LATENT INFERENCE-TIME GUIDANCE OF TIME SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time Series Foundation Models (TSFMs) currently provide state-of-the-art results in forecasting tasks. They are available out-of-the-box and rely on in-context learning to make their predictions, which makes the quality of their performance highly sensitive to the user-selected lookback, covariates, horizon and training data distributions. In practise, the quality of the forecasts are variable but complementary, which highlights the need for a principled ensembling approach, rather than selecting the best context. This paper introduces Latent Inference-Time Guidance for TSFMs, which adaptively combines a pool of TSFM forecasts through a time-dependent latent space with independent components. The framework comes equipped with identifiability and reconstruction guarantees, whilst maintaining the off-the-shelf aspect of foundation models. We provide experiments on datasets at various frequencies and from multiple domains: these show that the approach is competitive with traditional ensembling approaches.

1 INTRODUCTION

Time Series Foundation Models (TSFMs) have recently enabled competitive zero-shot and in-context forecasting across a wide range of domains. While deep learning methods for tabular and sequential data historically struggled to consistently outperform classical approaches (Shwartz-Ziv & Armon, 2022; Grinsztajn et al., 2022; McElfresh et al., 2023), recent TSFM architectures based on Transformers (Nie et al., 2023), in-context learning (Lu et al., 2025; Liu et al., 2025c) and large-scale pre-training have narrowed this gap (Liu et al., 2024). A wealth of these models are currently available (Grinsztajn et al., 2025; Ansari et al., 2025; Liu et al., 2025a; Auer et al., 2026; Qu et al., 2026), with benchmarks such as GIFT-Eval, fev-bench and Chronos available to compare them (Aksu et al., 2024; Shchur et al., 2025; Ansari et al., 2024). These models offer a compelling paradigm shift: rather than training a task-specific model, one can leverage a pre-trained model and adapt it to a chosen context at inference time.

Despite these advances, foundation models – and, more specifically, TSFMs – are sensitive to their input. For a given model, the prompt (which in the case of TSFMs encompasses the covariates, lookback and forecasting horizon choices) greatly affects the model accuracy (Michalkiewicz et al., 2025; Romanou et al., 2026). In short, different pre-trained TSFMs, or even different configurations of the same model, often exhibit complementary strengths across datasets and temporal regimes, due to their differences in architectures and pre-training data (Meyer et al., 2025; Berthelier et al., 2026). This variability suggests that the central challenge is not selecting a single model, but rather combining multiple forecasts in a principled and adaptive manner.

Classical approaches to forecasting with ensembles (Oliveira & Torgo, 2015), such as mixtures of experts and online aggregation methods (Gaillard et al., 2014; Wintenberger, 2017), as well as state-space techniques such as Kalman filtering (de Vilmarest et al., 2024), provide well-established solutions to this mixture of models problem. However, existing methods are typically restricted to linear aggregation schemes and do not exploit the representational structure of modern foundation models. Conversely, a foundation model can be fine-tuned on a specific dataset; but this is computationally expensive, statistically inefficient in low-data regimes, and can be impossible when the model is only accessible as a black box.

This motivates a different perspective, which we refer to as *inference-time guidance*. Combination mechanisms exist in generative modelling for time series and other modalities (Fedus et al., 2022; Liu

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.

et al., 2025a; Shi et al., 2025; Yang, 2026). These approaches combine expert models within a larger model. Rather than modifying or combining the existing foundation models themselves, we learn a lightweight mechanism that steers or combines the expert outputs using downstream observations. This paradigm has proved effective in generative modelling for time series and other modalities (Lee et al., 2025; Jiang et al., 2025; Bender & Morik, 2026), where external signals can guide pre-trained models without retraining. Beyond time series, approaches guiding existing foundation models exist, mostly using autoencoding architectures to identify relevant features at inference (Bricken et al., 2023; O'Neill et al., 2024; Le et al., 2024). However, to the best of our knowledge, and despite its conceptual appeal, there is no general probabilistic framework for inference-time latent guidance in time-series forecasting.

In this paper, we treat TSFM forecasts as a collection of experts, and use the combination of variational latent modelling with a structured state-space formulation to propose a principled probabilistic framework for combining TSFM forecasts, while capturing temporal dependencies and uncertainty. The resulting model introduces a time-dependent latent state that encodes how each expert should be weighted or corrected at each time step. In its simplest form, this reduces to a dynamic mixture of experts with time-varying weights; more generally, it defines a flexible nonlinear guidance mechanism acting on the forecasts. Importantly, our method leaves the underlying foundation models frozen: only the lightweight guidance mechanism is learned, without requiring access to or fine-tuning of the TSFM parameters. A key component of this approach is the structure given to the latent space, where rather than learning arbitrary hidden representations, the representations learned have an independent component decomposition, which connects the approach to work in nonlinear independent component analysis (ICA), where auxiliary variables such as temporal dependence and covariates enables identifiability of latent representations (Hälvä et al., 2021). In our setting, this allows guarantees on the induced denoised process.

Our contributions are threefold. First, we introduce a probabilistic latent guidance framework that adaptively combines and corrects forecasts from frozen TSFMs through a time-dependent latent state. Second, we connect this construction to structured nonlinear ICA and establish conditions for identifiability of the induced denoised process together with a reconstruction stability guarantee for the variational smoother. Third, we demonstrate the empirical benefits of this approach across datasets spanning multiple domains and sampling frequencies.

The remainder of the paper is organised as follows. In Section 2, we introduce the lines of research at the intersection of which this work lies. Section 3 introduces the proposed latent guidance model, its variational formulation, and its inference and forecasting procedures, as well as presenting identifiability analysis and theoretical guarantees. Experiments results are reported in Section 4, where we benchmark our proposed method with time series foundation models and classical aggregation methods. Section 5 discusses the avenues for future work. Additional technical details and proofs to our results are gathered in Appendix A and further experimental details are given in Appendix B.

2 Background and Related Works

Foundation models. Prior to 2023, a variety of deep learning architectures existed for tabular data and time series. However, these struggled to work better than smaller-scale statistical and machine learning methods (Grinsztajn et al., 2022; McElfresh et al., 2023). With Nie et al. (2023), the transformer architecture which had been introduced for computer vision and text-based tasks was adapted to tabular data, providing a deep learning architecture which was competitive with traditional methods. This was enabled the development of Tabular Foundation Models (TFMs), whose encoder or decoder architectures were pre-trained on large amounts of data. Encoder architectures mask parts of the time series and learn to reconstruct the missing segments (Woo et al., 2024; Liu et al., 2025a), whereas decoder architectures depend on causal modelling (Das et al., 2024). The first generation of TFMs were univariate, which limited their forecasting capacities. Hollmann et al. (2023); Qu et al. (2025) provided the TabPFN and TabICL models, which could learn tabular patterns in-context, allowing covariate data and multivariate forecasting. Alongside these in-context learning architectures, Ansari et al. (2025) released Chronos-2, an encoder architecture that takes context covariates and Das et al. (2024) have recently release TimesFM 3, which is the latest generation of a patch-based, transformer encoder architecture. Even in these latest generation of models, variability in performance can be observed due to the difference in synthetic data used at training, but these

models are now being used in many applications. Multiple TSFMs now combine multiple outputs to address this variability (Liu et al., 2025b). Benchmarks now exist to compare foundation models' performance (Aksu et al., 2024; Shchur et al., 2025), with new models coming out on an almost weekly basis.

TabICL is trained with in-context learning. To train the models, consider covariates $\mathbf{x}$ in input space $\mathcal{X} \subseteq \mathbb{R}^m$ (with $m > 1$) and target variables y in output space $\mathcal{Y} \subseteq \mathbb{R}$, from which n training points are sampled: $\mathcal{D}_{\text{train}} = (\mathbf{x}^i_{\text{train}}, y^i_{\text{train}})_{i=1}^n \sim p(\mathcal{D})$. Test points are sampled from the same distribution: $(\mathbf{x}_{\text{test}}, y_{\text{test}}) \sim p(\mathcal{D})$. Thus, predictions can be made with a single forward pass of a pre-trained neural network: $y_{\text{test}} \sim p(\cdot \mid \mathbf{x}_{\text{test}}, \mathcal{D}_{\text{train}})$. During training, the neural network parameters are updated with stochastic gradient descent. The models are trained with synthetic data (referred to as the *prior* in the literature), which is generated using additional preprocessing. Inference is then run on real-world datasets. The model now has variants which have been trained on time series following the same rationale, but can be used directly on time series in tabular format. We refer to both types as a TSFM.

Variational AutoEncoders. Variational Auto-Encoders (VAE) introduce approximations of a target conditional distribution in the context of latent data models, see Rezende et al. (2014); Kingma & Welling (2019). Consider target data $\mathbf{y}_{1:T} \in \mathbb{R}^{T\times m}$ with covariates $\mathbf{x}_{1:T} \in \mathbb{R}^{T\times d}$ and latent representation $\mathbf{z}_{1:T} \in \mathbb{R}^{T\times d_\ell}$. The latent variable generative model defines a joint density $(\mathbf{z}_{1:T}, \mathbf{y}_{1:T}) \mapsto p_\theta(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}, \mathbf{x}_{1:T})$ by specifying a prior $\mathbf{z}_{1:T} \mapsto p_\theta(\mathbf{z}_{1:T}|\mathbf{x}_{1:t})$ over the latent variable $\mathbf{z}_{1:T}$ and a conditional density $\mathbf{y}_{1:T} \mapsto p_\theta(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}, \mathbf{x}_{1:T})$ (also referred to as the decoder). The normalised log-likelihood of $(\mathbf{y}^i_{1:T})_{1\leq i\leq n}$ is therefore given by

$$\ell_n(\theta) = \frac{1}{n}\sum_{i=1}^n \log p_\theta(\mathbf{y}^i_{1:T}|\mathbf{x}_{1:T}) = \frac{1}{n}\sum_{i=1}^n \log \int p_\theta(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) p_\theta(\mathbf{y}^i_{1:T}|\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) \mathrm{d}\mathbf{z}_{1:T} \ ,$$

and the conditional distribution $p_\theta(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}, \mathbf{x}_{1:T}) \propto p_\theta(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) p_\theta(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}, \mathbf{x}_{1:T})$. Since the integral for the marginalising the latent variable is intractable, the marginal likelihood functions $p_\theta(\mathbf{y}^i_{1:T})$ for $1 \leq i \leq n$ are not available explicitly. So it is not possible to maximise the average marginal log-likelihood of the data. Since a maximum likelihood estimator cannot be computed simply, VAEs introduce a variational approach which aims to simultaneously provide a parameter estimate and an approximation of the conditional distribution of the latent variable given the observation. Consider a family of probability density functions $\{q_\varphi(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})\}_{\varphi\in\Phi}$. Then $\log p_\theta(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) \geq \mathcal{L}(\theta, \varphi, \mathbf{y}_{1:T}, \mathbf{x}_{1:T})$, where

$$\mathcal{L}(\theta, \varphi, \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) = \mathbb{E}_{q_\varphi(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})}\left[\log \frac{p_\theta(\mathbf{z}_{1:T}, \mathbf{y}_{1:T}|\mathbf{x}_{1:T})}{q_\varphi(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})}\right] \tag{1}$$

is the Evidence Lower BOund (ELBO). In this setting, $q_\varphi(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})$ is referred to as the *encoder*.

