task prior, and define the *price of generality* as the excess risk incurred on a target domain when using a rule optimised for a broader population. While the argument applies broadly to foundation models, it has practical relevance mostly for cases with a limited amount of information in the prompt. Consequently, we study yearly forecasting as a concrete application of this theory. Yearly series are practically important but systematically underrepresented in existing pretraining corpora, and their short histories make predictions more sensitive to prior misspecification. Following our developed theory, we introduce GRAIN, a specialist prior-data fitted network trained on simulations from a posterior-informed prior estimated from real yearly series. On the yearly subset of the GIFT-Eval, GRAIN outperforms generalist foundation models and statistical baselines without data leakage, supporting the benefit of aligning the training prior with the target task population.

1 INTRODUCTION

Recently, time series foundation models have achieved impressive zero-shot forecasting performance by leveraging large-scale pretraining on a diverse collection of data. Models and model families such as Chronos (Ansari et al., 2024; 2025), TimesFM (Das et al., 2024), Moirai (Woo et al., 2024) and TabPFN-TS (Hoo et al., 2025) demonstrate that a single pretrained foundation model can generalise across a wide range of forecasting tasks without retraining. These developments have established large-scale pretraining as a promising paradigm for learning transferable representations in time series forecasting.

However, strong average performance across heterogeneous tasks does not guarantee equally strong performance in every forecasting regime. Existing foundation models are usually trained by minimising average loss over the entire corpus, which can lead to regimes that contribute relatively little to the training objective being dominated by more prevalent data. This problem is likely to be more obvious for short time series, whose limited histories may provide insufficient evidence to distinguish the data generating processes from other unrelated processes encountered during pretraining. Yearly series, in particular, expose both limitations. They generally have short histories and are substantially underrepresented in existing pretraining corpora. For example, Chronos (Ansari et al., 2024, Table 3) includes no yearly data in its real-data pretraining corpus, although its zero-shot evaluation contains four yearly datasets. The model has to perform yearly forecasting entirely through cross-frequency transfer and frequency-agnostic synthetic series. Indeed, it loses against a simple combination of univariate statistical models (Petropoulos and Svetunkov, 2020) on low-frequency tasks, including those involving yearly series. It is not surprising to see current time series foundation models fail in such specific, underrepresented domains.

In fact, low-granular series remain an important use-case in real-world applications. The sampling frequency is often not a mere technical shortcoming, as many time series have an inherent frequency which cannot be increased by more frequent measurement. Take yearly or half-yearly school in-

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.

takes or yearly tax-related reporting as examples. Many economic, demographic, and administrative quantities are defined, collected, and reported according to annual cycles. One of the leading sources of macroeconomic data, the federal reserve economic data (FRED) database,[1] is dominated by low-frequency time series. It contains over 800,000 series across multiple temporal resolutions and over half of the series are annual. GIFT-Eval (Aksu et al., 2024), a commonly used benchmarking dataset, includes yearly time series from the M3 (Makridakis and Hibon, 2000) and M4 (Makridakis et al., 2020) competitions, as well as the tourism dataset (Hyndman and Athanasopoulos, 2021). These yearly series arise from diverse real-world applications, including demographic statistics, government expenditures, and veteran administration. Forecasts are produced naturally at an annual resolution because the relevant decisions are made annually, and each observation represents a complete yearly measurement. While high-frequency measurement in these environments is often neither natural nor practically available, many of these quantities could arguably in principle be measured at slightly higher frequencies such as quarterly or monthly. However, doing so may be administratively infeasible or inconsistent with the institutional processes that generate them, and is therefore not done in practice. Thus, we emphasise the importance and relevance of yearly forecasting tasks and argue that yearly series are substantially underrepresented in the existing foundation model space.

Our contribution then is twofold. First, we provide a formal theoretical argument for the limitations of generalist foundation models and the circumstances in which they cannot outperform specialist models, through a Bayesian decision-theoretic interpretation and the James-Stein phenomenon. While this theory applies broadly to foundation models, it is particularly relevant in the case of short, low-frequency series such as yearly data. Secondly, motivated by these theoretical results, we develop GRAIN (Grounded pRior-Aware INference), a specialist foundation model for yearly forecasting. Based on the prior-data fitted network (PFN, Müller et al., 2021) paradigm, GRAIN is trained on synthetic series generated from a posterior-informed specialist prior constructed by fitting a general-purpose Bayesian forecasting model to real yearly data. This alignment of the task prior with the target domain enables GRAIN to specialise in yearly forecasting. We show that GRAIN outperforms all other models not flagged for data leakage on the yearly subset of the GIFT-Eval benchmark (Aksu et al., 2024).

2 Background

In forecasting, foundation models include both pretrained large language models (LLMs) repurposed for time series forecasting, as well as specialised large models developed specifically for time series. Examples include, e.g., PromptCast (Xue and Salim, 2023), LLMTime (Gruver et al., 2023), and Time-LLM (Jin et al., 2024), and in the second group, TimeGPT-1 (Garza and Mergenthaler-Canseco, 2023), Lag-Llama (Rasul et al., 2023), Moirai (Woo et al., 2024), Chronos (Ansari et al., 2024; 2025), TimesFM (Das et al., 2024), and TinyTimeMixers (Ekambaram et al., 2024).

Prior-data fitted networks approximate Bayesian inference by training a neural network using data generated from a predefined prior (Hollmann et al., 2025). Large collections of synthetic data are sampled from this prior and a network is trained to learn a mapping that produces the corresponding posterior predictive distribution. Once trained, Bayesian prediction can be approximated for a new task in a single forward pass, without explicitly computing or sampling from the posterior. For example, Hollmann et al. (2022) introduced TabPFN, a transformer trained on synthetic tabular classification problems generated from a prior over structural causal models. Extending the same principle to time series, TabPFN-TS (Hoo et al., 2025) provides a wrapper for TabPFN to build suitable prompts for time series, and Dooley et al. (2024) proposed ForecastPFN, a transformer trained entirely on synthetic time series sampled from a predefined prior over forecasting processes.

A key challenge in developing PFNs is constructing suitable training data. Existing PFNs primarily rely on synthetic data generated from predefined probabilistic models such as Gaussian processes (Ansari et al., 2024). These approaches provide a powerful means of expanding the training corpus, yet the resulting models are inherently constrained by the assumptions of the simulation process. If the simulations do not sufficiently reflect real-world characteristics, the resulting model can fail to transfer effectively to practical forecasting scenarios. We instead develop a simulation procedure that preserves the characteristics of real-world time series, detailed in Section 4.

[1] https://fred.stlouisfed.org/

3 THEORY: THE PRICE OF GENERALITY IN FOUNDATION MODELS

Foundation models in forecasting are pretrained on broad and heterogeneous collections of time series. In this sense, they are technically global models (Montero-Manso and Hyndman, 2021) because a single shared model is learnt across many series. Yet they are rarely evaluated in the same way they are trained. Pretraining minimises aggregate forecasting loss over a broad task population, whereas downstream evaluation often focuses on a particular dataset or domain. This mismatch between training and evaluation population is central to the argument developed in this section.

This issue arises whenever the target population differs from the pretraining population and therefore applies broadly to foundation models, including LLM-based forecasters. In conventional LLM applications, however, rich prompts and large context windows can substantially reduce model-selection uncertainty. On the other hand, for time series foundation models, the observed series itself serves as the prompt. The available context is especially limited for short, low-frequency time series, and it is harder to eliminate the ambiguity about the underlying data-generating regime.

We develop this argument in two steps. We start from the classical James-Stein phenomenon to illustrate why global modelling and foundation models can be effective overall while worsening performance on particular components. We then formalise the argument using a Bayesian decision-theoretic framework and quantify the resulting excess risk, which we call the price of generality.

3.1 JAMES-STEIN AS A MOTIVATION FOR GLOBAL MODELS AND FOUNDATION MODELS

Global forecasting models are often effective because they pool information across related or even weakly related series, improving average performance through shared statistical strength (Montero-Manso and Hyndman, 2021). The James-Stein phenomenon offers a complementary explanation for this effect. Under total squared-error risk, a common shrinkage estimator can dominate separate estimation (Stein, 1956; James and Stein, 1961).

James-Stein also provides a good argument for why global models can fail in particular cases. The dominance of a common shrinkage estimator is under aggregate risk, which does not guarantee lower risk for every coordinate. As such, in a forecasting setting, this implies that global models, including foundation models, can be strongly preferred on average while remaining suboptimal for some series or subpopulations, particularly when they receive little effective weight in pretraining, and/or if short histories are insufficient to identify the correct latent regime.

