JOINT PRIOR-FITTED PREDICTION FOR PARTIALLY OBSERVED MULTIVARIATE TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series are rarely observed as a complete history followed by a fully unknown future. In healthcare, measurements can be missing within the context, channels can stop at different times, and some covariates may be already available over the forecast horizon. Many time-series foundation models operate on fixed forecast blocks and return point predictions or marginal predictive summaries, leaving cross-time and cross-variable dependence unavailable as an explicit joint density. We introduce JOINTTIMEPFN, a mask-conditional prior-data fitted network that predicts arbitrary missing cells in a fixed-length time-by-channel grid. Its query-size-agnostic low-rank Gaussian head returns a coherent joint posterior predictive distribution and its marginals in one forward pass. The model combines mask-aware bidirectional temporal encoding, rotary embeddings, and a broad linear model of coregionalization prior. This structured prior naturally motivates posterior predictive checks against exact and MCMC-based multi-output Gaussian process references, both in-prior and under distribution shift. When the reference posterior exhibits substantial dependence across prediction targets, the joint head closes 64% of the gap left by a correlation-free baseline and mostly outperforms autoregressive chaining of its marginals in joint likelihood on these synthetic references. On MIMIC-IV, a single model checkpoint performs competitively with baselines across zero-shot forecasting, known-future conditioning, ragged horizons, and interior imputation, while improving joint likelihood over its diagonal counterpart.

1 INTRODUCTION

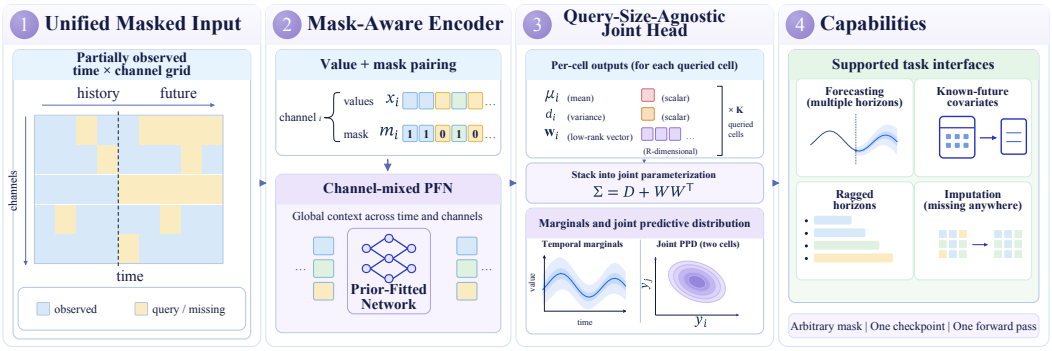
Multivariate temporal data often arrive as a partially observed grid instead of a complete prefix. For instance in intensive care, laboratory values and vital signs are measured at different times, treatments and planned interventions may be known ahead of some outcomes, and several missing values may need to be imputed together (Johnson et al., 2023). A model must therefore condition on every observed cell, regardless of its position, and quantify uncertainty over the full set of unknown values.

Prior-data fitted networks (PFNs) amortize zero-shot inference by approximating the posterior predictive of a synthetic pretraining distribution (Müller et al., 2022; 2025). While successfully adapted for multivariate time series (Dooley et al., 2023; Taga et al., 2025; Hoo et al., 2026; Potapczynski et al., 2026), current temporal PFNs retain the standard fixed-horizon setup, representing uncertainty through independent marginals, quantiles, or samples. Bayesian inference allows conditioning on arbitrary observation patterns to produce a joint predictive distribution, but existing PFNs do not combine arbitrary grid conditioning with an explicit joint density over queried cells.

Joint predictive distributions are needed when several variables are considered simultaneously. Two predictors can have identical marginals and still assign very different probabilities to a given trajectory shape. Multi-output Gaussian processes (MOGPs) provide a classical framework for such problems, conditioning on arbitrary observations and returning a coherent joint posterior over finite query sets (Álvarez & Lawrence, 2011). However, they are computationally expensive, since inference is cubic in the conditioned cells and the kernel must be fitted per dataset. This leaves a gap between the flexible joint interface of MOGPs and the scalable amortized inference offered by foundation models.

Recent work has leveraged MOGPs as a means of constructing synthetic pretraining distributions, in particular through linear models of coregionalization (LMCs, Álvarez et al. 2012), to train time-series PFNs (Taga et al., 2025). These models demonstrate effective amortized prediction for multivariate

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.



Method family	Temporal multivariate target	Arbitrary observed/query mask (▲: forecast-only masking.)	Prior-fitted	Tractable joint PPD
MOGP (Álvarez et al., 2012)	✓	✓	✗	✓
TACTiS-2 (Ashok et al., 2024)	✓	✓	✗	✓
Chronos-2 (Ansari et al., 2025)	✓	▲	✗	✗
TiRex-2 (Podest et al., 2026)	✓	▲	✗	✗
TimePFN (Taga et al., 2025)	✓	✗	✓	✗
TabPFN-TS (Hoo et al., 2026)	✗	▲	✓	✗
ApolloPFN (Potapczynski et al., 2026)	✗	▲	✓	✗
TabImpute (Feitelberg et al., 2026)	✗	✓	✓	✗
JointTimePFN (this work)	✓	✓	✓	✓

Figure 1: **Overview of the method and positioning against related work.**

time series, but focus on forecasting and are evaluated primarily via marginal predictions. Whether a PFN can also preserve the cross-target dependence of the underlying MOGP posterior, or more generally recover an accurate joint predictive over arbitrary query sets, remains less clear.

Contributions. We introduce JointTimePFN, a probabilistic PFN that unifies forecasting and imputation on a masked multivariate time-series grid by modeling marginal and joint predictive distributions. Figure 1 summarizes the interface and related methods. Our contributions are:

- **Conceptual.** We formulate forecasting, known-future conditioning, ragged horizons, and imputation as four prediction regimes within a single prior-fitted problem over arbitrary masks, bringing joint probabilistic prediction over queried cells to the zero-shot temporal PFN setting.
- **Methodological.** We combine mask-aware bidirectional temporal encoding with a query-size-agnostic low-rank Gaussian head, trained on a broadened LMC prior with continuously sampled kernel parameters and signed loadings. One forward pass returns an explicit joint Gaussian predictive over arbitrary queried cells and its analytic marginals.
- **Empirical & Evaluation framework.** We evaluate joint posterior fidelity against exact and MCMC-based MOGP references, using a diagonal baseline to reveal the amount of dependence in the true posterior, and evaluate a checkpoint zero-shot on MIMIC-IV under all four regimes, against an autoregressive baseline constructed from its own marginals and a cohort-trained attentional copula.

2 Related Work

Time-series foundation models. Existing large pretrained forecasters cover many variations, including tokenized probabilistic models (Ansari et al., 2024), decoder-only transformers (Das et al., 2024), universal patch-based models (Woo et al., 2024), and flow-based predictors (Liu et al., 2025), with some models, such as Chronos-2 and TiRex-2, adding multivariate targets and past or future covariates (Ansari et al., 2025; Podest et al., 2026). They usually forecast a fixed horizon from an observed context instead of conditioning on an arbitrary observed pattern; masked pretraining objectives such as MOMENT's do admit arbitrary missingness (Goswami et al., 2024), but none of these models outputs a coherent joint density over the targeted values.

PFNs for temporal prediction and missing data. PFNs approximate the Bayesian posterior predictive associated with a synthetic task prior (Müller et al., 2022; 2025). ForecastPFN uses a univariate prior (Dooley et al., 2023), while TimePFN, the closest predecessor and the backbone on which we build, pairs an LMC generator with channel-mixed patch tokens for multivariate

time-series point forecasting (Taga et al., 2025). TabPFN-TS utilizes tabular regression through temporal features (Hoo et al., 2026), and ApolloPFN adds a time-aware synthetic graph prior, rotary embeddings, and native exogenous conditioning (Potapczynski et al., 2026). TabImpute extends prior-fitted prediction to arbitrary missing entries in unordered tables, but it is not temporally native (Feitelberg et al., 2026). To the best of our knowledge, no existing method in this line combines temporal multivariate structure, arbitrary masking, and a coherent joint predictive head.