Variational excess-risk bounds exist for state-space models, and can be used to bound the risk. Chagneux et al. (2024b) use a backward factorisation of the variational distribution to provide bounds on the estimation error. Within the VAE literature, Chérief-Abdellatif et al. (2022) provide PAC-Bayesian error bounds on the reconstruction capacities of VAEs. Mbacke et al. (2023) develop conditional bounds which can then be applied to reconstruction, generation and re-generation. Hashimoto-Cullen et al. (2026) extend these conditional bounds to sequential data with a Markovian latent structure for reconstruction errors.

Structured Nonlinear Independent Component Analysis (ICA). Nonlinear ICA (Hyvärinen & Pajunen, 1999) assumes that observed data $\mathbf{x} \in \mathbb{R}^n$ is generated by some invertible nonlinear mixing function f from $\mathbf{s} \in \mathbb{R}^d$ latent independent components: $\mathbf{x} = f(\mathbf{s})$, where $p(\mathbf{s}) = \prod_{i=1}^d p(\mathbf{s}^{(i)})$. Hälvä et al. (2021) make additional assumptions on stationarity (for any t, t', then $(\mathbf{s}^{(i)}_t)_{1\leq i\leq d}$ and $(\mathbf{s}^{(i)}_{t'})_{1\leq i\leq d}$ are the same), conditional component independence, the nonlinear mixing function's injectivity and i.i.d. noise at each timestep, to propose Structured Nonlinear ICA (SNICA), where for independent latent components $\mathbf{s}_t = (\mathbf{s}^{(1)}_t, \ldots, \mathbf{s}^{(d)}_t)$, an observed $\mathbf{x}_t = f(\mathbf{s}_t) + \varepsilon_t$. Works proposing a structured VAEs which learns following a nonlinear ICA framework exist (Khemakhem et al., 2020; Brehmer et al., 2022). Connor et al. (2021) propose a VAE architecture with a learned latent structure.

3 LATENT GUIDANCE OF FOUNDATION MODELS

Recent advances in nonlinear ICA provide a principled framework for learning structured and identifiable latent representations in high-dimensional settings. In contrast to classical latent-variable models, nonlinear ICA leverages auxiliary structure (specifically for our setting; temporal dependencies and covariates) to ensure identifiability of the latent components. Each latent component exhibits its own dependency structure, while preserving conditional independence across components.

We introduce a nonlinear ICA perspective to introduce structured latent variables as a lightweight probabilistic interface for guiding pretrained foundation models. We call this framework **L**atent **I**nference-**Ti**me **G**uidance for **TSFM**s (LITiG-TSFM). The model is designed to operate at inference time without retraining the underlying foundation models. By modelling latent variables as time-dependent processes, the framework naturally captures sequential dependencies and allows for adaptive conditioning based on past observations or covariates.

This formulation is motivated by recent advances in nonlinear ICA, such as identifiable constructions (Hälvä et al., 2021, Δ-SNICA) and by the growing use of foundation models, for which lightweight probabilistic structures are needed to enable principled conditioning and guidance without retraining large-scale generative systems.

Notation. Consider $\mathbf{y}_{1:T} \in \mathbb{R}^{T\times m}$ a sequence of observations. Let $\mathbf{x}_{1:T} \in \mathbb{R}^{T\times d}$ be a sequence of covariates and $\mathbf{F} = (\mathbf{F}^j_{1:T})_{1\le j\le N} \in \mathbb{R}^{N\times T\times m}$ be a set of outputs of pre-trained foundation models. For all $t \ge 1$, we write $\mathbf{F}_t = F(\mathbf{x}_{1:t})$ the vector containing the predictions given by all foundation models at time t. We also write $\mathcal{N}(\cdot;\mu,\Sigma)$ for the Gaussian probability density function with mean μ and variance Σ.

Consider the following nonlinear ICA model LITiG-TSFM

$$y_t = h_\theta(\mathbf{F}_t, z_t) + \varepsilon_t \in \mathbb{R}^m$$
$$z_{t+1,i} = f_\theta(x_{t+1}, z_{t,i}) + \eta_t \in \mathbb{R}\ ,$$

with $z_t = (z_{t,i})_{1\le i\le p}, z_t \in \mathbb{R}^{d_\ell}$ the independent latent components, $(\varepsilon_t)_{t\ge1}$ and $(\eta_t)_{t\ge1}$ are i.i.d. with $\varepsilon_t \sim \mathcal{N}(0,\Sigma)$ and $\eta_t \sim \mathcal{N}(0,\rho^2)$. Since LITiG-TSFM is a nonlinear ICA model in which latent dynamics and observations are jointly parametrised and learned, its transition kernel $p_\theta(z_t \mid z_{t-1}, x_t)$ is modelled through a parametric map f_θ and the emission distribution $p_\theta(y_t \mid z_t)$ through a *decoder* h_θ.

The latent prior factorises as

$$p_\theta(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = p_\theta(z_1)\prod_{t=2}^{T}\prod_{i=1}^{p} p_\theta(z_{t,i} \mid z_{t-1,i}, x_t),$$

with Gaussian parameterisations $p_\theta(z_1) = \mathcal{N}(z_1;\mu_1,\rho_1^2)$ and $p_\theta(z_{t,i} \mid z_{t-1,i}) = \mathcal{N}(z_{t,i}; f_\theta(x_t, z_{t-1,i}), \rho^2)$ for each latent component p and time step t. The observation model satisfies

$$p_\theta(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_\theta(y_t \mid z_t), \quad p_\theta(y_t \mid z_t) = \mathcal{N}(y_t; h_\theta(\mathbf{F}_t, z_t), \Sigma).$$

To approximate the unknown posterior distribution, we consider a structured variational family:

$$q_\varphi(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) = q_{\varphi,T}(z_T \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) \prod_{t=1}^{T-1} q_{\varphi,t|t+1}(z_t \mid z_{t+1}, \mathbf{y}_{1:t}, \mathbf{x}_{1:t})\ .$$

A standard setting assumes that all conditional distributions are Gaussian with neural-network parametrisations. For instance, we may choose

$$q_{\varphi,T}(z_T \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) = \mathcal{N}(z_T; \mu_\varphi(\mathbf{y}_{1:T}, \mathbf{x}_{1:T}), \Sigma_\varphi(\mathbf{y}_{1:T}, \mathbf{x}_{1:T})) \tag{2}$$
$$q_{\varphi,t|t+1}(z_t \mid z_{t+1}, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \mathcal{N}(z_t; \mu_{\varphi,t}(z_{t+1}, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}), \Sigma_{\varphi,t}(z_{t+1}, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}))\ ,$$

where μ_φ, $\mu_{\varphi,t}$, Σ_φ, $\Sigma_{\varphi,t}$ are neural networks chosen depending on the applications.

The model is trained by maximising the ELBO given by (1) derived from the above variational decomposition. Rather than relying on its explicit closed form, we emphasise that the ELBO admits

a decomposition into three interpretable contributions: a reconstruction term associated with the observation model, a dynamical consistency term induced by the latent transition, and a regularisation term corresponding to the divergence between the variational posterior and the prior. This allows to derive an explicit loss function: the derivation is detailed in Appendix A.1.

Inference and forecasting. Once the model is trained, prediction and sampling rely on propagating the learned latent dynamics under the generative model p_θ, optionally combined with variational approximations. The predictive distribution is given by

$$p_\theta(y_{t+1} \mid \mathbf{y}_{1:t}) = \int p_\theta(y_{t+1} \mid z_{t+1})\, p_\theta(z_{t+1} \mid \mathbf{y}_{1:t})\, \mathrm{d}z_{t+1},$$

where

$$p_\theta(z_{t+1} \mid \mathbf{y}_{1:t}) = \int p_\theta(z_{t+1} \mid z_t, x_{t+1})\, p_\theta(z_t \mid \mathbf{y}_{1:t})\, \mathrm{d}z_t.$$

A practical approximation consists in initialising $z_t \sim q_\varphi(\cdot \mid \mathbf{y}_{1:t}, \mathbf{x}_{1:t})$, and propagating forward using the generative dynamics. To improve forecasting without retraining p_θ, one can refine the variational approximation by using a VAMP-type adaptation of the posterior at inference time (Tomczak & Welling, 2018). Further details are given in Appendix A.2.

Identifiability results. We now discuss identifiability of the latent guidance mechanism. Recall that, conditionally on the outputs of the foundation models, the observation equation of LITiG-TSFM can be written as

$$y_t = s_t + \varepsilon_t, \qquad \text{where } s_t = h_\theta(\mathbf{F}_t, z_t)\ .$$

In the following, all statements are made conditionally on $\sigma(\mathbf{F_t},\ t \geq 1)$.

A1 $(\mathbf{F}_t)_{t\geq 1}$ are assumed to be independent of $(z_t, \varepsilon_t)_{t\geq 1}$, and bounded.