3.1.1 THE JAMES-STEIN PHENOMENON

Consider estimating the mean vector of p independent normal distributions,

$$y_i \sim N(\mu_i, \sigma^2), \quad i = 1, \dots, p,$$

given the value of σ^2. We observe $\mathbf{y} = (y_1, \dots, y_p)$ with one observation per coordinate. The coordinate-wise maximum likelihood estimator and its total risk are

$$\hat{\boldsymbol{\mu}}^{\mathrm{MLE}} = (y_1, \dots, y_p), \text{ and } R(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}^{\mathrm{MLE}}) = \sum_{i=1}^{p} \mathbb{E}_{\mu_i}[(y_i - \mu_i)^2] = p\sigma^2 \tag{1}$$

under quadratic loss. For $p \geq 3$, the James-Stein estimator (Stein, 1956; James and Stein, 1961) has strictly smaller total risk,

$$\hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \left(1 - \frac{(p-2)\sigma^2}{s}\right)\mathbf{y}, \text{ and } R(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}^{\mathrm{JS}}) = p\sigma^2 - (p-2)^2\sigma^4\mathbb{E}_{\boldsymbol{\mu}}\left[\frac{1}{s}\right] < p\sigma^2, \tag{2}$$

where $s = \sum_{i=1}^{p} y_i^2$. The shrinkage factor improves the aggregate risk by shrinking all coordinates through the common factor determined by s. This improvement, however, does not imply lower risk for *every* coordinate. To see this, assume $\sigma^2 = 1$ and consider the true mean vector $\boldsymbol{\mu} = (\sqrt{p}, 0, \dots, 0)$. For a hypothetical realisation $\mathbf{y} = (\sqrt{p}, 0, \dots, 0)$,

$$\hat{\boldsymbol{\mu}}^{\mathrm{MLE}} = (\sqrt{p}, 0, \dots, 0), \text{ and } \hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \left(\frac{2}{\sqrt{p}}, 0, \dots, 0\right).$$

From (1), the risk of each coordinate for $\hat{\boldsymbol{\mu}}^{\mathrm{MLE}}$ is constant and equals one. For $\hat{\boldsymbol{\mu}}^{\mathrm{JS}}$, the first coordinate, which carries strong signal, is heavily shrunk towards the origin. The average risk and the first-coordinate risk of the James-Stein estimator for large p can be approximated by

$$\frac{1}{p}R(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}^{\mathrm{JS}}) \approx 1 - \frac{(p-2)^2}{2p^2}, \text{ and } R(\boldsymbol{\mu}, \hat{\mu}_1^{\mathrm{JS}}) \approx 1 - \frac{p-2}{p} + \frac{p^2-4}{4p},$$

respectively. Derivation details are provided in Appendix A. For example, at $p = 50$, an illustrative comparison is provided in Table 1. The James-Stein estimator performs better than the maximum likelihood estimator on average but has a much higher risk on the strongest coordinate. The distinction is especially important for foundation models. They are trained globally on a large pretraining corpus but are frequently used on one specialised downstream dataset. If that target dataset receives little effective weight in pretraining, or if short histories do not identify the correct latent regime, the global model remains influenced by irrelevant regions of the training population. This is exactly the setting in which, for example, underrepresented yearly series can be disadvantaged.

Estimator	Average risk	Risk on first coordinate
$\hat{\boldsymbol{\mu}}^{\mathrm{MLE}}$	1.00	**1.00**
$\hat{\boldsymbol{\mu}}^{\mathrm{JS}}$	$\approx$ **0.54**	$\approx$ 12.52

Table 1: Average risk per coordinate and risk on the first coordinate calculated for the two estimators when $p = 50$. The lowest value per column is presented in boldface.

3.1.2 An (Empirical) Bayesian View of James-Stein

The James-Stein estimator can be interpreted through a Bayesian lens; we find this very instructive, as it serves as a perfect motivation for our description of foundation models as (empirical) Bayesian procedures. Specifically, if we assign to the unknown μ_i independent normal distributions with unknown common variance ξ^2, we have the Bayesian hierarchy

$$y_i \,|\, \mu_i \sim N(\mu_i, \sigma^2) \text{ and } \mu_i \,|\, \xi^2 \sim N(0, \xi^2). \tag{3}$$

If ξ^2 is considered known (i.e., it is specified), then the posterior mean of μ_i is given

$$\mathbb{E}\left[\mu_i \,|\, y_i\right] = \left(\frac{\xi^2}{\sigma^2 + \xi^2}\right) y_i = \left(1 - \frac{\sigma^2}{\sigma^2 + \xi^2}\right) y_i. \tag{4}$$

This shrinks the regular least squares estimate towards zero by an amount determined by ξ^2. This procedure has the *potential* to improve average risk, and has a greater advantage if the μ_i happen to have a variance close to ξ^2. In general, this is not something we will know *a priori*.

The James-Stein procedure takes a different view: it treats the form of the prior as specified, but the specific prior distribution as an unknown that needs to be estimated from the collection of p problems. Specifically, under (3), $Y_i = \mu_i + \varepsilon_i$, with $\varepsilon_i \sim N(0, \sigma^2)$ and $\mu_i \sim N(0, \xi^2)$, so the marginal distribution is $Y_i \sim N(0, \xi^2 + \sigma^2)$. We may then leverage this fact to estimate the hyperparameter ξ^2 via marginal maximum likelihood:

$$\hat{\xi}^2 = \left(\frac{1}{p}\sum_{j=1}^{p} y_j^2 - \sigma^2\right)_+$$

which, when substituted into (4) yields an empirical-Bayes shrinkage estimator closely related to the James-Stein estimator. The role of ξ^2 reveals a trade-off between prior information and robustness to prior mismatch. That is, increasing ξ^2 makes the prior less informative and introduces less shrinkage. A smaller ξ^2 can place substantial mass around 0, and the shrinkage introduced will be beneficial if it matches the true population. When $\xi \to \infty$, the prior becomes arbitrarily diffuse and the estimator reduces to the least-squares estimator. An uninformative prior reduces sensitivity to severe prior mismatch, but progressively gives up the gains from pooling.

Inspired by this empirical Bayes viewpoint, we provide a definition of foundation models based on their asymptotic performance, and interpret their pretraining objective using Bayesian decision

theory. Here, pretraining can be viewed as Bayes-risk minimisation with respect to a task distribution induced by the pretraining corpus. We then compare the Bayes rules associated with a broad task population and a specific target domain, and define the price of generality as the excess risk incurred on the specialised domain when using a rule optimised for the broader population.

3.2 A BAYESIAN INTERPRETATION OF FOUNDATION MODELS

Foundation models are commonly characterised as models pretrained on broad and diverse datasets, typically at scale, and can subsequently be applied to a wide range of downstream tasks (Bommasani et al., 2021). While this description captures how foundation models are constructed, it does not identify the statistical object learnt during pretraining, or the criterion under which the resulting model is optimal. As such, we develop an interpretation of foundation models in a Bayesian manner. For completeness, Appendix B provides a brief introduction to Bayesian forecasting and decision theory. The purpose of this interpretation is not to claim that existing foundation models explicitly perform Bayesian inference, but to provide a framework for understanding their pretraining objectives and limitations. In short, pretraining determines an empirical task prior. Tasks that are weakly represented in pretraining receive little weight, undermining performance on those tasks.

We start by viewing a foundation model as a data-driven predictive universal model for a specified class of distributions

$$\mathcal{P}_\Theta = \{P_\theta : \theta \in \Theta\}.$$

Here, each P_θ represents a possible data generating process and p_θ denotes its density. The concept of universality is widely explored in information theory, particularly in universal coding, which addresses the challenge of encoding data generated by an unknown source from a specified family of distributions (Davisson, 1973; Rissanen, 1984; Barron et al., 1998). A universal code is constructed without knowing which member generated the observations, yet its code length asymptotically approaches that of a code designed with knowledge of the generating distribution. Let $\mathbf{z}_n = (z_1, \ldots, z_n)$ denote a sequence of observations generated from an unknown member P_θ, and $-\log p_\theta(\mathbf{z}_n)$ is its log loss (code length) under the generating model. Consider a probability model U with density $u(\cdot)$, $-\log u(\mathbf{z}_n)$ denotes the corresponding log loss. A universal model is defined as follows (Davisson, 1973; Rissanen, 1984; Barron et al., 1998).

Definition 1 (Universal model)**.** A probability model U is a universal model for $\mathcal{P}$ if

$$\lim_{n\to\infty} \sup_{\theta\in\Theta} \frac{1}{n} \mathbb{E}_{\mathbf{z}_n \sim P_\theta} \left[-\log u(\mathbf{z}_n) - (-\log p_\theta(\mathbf{z}_n)) \right] = 0.$$

This definition formalises the breadth of a universal model through its asymptotic performance over the reference class. It means the expected per-observation excess log loss of a foundation model U over data must be negligible for every $P_\theta \in \mathcal{P}$ uniformly. Along the same lines, an idealised foundation model should be universal over a specific class of problems, and able to approximate any data generating process in the class asymptotically well. A Bayes mixture (Clarke and Barron, 1990; Barron et al., 1998) provides an important construction of such a universal model.

Theorem 1. (Clarke and Barron, 1990; Barron et al., 1998) Let π be a weighting distribution over Θ with continuous and positive density, and suppose $\mathcal{P}$ satisfies the standard regularity conditions (Barron et al., 1998). Then the Bayesian mixture U_π, with density

$$u_\pi(\mathbf{z}) = \int p_\theta(\mathbf{z})\pi(\theta)d\theta$$

is a universal model for $\mathcal{P}$ on each compact subset of the parameter space Θ.