Joint probabilistic prediction. In the PFN literature, autoregressive chain-rule factorization is commonly used for joint prediction, but introduces order-dependence and may violate marginalization consistency (Bruinsma et al., 2023; Klötergens et al., 2026). Correlated trajectories can also be recovered from marginal forecasters with copulas (Baron et al., 2025). Both approaches construct cross-cell dependence post hoc from marginal models not explicitly trained to capture it. We instead output the full joint over queried values within a prior-fitted model, and our LMC pretraining prior also provides an exact joint posterior, enabling direct evaluation against this Bayesian reference (Jeong & Kim, 2023; Jiang et al., 2025). Outside the zero-shot setting, TACTiS-2 (Ashok et al., 2024) provides a related approach learning an attentional copula directly for multivariate forecasting and interpolation.

3 Background

3.1 Multivariate temporal prediction

Let $\boldsymbol{Y} \in \mathbb{R}^{T \times N}$ denote a multivariate series with N channels and T time indices. Standard multi-horizon prediction selects a forecast origin $t_\star < T$ and predicts a rectangular future block from the observed history. If $\mathcal{I}_{\text{tar}}$ denotes the target channels and $\mathcal{I}_{\text{cov}}$ denotes covariates whose future values are known, the probabilistic forecasting target is

$$p(\boldsymbol{Y}_{t_\star+1:T,\mathcal{I}_{\text{tar}}} \mid \boldsymbol{Y}_{1:t_\star,:}, \boldsymbol{Y}_{t_\star+1:T,\mathcal{I}_{\text{cov}}}) . \tag{1}$$

The future-covariate term is omitted in standard multivariate forecasting. A point forecaster learns

$$\widehat{\boldsymbol{Y}}_{t_\star+1:T,\mathcal{I}_{\text{tar}}} = f_\theta(\boldsymbol{Y}_{1:t_\star,:}, \boldsymbol{Y}_{t_\star+1:T,\mathcal{I}_{\text{cov}}}) . \tag{2}$$

Probabilistic forecasters commonly return one marginal distribution or set of quantiles per horizon step, or generate forecast trajectories through sampling. The interface in Equation 1 nevertheless assumes a shared cutoff and a designated future target block.

3.2 Prior-data fitted inference

PFNs amortize prediction over synthetically generated tasks (Müller et al., 2022; 2025). Let $\xi \sim p(\xi)$ denote a sampled data-generating process, $\mathcal{D} \sim p(\mathcal{D} \mid \xi)$ a dataset generated from it, and $s \sim p(s \mid \mathcal{D})$ a rule that partitions the dataset into a context $\mathcal{D}_{\text{ctx}}$ and query pairs $\mathcal{D}_{\text{qry}} = \{(x_i, y_i)\}_{i=1}^{K}$. For the PFN predictive distribution $q_\theta(y \mid x, \mathcal{D}_{\text{ctx}})$, the standard query-wise objective is

$$\mathcal{L}_{\text{PFN}}(\theta) = \mathbb{E}_{\xi,\mathcal{D},s}\left[-\frac{1}{K}\sum_{i=1}^{K} \log q_\theta(y_i \mid x_i, \mathcal{D}_{\text{ctx}})\right] . \tag{3}$$

Under the log score and sufficient capacity, each output approaches the corresponding posterior predictive marginal induced by the pretraining distribution (Müller et al., 2022; Reuter et al., 2025). The process distribution $p(\xi)$ and partition rule $p(s \mid \mathcal{D})$ jointly determine which inference problems are amortized. Temporal PFNs instantiate $\mathcal{D}$ as a time series and typically choose s to be a history–horizon split at a single time stamp, with past values as context and future values as targets.

3.3 Linear models of coregionalization and multi-output Gaussian processes

Gaussian processes (GPs) are a natural source of synthetic tasks for temporal PFNs. A GP places a distribution over functions through a mean function and covariance kernel, so any finite collection of function values is jointly Gaussian (Rasmussen & Williams, 2006). A PFN trained on GP-generated tasks amortizes the corresponding posterior predictive, while the exact GP provides a tractable Bayesian reference under Gaussian observation noise and fixed hyperparameters.

For multivariate series, linear models of coregionalization construct N outputs from Q independent latent GPs (Álvarez et al., 2012). Let $\boldsymbol{A} = (a_{nq}) \in \mathbb{R}^{N\times Q}$ be the loading matrix. Then

$$u_q \sim \mathcal{GP}(0, k_q), \qquad f_n(t) = \sum_{q=1}^{Q} a_{nq} u_q(t), \qquad y_n(t) = f_n(t) + \epsilon_n(t), \tag{4}$$

where $\epsilon_n(t) \sim \mathcal{N}(0, \sigma_n^2)$ independently, so that $\mathrm{Cov}[f_n(t), f_{n'}(t')] = \sum_q a_{nq} a_{n'q} k_q(t, t')$: one shared set of latent processes couples every pair of channels. This defines a MOGP, which at fixed parameters ϕ returns $\mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi)$ over any finite set of query cells by Gaussian conditioning. Integrating over uncertain kernel parameters, loadings, and observation noise gives the Bayesian posterior predictive mixture

$$p(\boldsymbol{y}_{\text{test}} \mid \boldsymbol{y}_{\text{obs}}) = \int \mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi)\, p(\phi \mid \boldsymbol{y}_{\text{obs}})\, \mathrm{d}\phi. \tag{5}$$

Note that Equation 5 is what a PFN pretrained on this family is trained to approximate, given that the kernel choice itself is also unknown. The fixed-ϕ and marginalized cases therefore yield two reference predictives: one without and one with hyperparameter uncertainty, the latter typically approximated with MCMC methods (Section 5). Note that this reference conditions on a supplied kernel structure and is therefore not the fully structure-marginalized predictive targeted by the PFN.

4 Method

We present JointTimePFN, a PFN that predicts the unknown entries of a partially observed multivariate temporal grid in one forward pass, while capable of returning a coherent joint distribution over them. Architecturally it is an encoder-only transformer whose tokens are convolutional patches of the series, attended over jointly across time and channels, and finally output heads through MLPs. We use the patch-based channel-mixing encoder and the LMC process family of TimePFN as a starting point (Taga et al., 2025), but then redefine the inference target, input encoding, local temporal processing, pretraining distribution, and probabilistic output.

4.1 Joint prediction on a partially observed temporal grid

Recall $\boldsymbol{Y} \in \mathbb{R}^{T\times N}$. We utilize a binary query mask $\boldsymbol{M} \in \{0,1\}^{T\times N}$, where $M_{tn} = 1$ means that Y_{tn} must be predicted. The mask partitions the grid into an observed and a queried index set,

$$\mathcal{O}(\boldsymbol{M}) = \{(t,n) : M_{tn} = 0\}, \qquad \mathcal{Q}(\boldsymbol{M}) = \{(t,n) : M_{tn} = 1\},$$

so that $\mathcal{O}(\boldsymbol{M}) \cup \mathcal{Q}(\boldsymbol{M})$ is the whole grid and the two are disjoint. Under any deterministic indexing of $\mathcal{Q}(\boldsymbol{M})$, let $\boldsymbol{y}_{\mathcal{Q}} \in \mathbb{R}^K$ contain its $K = |\mathcal{Q}(\boldsymbol{M})|$ values, and let $\boldsymbol{Y}_{\mathcal{O}}$ collect the values on $\mathcal{O}(\boldsymbol{M})$. The inference target is

$$p(\boldsymbol{y}_{\mathcal{Q}} \mid \boldsymbol{Y}_{\mathcal{O}}, \boldsymbol{M})\,. \tag{6}$$

Forecasting, known-future conditioning, ragged horizons, and imputation are all choices of $\boldsymbol{M}$ rather than separate model interfaces, though we do not focus on modeling missing-not-at-random mechanisms. Section 5 defines the four conditions we evaluate on and Figure 2 illustrates them.