The identifiability question is twofold. First, we ask whether the denoised signal $(s_t)_{t\geq 1}$ is identifiable from the noisy observations $(y_t)_{t\geq 1}$. Then, we ask whether the latent ICA components z_t are identifiable from s_t and $\mathbf{F}_t$. The first question follows from the noisy Structured Nonlinear ICA argument of Hälvä et al. (2021), while the second requires additional conditions on the decoder $(\mathbf{F}_t, z_t) \mapsto h_\theta(\mathbf{F}_t, z_t)$. Following Hälvä et al. (2021), we consider the conditions on the statistical properties of the signal for some $t_2 > t_1 \geq 1$.

A2 For some $\rho < 3$, there exist constants $A, B > 0$ such that, for all $\lambda \in \mathbb{R}^m$, $\mathbb{E}\left[\exp(\langle \lambda, s_{t_1}\rangle)\right] \leq A \exp(B\|\lambda\|^\rho)$.

A3 For all $\eta \in \mathbb{C}^m$, the random variable $\mathbb{E}[\exp(\langle \eta, s_{t_2}\rangle) \mid s_{t_1}]$ is not almost surely equal to zero.

A4 There does not exist $\eta \in \mathbb{R}^m$ and independent random variables $\tilde{s}, u$ such that u is a non-degenerate Gaussian random variable and $\langle \eta, s_{t_1}\rangle \overset{d}{=} \tilde{s} + u$.

Proposition 3.1 (Hälvä et al., 2021)**.** *Assume that A1–A4 hold. Then, up to translation, for all $k > 2$ and all $(t_3, \ldots, t_k)$, the map associating the distribution of $(s_{t_1}, \ldots, s_{t_k})$ to the distribution of $(y_{t_1}, \ldots, y_{t_k})$ is one-to-one. In particular, the distribution of the denoised latent guidance signal is identifiable from the noisy observations.*

We now derive a reconstruction guarantee. Following Chagneux et al. (2024a), our proposed structured ICA model provides reconstruction guarantees within a state-space modeling framework. In this setting, under a mixing-model assumption and assuming that the learned variational distribution is close in total variation distance to the true conditional distribution, yields a reconstruction risk that grows at most linearly with the time horizon. Let $\phi^\theta_{1:T}$ denote the true smoothing distribution of $\mathbf{z}_{1:T}$ given $(\mathbf{y}_{1:T}, \mathbf{x}_{1:T})$ and write ϕ^θ_t the true filtering distribution of z_t given $(\mathbf{y}_{1:t}, \mathbf{x}_{1:t})$ for $1 \leq t \leq T$. For all $2 \leq t \leq T$ and let $b^\theta_{t-1|t}(\cdot \mid z_{t+1}, \mathbf{x}_{1:t}, \mathbf{y}_{1:t})$ be the backward Markov kernel at time t:

$$b^\theta_{t-1|t}(z_{t-1} \mid z_t, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) \propto p_\theta(z_t|z_{t-1}, x_t)\phi^\theta_{t-1}(z_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{x}_{1:t-1})\ .$$

Therefore, $\phi^\theta_{1:T} = \phi^\theta_T \prod_{t=1}^{T-1} b^\theta_{t|t+1}$, which matches the factorisation of the variational family $q_\varphi = q_{\varphi,T} \prod_{t=1}^{T-1} q_{\varphi,t|t+1}$. Consider the following assumptions.

B1 There exist constants c_1, c_2 such that for all $t \geq 1$, θ, $\mathbf{F}_t$, z_t,

$$\|h_\theta(\mathbf{F}_t, z_t)\| \leq c_1, \quad |y_t| \leq c_2 \ .$$

B2 The proposed state-space model and the variational density satisfy a uniform mixing condition. There exist $0 < \sigma_- < \sigma_+ < \infty$ such that for all $1 \leq t \leq T-1$, θ, φ, $\mathbf{z}_{1:T}$, $\mathbf{x}_{1:T}$, $\mathbf{y}_{1:T}$,

$\sigma_- \leq p_\theta(z_{t+1}|z_t, x_{t+1}) p_\theta(y_{t+1}|z_{t+1}) \leq \sigma_+$ and $\sigma_- \leq q_{\varphi,t|t+1}(z_t \mid z_{t+1}, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) \leq \sigma_+$.

Assumption B2 is common in the nonlinear state space literature to obtain quantitative bounds for the approximation and control of joint smoothing distributions. This assumption does not hold on an unbounded state space. It can usually be satisfied by considering truncated latent spaces. In addition, note that compactly supported innovation law ensure A2 with $\rho < 2$.

B3 There exists $\varepsilon > 0$ such that

$$\|q_{\varphi,T}(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T}) - \phi_T^\theta(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})\|_{\mathrm{TV}} \leq \varepsilon,$$

and, for all $2 \leq t \leq T$ and all z_t,

$$\|q_{t-1|t}^\phi(\cdot \mid z_t, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) - b_{t-1|t}^\theta(\cdot \mid z_t, \mathbf{x}_{1:t}, \mathbf{y}_{1:t})\|_{\mathrm{TV}} \leq \varepsilon,$$

where $b_{t-1|t}^\theta$ is the backward smoothing kernel.

Proposition 3.2 turns a local variational approximation guarantee into a global smoothing guarantee for our latent mixture of foundation models. If the backward variational kernel are uniformly close in total variation to their exact smoothing counterparts, with local error at most ε, then the error on additive smoothing functionals, and classical reconstruction errors, grow at most linearly with the time horizon.

Proposition 3.2. *Assume that B1–B3 hold. Let*

$$\ell_t(z_t) = \|y_t - h_\theta(\mathbf{F}_t, z_t)\|^2.$$

Then there exists a constant $C > 0$ such that

$$\left| \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{q_\varphi(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})}[\ell_t(z_t)] - \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\phi_{1:T}^\theta(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})}[\ell_t(z_t)] \right| \leq C\varepsilon \ .$$

The excess reconstruction risk of the variational smoother with respect to the oracle smoother is controlled linearly by the approximation error of the backward variational kernels.

Proof. The proof is deferred to Section A.3. □

Proposition 3.2 bounds the gap between the variational reconstruction risk and the risk of the oracle smoother. The linear growth in T of the unnormalised bound is important: it shows that structured backward variational families can preserve the stability properties of exact smoothing and that local approximation errors do not compound exponentially.

4 EXPERIMENTS

To evaluate the performance of our proposed framework LITiG-TSFM, we implement it on datasets covering multiple domains, frequencies and levels of noise. All experiments are implemented in PyTorch and run on a MacBook Air with an M2 chip and 16GB RAM. Non-deterministic methods are each run ten times, and results are given as mean (std). We report compute cost for our method compared to its baselines for each dataset in Appendix B. The code is available on GitHub[1].

Metrics. To evaluate the reconstruction and forecasting tasks, we use the Root Mean Square Error and the Mean Absolute Error. Their expressions are given in Appendix B.

Datasets. To evaluate the LITiG-TSFM framework, we evaluate its forecasting capacities on datasets from multiple domains and at a variety of frequencies. This provides a wealth of model behaviours with varying levels of noise and seasonality, and from hourly to monthly timesteps. Further information on these datasets and how they are preprocessed is available in Appendix B.

[1] https://anonymous.4open.science/r/LITiG-TSFM-77FD/README.md

Dataset	Domain	Freq	Length	# Covariates	# Series
London Smart Meter (UK Power Networks, 2015)	Energy	D	606	17	5
Mauna Loa (Lan & Keeling, 2026)	Climate	M	627	9	1
NN5 Weekly (Godahewa et al., 2020)	Finance	W	837	4	111
Washington Bicycle Share (Fanaee-T, 2013)	Traffic	H	17378	30	1

Table 1: Datasets used to evaluate LITIG-TSFM and their key characteristics.

4.1 BASELINES

4.1.1 IDEAL SETTINGS BASELINES

Highest-performing TSFM context. We prompt TabICLv2, a foundation model with strong performances in the GIFT-Eval (Aksu et al., 2024) and fev-bench (Shchur et al., 2025) baselines. Rather than looking to find the best TSFM and its best configuration, we choose TabICLv2 as a representative sample of the current state of the art for TSFMs and use a set of its forecasts. To replicate implementation conditions where performance is evaluated on a static training dataset to select a model, the best TSFM from the validation fold of the data is retained as the best TSFM, and evaluated on the test fold of the data. Further details on how the TSFMs are prompted for each dataset are given in Appendix B. The group of high-performing forecasts is called *experts* for the rest of this work, and is the same for each of the methods implemented.

Oracle mixture. Consider a set of N experts $\{\mathbf{F}^k_{1:T}\}_{k=1}^N$ for a time series $\mathbf{y}_{1:T}$. At each time step $1 \leq t \leq T$, these experts have loss $p_{k,t}$. The oracle forecast of these experts is given by $\hat{y}_t = F_{\hat{k},t}$, where $\hat{k} = \arg\min_k p_{k,t}$ for $p_{k,t}$ the loss of an expert $\mathbf{F}^k_{1:T}$ at time step t. This benchmark is inaccessible at inference, since it requires having access to the ground truth to select the best expert at each time step. So we use it as a performance floor, rather than a baseline to beat.

4.1.2 MIXTURE OF EXPERTS BASELINES

Foundation models. We use TabICLv2 raw quantiles and mean forecasts as the experts which will be aggregated. The mean and standard deviation of these forecasts is reported (under the label *All TSFMs*), with the same context given for raw quantile and mean outputs. More detail on the contexts given to the TSFM is available in Appendix B.