The weighting distribution π need not be unique; different weighting distributions may induce universal models satisfying Definition 1. This flexibility allows the mixture weights to be learnt from data to adapt predictive performance to a specific problem set.

Definition 2 (Ideal foundation model)**.** An *ideal foundation model* is a Bayesian mixture $u_{\hat{\pi}}(\mathbf{z})$ over some set of possible generating processes, $\mathcal{P}_\Theta$, with a prior $\hat{\pi}$ learned from a training corpus.

Such a foundation model is "ideal" in several aspects. Consider a forecasting rule for forecasting futures $\mathbf{f} = (f_1, \ldots, f_h)$, given an observed series $\mathbf{y}$, induced by our Bayesian model $u_{\hat{\pi}}$, under a

given loss $L(\cdot)$:

$$\delta_{\hat{\pi},L}(\mathbf{y}) = \arg\min_{\mathbf{d}} \left\{ \int L(\mathbf{d}, \mathbf{f}) u_{\hat{\pi}}(\mathbf{f} \mid \mathbf{y}) d\mathbf{f} \right\}.$$

First, under the conditions of Theorem 1, the Bayesian mixture guarantees optimal large-sample performance over its effective model class, in a specific sense. Second, under mild regularity conditions regarding $\hat{\pi}$, the rule $\delta_{\hat{\pi},L}$ is *admissible* for $\mathcal{P}_\Theta$. Let $R_{p_\theta}(\delta) = \mathbb{E}_{(\mathbf{y},\mathbf{f})\sim p_\theta}\left[L(\delta(\mathbf{y}), \mathbf{f})\right]$ denote the point-wise (at p_θ) risk of a forecasting rule δ; this means that there is no alternative rule δ_A such that $R_{p_\theta}(\delta_A) \leq R_{p_\theta}(\delta_{\hat{\pi},L})$ for every $\theta \in \Theta$, with strict inequality for at least one $\theta \in \Theta$ (Robert, 2007). That is, for the space of processes $\mathcal{P}_\Theta$ there can exist no other procedure, Bayesian or otherwise, that uniformly outperforms the rule $\delta_{\hat{\pi},L}$. Third, the decision rule $\delta_{\hat{\pi},L}$ is Bayes optimal against $\hat{\pi}$, i.e., it attains $\min_\delta \int \hat{\pi}(\theta) R_{p_\theta}(\delta) d\theta$. We note that limiting ourselves to Bayesian mixtures is not restrictive in any real sense, as under standard complete-class conditions, admissible procedures are Bayes, generalized Bayes, or limits of Bayes procedures (Ferguson, 1967; Lehmann and Casella, 1998). Further, this Bayesian view captures the essential sources of inductive bias: the support of the prior, $\hat{\pi}$, determines the effective *model class* within P_Θ, potentially including choices between different model families or architectures, while the way prior probability is distributed over this support captures the effects of *regularisation*. Thus, ideal foundation models represent essentially the broad class of models we would ideally consider in practice.

This leaves open the question of *how* to choose $\hat{\pi}$. This is essentially a type of *empirical Bayes* problem, and as such can be tackled in a variety of ways. For example, if we let $\hat{\pi}$ be a member of a family of prior distributions indexed by some hyperparameter α, so that $\hat{\pi} \equiv \pi_{\hat{\alpha}}$, then α can be estimated via a standard technique such as marginal maximum likelihood (Efron, 2019).

3.3 Price of generality

Consider a rich distribution set $\mathcal{P}_\Theta$, for example, a sufficiently complex deep neural network, and a corpus of data $\left\{\mathbf{y}^{(i)}, \mathbf{f}^{(i)}\right\}$, $i = 1, \ldots, n$, where $\mathbf{y}^{(i)}$ denotes a training series and $\mathbf{f}^{(i)}$ a future series for forecast evaluation. For a given prior distribution π over Θ the (empirical) risk for a chosen loss, and a chosen subset of the corpus, say $I \subseteq \{1, \ldots, n\}$ is

$$\bar{R}_{\pi,L}(I) = \frac{1}{|I|} \sum_{i \in I} L\left(\delta_{\pi,L}(\mathbf{y}^{(i)}), \mathbf{f}^{(i)}\right). \tag{5}$$

Letting $G = \{1, \ldots, n\}$ denote the entire corpus, we can then define the corpus oracle prior as

$$\pi^*_G \in \arg\min_{\pi \in \Pi} \bar{R}_{\pi,L}(G) \tag{6}$$

and $\bar{R}_{\pi^*_G,L}(G)$ as the corpus oracle risk. This represents an ideal lower bound, within the specified model class $\mathcal{P}_\Theta$ and prior class Π, on the empirical risk achievable over the corpus. It also represents the limiting objective implicitly targeted when methods are developed and compared on a fixed benchmark corpus. Of course, researchers do not usually explicitly optimise (6), as doing so directly would constitute "data leakage". In practice, research on benchmark corpora proceeds by modifying the model or prior class and minimising some surrogate objective, usually based on training data, to obtain some data-driven model $\hat{\pi}$, while benchmark performance $\bar{R}_{\hat{\pi},L}(G)$ remains the ultimate criterion by which the resulting method is evaluated. We call a model designed to perform well across the entirety of G a *generalist* model.

Common foundation model wisdom says that more training data will produce a better foundation model. However, practical applications are frequently concerned with only a restricted subset of tasks. Instead of achieving strong average performance across all tasks represented by G, the aim is usually to perform well in a specific application domain; for example, yearly time series in forecasting. A generalist foundation model is optimised for average performance over the complete pretraining task population, potentially at the expense of performance on any one task, just as a specialist model may forego average performance in exchange for lower Bayes risk on a particular task. To see this, divide our corpus into two subgroups of tasks: let T denote the tasks of interest (say, yearly time series) and N denote the tasks we deem to be irrelevant, or nuisance tasks. Then, we can form oracle priors for both sub-tasks using

$$\pi^*_T \in \arg\min_{\pi \in \Pi} \bar{R}_{\pi,L}(T) \quad \text{and} \quad \pi^*_N \in \arg\min_{\pi \in \Pi} \bar{R}_{\pi,L}(N),$$

respectively. We can then define an ideal two-expert generalist model, π^*_G, as an optimal mixture of these two, i.e.,

$$\pi^*_G \equiv \pi_{\alpha^*} = \alpha^* \pi^*_T + (1-\alpha^*)\pi^*_N, \;\; \text{with} \;\; \alpha^* \in \arg\min_{\alpha \in [0,1]} \bar{R}_{\pi_\alpha, L}(G). \tag{7}$$

This mixture is optimal for the complete corpus, G, within the class of mixtures of the two oracle priors, and need not coincide with the unrestricted corpus oracle π^*_G in (6). The key question then is: how will the introduction of additional (nuisance) data to our foundation model degrade the performance of our specialist model π^*_T? Let

$$D_L(T||G) = \bar{R}_{\pi^*_G, L}(T) - \bar{R}_{\pi^*_T, L}(T)$$

denote the regret of using the generalist model π^*_G over specialist model π^*_T when evaluating on the target task set T. This is the *price of generality*, and as π^*_T is optimal for T we have $D_L(T||G) \geq 0$, with strict inequality whenever the generalist rule has strictly greater risk on T. Consider a time series $\mathbf{y}$; under the mixture (7) the posterior predictive distribution under the generalist model is a weighted mixture of the posterior predictive distributions under the specialist and nuisance models:

$$u_G(\mathbf{f} \,|\, \mathbf{y}) = w_T(\mathbf{y}) u_T(\mathbf{f} \,|\, \mathbf{y}) + (1 - w_T(\mathbf{y})) u_N(\mathbf{f} \,|\, \mathbf{y}) \tag{8}$$

where

$$w_T(\mathbf{y}) = \frac{\alpha^* u_T(\mathbf{y})}{\alpha^* u_T(\mathbf{y}) + (1-\alpha^*) u_N(\mathbf{y})}$$

is the posterior weight for model u_T. The rule (8) is clearly a form of mixture-of-experts, with the "expert" weight being given by the posterior probability of the expert given the data. Specialising our loss to squared error yields the following useful result.