A fixed-horizon point forecaster is recovered by masking every channel after a shared origin and reporting only a deterministic summary $\widehat{\boldsymbol{Y}}_{t_\star+1:T,:} = f_\theta(\boldsymbol{Y}_{1:t_\star,:})$, while a probabilistic model that returns one distribution per queried cell instead provides $\{q_{\theta,i}(y_i \mid \boldsymbol{Y}_{\mathcal{O}}, \boldsymbol{M})\}_{i=1}^K$. Yet these marginals do not determine a joint distribution; two simple approaches are available without changing the model. The independent marginal approximation simply multiplies them, $q_\theta^{\text{ind}}(\boldsymbol{y}_{\mathcal{Q}} \mid \boldsymbol{Y}_{\mathcal{O}}, \boldsymbol{M}) = \prod_i q_{\theta,i}(y_i \mid \boldsymbol{Y}_{\mathcal{O}}, \boldsymbol{M})$, assuming strict conditional independence across all queries. Autoregressive extraction instead reveals cells one at a time in some order π, giving $q_{\theta,\pi}^{\text{AR}} = \prod_j q_\theta(y_{\pi_j} \mid \boldsymbol{Y}_{\mathcal{O}}, \boldsymbol{y}_{\pi_{<j}}, \boldsymbol{M}^{(j)})$, where $\boldsymbol{M}^{(j)}$ indicates the cells still unrevealed at step j. For any fixed order π this defines a valid joint density that need not be multivariate Gaussian, even when the individual conditionals are Gaussian. Generating a completion requires K sequential model calls, while its density at a specified completion can be evaluated directly from the conditional factors. Moreover, whenever learned conditionals are mutually incompatible, the resulting joint depends on the order π (Bruinsma et al., 2023; Klötergens et al., 2026). Our objective instead focuses on a single order-independent distribution for Equation 6, produced in one forward pass.

4.2 PRETRAINING OVER PROCESSES AND OBSERVATION PATTERNS

The pretraining prior we use is adapted from TimePFN's (Taga et al., 2025), which we broaden and combine with a mixture of varying mask conditions. For each generated series, we sample a latent count $Q \sim p_Q$ and independent latent processes

$$u_q \sim \mathcal{GP}(0, k_q), \qquad k_q \sim p_{\text{comp}}(k). \tag{7}$$

Each k_q is a random sum or product of one to five base kernels drawn from the RBF, Matérn, periodic, changepoint, linear and rational-quadratic families, with continuous hyperparameters drawn mostly log-uniformly from the ranges in Table S5. This largely extends TimePFN's fixed bank of hardcoded kernel instances, which includes neither Matérn nor changepoint components (Taga et al., 2025). We add these to expose the model to rougher and non-stationary behavior.

Moreover, we replace TimePFN's nonnegative Dirichlet loadings with signed loadings, allowing both positive and negative prior cross-channel covariance at a shared time. For channel n,

$$\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_Q), \qquad \boldsymbol{a}_n = s_n \frac{\boldsymbol{z}_n}{\|\boldsymbol{z}_n\|_2}, \qquad s_n \sim p_s, \tag{8}$$

where s_n is the ℓ_2-norm of a draw from the Dirichlet loading prior it replaces (Appendix B.2). We include also private latent channels to expose the model to some weak or non-existing cross-channel dependence, and add observation noise after latent mixing.

To train a single model across diverse masking conditions, we use a mixture of nine masking patterns, listed with their sampling weights in Table S2. The pretraining distribution is intentionally broader than the evaluation settings and is not matched to the four test conditions, following evidence that task diversity drives in-context generalization (Shaheen et al., 2026).

4.3 MASK-AWARE TEMPORAL ENCODER

The original backbone (Taga et al., 2025) tokenizes each channel with multi-scale one-dimensional convolutions, pools them into overlapping patches in the manner of patch-based forecasters (Nie et al., 2023), and applies L layers of unmasked self-attention over the flattened $(N + K_{\text{cal}})P$ tokens, where P is the number of temporal patches and K_{cal} is the number of calendar features (Appendix A.3). We retain this structure and make four changes, so that the encoder accepts queries anywhere in the grid, separates observed from missing cells, and sees context from both temporal directions.

Full-grid encoding and decoding. The encoder takes the entire $T \times N$ grid and the head predicts every cell. TimePFN embeds only the $t_\star$ history positions and projects to the $T - t_\star$ horizon, so a future cell is an output and never an input. Attention is unmasked in both models, but TimePFN has no post-context tokens to attend to. Encoding all T positions supports known future covariates and queries at arbitrary interior positions. This requires evaluating the loss strictly on the queried cells.

Mask-augmented channel encoding. The query mask enters inside the feature: values and indicators are paired per channel as $\boldsymbol{X}_n = [\widetilde{\boldsymbol{Y}}_{:,n}, \boldsymbol{M}_{:,n}] \in \mathbb{R}^{T\times 2}$, so the channel count stays N instead of doubling and each indicator is locally aligned with its value.

Observed-cell normalization. Inputs are normalized from observed cells only. With per-channel location μ_n^{obs} and scale s_n^{obs} computed over $M_{tn} = 0$, we define $\widetilde{Y}_{tn} = (Y_{tn} - \mu_n^{\text{obs}})/s_n^{\text{obs}}$ for $M_{tn} = 0$ and $\widetilde{Y}_{tn} = 0$ otherwise, so unavailable values enter neither the normalization nor the encoder. If a channel has no observed cells, we set $\mu_n^{\text{obs}} = 0$ and $s_n^{\text{obs}} = 10^{-2}$, so all of its unavailable inputs enter the encoder as zero. Output heads are transformed back to data units using μ_n^{obs} and s_n^{obs}.

Bidirectional local context. The local convolutional stage should also look forward when observations and queries are interleaved. We replace TimePFN's causal local padding with symmetric padding, allowing convolutional and pooling features to incorporate available observations on both sides before global self-attention. We further add rotary position embeddings along the patch axis with base 10, which TimePFN does not use (Su et al., 2024).

The prediction heads process channels independently with shared weights, so one checkpoint handles variable channel counts N at inference. The temporal length $T = 192$ remains fixed, as in (Taga et al., 2025). Appendix A.3 gives the layer shapes, denormalization identities, and calendar features.

4.4 PREDICTIVE HEAD AND OBJECTIVE

The choice of output family follows directly from the LMC pretraining target. Assuming Gaussian likelihood and fixed hyperparameters ϕ, the LMC posterior predictive is exactly Gaussian, so a Gaussian head returns a closed-form joint density and thus the exact marginals of every subset. Empirically marginalizing the LMC posterior over MCMC samples of ϕ makes the target a mixture of Gaussians, which a Gaussian head generally cannot represent exactly, and thus we quantify that structural gap (Appendix B.3 and Section 5.1).