Mixture of experts. In time series forecasting, principled mixtures of high-experts is a traditional method to improve forecasts with complementary strengths. For a set of forecasts $\mathbf{F}_t = \{F_{k,t}\}_{k=1}^K$, each with weight $\pi_{k,t}$ depending on historical performance and with $\sum_{k=1}^K \pi_{k,t} = 1$, the mixture of experts prediction is given by $y_t = \sum_{k=1}^K \pi_{k,t} F_{k,t}$. The ML-Poly algorithm (Gaillard et al., 2014) from the Opera package[2] is implemented on the experts obtained by prompting TabICLv2. The ML-Poly algorithm differs from mixture-of-expert TSFMs. Indeed, the weights are learned in a linear way, without additional neural network architectures (Liu et al., 2025b); also, they do not necessitate agentic evaluation (Cao et al., 2025). This approach comes with robustness guarantees to drifts in the data distribution, as does the oracle mixture. This makes ML-Poly an attractive method for time series forecasting, where distribution drift is a common challenge.

Kalman Filter. A Kalman Filter updates the weights of a state space model following the equations

$$\mathbf{y}_{t+1:t+H} = \alpha_t \mathbf{F}_t + \beta_t + \varepsilon_t, \quad \text{where} \quad \begin{pmatrix} \alpha_{t+1} \\ \beta_{t+1} \end{pmatrix} = A \begin{pmatrix} \alpha_t \\ \beta_t \end{pmatrix} + \eta_t,$$

α_t is the weights for the model output, β_t is the bias of the model, $\varepsilon \sim \mathcal{N}(0, Q)$ and $\eta_t \sim \mathcal{N}(0, R)$. In our experiment, we fix $A = \mathrm{Id}$ and learn Q, R using an expectation maximisation implemented in the viking_kalman package[3].

[2] https://github.com/Dralliag/opera-python

[3] https://github.com/EDF-Lab/viking_kalman

4.2 RESULTS

For each dataset, we train LITIG-TSFM with the objective derived in Appendix A. Details of the implemented neural networks are given in Appendix B. The encoder, decoder and prior architectures used are flexible and can be adapted to the specificities of each dataset.

Forecasting. We provide results for the London Smart Meter and NN5 datasets in Figure 1. Figure 2 provides results for the Washington Bicycle Share and Mauna Loa datasets. In both figures, the ML-Poly and Kalman Filter mean and standard deviation are reported for datasets with multiple time series; variance is not available for these models on single-series datasets, since the methods are deterministic. The oracle and best TSFM performances given for the London Smart Meter and NN5 datasets are given as the average oracle and best TSFM metrics over the series in the dataset. Figure 1 shows that when TSFM forecasts have a good performance, a mixture of experts method can still provide an improvement. Furthermore, LITIG-TSFM is competitive with its baseline methods.

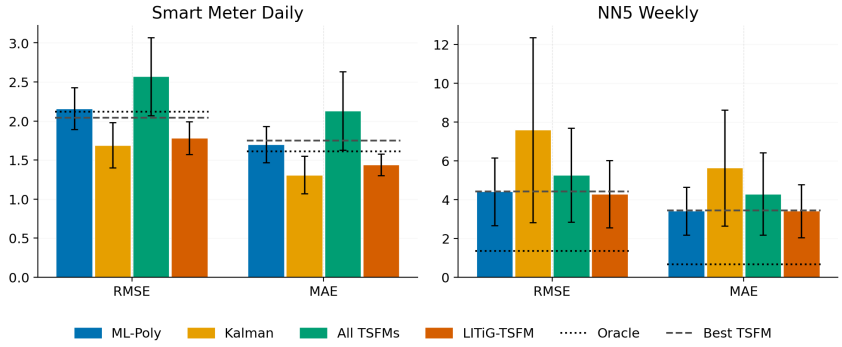

Figure 1: Average forecasting performance $\times 10^2$ on datasets with strong TSFM forecasts. Lower is better. Note that the y axes are on different scales.

The TSFM forecasts are much weaker on the Washington Bicycle Share and Mauna Loa datasets. On the Washington Bicycle Share dataset, the experts have an average (std) RMSE of 19.45 (6.61) and an average (std) MAE of 14.70 (5.65). For the Mauna Loa dataset, the average (std) RMSE is 45.27 (17.33) and the average (std) MAE is 35.34 (13.84). Therefore, Figure 2 does not show these metrics, to compare competitive mixture of experts methods only. For the Washington Bicycle Share dataset, the weak TSFM forecasts are due to the high frequency and levels of noise in the data. For the Mauna Loa data, the weak TSFM forecasts can be put down to the strong trend component of the data, which is monotonically increasing: the TSFM fits on the train fold of the data, and the test set is sequentially after the training fold, but the time series continues to grow beyond the domain originally seen. Whilst the TSFM can still detect the periodicity in the dataset, it will go back to the mean of the training fold, rather than correctly continue the growth in the training set. These two datasets present compelling use-cases for mixture-of-experts methods.

Expert ablation. Due to its formulation, the ML-Poly algorithm gives a forecast contained in the convex hull given by the expert forecasts. Thus, when an oracle forecast is not particularly strong, ML-Poly can only prove to a certain extent. This motivates the use of a Kalman Filter in the first instance. LITIG-TSFM then intervenes to go beyond the linear setting provided by the Kalman Filter. To illustrate this, we work in a setting where the performance of the TSFMs is known. We control the quality of the experts used in the ensemble methods to study whether LITIG-TSFM is sensitive to poor-performing experts, and whether its performance deteriorates faster than other ensembling methods. After ranking the TSFMs' performance on the test segments, the strongest nine experts are chosen as the set of best experts; the five weakest experts are retained as the set of poor experts. The worst is taken for the setting with one poor expert, and the three worst are taken for the setting with three poor experts. Figure 3 compares LITIG-TSFM to its ensembling baselines' performances when these poor experts are added to a strong set of forecasts. The error of the TSFMs grows with the addition of weaker experts; LITIG-TSFM stays robust to this and does not deteriorate, which makes it competitive with the ML-Poly and Kalman Filter baselines.

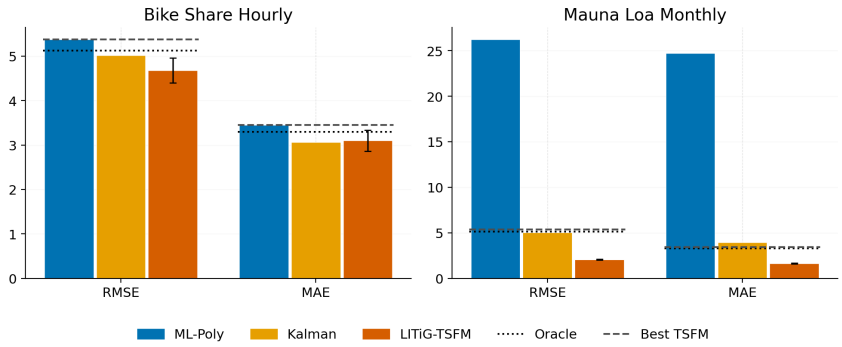

Figure 2: Average forecasting performance $\times 10^2$ on datasets with poor TSFM forecasts: for the Washington Bicycle Share dataset, mean RMSE (std) is 19.45 (6.61) and mean MAE (std) is 14.70 (5.65); for the Mauna Loa dataset, mean RMSE (std) is 45.27 (17.33) and mean MAE (std) is 35.34 (13.84). Lower is better. Note that the y axes are on different scales.

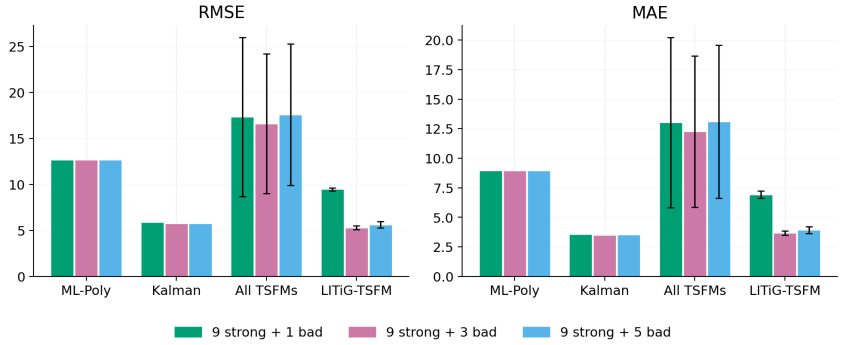

Figure 3: Experts ablation: LITiG-TSFM performance compared to baseline ensemble methods on the Washington Bicycle Share dataset, with nine strong experts and one, three and five poor ones respectively. Metrics given $\times 10^{-2}$, lower is better.

Appendix B provides a further ablation study on the architecture of the encoder and decoder, which highlights the possible gains from making relevant architectural design choices. Table 7 shows that for a set of prompted experts, running a Kalman Filter on a dataset with multiple time series or with high frequency time steps will take longer than training and prompting LITiG-TSFM. Furthermore, LITiG-TSFM does not require a warm-up window as the Kalman Filter or ML-Poly algorithms do; this means that in fewer steps at inference, the forecast becomes useful, and we do not need to discard the initial steps.

5 Discussion, Limitations and Conclusion

This work lies at the intersection of non-linear ICA, variational inference and TSFMs, providing a probabilistic inference-time guidance framework (LITiG-TSFM) for combining TSFM forecasts through latent structured state-space modelling. This is accompanied by existing guarantees on the latent state dynamics, and adapts reconstruction guarantees to the proposed frameworks, as well as providing novel forecasting guarantees. Whilst we use forward factorisations of the time series when implementing LITiG-TSFM, the reconstruction guarantee is given in a backward factorisation, as it builds on existing works. Furthermore, some of the assumptions made to achieve the results can be too restrictive for a real-life dataset. Our experiments use TabICLv2 as an example of a TSFM to create experts; however, it would be possible to implement the same experiments with any existing out-of-the-box TSFM.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, to clean and reformat datasets and to support qualitative and thematic data analysis. We have not used generative AI tools for dataset generation, help in developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments or interpreting results, and assisting with translation is not applicable to this work. Additionally, we used generative AI tools for creating and editing software code. We have reviewed all AI-assisted work: all LLM-generated code was verified and tested for correctness by the lead author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The results given in Section 3 are derived from cited works. Assumptions made to apply the results are given in Section 3. The proofs for further theoretical guarantees developed for LITiG-TSFM are given in Appendix A. The code for LITiG-TSFM is linked to an anonymous Github in Section 4. Further empirical details on the empirical implementation are given in Appendix B.