Proposition 1. Let π^*_T and π^*_N denote the oracle specialist and nuisance priors, and π^*_G the optimal two-expert generalist mixture. Suppose that the tasks are independent and the specialist model is correctly specified and choose L to be squared-error, so that $\delta_{\pi^*_T}(\mathbf{y}) = \mathbb{E}[\mathbf{f} \,|\, \mathbf{y}, T]$. Then

$$\mathbb{E}\left[D_{\mathrm{SE}}(T||G) \,|\, \left\{ \mathbf{y}^{(i)} \right\}_{i \in T} \right] = \frac{1}{|T|} \sum_{i \in T} \left\{ 1 - w_T(\mathbf{y}^{(i)}) \right\}^2 \left\| \delta_{\pi^*_N}(\mathbf{y}^{(i)}) - \delta_{\pi^*_T}(\mathbf{y}^{(i)}) \right\|^2. \tag{9}$$

Proposition 1 characterises the conditions under which augmentation of a foundation model by nuisance time series can cause the risk of the specialist model to increase. The first term in the right-hand-side captures how much posterior weight remains on the nuisance expert after observing the time series. Through $w_T(y)$, this depends both on the generalist model's prior weighting of the specialist expert, α^*, and on how informative the observed series is for distinguishing between the two experts. The second term captures the inflation in prediction risk that occurs if we use the nuisance model in place of the optimal specialist model. Thus, the price of generality will be large if simultaneously the time series themselves provide insufficient evidence for model selection, *and* similar-looking series in the target and nuisance tasks have very different future behaviour. This is precisely the failure mode we expect for yearly time series; these are low-granularity, high-noise series. The problem is not that non-yearly series are "irrelevant" *per se*; rather, it is that short yearly series may be insufficiently informative to distinguish useful transfer from harmful transfer. We conclude by noting that (9) is *optimistic*, as it is based on oracle (data leakage) procedures; in practice, we must replace optimal priors with priors estimated from data, which can inflate the risk, particularly if the fraction of relevant tasks in the corpus is small.

4 Application: A specialist yearly foundation model

Following the above theory, this section proposes a specialist foundation model for yearly forecasting. First, we estimate a yearly specialist prior from real-world series and simulate synthetic training data based on a model for general purposes. Second, we train a PFN to approximate the posterior predictive distribution induced by this prior.

4.1 Yearly specialist prior estimation and synthetic series simulation

The LGT model is a flexible Bayesian exponential smoothing model that captures a broad spectrum of trend patterns and heteroscedasticity. It is a general forecasting tool and has demonstrated state-of-the-art performance on the M3 competition dataset (Makridakis and Hibon, 2000). To estimate

a yearly specialist task prior, we fit the LGT model to the training portion of each series in the M3 yearly subset. The resulting collection of parameter posteriors provides an empirical approximation $\widehat{\pi}_T$, making it a *posterior-informed prior* based on a general method fitted on this domain. This allows us to exploit the weakly informative priors of the LGT model, and use the resulting posteriors to *concentrate* probability mass in regions relevant to yearly forecasting tasks. We then draw model parameters from these posterior distributions and simulate synthetic series conditional on these draws. This simulation procedure preserves important characteristics learnt from real-world yearly data, and produces a posterior-informed synthetic training distribution that is aligned with the target domain. The LGT specification and simulation setup are provided in Appendix D.

4.2 The GRAIN architecture and training

We construct input matrix $\mathbf{V}$ from standardised observations and temporal information indicating their order in the series, under a fixed-width context of 200 observations. Shorter series are left-padded with an indicator distinguishing padded positions from observed values. GRAIN [2] learns a mapping from $\mathbf{V}$ to a grid of predictive quantiles on the standardised scale, $g_{\boldsymbol{\psi}}(\mathbf{V}) = \tilde{\mathbf{Q}} \in \mathbb{R}^{H\times K}$. Each entry $\tilde{q}_{h,k}$ in $\tilde{\mathbf{Q}}$ denotes the estimated conditional quantile at level α_k for future step h over the horizon H. The quantile levels of interest are denoted by $\mathcal{Q} = \{\alpha_1, \ldots, \alpha_K\}$, and $\boldsymbol{\psi}$ denotes the parameters of GRAIN. We use an encoder-only transformer with dual-scale patching, with details provided in Appendix E.

Training series are sampled under $\widehat{\pi}_T$. We draw $\boldsymbol{\theta} \sim \widehat{\pi}_T$ and simulate series from $P_{\boldsymbol{\theta}}$ (i.e., the LGT model). The simulated history $\mathbf{y}$ is converted into $\mathbf{V}$, and the subsequent H observations (standardised), denoted by $\tilde{\mathbf{f}}$, form the forecasting targets. GRAIN is trained to approximate the posterior predictive quantiles induced by $\widehat{\pi}_T$, with an additional smoothness regulariser:

$$\boldsymbol{\psi}^\star = \operatorname*{argmin}_{\boldsymbol{\psi}} \mathbb{E}_{\theta\sim\widehat{\pi}_T}\left[\mathbb{E}_{(\mathbf{y},\mathbf{f})\sim P_\theta}\left[L_{\boldsymbol{\psi}}(\mathbf{V},\tilde{\mathbf{f}})\right]\right].$$

The loss function L combines a predictive component with a regulariser that encourages smooth trend,

$$L_{\boldsymbol{\psi}}(\mathbf{V},\tilde{\mathbf{f}}) = L_q(g_{\boldsymbol{\psi}}(\mathbf{V}),\tilde{\mathbf{f}}) + \lambda\Omega_{\boldsymbol{\psi}}(\mathbf{V}),$$

where

$$L_q(g_{\boldsymbol{\psi}}(\mathbf{V}),\tilde{\mathbf{f}}) = \frac{1}{HK}\sum_{h=1}^{H}\sum_{k=1}^{K}\Lambda_{\alpha_k}\left(\tilde{q}_{h,k},\tilde{f}_h\right) \quad \text{and} \quad \Omega_{\boldsymbol{\psi}}(\mathbf{V}) = \frac{1}{(J-2)D}\sum_{j=3}^{J}||\Delta^2\mathbf{r}_j||_2^2,$$

respectively. Here, $\Lambda_\alpha(q,y) = (\alpha - \mathbf{1}\{y<q\})(y-q)$ is the pinball loss, $\mathbf{r}_j \in \mathbb{R}^D$ denotes the trend representation at token position $j = 1,\ldots,J$, where $J = 120$ is the number of tokens produced by dual-scale patching and $D = 128$ is the token dimension, and $\Delta^2\mathbf{r}_j = \mathbf{r}_j - 2\mathbf{r}_{j-1} + \mathbf{r}_{j-2}$ denotes the second difference of the trend token sequence.

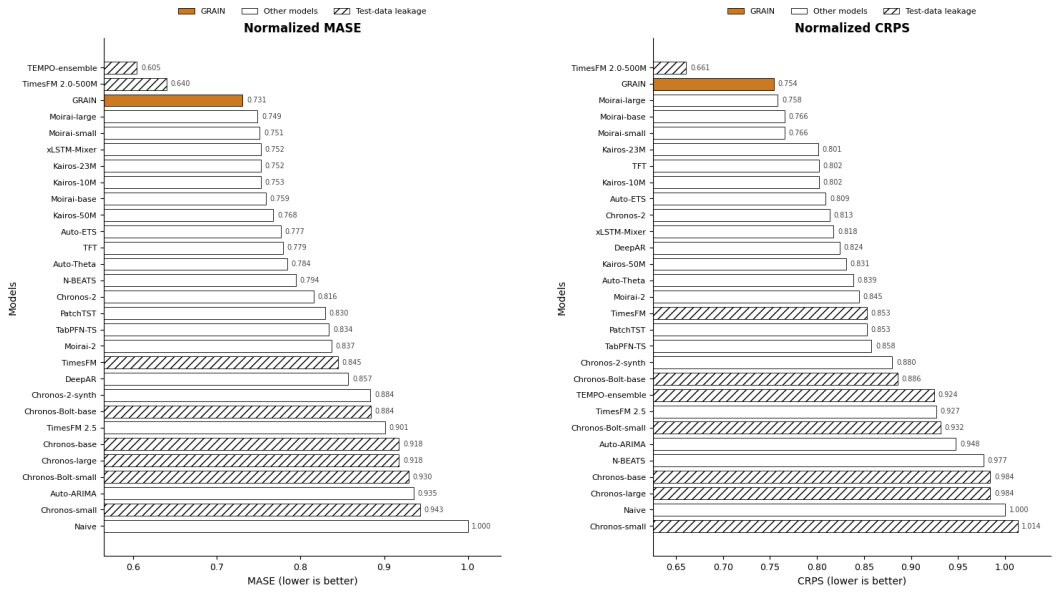
Once trained, GRAIN produces the predictive quantiles through a single forward pass without per-series parameter estimation or posterior sampling. Viewed sequentially, the LGT posteriors fitted to the original yearly series are repurposed as a prior for new forecasting tasks, while GRAIN approximates the subsequent Bayesian update induced by a new history. It provides efficient Bayesian prediction while retaining the information encoded by the estimated task prior.

5 Experiments

We evaluate GRAIN on the yearly subset of the GIFT-Eval benchmark (Aksu et al., 2024), which consists of the yearly series of the M4 competition (Makridakis et al., 2020) for evaluation, and of the M3 competition (Makridakis and Hibon, 2000) series (among others) for pretraining. This way, we prevent data leakage as our simulation uses only the M3 series. Following the GIFT-Eval protocol, we report mean absolute scaled error (MASE, Hyndman and Koehler, 2006) and continuous ranked probability score (CRPS, Gneiting and Raftery, 2007). Both metrics are normalised by the corresponding score of the naïve forecast, indicating improvement over the naïve baseline. Definitions of the metrics are provided in Appendix G.