For each queried cell $i \in \mathcal{Q}(\boldsymbol{M})$, the output is $(\mu_i, \ell_i, \boldsymbol{w}_i)$, where $\boldsymbol{w}_i \in \mathbb{R}^R$ is a low-rank factor and ℓ_i an unconstrained log-variance whose exponential $d_i = e^{\ell_i} > 0$ is a cell-specific private variance. Stacking these K queried outputs yields the joint predictive parameterization

$$\begin{aligned} \boldsymbol{\mu} \in \mathbb{R}^K, \qquad \boldsymbol{D} = \operatorname{diag}(d_1, \ldots, d_K), \qquad \boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]^\top \in \mathbb{R}^{K \times R}, \\ q_\theta(\boldsymbol{y}_\mathcal{Q} \mid \boldsymbol{Y}_\mathcal{O}, \boldsymbol{M}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \boldsymbol{\Sigma} = \boldsymbol{D} + \boldsymbol{W}\boldsymbol{W}^\top. \end{aligned} \tag{9}$$

The diagonal $\boldsymbol{D}$ represents independent cell-specific noise, while the low-rank term $\boldsymbol{W}\boldsymbol{W}^\top$ captures cross-cell dependencies ($\Sigma_{ij} = \boldsymbol{w}_i^\top \boldsymbol{w}_j$ for $i \neq j$). This requires only $R + 2$ outputs per cell, keeping the head dimension independent of the varying query size K, and avoids the $O(K^2)$ parameterization of a full covariance matrix. Because $d_i > 0$, $\boldsymbol{\Sigma}$ is positive definite by construction and requires no projection. We use rank $R = 32$, see Appendix D.5 for ablations.

Since for any subset $S \subseteq \mathcal{Q}(\boldsymbol{M})$, the covariance submatrix is exactly $[\boldsymbol{\Sigma}]_{S,S} = \boldsymbol{D}_S + \boldsymbol{W}_S \boldsymbol{W}_S^\top$, every subset marginal is obtained exactly from the same joint Gaussian. Taking $S = \{i\}$ gives the implied univariate marginal:

$$q_\theta^{\text{marg}}(y_i \mid \boldsymbol{Y}_\mathcal{O}, \boldsymbol{M}) = \mathcal{N}\big(y_i \mid \mu_i,\ d_i + \|\boldsymbol{w}_i\|^2\big)\,, \tag{10}$$

so the full predictive variance is $d_i + \|\boldsymbol{w}_i\|^2$ and one checkpoint supports both marginal and joint evaluation. Let $\boldsymbol{r} = \boldsymbol{y}_\mathcal{Q} - \boldsymbol{\mu}$. We use query-normalized Gaussian NLLs for the joint predictive and its implied marginals:

$$\mathcal{L}_{\text{joint}} = \frac{1}{2K}\left[K \log(2\pi) + \log|\boldsymbol{\Sigma}| + \boldsymbol{r}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{r}\right], \tag{11}$$

$$\mathcal{L}_{\text{marg}} = \frac{1}{2K} \sum_{i=1}^{K} \left[\log(2\pi\Sigma_{ii}) + \frac{r_i^2}{\Sigma_{ii}}\right]. \tag{12}$$

$$\mathcal{L} = \mathcal{L}_{\text{joint}} + \lambda \mathcal{L}_{\text{marg}}. \tag{13}$$

Both $\mathcal{L}_{\text{joint}}$ and $\mathcal{L}_{\text{marg}}$ are strictly proper for their respective targets, so their nonnegatively weighted sum retains the true predictive as its population minimizer whenever it lies in the model family (Gneiting & Raftery, 2007). We use $\lambda = 0.5$, see Appendix D.5 for ablations.

Tractable low-rank likelihood. Let

$$\boldsymbol{B} = \boldsymbol{I}_R + \boldsymbol{W}^\top \boldsymbol{D}^{-1} \boldsymbol{W}. \tag{14}$$

The Woodbury identity and matrix determinant lemma give

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{D}^{-1} - \boldsymbol{D}^{-1} \boldsymbol{W} \boldsymbol{B}^{-1} \boldsymbol{W}^\top \boldsymbol{D}^{-1}, \tag{15}$$

$$\log|\boldsymbol{\Sigma}| = \sum_{i=1}^{K} \log d_i + \log|\boldsymbol{B}|. \tag{16}$$

Likelihood evaluation therefore costs $O(KR^2 + R^3)$ time and stores $O(KR)$ head outputs rather than a dense $K \times K$ covariance. Appendix A.4 also provides the Cholesky implementation, numerical safeguards, and treatment of large query sets.

5 EXPERIMENTS

The experiments address three questions. First, what is the fidelity of JOINTTIMEPFN to the posterior predictive induced by an LMC prior, both marginally and jointly? Second, does one mask-conditional checkpoint remain competitive across forecasting, covariate-informed forecasting, ragged horizons, and imputation? Third, which prior, architecture, and head components drive the observed behavior?

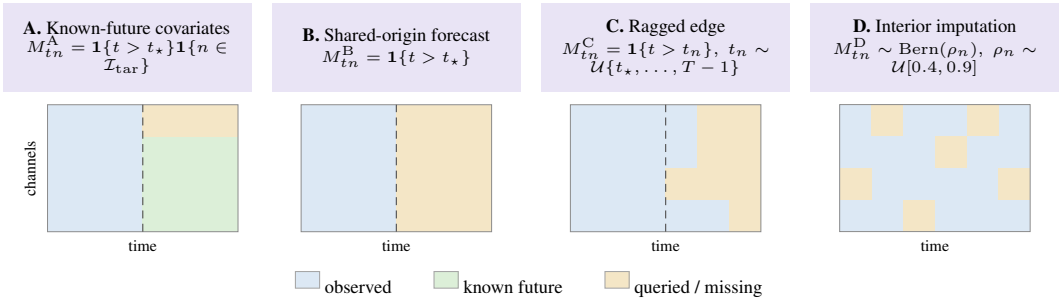


Figure 2: **Observation conditions.** Four query patterns under the unified mask formulation. A retains future covariates, B uses a shared forecast origin, C uses channel-specific origins, and D masks cells throughout the grid. Here $t_\star$ denotes the shared origin and $\mathcal{I}_{\text{tar}}$ the target-channel set.

Experimental setting. Each task is a $T \times N$ window with $T = 192$, forecast origin $t_\star = 96$, and a mask drawn from conditions A–D (detailed in Figure 2 and Table S4). Fidelity checks use independently generated synthetic LMC data, either in-prior using the pretraining hyperparameter ranges, or off-prior using shifted kernel and noise ranges (Appendix B.2 and Table S5). Each LMC realization contributes five non-overlapping windows, with uncertainty reported across realizations. The real-data experiments use the same four observation patterns, with MIMIC-IV (Johnson et al., 2023) as the main evaluation. See Appendix B for details on data generation and preprocessing. Every experiment evaluates the same checkpoint under four query masks, detailed in Figure 2.

Baselines. The exact MOGP oracle (orc) receives the true latent kernels, hyperparameters, loadings, and noise variance. A No-U-Turn sampler (NUTS, Hoffman & Gelman 2014) based MOGP receives the kernel structure but infers continuous hyperparameters and loadings, retaining posterior uncertainty via Equation 5. For real-world data, we compare against TimePFN (Taga et al., 2025), TabPFN-TS (Hoo et al., 2026), TabImpute (Feitelberg et al. 2026, not temporally native), Chronos-2 at two scales (Ansari et al., 2025), and TiRex-2 (Podest et al., 2026), wherever their interfaces apply, and against a cohort-trained TACTiS-2 (Ashok et al., 2024), which is not a zero-shot model. We do not force forecast-only methods onto interior imputation tasks. See Appendix C for baseline details.

5.1 Fidelities

Table 1 measures how well JointTimePFN approximates the ϕ-marginalized posterior predictive across the LMC kernel suites. The reported pool covers $N \in \{2, 4, 6\}$, where the NUTS reference remains affordable, with ten data seeds per band. Entries are medians over configurations, with each row evaluated on matched cells and windows, see Appendix B.3 for details.

Fidelity under prior shift. The first two divergence columns of Table 1(a) compare the MCMC-based MOGP and JointTimePFN to the same true-ϕ oracle. In-prior, the MOGP is closer to the oracle, with $D_{\text{KL}}(\text{orc}\|\text{N}) = 0.023$–$0.076$ versus $D_{\text{KL}}(\text{orc}\|\text{Ours}) = 0.192$–$0.347$ nats per cell. Off-prior, this reverses: the corresponding ranges are 0.260–0.622 for the MOGP and 0.069–0.078 for JointTimePFN, suggesting greater robustness of the amortized predictor to this prior shift. The marginal results in Table S12 show the same pattern: in-prior, the corresponding ranges are 0.030–0.156 for the MOGP and 0.282–0.716 for JointTimePFN, while off-prior they are 0.113–0.573 and 0.080–0.129, respectively. Absolute KL values are compared only within each regime, since the higher off-prior noise changes their scale.