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. 2024. URL https://openreview.net/forum?id=Z2cMOOANFX.

Abdul Fatir Ansari, Lorenzo Stella, Ali 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 Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

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, October 2025. URL http://arxiv.org/abs/2510.15821. arXiv:2510.15821 [cs].

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=v7UqniC9pF.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Sidney Bender and Marco Morik. Visual disentangled diffusion autoencoders: Scalable counterfactual generation for foundation models. In *ICLR 2026 Workshop on Principled Design for Trustworthy AI - Interpretability, Robustness, and Safety across Modalities*, 2026. URL https://openreview.net/forum?id=tKA7DqmEI1.

Gaspard Berthelier, Mariia Baranova, Andrei-Tiberiu Pantea, Etienne Le Naour, Adrien Petralia, Tahar Nabil, and Themis Palpanas. Investigating simple target-covariate relationships for Chronos-2 and TabPFN-TS. In *1st ICLR Workshop on Time Series in the Age of Large Models*, 2026. URL https://openreview.net/forum?id=H8p5Yc0c3W.

Johann Brehmer, Pim de Haan, Phillip Lippe, and Taco Cohen. Weakly supervised causal representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

Defu Cao, Michael Gee, Jinbo Liu, Hengxuan Wang, Wei Yang, Rui Wang, and Yan Liu. Conversational Time Series Foundation Models: Towards Explainable and Effective Forecasting, December 2025. URL http://arxiv.org/abs/2512.16022. arXiv:2512.16022 [cs].

Mathis Chagneux, Elisabeth Gassiat, Pierre Gloaguen, and Sylvain Le Corff. Additive smoothing error in backward variational inference for general state-space models. *Journal of Machine Learning Research*, 25(28):1–33, 2024a. URL http://jmlr.org/papers/v25/22-1392.html.

Mathis Chagneux, Pierre Gloaguen, Sylvain Le Corff, and Jimmy Olsson. Importance sampling for online variational learning. *arXiv preprint arXiv:2402.02859*, 2024b.

Badr-Eddine Chérief-Abdellatif, Yuyang Shi, Arnaud Doucet, and Benjamin Guedj. On PAC-Bayesian reconstruction guarantees for VAEs. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 3066–3079. PMLR, May 2022. URL https://proceedings.mlr.press/v151/cherief-abdellatif22a.html.

Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*. PMLR, 13–15 Apr 2021.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, pp. 10148–10167. PMLR, 2024.

Joseph de Vilmarest, Jethro Browell, Matteo Fasiolo, Yannig Goude, and Olivier Wintenberger. Adaptive probabilistic forecasting of electricity (net-)load. *IEEE Transactions on Power Systems*, 39(2):4154–4163, 2024.

Hadi Fanaee-T. Bike sharing dataset, 2013. URL https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pp. 176–196. PMLR, 2014.

Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. NN5 weekly dataset, June 2020. URL https://doi.org/10.5281/zenodo.4656125.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models, November 2025. URL http://arxiv.org/abs/2511.08667. arXiv:2511.08667 [cs].

Hermanni Hälvä, Sylvain Le Corff, Luc Lehéricy, Jonathan So, Yongjie Zhu, Elisabeth Gassiat, and Aapo Hyvarinen. Disentangling identifiable features from noisy data with structured nonlinear ICA. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Chloé Hashimoto-Cullen, Ghislain Agoua, Benjamin Guedj, and Sylvain Le Corff. PAC-Bayesian reconstruction guarantees for time series variational autoencoders, 2026. URL https://arxiv.org/abs/2609.05212.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*. PMLR, 26–28 Aug 2020.

Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Dr. Xin Lan and Dr. Ralph Keeling. Mauna loa co_2 monthly mean data, 2026. URL https://gml.noaa.gov/ccgg/trends/data.html.

Nhat Minh Le, Neel Patel, Ciyue Shen, Blake Martin, Alfred Eng, Chintan Shah, Sean Grullon, and Dinkar Juyal. Learning biologically relevant features in a pathology foundation model using sparse autoencoders. In *Advancements In Medical Foundation Models: Explainability, Robustness, Security, and Beyond*, 2024. URL https://openreview.net/forum?id=daV16mhUBd.

Thomas L Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight online adaption for time series foundation model forecasts. In *Forty-second International Conference on Machine Learning*, 2025.

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, November 2025a. URL http://arxiv.org/abs/2511.11698.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-MoE: Empowering time series foundation models with sparse mixture of experts. In *Forty-second International Conference on Machine Learning*, 2025b.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *International Conference on Machine Learning*, pp. 32369–32399. PMLR, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=KMCJXjlDDr.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Jiecheng Lu, Yan Sun, and Shihao Yang. In-context time series predictor. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dCcY2pyNIO.

Sokhna Diarra Mbacke, Florence Clerc, and Pascal Germain. Statistical Guarantees for Variational Autoencoders using PAC-Bayesian Theory, December 2023. URL http://arxiv.org/abs/2310.04935. arXiv:2310.04935 [cs].

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Marcel Meyer, David Zapata Gonzalez, Sascha Kaltenpoth, and Oliver Müller. Benchmarking time series foundation models for short-term household electricity load forecasting. *IEEE Access*, 13: 218141–218153, 2025.

Mateusz Michalkiewicz, Sheena Bai, Mahsa Baktashmotlagh, Varun Jampani, and Guha Balakrishnan. Not all views are created equal: Analyzing viewpoint instabilities in vision foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9113–9123, 2025.

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

Mariana Oliveira and Luis Torgo. Ensembles for time series forecasting. In *Proceedings of the Sixth Asian Conference on Machine Learning*, volume 39 of *Proceedings of Machine Learning Research*, Nha Trang City, Vietnam, 26–28 Nov 2015. PMLR.

Charles O'Neill, Christine Ye, Kartheik G. Iyer, and John F Wu. Towards interpretable scientific foundation models: Sparse autoencoders for disentangling dense embeddings of scientific concepts. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024. URL https://openreview.net/forum?id=mPq3R6jdtD.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning*. PMLR, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SxsyLjIfWB.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2014.

Angelika Romanou, Mark Ibrahim, Candace Ross, Chantal Shaib, Kerem Oktar, Samuel J Bell, Anaelia Ovalle, Jesse Dodge, Antoine Bosselut, Koustuv Sinha, et al. Brittlebench: Quantifying llm robustness via prompt sensitivity. *arXiv preprint arXiv:2603.13285*, 2026.

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

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022. ISSN 1566-2535.

Leslie N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, 2017. doi: 10.1109/WACV.2017.58.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pp. 1214–1223. PMLR, 2018.

UK Power Networks. SmartMeter Energy Consumption Data in London Households, 2015. URL https://gml.noaa.gov/ccgg/trends/data.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Olivier Wintenberger. Optimal learning with Bernstein online aggregation. *Machine Learning*, 106 (1):119–141, 2017.

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*, Proceedings of Machine Learning Research. PMLR, 2024.

Janghoon Yang. Expert transformer: Fine-grained mixture-of-experts for time-series forecasting. *Expert Systems with Applications*, pp. 132634, 2026.

A Technical Results

A.1 Derivation of the Evidence Lower Bound

Recall that the target data is $\mathbf{y}_{1:T} \in \mathbb{R}^{T \times m}$, with covariate data $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times d}$ and the N expert forecasts for $\mathbf{y}_{1:T}$ are $\mathbf{F}^i_{1:T} \in \mathbb{R}^{N \times T \times m}$, where $1 \leq i \leq N$. For ease of notation, denote $\mathbf{x}_{1:T} = (\mathbf{x}_{1:T}, \mathbf{F}^{1:N}_{1:T})$ for the rest of this derivation. Consider the joint distribution $p_\theta(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}, \mathbf{y}_{1:T})$ whose parameters $\theta \in \Theta$ we would like to learn. Since the posterior $p_\theta(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:T})$ is intractable, we use variational inference to approximate it, and maximise the ELBO on parameters (θ, φ), which is given by

$$\mathcal{L}(\mathbf{y}_{1:T}, \mathbf{x}_{1:T}; \theta, \varphi) = \mathbb{E}_{q_\varphi(\cdot \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})} \left[\log \frac{p_\theta(\cdot, \mathbf{y}_{1:T} \mid \mathbf{x}_{1:T})}{q_\varphi(\cdot \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})} \right] .$$

With the definition of the joint distribution and the properties of the logarithm, this can be written as:

$$\begin{aligned} &\mathcal{L}(\mathbf{y}_{1:T}, \mathbf{x}_{1:T}; \theta, \varphi) \\ &\qquad = \mathbb{E}_{q_\varphi(\cdot \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})} \left[\log p_\theta(\cdot) + \log p_\theta(\mathbf{y}_{1:T} \mid \mathbf{x}_{1:T}, \cdot) - \log q_\varphi(\cdot \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) \right] . \end{aligned} \tag{3}$$

Within the expectation, each term can be derived in closed form. The *prior* is factorised as $p_\theta(\mathbf{z}_{1:T}) = p_\theta(z_1) \prod_{t=2}^T p_\theta(z_t \mid z_{t-1})$, where we parametrise $p_\theta(z_1)$ with $\mathcal{N}(z_1; \mu_1, \rho_1)$ and $p_\theta(z_t \mid z_{t-1})$ is parametrised by $\mathcal{N}(z_{t+1}; f_\theta(x_{t+1}, z_t), \rho^2)$. The logarithm of the prior distribution is therefore given by:

$$\begin{aligned} \log p_\theta(\mathbf{z}_{1:T}) &= \log \left(p_\theta(z_1) \prod_{t=2}^T p_\theta(z_t \mid z_{t-1}) \right) \\ &= \log p_\theta(z_1) + \sum_{t=2}^T \log p_\theta(z_t \mid z_{t-1}) \\ &= -\frac{1}{2} \log(2\pi\rho_1^2) - \frac{(z_1 - \mu_1)^2}{2\rho_1^2} + \sum_{t=2}^T \left(-\frac{1}{2} \log(2\pi\rho^2) - \frac{(z_t - f_\theta(x_t, z_{t-1}))^2}{2\rho^2} \right) \\ &= -\frac{1}{2} \log(2\pi\rho_1^2) - \frac{(z_1 - \mu_1)^2}{2\rho_1^2} - \frac{T}{2} \log(2\pi\rho^2) - \sum_{t=2}^T \frac{(z_t - f_\theta(x_t, z_{t-1}))^2}{2\rho^2} \\ &= -\frac{1}{2} \left(\log(2\pi\rho_1^2) + T \log(2\pi\rho^2) - \frac{(z_1 - \mu_1)^2}{2\rho_1^2} - \sum_{t=2}^T \frac{(z_t - f_\theta(x_t, z_{t-1}))^2}{2\rho^2} \right) . \end{aligned}$$