[2] Code and implementation details are available at https://github.com/anonymousresearcher5678/GRAIN

We compare GRAIN with a broad collection of statistical, deep-learning and foundation model baselines. Baseline results are computed from the forecasts released by GIFT-Eval, restricted to its yearly subset using the same metrics. Figure 1 reports the main results, while complete rankings are reported in Appendix H. GRAIN achieves a normalised MASE of 0.731 and a normalised CRPS of 0.754, ranking first on both metrics among models not flagged by the GIFT-Eval leaderboard for data leakage. These results indicate that the yearly specialist task prior estimated from the M3 dataset transfers effectively to unseen series from the M4 dataset without dataset-specific fine-tuning.



(a) Normalised MASE

(b) Normalised CRPS

Figure 1: Normalised MASE and CRPS on the GIFT-Eval yearly benchmark, relative to the naïve baseline. Models with test data leakage are marked with stripes. GRAIN achieves the best among models not flagged for test-data leakage on both metrics.

6 Conclusion

This paper developed a theory of why foundation models can be strong generalists yet still underperform on specific domains. The classical James–Stein phenomenon provides the motivating intuition that gains under an aggregate objective need not translate to each component task. We then interpreted pretraining as empirical Bayes-risk minimisation under a corpus-induced task prior, and defined the *price of generality* as the resulting excess risk due to prior mismatch.

We instantiated this theory in yearly forecasting, where short histories and corpus underrepresentation make prior mismatch particularly consequential. We proposed a specialist yearly foundation model, GRAIN, trained via prior-data fitting on synthetic annual series. It constructs a specialist task prior by repurposing posteriors based on a general method fitted to real yearly series. On the yearly subset of GIFT-Eval, GRAIN ranked best among leakage-free baselines, consistent with the developed theory that aligning the training prior with the target task population can reduce excess risk. More broadly, our results suggest that foundation models should be assessed not only by average benchmark performance, but also by alignment with their intended deployment domain. Future work can extend this framework to other underrepresented regimes (e.g., intermittent demand or domain-specific low-data settings), investigate hybrid specialist-generalist adaptation strategies, and develop practical diagnostics to estimate the expected price of generality before deployment.

7 USE OF AI TOOLS

In this work, we used generative AI tools to assist with language polishing and proof verification. All AI-assisted content was carefully reviewed and verified by the authors.

REFERENCES

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

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

A. F. Ansari, O. Shchur, J. Küken, A. Auer, B. Han, P. Mercado, S. S. Rangapuram, H. Shen, L. Stella, X. Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, 16(4):521–530, 2000.

A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

B. S. Clarke and A. R. Barron. Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, 36(3):453–471, 1990.

A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=jn2iTJas6h.

L. Davisson. Universal noiseless coding. *IEEE Transactions on Information Theory*, 19(6):783–795, 1973.

S. Dooley, G. S. Khurana, C. Mohapatra, S. V. Naidu, and C. White. ForecastPFN: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

B. Efron. Bayes, oracle Bayes and empirical Bayes. *Statistical Science*, 34(2):177–201, 2019.

B. Efron and C. Morris. Combining possibly related estimation problems. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 35(3):379–402, 1973a.

B. Efron and C. Morris. Stein's estimation rule and its competitors—an empirical Bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973b.

V. Ekambaram, A. Jati, P. Dayama, S. Mukherjee, N. H. Nguyen, W. M. Gifford, C. Reddy, and J. Kalagnanam. Tiny Time Mixers (TTMs): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series, 2024. URL https://arxiv.org/abs/2401.03955.

T. S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York, 1967.

A. Garza and M. Mergenthaler-Canseco. TimeGPT-1. arXiv preprint arXiv:2310.03589, 2023.

A. Garza, M. M. Canseco, C. Challú, and K. G. Olivares. StatsForecast: Lightning fast forecasting with statistical and econometric models, 2022. URL https://github.com/Nixtla/statsforecast. Published: PyCon Salt Lake City, Utah, US 2022.

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

N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large language models are zero-shot time series forecasters. In *NeurIPS*, 2023.

N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *NeurIPS 2022 First Table Representation Workshop*, October 2022.

N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S. B. Hoo, R. T. Schirrmeister, and F. Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, Jan. 2025. ISSN 1476-4687. URL https://doi.org/10.1038/s41586-024-08328-6.

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

R. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice, 3rd edition*. OTexts: Melbourne, Australia, 2021. URL OTexts.com/fpp3.

R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

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

W. James and C. Stein. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379. University of California Press, 1961.

M. Jin, S. Wang, L. Ma, Z. Chu, J. Zhang, X. Shi, and Q. Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978.

E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer, New York, 2 edition, 1998.

X. Long, D. F. Schmidt, C. Bergmeir, and S. Smyl. Fast Gibbs sampling for the local-seasonal-global trend Bayesian exponential smoothing model. *Statistics and Computing*, 35(3):77, 2025.

S. Makridakis and M. Hibon. The M3-competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000.

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

P. Montero-Manso and R. Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, 37(4):1632–1653, 2021. doi: 10.1016/j.ijforecast.2021.03.004.

S. Müller, N. Hollmann, S. P. Arango, J. Grabocka, and F. Hutter. Transformers can do Bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Y. Park, D. Maddix, F.-X. Aubet, K. Kan, J. Gasthaus, and Y. Wang. Learning quantile functions without quantile crossing for distribution-free time series forecasting. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 8127–8150, 2022.

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

K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, et al. Lag-llama: Towards foundation models for time series forecasting. arXiv preprint arXiv:2310.08278, 2023.

J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4):629–636, 1984.

C. P. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer, 2007.

S. Smyl, C. Bergmeir, A. Dokumentov, X. Long, E. Wibowo, and D. Schmidt. Local and global trend Bayesian exponential smoothing models. *International Journal of Forecasting*, 2024.

C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the third Berkeley symposium on mathematical statistics and probability, volume 1: Contributions to the theory of statistics*, pages 197–207. University of California Press, 1956.

C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, pages 1135–1151, 1981.

G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=Yd8eHMY1wz`.

H. Xue and F. D. Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14, 2023. doi: 10.1109/TKDE.2023.3342137.

A The James-Stein estimator and the derivation of its estimated risk

Stein (1956) demonstrated a counter-intuitive phenomenon that, when estimating multiple parameters jointly, one can achieve uniformly lower risk than independent estimation. Specifically, when estimating the mean vector for a p-dimensional multivariate normal distribution, the standard maximum likelihood estimator (i.e., the sample mean of each coordinate) is no longer admissible when $p \geq 3$, meaning that there exist alternative estimators with strictly smaller risk everywhere in the parameter space. This result reveals that joint estimation can improve accuracy by borrowing information across seemingly unrelated problems. The classical example from the literature (see, e.g., Efron and Morris, 1973a) illustrates that one may simultaneously estimate quantities as diverse as "tea consumption in Taiwan ... [and] the speed of light and weight of hogs in Montana" when all of these problems are of interest. Although they are not semantically related, the connection is built through the evaluation regime, where the model performance is assessed on average error across all the problems. James and Stein (1961) later provided a shrinkage estimator that dominates the maximum likelihood estimator by pulling the estimator towards the shared population centre. Efron and Morris (1973b) interpreted the James-Stein estimator through the lens of empirical Bayes estimation, and showed that the shrinkage target itself can be estimated from the observations. Interestingly, the James-Stein estimator provides a useful perspective for understanding existing foundation models, which are jointly optimised for a diverse range of tasks and similarly aimed for best average performance over the task population. One can improve the overall performance by exploiting shared structure at a population level. However, it is important to emphasise that such improvements are measured with respect to the overall average risk, rather than implying pointwise optimality for every individual task.

In the illustrative example given in Section 3.1, we consider the true mean vector

$$\boldsymbol{\mu} = (\sqrt{p}, 0, \ldots, 0),$$

and a hypothetical realisation $\mathbf{y} = (\sqrt{p}, 0, \ldots, 0)$. The James-Stein estimator is given by

$$\hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \left(\frac{2}{\sqrt{p}}, 0, \ldots, 0\right).$$

From (2), it has average risk

$$\frac{1}{p} R(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}^{\mathrm{JS}}) = 1 - \frac{(p-2)^2}{p} \mathbb{E}_{\boldsymbol{\mu}} \left[\frac{1}{s}\right],$$

Although $s = p$ at the illustrative realisation above, the risk averages over random Y. For large p, we have

$$\frac{y_1^2}{p} \to 1 \ \text{ and } \ \frac{1}{p} \sum_{i=2}^{p} y_i^2 \to 1,$$

then we can approximate $\mathbb{E}_{\boldsymbol{\mu}} \left[\frac{1}{s}\right] \approx \frac{1}{2p}$ and

$$\frac{1}{p} R(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}^{\mathrm{JS}}) \approx 1 - \frac{(p-2)^2}{2p^2}.$$