Recovering joint dependence. The last two in-prior columns of Table 1(a) compare JointTimePFN with the *diag null*, which retains the reference marginals but removes all cross-cell correlation. Under condition B, the diag null divergence is 0.799 nats per cell, versus 0.286 for JointTimePFN, so the joint head removes 64% of the dependence-free gap. Under A, C and D, dependence is weaker (0.067, 0.312 and 0.034), and JointTimePFN is worse than the dependence-free baseline by 0.245, 0.192 and 0.299 nats per cell, reflecting some combination of spurious correlations and marginal error. Our joint head also achieves lower joint NLL than autoregressive extraction from the same marginals on conditions A–C, while requiring one forward pass instead of K sequential calls (Table 1(b); more results in Table S16 and Appendix D.4).

External prior-fitted baselines. On the same evaluation design as Table 1, JointTimePFN is closer to the exact oracle than moment-matched diagonal approximations of TabImpute in all eight settings

Table 1: **Fidelity to the ϕ-marginalized posterior predictive** (nats per cell, **lower is better**). orc is the exact true-hyperparameter predictive and N the moment-matched Gaussian of the NUTS mixture, used because D_{KL} against the mixture has no closed form. The first two divergence columns compare the MOGP reference and JOINTTIMEPFN against the same true-ϕ oracle, while $D_{\mathrm{KL}}(\mathrm{N}\|\mathrm{Ours})$ measures fidelity to the Bayesian reference. In panel (a) *diag null* is $D_{\mathrm{KL}}(\mathrm{N}\|\mathrm{diag\,N})$, the dependence available to be captured. In- and off-prior results are compared only within each regime, since the off-prior noise changes the scale of the divergence (Appendix B.3). Panel (b) is the joint NLL of four constructions on the in-prior pool, where AR is the chain-rule density of our own conditionals under one fill order (Appendix D.4).

(a) Joint	**In-prior**				**Off-prior**			
Cond	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{N})$	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{Ours})$	$D_{\mathrm{KL}}(\mathrm{N}\|\mathrm{Ours})$	diag null	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{N})$	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{Ours})$	$D_{\mathrm{KL}}(\mathrm{N}\|\mathrm{Ours})$	diag null
A	0.029	0.192	0.312	0.067	0.622	0.077	0.415	0.074
B	0.076	0.263	0.286	0.799	0.260	0.071	0.232	0.270
C	0.031	0.294	0.504	0.312	0.440	0.069	0.377	0.249
D	0.023	0.347	0.333	0.034	0.586	0.078	0.374	0.062

(b) Joint NLL, in-prior	Oracle (orc)	Ours	AR extraction	Independent marginals
A	-1.10	-0.77	-0.72	-0.14
B	-0.75	-0.47	-0.29	0.73
C	-0.91	-0.63	-0.57	0.10
D	-0.99	-0.60	-0.61	-0.59

and TabPFN-TS in all six supported settings (Table S17). In-prior under B, the KL values are 0.251 for JOINTTIMEPFN, 2.282 for TabImpute, and 1.768 for TabPFN-TS nats per queried cell.

5.2 UNIFIED PREDICTION ON REAL SERIES

Zero-shot transfer on MIMIC-IV. We then benchmark the same checkpoint zero-shot on MIMIC-IV against published foundation models, under the same four masks (Table 2). It outperforms the backbone we build on, TimePFN, on its only available condition, and TabImpute in condition D, and is competitive in all four. The joint head gains over its own diagonal in every condition, with broadly similar marginal performance. Both autoregressive extraction from our own marginals and a TACTiS-2 trained on this cohort nevertheless reach a lower joint NLL (Table S10). Figure 3 illustrates the same checkpoint's predictions under three observation masks on a single test window. Appendix B.4 gives the cohort, and Appendix C gives the baselines.

Cross-dataset generalization. We further test whether the same zero-shot checkpoint generalizes beyond MIMIC-IV (Appendix D.2). On ETTh1 and exchange rate, JOINTTIMEPFN improves on TimePFN's forecasting MSE under B and TabImpute's imputation MSE and CRPS_9 under D. On PhysioNet, it handles 36 channels, beyond the pretraining range of 2–20, and achieves the lowest reported MSE under B. Specialised forecasters remain stronger in some settings, particularly with known-future covariates (A). Joint likelihood gains over the model's diagonal are positive throughout ETTh1 and under A–C on PhysioNet, but negative under B–C on exchange rate and D on PhysioNet.

Table 2: **Real-data evaluation on MIMIC-IV** (per queried cell, 400 windows per condition). **(a)** Marginal CRPS_9 and MSE (**lower is better**). Dashes (–) denote unsupported configurations, and † indicates results based on fewer than 400 scored windows. Reported scores are averaged over the windows successfully scored by each arm. Bootstrap intervals, window counts, and coverages are in Table S9. **(b)** Joint likelihood gain over the model's own diagonal, $\Delta = \mathrm{NLL}_{\mathrm{diag}} - \mathrm{NLL}_{\mathrm{joint}}$.

(a) Marginal	$\mathrm{CRPS}_9 \downarrow$				MSE$\downarrow$			
Model	A	B	C	D	A	B	C	D
TimePFN	–	–	–	–	–	0.7835	–	–
TabPFN-TS	**0.3210**	0.4288†	0.3619†	–	0.5339	0.8098†	0.6109†	–
TabImpute	0.3526	0.4322	0.3875	0.3440	0.5650	0.7558	0.6399	0.5316
Chronos-2 (120M)	0.9885	3.6180	0.5427	–	**0.5247**	0.7179	0.5715	–
Chronos-2 (28M)	0.6232	2.1598	0.4358	–	0.5340	0.7351	0.5815	–
TiRex-2	0.3382	**0.4121**	**0.3522**	–	0.5420	0.7241	**0.5647**	–
JOINTTIMEPFN (marginal)	0.3798	0.4264	0.3788	0.2733	0.6168	0.7213	0.6083	0.3487
JOINTTIMEPFN (joint)	0.3647	0.4292	0.3749	**0.2698**	0.5778	**0.7123**	0.5920	**0.3420**

(b) Joint

Cond	$\mathrm{NLL}_{\mathrm{diag}}$	$\mathrm{NLL}_{\mathrm{joint}}$	$\Delta \uparrow$
A	2.6603	1.6089	1.0514
B	2.3763	2.0789	0.2974
C	2.5589	1.9623	0.5966
D	1.8940	1.5506	0.3434

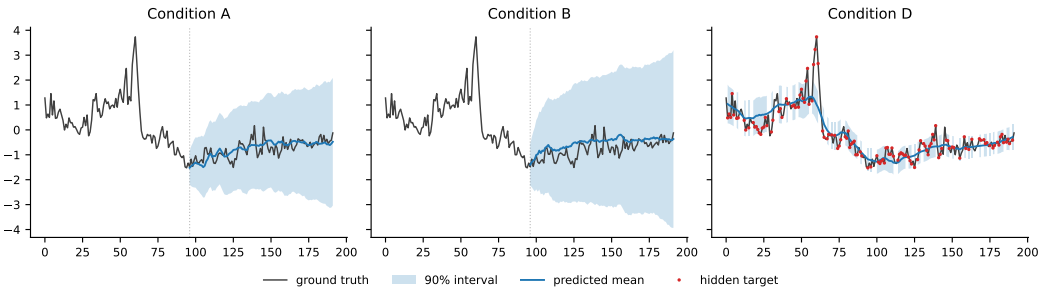


Figure 3: **Temporal predictions under three observation masks.** Heart-rate predictions from the same JOINTTIMEPFN checkpoint on one MIMIC-IV test window. Condition A uses future observations from other channels, B hides all future channels, and D masks scattered target cells. Black shows ground truth, blue the predictive mean with pointwise 90% marginal intervals, and red dots the hidden targets. Future covariates in A yield more confident and better aligned predictions.