The Gaussian *decoder* $p_\theta(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}, \mathbf{x}_{1:T})$ factorises into $p_\theta(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) = \prod_{t=1}^T p_\theta(y_t \mid z_t)$, where the conditioning on x_t is implicit from the dependency on z_t, and where $p_\theta(y_t \mid z_t)$ is parametrised by $\mathcal{N}(y_t; h_\theta(F_t, z_t), \Sigma)$. The logarithm of this distribution is given by:

$$\begin{aligned} \log p_\theta(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) &= \log \prod_{t=1}^T p_\theta(y_t \mid z_t) \\ &= \sum_{t=1}^T \log p_\theta(y_t \mid z_t) \\ &= -\frac{T}{2} \log \left(\det(2\pi\Sigma) \right) - \frac{1}{2} \sum_{t=1}^T \left(y_t - h_\theta(z_t, F_t) \right)^\top \Sigma^{-1} \left(y_t - h_\theta(z_t, F_t) \right) . \end{aligned}$$

The posterior $p_\theta(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:T})$ is intractable. We estimate it with the variational family $q_\varphi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = q_\varphi(z_1 \mid x_1, y_1) \prod_{t=2}^T q_\varphi(z_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}, z_{t-1})$, where as per (2), $q_\varphi(z_1 \mid x_1, y_1)$ and $q_\varphi(z_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}, z_{t-1})$ are Gaussians parametrised by $\mathcal{N}(z_1; \mu_\varphi(y_1, x_1); \Sigma_\varphi(y_1, x_1))$ and $\mathcal{N}(z_t; \mu_{\varphi,t}(z_{t-1}, \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}), \Sigma_{\varphi,t}(z_{t-1}, \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}))$ respectively. The logarithm of the

variational family is therefore:

$$\begin{aligned}
\log q_\varphi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:T}) &= \log\left(q_\varphi(z_1 \mid x_1, y_1)\prod_{t=2}^{T} q_\varphi(z_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}, z_{t-1})\right) \\
&= \log q_\varphi(z_1 \mid x_1, y_1) + \sum_{t=2}^{T} \log q_\varphi(z_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}, z_{t-1}) \\
&= \log\left[\frac{1}{\sqrt{2\pi\sigma_\varphi^2(x_1)}}\exp\left(-\frac{(z_1-\mu_\varphi(y_1))^2}{2\sigma_\varphi^2(x_1)}\right)\right] + \sum_{t=2}^{T}\log\left[\frac{1}{\sqrt{2\pi\sigma_{\varphi,t}^2}}\exp\left(-\frac{(z_t-\mu_{\varphi,t}(\mathbf{y}_{1:t}))^2}{2\sigma_{\varphi,t}^2}\right)\right] \\
&= -\frac{1}{2}\left(\log(2\pi\sigma_\varphi^2(x_1)) - \frac{(z_1-\mu_\varphi(y_1))^2}{2\sigma_{\varphi,t}^2(x_1)}\right) + \sum_{t=2}^{T}\left[\log(2\pi\sigma_{\varphi,t}^2) - \frac{(z_t-\mu_{\varphi,t}(\mathbf{y}_{1:t}))^2}{2\sigma_{\varphi,t}^2}\right] .
\end{aligned}$$

Plugging these values into (3) gives the following ELBO:

$$\begin{aligned}
\mathcal{L}(\theta, \phi; \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) = \mathbb{E}_{q_\varphi(\cdot|\mathbf{y}_{1:T}, \mathbf{x}_{1:T})}\Bigg[&-\frac{1}{2}\log(2\pi\rho_1^2) + T\log(2\pi\rho^2) - \frac{(z_1-\mu)^2}{2\rho_1^2} \\
&- \sum_{t=2}^{T}\frac{(z_t - f_\theta(x_t, z_{t-1}))^2}{2\rho^2} + \frac{1}{2}\Bigg(-T\log(2\pi\sigma^2) + \sum_{t=1}^{T}\left(-\frac{(y_t - h_\theta(z_t, F_t))^2}{2\sigma^2}\right) \\
&+ \frac{1}{2}\log(2\pi\rho_1^2) - T\log(2\pi\rho^2) + \frac{(z_1-\mu)^2}{2\rho_1^2} + \sum_{t=1}^{T}\frac{(z_t - f_\theta(x_t, z_{t-1}))^2}{2\rho^2}\Bigg)\Bigg] .
\end{aligned}$$

A.2 Derivation of the forecasting distribution

The conditional expectation of a new observation given past observations and covariates can be written as:

$$\begin{aligned}
\mathbb{E}_{p_\theta(\cdot|\mathbf{y}_{1:t}, \mathbf{x}_{1:t+1})}\left[y_{t+1}\right] &= \int p_\theta(y_{t+1} \mid \mathbf{y}_{1:t}, \mathbf{x}_{1:t+1})\mathrm{d}y_{t+1} \\
&= \int p_\theta(y_{t+1} \mid z_{t+1})p_\theta(z_{t+1} \mid z_t, x_{t+1})p_\theta(z_t \mid \mathbf{y}_{1:t}, \mathbf{x}_{1:t})\mathrm{d}y_{t+1}\mathrm{d}z_{t+1}\mathrm{d}z_t .
\end{aligned}$$

This predictive expectation can be approximated using the trained variational family:

$$q_\varphi(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T}) = q_{\varphi,T}(z_T \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})\prod_{t=1}^{T-1} q_{\varphi,t|t+1}(z_t \mid z_{t+1}, \mathbf{y}_{1:t}, \mathbf{x}_{1:t}) .$$

In this setting, the expectation under $p_\theta(z_t \mid \mathbf{y}_{1:t}, \mathbf{x}_{1:t})$ can be approximated either by an expectation under $q_\varphi(z_t \mid \mathbf{y}_{1:t}, \mathbf{x}_{1:t})$ or using a VAMP-like sampler (Tomczak & Welling, 2018):

$$\mu_{\text{vamp}}(z_t) = \frac{1}{n}\sum_{i=1}^{n} q_\varphi(z_t \mid \mathbf{y}_{1:t}^i, \mathbf{x}_{1:t}) .$$

A.3 Proof of Proposition 3.2

Proof. Conditionally on $(\mathbf{x}_{1:T}, \mathbf{F}_{1:T})$ the proposed nonlinear ICA model is a general state space model. Under the strong mixing assumption **B2**, we can apply Proposition 1 of Chagneux et al. (2024a) using the additive state functional

$$h_{1:T}(\mathbf{z}_{1:T}) = \sum_{t=1}^{T}\ell_t(z_t) = \sum_{t=1}^{T}\|y_t - h_\theta(\mathbf{F}_t, z_t)\|^2 .$$

Therefore, for all $1 \leq k \leq T-1$, and all probability densities $\tilde{q}_k$,

$$\left|\sum_{t=1}^{T} \mathbb{E}_{q_\varphi(\cdot|\mathbf{y}_{1:T},\mathbf{x}_{1:T})}[\ell_t(z_t)] - \sum_{t=1}^{T} \mathbb{E}_{\phi^\theta_{1:T}(\cdot|\mathbf{y}_{1:T},\mathbf{x}_{1:T})}[\ell_t(z_t)]\right| \leq 2\frac{\sigma_+}{\sigma_-}\sum_{t=1}^{T-1}\|\ell_t\|_\infty \times \left(c_1(\theta) + \sum_{m=2}^{k}\rho^{k-m+1}c_m(\theta,\varphi) + c_{k+1}(\theta,\varphi) + \sum_{m=k+2}^{T}\rho^{m-k-1}c_m(\theta,\varphi)\right) ,$$

where $\rho = 1 - \sigma_-/\sigma_+$ and $c_1(\theta) = \|\tilde{q}_1 - \phi^\theta_1\|_{\mathrm{tv}}$ and for $2 \leq k \leq T$,

$$c_k(\theta,\varphi) = \left\|\tilde{\phi}^\theta_{k-1|k} - \tilde{\nu}^\varphi_{k-1|k}\right\|_{\mathrm{tv}} ,$$

where $\tilde{\phi}^\theta_{k-1|k}(z_{k-1},z_k) \propto \tilde{q}_{k-1}(z_{k-1})p_\theta(z_k|z_{k-1},x_k)p_\theta(y_k|z_k)$ and $\tilde{\nu}^\varphi_{k-1|k}(z_{k-1},z_k) = \tilde{q}_k(z_k)q_{\varphi,k-1|k}(z_{k-1} \mid z_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$. By choosing for all $1 \leq k \leq T$, $\tilde{q}_k = \phi^\theta_k$, yields $c_1(\theta) = 0$ and, dropping the dependency on the covariates $\mathbf{x}_{1:T}$ for better clarity,

$$\begin{aligned} c_k(\theta,\varphi) &= \left\| \frac{\phi^\theta_{k-1}(z_{k-1})p_\theta(z_k|z_{k-1})p_\theta(y_k|z_k)}{\int \phi^\theta_{k-1}(z_{k-1})p_\theta(z_k|z_{k-1})p_\theta(y_k|z_k)\mathrm{d}z_{k-1}\mathrm{d}z_k} - \phi^\theta_k(z_k)q_{\varphi,k-1|k}(z_{k-1} \mid z_k, \mathbf{y}_{1:k-1})\right\|_{\mathrm{tv}} \\ &\leq \left\| \frac{\phi^\theta_{k-1}(z_{k-1})p_\theta(z_k|z_{k-1})p_\theta(y_k|z_k)}{\int \phi^\theta_{k-1}(z_{k-1})p_\theta(z_k|z_{k-1})p_\theta(y_k|z_k)\mathrm{d}z_{k-1}\mathrm{d}z_k} - \phi^\theta_k(z_k)b^\theta_{k-1|k}(z_{k-1} \mid z_k, \mathbf{y}_{1:k})\right\|_{\mathrm{tv}} \\ &\quad + \left\|\phi^\theta_k(z_k)b^\theta_{k-1|k}(z_{k-1} \mid z_k, \mathbf{y}_{1:k}) - \phi^\theta_k(z_k)q_{\varphi,k-1|k}(z_{k-1} \mid z_k, \mathbf{y}_{1:k-1})\right\|_{\mathrm{tv}} . \end{aligned}$$