Its risk on the first coordinate is

$$\begin{aligned} R(\boldsymbol{\mu}, \hat{\mu}_1^{\mathrm{JS}}) &= \mathbb{E}_{\boldsymbol{\mu}} \left[\left(\left(1 - \frac{p-2}{s}\right) y_1 - \mu_1 \right)^2 \right] \\ &= \mathbb{E}_{\boldsymbol{\mu}} \left[(y_1 - \mu_1)^2\right] - 2\mathbb{E}_{\boldsymbol{\mu}} \left[(y_1 - \mu_1) \frac{p-2}{s} y_1\right] + (p-2)^2 \mathbb{E}_{\boldsymbol{\mu}} \left[\frac{y_1^2}{s^2}\right] \\ &= 1 - 2(p-2)\mathbb{E}_{\boldsymbol{\mu}} \left[\frac{1}{s}\right] + (p^2 - 4)\mathbb{E}_{\boldsymbol{\mu}} \left[\frac{y_1^2}{s^2}\right] \end{aligned}$$

using Stein's identity (Stein, 1981). We approximate $\mathbb{E}_{\boldsymbol{\mu}} \left[\frac{y_1^2}{s^2}\right] \approx \frac{1}{4p}$ such that

$$R(\boldsymbol{\mu}, \hat{\mu}_1^{\mathrm{JS}}) \approx 1 - \frac{p-2}{p} + \frac{p^2 - 4}{4p}.$$

B AN INTRODUCTION TO BAYESIAN FORECASTING AND DECISION THEORY

Consider a family of distributions $\mathcal{P}_\Theta = \{P_\theta : \theta \in \Theta\}$ and let π denote a prior distribution over Θ. Let $\mathbf{y}$ denote the observed history and $\mathbf{f}$ the future values over a forecast horizon, that is

$$\theta \sim \pi, \quad (\mathbf{y}, \mathbf{f}) \sim P_\theta.$$

Using Bayes' rule, we form the posterior distribution,

$$\pi(\theta \,|\, \mathbf{y}) \propto p_\theta(\mathbf{y}) \pi(\theta),$$

and the posterior predictive distribution

$$p(\mathbf{f} \,|\, \mathbf{y}) = \int p_\theta(\mathbf{f} \,|\, \mathbf{y}) \pi(\theta \,|\, \mathbf{y}) d\theta$$

after observing $\mathbf{y}$. For a chosen loss function $L(\cdot, \cdot)$, the posterior expected loss for a candidate forecast $\mathbf{d}$ is defined as

$$\rho_\pi(\mathbf{d} \,|\, \mathbf{y}) = \int L(\mathbf{d}, \mathbf{f}) p(\mathbf{f} \,|\, \mathbf{y}) d\mathbf{f}.$$

A Bayes forecasting rule $\delta_{\pi,L}(\mathbf{y})$ is obtained by minimising the above posterior expected loss, i.e.,

$$\delta_{\pi,L}(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{d}} \rho_\pi(\mathbf{d} \,|\, \mathbf{y}).$$

Thus, $\delta_{\pi,L}(\mathbf{y})$ gives the optimal forecast under the chosen loss conditioned on the observed history $\mathbf{y}$. The same problem admits a population-level representation for a given prior π. Define the risk

$$R(\theta, \delta) = \mathbb{E}_{(\mathbf{y},\mathbf{f}) \sim P_\theta} \left[L(\delta(\mathbf{y}), \mathbf{f})\right]$$

as the expected loss over all possible realised histories and future values using a forecasting rule δ. Averaging $R(\theta, \delta)$ over the parameter space under the prior π gives the Bayes risk

$$r(\pi, \delta) = \mathbb{E}_{\theta \sim \pi} \left[R(\theta, \delta) \right].$$

By conditioning on $\mathbf{y}$, the Bayes risk can be equivalently written as

$$r(\pi, \delta) = \mathbb{E}_{\mathbf{y} \sim m(\mathbf{y})} \left[\rho_\pi(\delta(\mathbf{y}) \,|\, \mathbf{y}) \right],$$

where

$$m(\mathbf{y}) = \int p_\theta(\mathbf{y}) \pi(\theta) d\theta$$

denotes the marginal distribution of data, which is also known as the prior predictive distribution. Because the Bayes risk is an average of posterior expected losses over the data space under $m(\mathbf{y})$, minimising the latter pointwise yields a rule that minimises the former (Robert, 2007). Hence,

$$\delta_{\pi, L}(\cdot) = \operatorname*{argmin}_{\delta} r(\pi, \delta).$$

In other words, minimising the posterior expected loss and minimising the above Bayes risk lead to the same optimal Bayes forecasting rule. This Bayes rule, importantly, is *admissible* for $\mathcal{P}_\Theta$ under standard regularity conditions. That is, there is no other alternative δ_A such that

$$R(\theta, \delta_A) \leq R(\theta, \delta_{\pi, L})$$

for every $\theta \in \Theta$, with strict inequality for at least one $\theta \in \Theta$ (Robert, 2007). Loosely speaking, no alternative rule can have strictly lower risk everywhere in the parameter space.

For the Bayesian mixture in Theorem 1 and $\mathbf{z} = (\mathbf{y}, \mathbf{f})$, we have

$$u_\pi(\mathbf{y}, \mathbf{f}) = \int p_\theta(\mathbf{y}, \mathbf{f}) \pi(\theta) d\theta.$$

Interpreting the weighting distribution π as a prior over θ, conditioning on history $\mathbf{y}$ gives

$$\begin{aligned} u_\pi(\mathbf{f} \,|\, \mathbf{y}) &= \frac{u_\pi(\mathbf{y}, \mathbf{f})}{u_\pi(\mathbf{y})} \\ &= \int p_\theta(\mathbf{f} \,|\, \mathbf{y}) \pi(\theta \,|\, \mathbf{y}) d\theta. \end{aligned}$$

Thus, the conditional distribution induced by the Bayes mixture is exactly the posterior predictive distribution. The forecasting rule obtained by minimising posterior expected loss is therefore minimising Bayes risk against the prior π, and, under the stated regularity conditions, admissible.

Typically the prior π is not known in practice; different choices of π induce different forecasting rules against π. When interest is restricted to a target domain represented by a region $\Theta_T \subseteq \Theta$, a rule optimised over a broad corpus can have higher risk on that domain than a specialist rule. This motivates the comparison between generalist and target-domain specialist priors in Section 3.3.

C Proof of Proposition 1

Under squared-error loss, the Bayes rule is the posterior predictive mean. From the mixture representation in (8), the generalist rule is therefore

$$\delta_{\pi_G^*}(\mathbf{y}) = w_T(\mathbf{y}) \delta_{\pi_T^*}(\mathbf{y}) + \{1 - w_T(\mathbf{y})\} \delta_{\pi_N^*}(\mathbf{y}).$$

Hence,

$$\delta_{\pi_G^*}(\mathbf{y}) - \delta_{\pi_T^*}(\mathbf{y}) = \{1 - w_T(\mathbf{y})\} \left\{ \delta_{\pi_N^*}(\mathbf{y}) - \delta_{\pi_T^*}(\mathbf{y}) \right\}.$$

For any $i \in T$, write

$$\delta_{\pi_G^*}(\mathbf{y}^{(i)}) - \mathbf{f}^{(i)} = \left\{ \delta_{\pi_G^*}(\mathbf{y}^{(i)}) - \delta_{\pi_T^*}(\mathbf{y}^{(i)}) \right\} + \left\{ \delta_{\pi_T^*}(\mathbf{y}^{(i)}) - \mathbf{f}^{(i)} \right\}.$$

Expanding the squared norm and taking expectation conditional on $\mathbf{y}^{(i)}$ gives

$$\begin{aligned} &\mathbb{E} \left[\left\| \delta_{\pi_G^*}(\mathbf{y}^{(i)}) - \mathbf{f}^{(i)} \right\|^2 - \left\| \delta_{\pi_T^*}(\mathbf{y}^{(i)}) - \mathbf{f}^{(i)} \right\|^2 \,\middle|\, \left\{ \mathbf{y}^{(i)} \right\}_{i \in T} \right] \\ &\quad = \left\| \delta_{\pi_G^*}(\mathbf{y}^{(i)}) - \delta_{\pi_T^*}(\mathbf{y}^{(i)}) \right\|^2, \end{aligned}$$

	Description
ν	degrees-of-freedom parameter in the Student t-distribution
γ	coefficient of the global trend
ρ	power coefficient of the global trend, in $[-0.5, 1]$
λ	damping coefficient of the local trend, in $[-100, 1]$
α	level smoothing parameter, in $[0, 1]$
β	local trend smoothing parameter, in $[0, 1]$
χ	scale of error, positive, constant for each time period
ϕ	mixing parameter of homoscedastic error and heteroscedastic error, in $[0, 1]$
τ	power coefficient of the heteroscedastic error, in $[0, 1]$
b_1	initial local trend

Table 2: Parameters of the LGT model (Smyl et al., 2024; Long et al., 2025).