5.3 EFFECT OF PRETRAINING PRIOR AND ARCHITECTURE CHOICES

To assess which design choices are most important for JOINTTIMEPFN, we ablate the broadened pretraining prior, signed loadings, rotary embeddings, and joint predictive training. Table 3 summarizes the main effects on MIMIC-IV, while Appendix D.5 reports the full rank and loss-weight sweeps. The largest degradation comes from replacing the broadened prior with TimePFN's fixed kernel bank: joint NLL rises from 1.945 to 8.273, alongside higher MSE and CRPS_9. Removing RoPE or using unsigned loadings also worsens these metrics, although the loading ablation has a smaller effect on joint NLL. The joint checkpoint additionally achieves lower MSE and CRPS_9 than marginal-only training, showing that modeling dependence need not sacrifice marginal accuracy in this evaluation.

Table 3: **Component ablations on MIMIC-IV**, averaged over conditions A–D (**lower is better**).

Variant	MSE	CRPS_9	Joint NLL
Full JOINTTIMEPFN	0.5707	0.3653	1.945
TimePFN's fixed kernel bank	0.6506	0.4048	8.273
Unsigned loadings	0.5915	0.3714	1.998
No RoPE	0.5912	0.3703	2.105
Marginal-only (log score)	0.5867	0.3698	–

6 CONCLUSION

We introduced JOINTTIMEPFN, a mask-conditional PFN that unifies forecasting and imputation and returns a low-rank Gaussian joint with analytic marginals in one forward pass. More broadly, this casts prior fitting as amortization of a joint conditional predictive law over partially observed time series, with fixed-horizon forecasting as one observation pattern among many. The MOGP experiments further indicate that this amortization need not simply reproduce the pretraining prior: although the model gives up some fidelity in-prior, it can be more robust when the data-generating hyperparameters shift. Explicit joint prediction therefore provides a practical middle ground between marginal-only foundation models and more expressive alternatives. This being said, autoregressive extraction and cohort-trained TACTiS-2 can achieve better joint likelihood in some settings, but require respectively K sequential passes or dataset-specific training, whereas JOINTTIMEPFN produces the full predictive in a single zero-shot pass.

Limitations. At a fixed observed context, the Gaussian head cannot represent skewness, multimodality, or non-Gaussian dependence, even if its mean and covariance are nonlinear functions of the context. The fixed rank further restricts covariance expressivity, with temporal and cross-channel dependence represented through the same low-rank factors rather than separate structures. More expressive heads, such as Gaussian mixtures, could relax these restrictions, but at substantially higher cost for high-dimensional joint predictions (Whittle et al., 2026). The joint head can also underperform a diagonal reference when queried cells are nearly conditionally independent, suggesting again future work on richer output families and alternative pretraining priors. From an evaluation perspective, posterior fidelity is assessed only within synthetic LMC settings, so it primarily measures amortization of the training prior rather than its adequacy on real-world data. Lastly, our experiments on clinical datasets do not explicitly model missing-not-at-random observation mechanisms.

AI USE STATEMENT

Generative AI tools were used to assist with prose editing, to draft passages, and with parts of the implementation and experimental code. All methodological choices, experiments, reported results, and claims were reviewed and validated. Generative AI was not used to generate experimental data or to define the core method.

ETHICS STATEMENT

The healthcare experiments use the deidentified MIMIC-IV database under its credentialing and data-use requirements (Johnson et al., 2023). All inference is run locally on controlled compute.

REPRODUCIBILITY STATEMENT

Appendix B specifies the synthetic generators, masks, metrics, data splits, and reference predictors. Appendix A gives the pretraining algorithm, model shapes, covariance computation, and numerical safeguards. Appendices C and E document baseline configurations, training hyperparameters, evaluation settings, and compute. An anonymized code bundle, including the model implementations, configuration files, and evaluation scripts, will be made publicly available upon acceptance.

REFERENCES

Mauricio A. Álvarez and Neil D. Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 2011.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 2012.

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

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. TACTiS-2: Better, faster, simpler attentional copulas for multivariate time series. *International Conference on Learning Representations*, 2024.

Ethan Baron, Boris Oreshkin, Ruijun Ma, Hanyu Zhang, Kari Torkkola, Michael W. Mahoney, Andrew Gordon Wilson, and Tatiana Konstantinova. Efficiently generating correlated sample paths from multi-step time series foundation models. *arXiv preprint arXiv:2510.02224*, 2025.

Wessel P. Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner. Autoregressive conditional neural processes. *International Conference on Learning Representations*, 2023.

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

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V. Naidu, and Colin White. ForecastPFN: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 2023.

Jacob Feitelberg, Dwaipayan Saha, Kyuseong Choi, Zaid Ahmad, Anish Agarwal, and Raaz Dwivedi. One pipeline, many transformers: Pattern-specific imputation specialists for tabular missing data. *arXiv preprint arXiv:2510.02625*, 2026.

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

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 2014.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending TabPFN-v2 to time series forecasting. *arXiv preprint arXiv:2501.02945*, 2026.

Daniel P. Jeong and Seyoung Kim. Factorial sde for multi-output gaussian process regression. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, 2023.

Xiaoyu Jiang, Sokratia Georgaka, Magnus Rattray, and Mauricio A. Álvarez. Scalable multi-output gaussian processes with stochastic variational inference. *Transactions on Machine Learning Research*, 2025.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 2023.

Christian Klötergens, Vijaya Krishna Yalavarthi, Lars Schmidt-Thieme, and Tom Hanika. Do tabular foundation models agree with themselves? *arXiv preprint arXiv:2608.06004*, 2026.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. *International Conference on Learning Representations*, 2022.

Samuel Müller, Arik Reuter, Noah Hollmann, David Rügamer, and Frank Hutter. Position: The future of bayesian prediction is prior-fitted. *International Conference on Machine Learning*, 2025.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *International Conference on Learning Representations*, 2023.

Patrick Podest, Marco Pichler, Elias Bürger, Levente Zólyomi, Bernhard Voggenberger, Wilhelm Berghammer, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex-2: Generalizing tirex to multivariate data and streaming. *arXiv preprint arXiv:2607.01204*, 2026.

Andres Potapczynski, Ravi Kiran Selvam, Tatiana Konstantinova, Malcolm Wolff, Kin G. Olivares, Ruijun Ma, Michael W. Mahoney, Andrew G. Wilson, Boris N. Oreshkin, and Dmitry Efimov. Time-aware prior fitted networks for zero-shot forecasting with exogenous variables. *Transactions on Machine Learning Research*, 2026.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.* MIT Press, 2006.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full bayesian inference in context? *International Conference on Machine Learning*, 2025.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *International Conference on Learning Representations*, 2022.

Nour Shaheen, Junwei Ma, Alex Labach, Frank Hutter, Valentin Thomas, and Anthony L. Caterini. Understanding the surprising generalization properties of tabular foundation models. *arXiv preprint arXiv:2608.17957*, 2026.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. TimePFN: Effective multivariate time series forecasting with synthetic data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 2021.

George Whittle, Juliusz Ziomek, Jacob Henry Rawling, and Michael A Osborne. Distribution transformers: Fast approximate bayesian inference with on-the-fly prior adaptation. In *Forty-third International Conference on Machine Learning*, 2026.

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

The appendix is organized as follows:

- **Appendix A** gives the pretraining distribution and mask mixture, one pretraining step, the architecture and its parameterization, and the Woodbury likelihood with its sampling and numerical safeguards.
- **Appendix B** specifies the mask families, the synthetic fidelity suites, the exact and NUTS MOGP references, the divergence and calibration metrics, and the MIMIC-IV cohort.
- **Appendix C** describes every baseline, the information and targets each receives, and the conditions its interface can express.
- **Appendix D** reports the full fidelity tables with dispersion, the autoregressive coherence probes, and the ablations.
- **Appendix E** reports model, training, and evaluation hyperparameters.