By definition of the backward kernel, the first term in the last inequality is 0 and therefore, by assumption **B3**,

$$c_k(\theta,\varphi) \leq \varepsilon .$$

Then, by assumption **B1** which gives $\|y_t - h_\theta(\mathbf{F}_t, z_t)\|^2 \leq \|y_t\| + \|h_\theta(\mathbf{F}_t, z_t)\| \leq c_2 + c_1$, we can bound the square norm:

$$\|y_t - h_\theta(\mathbf{F}_t, z_t)\|^2 \leq 2c_2^2 + 2c_1^2 ,$$

so that $\|\ell_t\|_\infty \leq 2c_1^2 + 2c_2^2$, which gives $C \coloneqq 2\frac{\sigma_-}{\sigma_+}(2c_2^2 + 2c_1^2)$. This concludes the proof. □

B Empirical Details

Metrics. For a time series $\mathbf{y}_{1:T}$ and a sequence of predictions $\hat{\mathbf{y}}_{1:T}$, the *Root Mean Square Error (RMSE)* and *Mean Absolute Error (MAE)* are given by

$$\mathrm{RMSE}(\mathbf{y}_{1:T}, \hat{\mathbf{y}}_{1:T}) = \left(\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_t - y_t)^2\right)^{1/2} , \quad \mathrm{MAE}(\mathbf{y}_{1:T}, \hat{\mathbf{y}}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}|\hat{y}_t - y_t| .$$

TSFM prompts for expert creation. We use TabICLv2[4] (Qu et al., 2026) in regression mode to predict the next time step. For this appendix, we call the *training fold* of the dataset the initial part of the time series which is given as context to the TSFM. The TSFM then makes single-step forecasts using that original context, and uses those forecasts for the following steps. At a chosen frequency, the ground truth for the forecast made replaces the forecast in the context for the following time steps. So for example, for a daily dataset with monthly prompting, the TSFM will use the original training fold, then make forecasts on a daily basis for a month; at the end of that month, the true value of the time series will be revealed and added to the context, replacing the last month's forecasts. For the NN5, Mauna Loa and Washington Bicycle Share datasets, we use a high-frequency prompt and a lower-frequency prompt. For the London Smart Meter dataset, we only use low-frequency prompts. For each frequency used, we use both the mean and the raw quantile outputs from TabICLv2 as our experts.

[4] https://tabicl.readthedocs.io/en/latest/

Datasets. *Individual consumption aggregated by socio-economic profile.* The London Smart Meter dataset (UK Power Networks, 2015) records the electricity consumption of households with a smart meter in London between 2012 and 2014. The covariates include Acorn segments, which provide a socio-demographic segmentation[5] for each household. The dataset is aggregated by Acorn segments: experiments are run on a group of these segment (segments K, L, M, N and O, which are part of the 'Steadfast Communities' group), to see how LITiG-TSFM handles energy data in a noisy setting. It also evaluates how the model performs when learning to forecast for multiple categories at once. Electricity consumption is the target variable; more information on the available covariates and how they are preprocessed is given in Table 2.

Feature	Preprocessing
Acorn segment	One-hot encoded
Number of clients	Normalised between 0 and 1
Date	Year cyclically encoded
Weekday	One-hot encoded
Temperature	Normalised between 0 and 1
Consumption (target)	Normalised between 0 and 1

Table 2: Preprocessing for variables in the London Smart meter dataset.

UK ATM demand. The NN5 dataset (Godahewa et al., 2020) provides weekly demand for 111 ATMs in the UK. Besides the demand (which is the target variable), only the date is known – as noted in Table 3 – the month and week of the year are cyclically encoded. This tests LITiG-TSFM in a setting with multiple series learned at once, despite having very little covariate information and a large amount of noise.

Feature	Preprocessing
Demand (target)	Normalised between 0 and 1
Month of year	Cyclically encoded
Week of year	Cyclically encoded

Table 3: Preprocessing for variables in the NN5 weekly dataset.

Atmospheric CO2 levels. Lan & Keeling (2026) provide monthly mean measurements of atmospheric CO_2 at Mauna Loa. It has strong seasonal and trend components. The target data is detrended by differencing (each y_t is replaced by $y_t^{\text{detrended}} = y_t - y_{t-1}$), to avoid testing on out-of-domain data. Preprocessing information for available covariates is given in Table 4. This datasets tests LITiG-TSFM in a setting where the distribution of the test fold is different from that of the training fold.

Feature	Preprocessing
Positional encoding	Encoded between 0 and 1
Year	Normalised between 0 and 1
Month	Cyclically encoded
Mean monthly concentration (target)	Detrended, normalised between 0 and 1; lagged
Deseasonalised mean monthly concentration	Detrended, normalised between 0 and 1; lagged
Number of days with measurements	Normalised between 0 and 1
Standard deviation	Normalised between 0 and 1
Uncertainty	None

Table 4: Preprocessing for variables in the Mauna Loa monthly dataset.

Bicycle share dataset. The Washington Bicycle Share dataset (Fanaee-T, 2013) provides the number of rented bicycles on an hourly time step over 2011 – 2012, alongside weather and calendar covariates. Two categories of users exist: subscribed users and casual users. The total number of users is given

[5] https://acorn.caci.co.uk/how-acorn-works/

as a target variable. The lags of subscribed users and casual users are given as covariates, as well as the lag for the total number of users. This dataset tests LITiG-TSFM on a noisy, high-frequence dataset. Further information on how these variables were preprocessed is given in Table 5.

Feature	Preprocessing
Season	One-hot encoded
Bank holiday	One-hot encoded
Day of week	One-hot encoded
Hour	Cyclically encoded
Month	Cyclically encoded
Working day	One-hot encoded
Temperature	Normalised between 0 and 1
Average temperature	Normalised between 0 and 1
Humidity	Normalised between 0 and 1
Windspeed	Normalised between 0 and 1
Weather conditions	One-hot encoded
Number of non-subscription users	Normalised between 0 and 1, lag taken
Number of subscribed users	Normalised between 0 and 1, lag taken
Total users (target)	Normalised between 0 and 1, lag added

Table 5: Preprocessing for variables in the Washington Bicycle Share dataset.

Model architecture. LITiG-TSFM is implemented in PyTorch. Training is conducted with the AdamW optimiser (Loshchilov & Hutter, 2019) that uses a cyclic learning rate scheduler (Smith, 2017). The components of the LITiG-TSFM model are implemented as follows:

- *Encoder.* An attention layer (Vaswani et al., 2017) is preceded and followed by a normalisation layer (Ba et al., 2016). This is followed by two linear layers, separated by a GELU activation function. The output mean has a sigmoid activation function; the output variance is clamped between -6 and 0.
- *Prior.* The prior is made up of a linear input layer, a GELU activation function and an output layer which feeds through a sigmoid activation function.
- *Decoder.* The decoder is made up of a linear input layer, a GELU activation function, a linear output layer with a sigmoid activation function.

Hyperparameters. Each dataset requires domain- and frequency-aware hyperparameter selection. These modelling choices enable strong forecasting performance. Table 6 gives the main hyperparameters which required per-dataset tuning.

Hyperparameter	Smart Meter	Bike Share	NN5 Weekly	Mauna Loa
Sequence length	64	96	16	196
Batch size	128	256	64	128
Learning rate	5×10^{-3}	1×10^{-3}	1×10^{-3}	5×10^{-3}
Latent dimension	64	64	32	64
Epochs	200	20	60	500
Number heads in encoder	4	4	4	4
Encoder hidden layer	128	256, 128	128	128
Prior hidden layer	64	64	64	64
Decoder hidden layer	128	128	128	64

Table 6: Hyperparameters chosen for each dataset.

Inference-time compute. In our setting, implementing the ML-Poly mixture of experts method requires almost no additional computational overhead. Calculating the oracle forecast and the best expert likewise does not require additional time. Table 7 provides compute times for the TSFM

inference (on higher and lower frequency of prompts, for mean and quantile outputs respectively), the expectation maximisation for the Kalman Filter and training and inference time for the LITiG-TSFM algorithm. The additional compute overhead needed to run expectation maximisation on the Kalman Filter further justifies using a non-linear encoder for LITiG-TSFM: as stated in Section 3, the linear latent state cannot be disentangled. This is especially true for the London Smart Meter and NN5 datasets, where the Kalman Filter needs to be refit for each time series in the dataset, whereas LITiG-TSFM only requires to be trained once and can then run inference for every series in the dataset.

Dataset	TSFM low freq.	TSFM high freq.	Kalman	LITiG-TSFM train + infer
Smart Meter*	1'44, 1'27	Not used	35"	3'04" + 1"
Bike Share	39', 10'	60', 10'	13'02"	9'48" + 33"
NN5*	10", 9"	40", 59"	5"	37" + 1"
Mauna Loa	1', 1'	14', 16'	1'	4' + 2"

Table 7: Compute cost for LITiG-TSFM compared to the baseline models. Datasets with a * show cost for fitting the Kalman Filter and running LITiG-TSFM inference a single series. TSFM prompting times show the time needed to make a mean forecast and a raw quantiles forecast.

Qualitative samples. We provide sample time series and their forecasts for the éCO$_2$mix, London Smart Meter (Acorn segment K) and Washington Bicycle share datasets. This provides an additional qualitative perspective on where a mixture of experts method is a valuable addition to TSFM forecasts.