since, by correct specification of the specialist model,

$$\delta_{\pi_T^*}(\mathbf{y}^{(i)}) = \mathbb{E}[\mathbf{f}^{(i)} \mid \mathbf{y}^{(i)}, T],$$

and therefore the cross term vanishes. Substituting the expression for $\delta_{\pi_G^*}(\mathbf{y}) - \delta_{\pi_T^*}(\mathbf{y})$ yields

$$\mathbb{E}\left[\left\|\delta_{\pi_G^*}(\mathbf{y}^{(i)}) - \mathbf{f}^{(i)}\right\|^2 - \left\|\delta_{\pi_T^*}(\mathbf{y}^{(i)}) - \mathbf{f}^{(i)}\right\|^2 \mid \left\{\mathbf{y}^{(i)}\right\}_{i\in T}\right] = \{1-w_T(\mathbf{y}^{(i)})\}^2 \left\|\delta_{\pi_N^*}(\mathbf{y}^{(i)}) - \delta_{\pi_T^*}(\mathbf{y}^{(i)})\right\|^2.$$

Averaging over $i \in T$ gives (9). □

D Detailed simulation setup based on the Bayesian LGT model

The LGT model (Smyl et al., 2024; Long et al., 2025) assumes each observation is a realisation from a Student t-distribution,

$$y_{t+1} \mid \hat{y}_{t+1}, \hat{\sigma}_{t+1}, \nu \sim t(\nu, \hat{y}_{t+1}, \hat{\sigma}_{t+1}),$$

where

$$\begin{aligned}
\hat{y}_{t+1} &= l_t + \gamma l_t^{\rho} + \lambda b_t, \\
l_t &= \alpha y_t + (1-\alpha) l_{t-1}, \\
b_t &= \beta(l_t - l_{t-1}) + (1-\beta) b_{t-1}, \\
\hat{\sigma}_{t+1}^2 &= \chi^2(\phi^2 + (1-\phi)^2 l_t^{2\tau}),
\end{aligned}$$

A detailed description of model parameters is provided in Table 2. The LGT model provides flexible trend and time-varying heteroscedastic error modelling. Instead of a normal assumption, the heavy-tailed Student t-distribution accommodates the volatility in the series while being more resistant to outliers.

We fit the LGT model to each yearly series (in-sample portion only) in the M3 dataset (Makridakis and Hibon, 2000), with 645 series in total, using the Gibbs sampler developed in Long et al. (2025). For each fitted series, we take 250 posterior draws, and simulate one new series for each draw under the corresponding LGT model. The simulated length is 200. This procedure produces 161,250 synthetic yearly series and allows the synthetic training data to reflect the behaviours learnt from real yearly series.

E GRAIN architecture and training details

This section gives the full specification of GRAIN,[3] introduced in Section 4.2. GRAIN uses a compact encoder-only transformer to map a fixed-width representation of a historical series to predictive quantiles over a given forecast horizon. The model contains 1,540,772 trainable parameters and produces an $H \times K$ matrix of quantiles, where $H = 6$, $K = 11$ and $\mathcal{Q} = \{0.05, 0.1, 0.2, \ldots, 0.9, 0.95\}$. Figure 2 summarises the data flow. In the following, we describe the input construction, patching scheme, internal representations, transformer encoder, prediction head, and training procedure.

[3]Code and implementation details are available at https://github.com/anonymousresearcher5678/GRAIN

E.1 Input construction and standardisation

Let $y_1, \ldots, y_T$ denote the observed history for a series. The sample mean and standard deviation (numerically stabilised) are computed from these observations as

$$\mu_y = \frac{1}{T}\sum_{t=1}^{T} y_t, \quad \text{and} \quad s_y = \left\{ \frac{1}{T}\sum_{t=1}^{T} (y_t - \mu_y)^2 + \epsilon \right\}^{1/2}, \tag{10}$$

where $\epsilon = 10^{-4}$ provides numerical stability. The observed history is standardised according to

$$\tilde{y}_t = \frac{y_t - \mu_y}{s_y}, \qquad t = 1, \ldots, T. \tag{11}$$

The forecasting targets are transformed using the same procedure

$$\tilde{f}_h = \frac{f_h - \mu_y}{s_y}, \qquad h = 1, \ldots, H. \tag{12}$$

Yearly series can differ substantially in their levels and scales. This transformation removes series-specific location and scale information and allows GRAIN to learn patterns shared across trajectories of different magnitudes. We use a fixed input width $T_{\max} = 200$ and histories shorter than $T_{\max}$ are left-padded. A binary observation indicator is used to distinguish observed values from padded positions. Input positions are also associated with temporal information derived from the observation timestamps indicating their order.

E.2 Dual-scale patching

The input matrix $\mathbf{V}$ is partitioned into non-overlapping patches at two temporal scales. The long scale uses patches of length 10 and produces 20 tokens. The short scale uses patches of length 2 and produces 100 tokens. Each patch is linearly projected to width $d = 128$ and receives a learned positional embedding. The two token sequences are concatenated to form 120 tokens. This construction gives the encoder simultaneous access to both long-range structure and short-range year-to-year variation. Padded observations remain identified by the corresponding indicator and are excluded from attention.

E.3 Representations and gated fusion

GRAIN constructs three representations from the patch tokens. The first is the base patch representation. The second is a temporal representation derived from the observation timestamps. The third is a causal trend representation obtained through a depthwise-separable convolution followed by short causal smoothing. The trend convolution has kernel width 15 and dilation 2. The smoothing kernel has width 3.

A softmax gate combines the three representations. Its weights are computed from pooled summaries and are learned separately for each series and channel. The gate therefore allows the relative contributions of the base, temporal, and trend representations to vary across inputs. The trend representation is additionally regularised by the smoothness penalty defined in Section 4.2, with weight $\lambda = 10^{-4}$.

E.4 Transformer encoder and prediction head

The fused tokens are processed by four pre-normalisation transformer encoder blocks. Each block has model width 128, eight attention heads, feed-forward width 512, and dropout rate 0.1. Attention pooling with four learned queries and four pooling heads converts the token sequence into a fixed-dimensional representation. A two-layer multilayer perceptron then produces $H \times K = 66$ outputs as quantile forecasts over the horizon.

The network outputs quantiles on the standardised scale. Each predicted quantile is mapped back to the original scale using

$$\hat{q}_{h,k} = s_y \tilde{q}_{h,k} + \mu_y. \tag{13}$$

At prediction time, the K outputs are sorted at each forecast horizon to impose non-decreasing quantiles (Park et al., 2022).

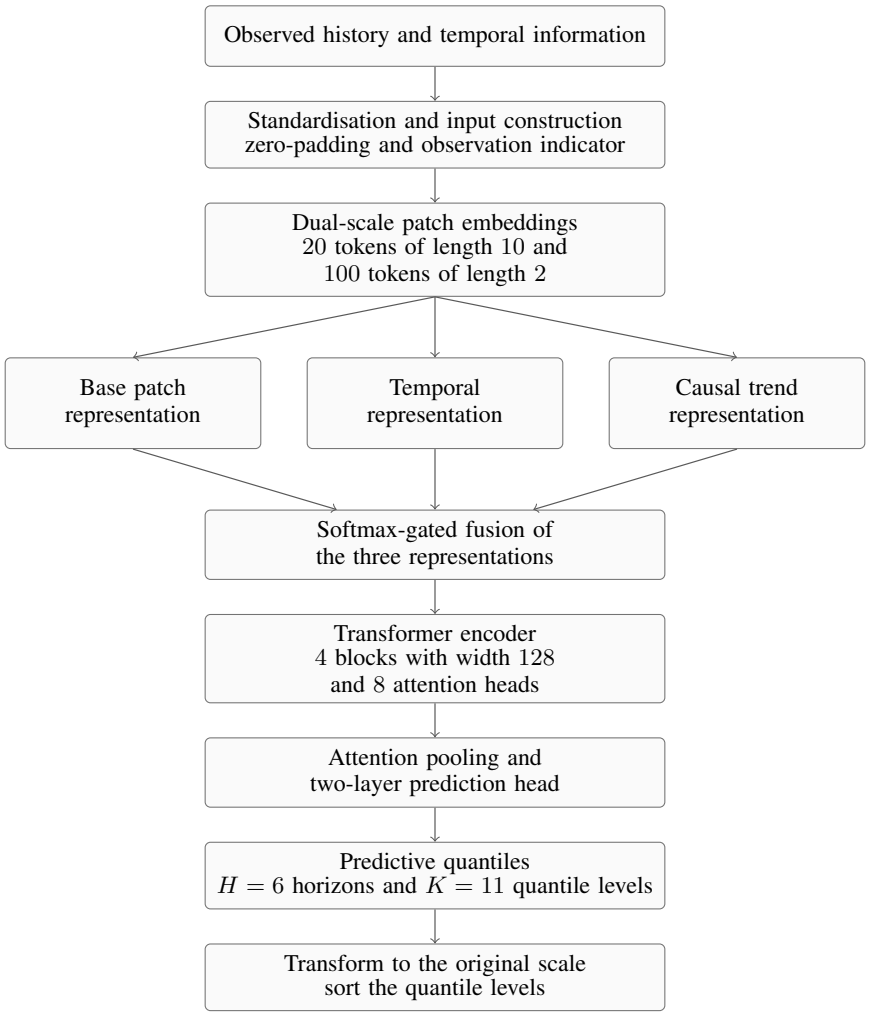


Figure 2: Overview of the GRAIN architecture. The standardised input is partitioned into patches at two scales. The resulting representations are fused and processed by a transformer encoder to produce predictive quantiles.