A Method details

A.1 Pretraining distribution

Table S1: Pretraining-generator components.

Component	Distribution or operation	Purpose
Generated length	$L_{\text{gen}} = 1024$	Several overlapping windows per independent series.
Generated channels	$C = 20$; subsample $N \in [2, 20]$	Avoid pretraining at one fixed channel count.
Latent count	hierarchical Weibull: shape $\mathcal{U}[1.5, 2.5]$, scale log-uniform on $[2, 10]$, rounded and clipped to $[2, C]$	Variable effective rank and coupling strength.
Kernel composition	1–5 bases, random sum/product	Diverse stationary and nonstationary temporal structure.
Kernel parameters	mostly log-uniform; see Section 4.2 and Table S5	Broaden support beyond TimePFN's fixed kernel bank.
Loadings	Equation 8	Signed isotropic channel dependence.
Private channels	fraction $\pi_{\text{priv}} = 0.2$, selected uniformly	Teach the model to ignore uninformative channels.
Observation noise	sampled after LMC mixing	Match standard MOGP observation semantics.
Mask	mixture over nine patterns (Table S2)	One checkpoint for forecasting and imputation.

Table S2: Pretraining mask mixture. The evaluation conditions of Table S4 are a narrower family than this: the model is trained on patterns it is never tested on.

Pattern	Weight	Description
Shared-origin forecast	0.10	Classic forecast from a fixed origin.
Partial-horizon forecast	0.10	Shared origin, randomized horizon length.
Covariate forecast	0.15	Future partially known on a subset of channels.
Single-channel forecast	0.05	One target future hidden, matching condition A.
Uniform interior missingness	0.10	Cells hidden independently across the grid.
Per-channel block outage	0.15	Contiguous outage on individual channels.
Global blackout	0.10	All channels unobserved over a shared interval.
Heterogeneous sparse	0.15	Strongly uneven per-channel rates, resembling irregular charting.
Mixture of the above	0.10	Random combinations within one window.

A.2 One pretraining step

Algorithm 1 One JointTimePFN pretraining step

Require: Batch size B, generated length L_{gen}, maximum channels C, mask distribution $p(\boldsymbol{M})$

1: **for** $b = 1, \ldots, B$ **do**
2: Sample latent rank Q, kernel compositions $\{k_q\}_{q=1}^{Q}$, and their continuous hyperparameters.
3: Draw latent paths $u_q \sim \mathcal{GP}(0, k_q)$ on L_{gen} time points.
4: Draw signed loadings, optional private channels, and observation noise; construct $\boldsymbol{Y}^{(b)} \in \mathbb{R}^{L_{\text{gen}} \times C}$.
5: Sample a window, a channel subset, and a query mask $\boldsymbol{M}^{(b)} \sim p(\boldsymbol{M})$.
6: Compute observed-cell channel statistics and form mask-augmented per-channel inputs.
7: Encode all tasks with the shared patch-based, channel-mixed transformer.
8: Assemble marginal or joint predictions on each $\mathcal{Q}(\boldsymbol{M}^{(b)})$.
9: Average the task-normalized losses in Equation 13; update θ.

A.3 Architecture details and parameterization

For a batch element with N data channels and K_{cal} pseudo-channels for calendar features (following TimePFN Taga et al. 2025), the encoder and patch projection produce tokens $\boldsymbol{H}^{(0)} \in \mathbb{R}^{(N+K_{\text{cal}})P \times d}$, where P is the number of temporal patches.

Calendar features and alignment. To ensure that the exact oracle and MCMC references are not strictly disadvantaged against the neural models by missing side information, calendar features ($K_{\text{cal}} = 0$) are disabled across all experiments. For the real-data evaluation on MIMIC-IV, we supply a regular hourly grid with an arbitrary origin. While real-world phase alignment (e.g., matching a timestamp to exactly 2:00 AM) is omitted, strict periodicity is fully preserved and remains accessible to all models via positional encodings (Ours), seasonal features (TabPFN-TS), or inferred frequencies (Chronos-2).

Mask-aware tokenization. The tokenizer applies three parallel 1D convolutions (widths $5, 9, 15$) mapping the two input features (value and query mask) to three output channels, followed by magnitude-preserving max pooling (widths $3, 5, 7$), which keeps the element of largest absolute value in each window together with its sign. The nine convolution outputs are then concatenated with the two original features, giving 11 feature channels per timestep, so a patch of 32 timesteps flattens to the inner dimension $32 \cdot 11$ of Table S19 before projection; that 11 counts features and coincides only numerically with the $P = 11$ patches below. To allow bidirectional context, we replace the causal left-only padding of the original TimePFN with symmetric padding: $(2, 2)$, $(4, 4)$, and $(7, 7)$ for convolutions, and $(1, 1)$, $(2, 2)$, and $(3, 3)$ for pooling. Patches of length 32 with stride 16 yield $P = 11$ patches for a $T = 192$ window.

Channel-mixed encoder. Each patch is projected to dimension $d = 256$. The encoder consists of $L = 8$ transformer blocks at this width (8 attention heads, feed-forward width 512); the $d_{\text{model}} = 1024$ below is the hidden width of the prediction heads, not of the encoder. Rotary Position Embeddings (RoPE) are applied to the patch axis at a modified base of 10.

Prediction heads. After the transformer blocks, tokens are reshaped per channel into a flattened vector of width $Pd = 2816$. Each head is a two-layer perceptron ($\mathbb{R}^{Pd} \to \mathbb{R}^{d_{\text{model}}} \xrightarrow{\text{GELU}} \mathbb{R}^{T}$ or $\mathbb{R}^{TR}$) that emits predictions for all T positions simultaneously. Three parallel heads output the parameters for the Gaussian predictive:

$$\boldsymbol{h}_{tn} \mapsto (\mu_{tn},\ \ell_{tn},\ \boldsymbol{w}_{tn}), \qquad d_{tn} = \max\{\exp(\text{clip}(\ell_{tn}, -\ell_{\max}, \ell_{\max})),\ d_{\min}\} \tag{S1}$$

The factor projection $\boldsymbol{W}$ is initialized via $\mathcal{N}(0, 10^{-4})$ with zero bias. Exact zero initialization is avoided because $\partial\mathcal{L}/\partial\boldsymbol{W}$ vanishes identically at $\boldsymbol{W} = \boldsymbol{0}$ for $\boldsymbol{\Sigma} = \boldsymbol{D} + \boldsymbol{W}\boldsymbol{W}^{\top}$, which would stall learning.

Denormalization. Head outputs are mapped back to data units by inverting the input normalization per channel: $\mu \mapsto s_n\mu + \mu_n$, $\ell \mapsto \ell + 2\log s_n$, and $\boldsymbol{w} \mapsto s_n\boldsymbol{w}$. Thus, the covariance between queried cells from channels n and m is scaled by $s_n s_m$ (and the variance within channel n by s_n^2), preserving the unitless correlation structure.

Pretraining configuration. Training runs for a single pass over the synthetic corpus at batch size 16 using a learning rate of 3×10^{-4} with a one-cycle schedule ($\mathrm{pct_start} = 0.3$, decay 0.8). The marginal loss weight is set to $\lambda = 0.5$. Windows are cut at stride 8 from length-1024 generated series. During pretraining, the channel count N is drawn uniformly per batch, since a $[B, T, N]$ batch cannot stack ragged N.