Figure 4 shows a sample from the London Smart Meter dataset, which provides a challenging case to the experts, where they consistent predict smaller quantities than the actual outcome. LITiG-TSFM is able to correct for this, and has more occurrences of clearly identifying the extreme values of peaks and valleys than the Kalman Filter. It also it has the advantage of only needing to be trained once for all the segments in the time series; whereas the Kalman Filter would require refitting for each time series in the dataset.

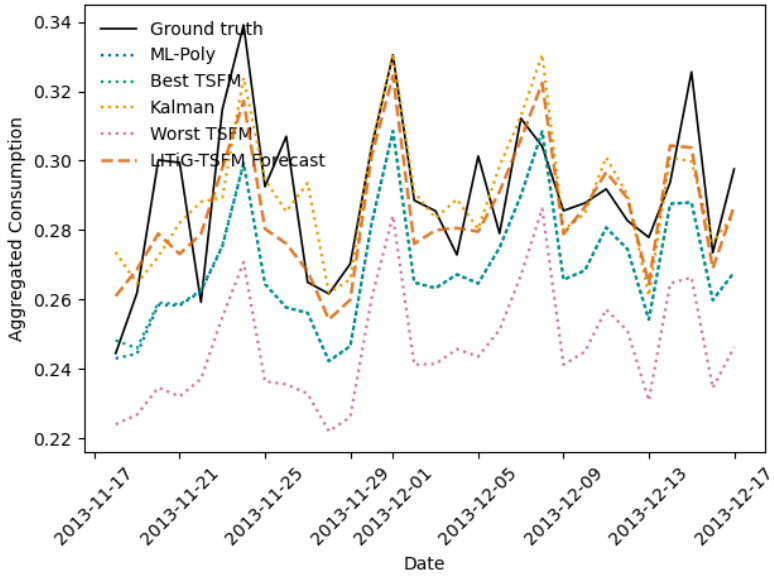

Figure 4: London Smart Meter, Acorn Segment K forecast sample of LITiG-TSFM and its baselines.

The sample from the Washington Bicycle Share dataset in Figure 5 is similar to the London Smart Meter segment shown, where there is a large gap between the worst performing expert and target value. LITiG-TSFM is again capable of fitting the peaks and valleys, and it is faster at training and inference combined, compared to the Kalman Filter. This is due to the length of the time series to forecast, which is at much higher frequency than the London Smart Meter sample provided above.

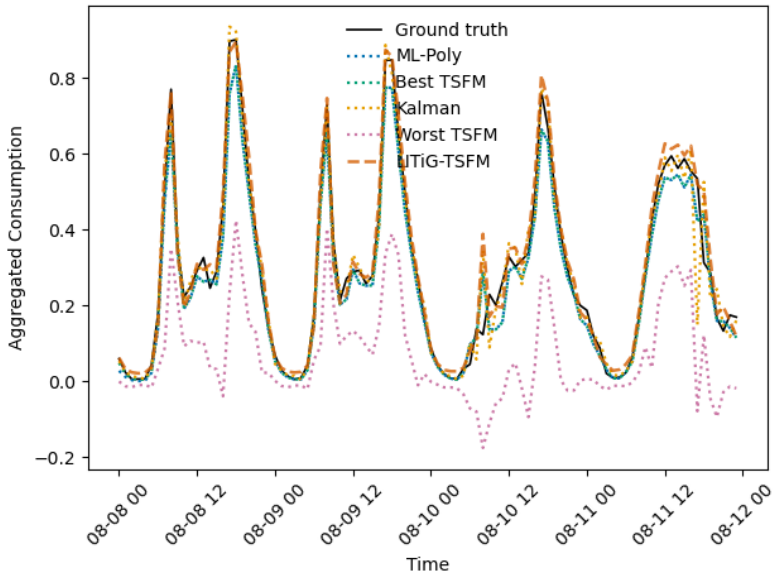

Figure 5: Washington Bike Share forecast sample of LITiG-TSFM and its baselines.

Figure 6 provides a sample for a sample from the NN5 dataset, alongsides the forecasts from the baseline approaches. This shows that ML-Poly and Kalman either other-smooth the mixture of predictions, or are not able to learn the pattern consistently. LITiG-TSFM bridges these two approaches.

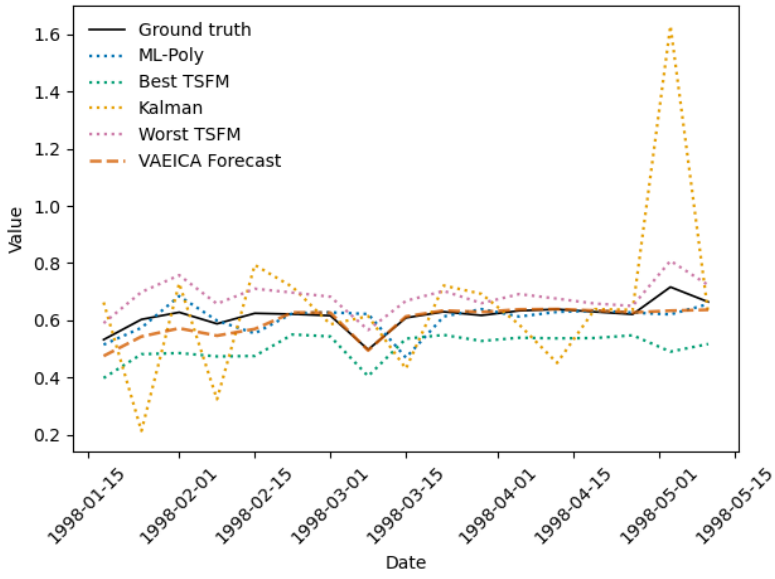

Figure 6: NN5 forecast sample of LITiG-TSFM and its baselines.

Figure 7 provides a forecast from the detrended Mauna Loa time series.

Architecture ablation. We conduct an ablation study on LITiG-TSFM's reconstruction performance, by working with different encoder and decoder architectures for the NN5 dataset. The trained models are evaluated on their reconstruction performance on the training data. The performance for each of these models is given in Figure 8, where the MLP Decoder and Attention Encoder are as

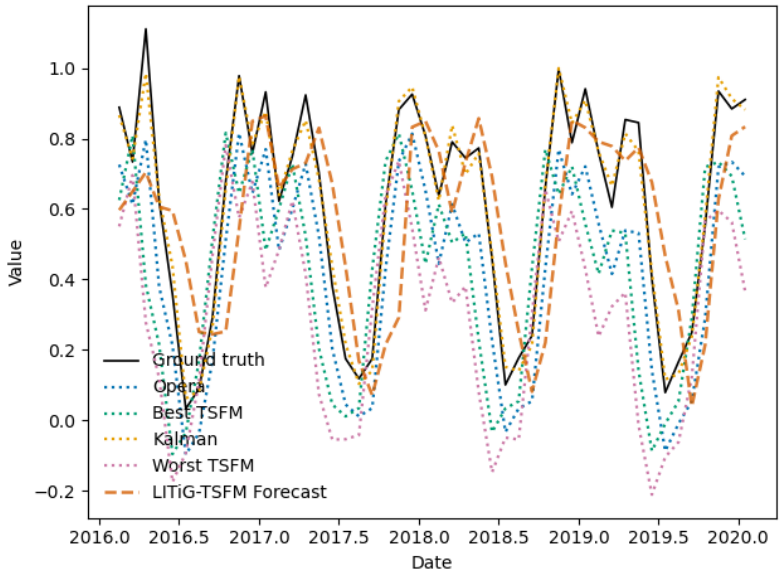

Figure 7: Detrended Mauna Loa forecast sample of LITiG-TSFM and its baselines.

described in Appendix B. The *MLP Encoder* is made up of a linear layer of size 32 and positional encoding of the position in the sequence, a LeakyReLU activation with slope 1.15, a linear layer of size 16 and a linear output for mean and covariance, where the mean has a Sigmoid activation layer and the variance is clamped between -6 and 2. The *dot product Decoder* constrains the latent space to be of the same size as the number of experts used, as it simply takes the dot product of the latent state z_t by the experts $\mathbf{F}_t$, with Gaussian noise $\varepsilon \sim \mathcal{N}(0, 1)$: $\mathbf{y_t} = \mathbf{F}_t \cdot z_t + \varepsilon$. These experiments use a latent space of dimension 100, which corresponds to the number of experts: this is a constraint when using the dot product Decoder. The attention Encoder and MLP Decoder use a latent space of dimension 64.

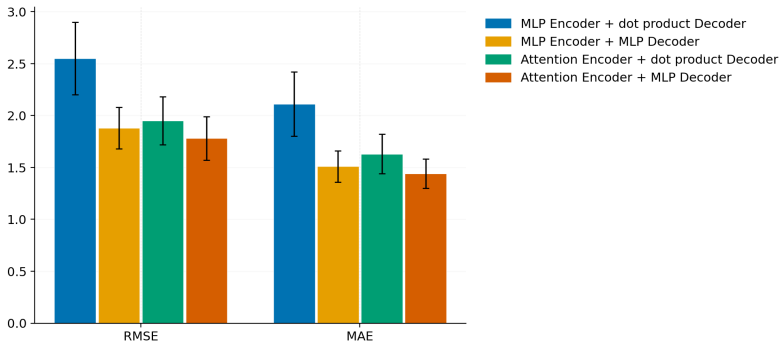

Figure 8: Architecture ablation: LITiG-TSFM performance on the London Smart Meter dataset with different encoder and decoders. Metrics given $\times 10^{-2}$, lower is better.

Using a Multi-Layered Perceptron (MLP) in the decoder enables a stronger performance. Similarly, an attention-based encoder allows to better embed the sequential dependencies in the dataset. This is due to two factors: first, the most sophisticated decoder architecture means that the decoding can be learned more efficiently. Second, it means that the dimension of the latent space no longer depends on the number of experts, which gives choice for the dimension of the latent space. This gives more possibilities for model tuning, thus making it stronger.