E.5 Optimisation, initialisation, and numerical precision

We use the Adam optimiser (Kingma and Ba, 2014) with learning rate 10^{-4}, and a batch size of 1024. All training computations use double precision. Embeddings are initialised independently from a uniform distribution in $[-0.05, 0.05]$. The depthwise convolution in the trend branch has 15 trainable filter coefficients per channel. These coefficients are initialised to $1/15$, so the branch initially acts as a moving-average smoother. During training, the coefficients can adapt away from this initialisation. Table 3 reports the complete set of architecture and optimisation settings used in the experiments.

F Evaluation on the M3 yearly data

The LGT model has previously achieved state-of-the-art performance on the yearly subset of the M3 competition dataset (Smyl et al., 2024; Long et al., 2025), which is also the model used to construct the estimated yearly task prior for training GRAIN. We therefore perform an in-domain evaluation on the M3 dataset. Following the original competition setup, we use the symmetric mean absolute percentage error (sMAPE, Makridakis and Hibon, 2000), and mean absolute scaled error (MASE,

Table 3: GRAIN architecture and optimisation settings.

	Setting	Value
Input	Maximum input length	200
	Long patch size (tokens)	10 (20)
	Short patch size (tokens)	2 (100)
	Total tokens	120
	Standardisation stabiliser	10^{-4}
Trend view	Convolution kernel / dilation	15 / 2
	Smoothing kernel	3
	Roughness penalty weight	10^{-4}
	Depthwise initialisation	All coefficients 1/15
Encoder	Model width	128
	Blocks	4
	Attention heads	8
	Feed-forward width	512
	Layer-normalisation stabiliser	10^{-6}
Head	Pooling queries / heads	4 / 4
	Quantile levels K	11
	Horizon H	6
	Output units	66
	Total parameters	1,540,772
Training	Optimiser	Adam
	Learning rate	10^{-4}
	Adam stabiliser	10^{-7}
	Batch size	1024
	Gradient clipping	None
	Dropout	0.1
	Precision	Float64

Hyndman and Koehler, 2006). They are calculated by

$$\text{sMAPE} = \frac{200}{H} \sum_{h=1}^{H} \frac{|f_h - \hat{f}_h|}{|f_h| + |\hat{f}_h|}, \tag{14}$$

$$\text{MASE} = \frac{H^{-1} \sum_{h=1}^{H} |f_h - \hat{f}_h|}{(T-1)^{-1} \sum_{t=2}^{T} |y_t - y_{t-1}|}. \tag{15}$$

In particular, the denominator of MASE is the mean absolute error of the in-sample one-step-ahead naïve forecast. The point forecast produced by GRAIN is taken to be the predictive median, that is, $\hat{f}_h = \widehat{q}_{h,k^\star}$ where $\alpha_{k^\star} = 0.5$. We additionally compare GRAIN with several established univariate forecasting methods, namely, Auto-ARIMA (Hyndman and Khandakar, 2008), Auto-ETS (Hyndman and Khandakar, 2008), Auto-Theta (Assimakopoulos and Nikolopoulos, 2000), and the naive forecast. The statistical methods are implemented using the StatsForecast library (Garza et al., 2022).

Table 4 reports point forecast accuracy and average inference time per series on the M3 yearly subset. GRAIN achieves lower MASE and sMAPE than all statistical baselines, other than MCMC-based LGT. Its forecast accuracy is also close to that of the full LGT model fitted separately to each series using Gibbs sampling. However, GRAIN requires only 11.9 ms per series compared with 4630.0 ms for LGT. This corresponds to an approximately 390-fold reduction in inference time. These results indicate that GRAIN can retain most of the predictive accuracy of per-series Bayesian estimation while substantially reducing its computational cost. Once trained, GRAIN can generate forecasts for new yearly series through a single forward pass without per-series parameter estimation or posterior sampling.

Table 4: Accuracy results and average inference time per series in milliseconds on the M3 yearly subset.

Model	MASE	sMAPE	Average Runtime (ms)
GRAIN	2.58	15.33	11.9
LGT	**2.50**	**14.99**	4630.0
AutoETS	2.70	16.19	4.4
AutoARIMA	2.89	16.77	106.5
AutoTHETA	2.60	16.62	4.3
Naive	3.17	17.88	**2.2**

G Evaluation metrics on the GIFT-Eval dataset

GIFT-Eval benchmark (Aksu et al., 2024) evaluates point and probabilistic forecast accuracy using MASE (Hyndman and Koehler, 2006) and CRPS (Gneiting and Raftery, 2007), respectively. Both metrics are normalised by the error of the naïve forecast. The formula to compute MASE is provided in (15). The CRPS is approximated by the mean weighted quantile loss,

$$\text{CRPS} \approx \frac{1}{K} \sum_{k=1}^{K} \text{wQL}[\alpha_k],$$

where $K = 9$ quantiles are selected for evaluation and $\alpha_k = 0.1k$ for $k = 1, \ldots, 9$. The weighted quantile loss $\text{wQL}[\alpha]$ is

$$\text{wQL}[\alpha] = 2 \frac{\sum_h \Lambda_\alpha(\hat{q}_{h,k}, f_h)}{\sum_h |f_h|},$$

where $\Lambda_\alpha(q, y)$ is the pinball loss (Koenker and Bassett Jr, 1978).

H Detailed results on the GIFT-Eval benchmark

In this section we provide detailed accuracy results of GRAIN compared with other baselines on the GIFT-Eval dataset in Table 5 and Table 6.

Table 5: MASE results on the GIFT-Eval yearly benchmark. A check mark ✓ indicates that GIFT-Eval flags the model for potential test-data leakage.

Rank	Model	Flagged for test-data leakage	MASE	Normalised MASE
1	TEMPO-ensemble	✓	2.398	0.605
2	TimesFM 2.0-500M	✓	2.538	0.640
3	GRAIN	-	**2.899**	**0.731**
4	Moirai-large	-	2.970	0.749
5	Moirai-small	-	2.980	0.751
6	xLSTM-Mixer	-	2.984	0.752
7	Kairos-23M	-	2.984	0.752
8	Kairos-10M	-	2.985	0.753
9	Moirai-base	-	3.010	0.759
10	Kairos-50M	-	3.044	0.768
11	Auto-ETS	-	3.080	0.777
12	TFT	-	3.090	0.779
13	Auto-Theta	-	3.110	0.784
14	N-BEATS	-	3.150	0.794
15	Chronos-2	-	3.236	0.816
16	PatchTST	-	3.290	0.830
17	TabPFN-TS	-	3.308	0.834
18	Moirai-2	-	3.320	0.837
19	TimesFM	✓	3.350	0.845
20	DeepAR	-	3.400	0.857
21	Chronos-2-synth	-	3.505	0.884
22	Chronos-Bolt-base	✓	3.507	0.884
23	TimesFM 2.5	-	3.575	0.901
24	Chronos-base	✓	3.640	0.918
25	Chronos-large	✓	3.640	0.918
26	Chronos-Bolt-small	✓	3.687	0.930
27	Auto-ARIMA	-	3.710	0.935
28	Chronos-small	✓	3.740	0.943
29	Naive	-	3.966	1.000

Table 6: CRPS results on the GIFT-Eval yearly benchmark. A check mark ✓ indicates that GIFT-Eval flags the model for potential test-data leakage.

Rank	Model	Flagged for test-data leakage	CRPS	Normalised CRPS
1	TimesFM 2.0-500M	✓	0.091	0.661
2	GRAIN	-	**0.103**	**0.754**
3	Moirai-large	-	0.104	0.758
4	Moirai-small	-	0.105	0.766
5	Moirai-base	-	0.105	0.766
6	Kairos-23M	-	0.110	0.801
7	TFT	-	0.110	0.802
8	Kairos-10M	-	0.110	0.802
9	Auto-ETS	-	0.111	0.809
10	Chronos-2	-	0.112	0.813
11	xLSTM-Mixer	-	0.112	0.818
12	DeepAR	-	0.113	0.824
13	Kairos-50M	-	0.114	0.831
14	Auto-Theta	-	0.115	0.839
15	Moirai-2	-	0.116	0.845
16	PatchTST	-	0.117	0.853
17	TimesFM	✓	0.117	0.853
18	TabPFN-TS	-	0.118	0.858
19	Chronos-2-synth	-	0.121	0.880
20	Chronos-Bolt-base	✓	0.121	0.886
21	TEMPO-ensemble	✓	0.127	0.924
22	TimesFM 2.5	-	0.127	0.927
23	Chronos-Bolt-small	✓	0.128	0.932
24	Auto-ARIMA	-	0.130	0.948
25	N-BEATS	-	0.134	0.977
26	Chronos-large	✓	0.135	0.984
27	Chronos-base	✓	0.135	0.984
28	Naive	-	0.137	1.000
29	Chronos-small	✓	0.139	1.014