A.4 Woodbury likelihood, sampling, and safeguards

In practice we never form the $\boldsymbol{\Sigma}^{-1}$ of Equation 15. Let $\boldsymbol{u} = \boldsymbol{D}^{-1}\boldsymbol{r}$. We compute a Cholesky factor $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{L}^\top$ and obtain

$$\boldsymbol{r}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{r} = \boldsymbol{r}^\top \boldsymbol{u} - (\boldsymbol{W}^\top \boldsymbol{u})^\top \boldsymbol{B}^{-1} (\boldsymbol{W}^\top \boldsymbol{u}), \tag{S2}$$

$$\log |\boldsymbol{\Sigma}| = \sum_i \log d_i + 2 \sum_{r=1}^{R} \log L_{rr}. \tag{S3}$$

Numerical safeguards and batching. Table S3 lists the constants.

Table S3: Implementation details for stable joint likelihood evaluation and variable-K minibatch processing.

Parameter / Operation	Value	Purpose
Log-variance clamp ($\ell_{\max}$)	$[-10, 10]$	Bounds the predicted variance. Only the upper bound binds: $\exp(-10) < d_{\min}$, so the lower clip in Equation S1 is superseded by the variance floor.
Variance floor ($d_{\min}$)	10^{-4}	Additional strict safeguard against singular covariance matrices.
Ridge on $\boldsymbol{d}$	10^{-6}	Added to the private variances, so the covariance scored is exactly $\boldsymbol{\Sigma} + 10^{-6}\boldsymbol{I}$.
Normalization scale	$s_n^{\mathrm{obs}} = \sqrt{v_n^{\mathrm{obs}} + 10^{-4}}$	Handles channels with zero or undefined observed variance.
Variable K padding	$d_i = 1$, $\boldsymbol{w}_i = \boldsymbol{0}$, $y_i = \mu_i$	Nullifies padded cells' contributions to the log-determinant, quadratic form, and $\boldsymbol{B}$.
Exact K tracking	Evaluated pre-padding	Ensures the $K \log(2\pi)$ normalizing constant remains strictly exact for every sample.

Large query sets. A full condition-B horizon can contain $K = (T - t_\star)N_{\mathrm{target}}$ cells. Task normalization keeps its loss scale comparable with sparse masks, but memory still grows linearly in KR. In practice no cap was needed: at $N = 20$ a full grid is $TN = 3840$ cells and the factors fit, so every queried cell enters the joint loss. Evaluation assembles the covariance factors directly because it stores $\boldsymbol{W}$ instead of the large $\boldsymbol{\Sigma}$.

B EXPERIMENTAL SETUP

B.1 MASK FAMILIES IN EXPERIMENTS

Table S4 defines the four tasks. The same sampled series can be paired with several masks, but train, validation, and test series remain disjoint.

Table S4: Observation and query masks. $t_\star$ is a shared forecast origin and t_n a channel-specific origin. The scored set contains only cells newly hidden by the evaluation mask.

ID	Name	Query rule	Main capability isolated
A	Known-future covariates	Mask the future of one target channel for $t > t_\star$; retain the future of every covariate.	Use cross-channel information beyond the target origin.
B	Shared-origin forecast	Mask all channels for $t > t_\star$.	Standard multivariate forecasting
C	Ragged edge	Draw an origin $t_n \sim \mathcal{U}\{t_\star, \dots, T-1\}$ independently for every channel and mask Y_{tn} for $t > t_n$, so no channel's future stays complete.	Heterogeneous horizons with a staggered, partially observed future on every channel.
D	Interior imputation	Mask cells independently within each channel at rate $\rho_n \sim \mathcal{U}[0.4, 0.9]$, drawn per channel.	Imputation from context on both sides of each gap.

Condition D is missing-at-random within each channel, with the per-channel rate varying the amount of missingness. None of the four conditions is missing-not-at-random.

B.2 SYNTHETIC FIDELITY SUITES AND MOGP REFERENCES

The fidelity suite uses an LMC with no private channels, so that the oracle and the NUTS reference have the same latent structure as the generator. Each realization has length 1024, giving five disjoint evaluation windows of $T = 192$. The suite varies:

- channel count $N \in \{2, 4, 6\}$ restricted by the MCMC reference's computational cost.
- latent rank fixed at $Q = 2$ throughout the grid unless explicitly stated differently.
- composed kernels drawn exactly as in Section 4.2, with the count per latent uniform on $\{1, \dots, 5\}$ and $\nu \in \{1/2, 3/2, 5/2\}$ for Matérn, so roughly a fifth of latents are a single kernel and the rest are compositions.
- signed isotropic loadings, direction drawn uniformly on the unit sphere and magnitude from a Dirichlet draw whose concentration is itself sampled, $\alpha \sim \mathcal{U}[0.1, 10]$, so cross-channel dependence takes either sign.
- no private latents ($\pi_{\text{priv}} = 0$) and observation noise added after LMC mixing, so the oracle and NUTS model exactly the generator's latent structure.
- the four mask conditions in Table S4.

Exact oracle. The oracle receives Q, every kernel tree and hyperparameter, the loading matrix, and observation noise. Standard Gaussian conditioning gives $(\boldsymbol{\mu}_\star, \boldsymbol{\Sigma}_\star)$ at the query cells. It is therefore a theoretical optimum, not a fair comparison.

NUTS MOGP. Table S6 lists what each of the three is told. Its priors over the continuous parameters are the pretraining ones: log-uniform length scale and noise on the in-prior bands of Table S5, rescaled to per-window units, and an isotropic Gaussian on the loadings matched to the second moment of the generator's Dirichlet magnitudes. Like JOINTTIMEPFN the sampler therefore cannot widen its prior off-prior, which is what makes the two comparable there.

Table S5: **Hyperparameter ranges of the two prior bands**. Continuous positive parameters are drawn log-uniformly within the listed ranges unless stated otherwise. *In-prior* uses the pretraining hyperparameter ranges, while *off-prior* uses the shifted ranges listed below. Lengths are relative to a realization of length 1024, so the off-prior floor $\ell = 0.005$ is about five steps and the data remains a resolved Gaussian process.

Hyperparameter	In-prior	Off-prior (shifted band)
Squared-exponential length scale ℓ	$(0.05, 3.0)$	$(0.005, 0.04)$
Periodic length scale	$(0.005, 1.0)$	$(0.0005, 0.004)$
Periodic period	$(0.05, 1.0)$	$(0.005, 0.04)$
Canonical calendar periods	used	not used
Observation noise σ	$(0.01, 0.5)$	$(0.6, 1.5)$
Changepoint steepness	$(5.0, 50.0)$	$(60.0, 500.0)$
Changepoint base variance	$(0.01, 0.5)$	$(0.6, 2.0)$

We use NUTS (Hoffman & Gelman, 2014) with 2 chains, 600 warm-up iterations and 400 retained draws per chain, over 5 evaluation windows per cell. The marginal evaluation thins by 2, giving $J = 400$ draws; the joint evaluation retains every draw, giving $J = 800$. Each draw produces one Gaussian predictive, so the empirical predictive is a uniform Gaussian mixture with J components.

Table S6: What each of the three models gets.

Model	Kernel tree	Continuous hyperparameters	Loading matrix	Inference
Exact MOGP oracle	true	true	true	closed-form conditioning
NUTS MOGP	composition tree and kernel families supplied	inferred	inferred	posterior samples
JOINTTIMEPFN	unknown	amortized over pretraining prior	amortized over pretraining prior	one forward pass

B.3 FIDELITY AND PROBABILISTIC METRICS

Cell sets. The joint reference needs a full $K \times K$ covariance per window, so we score joint fidelity on query subsets of at most $K = 60$ cells, drawn uniformly without replacement from that window's scored cells. The exact oracle, the moment-matched NUTS reference, JOINTTIMEPFN, and the diagonal null are all scored on exactly the same drawn cells, making results comparable across columns. Marginal fidelity is measured on a separately selected, larger pool, so the two panels are not compared numerically with one another: for a correlated $\boldsymbol{\Sigma}$ the joint divergence per queried cell is not the mean of the per-cell marginal divergences.

Gaussian KL. For $p = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $q = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ in K dimensions,

$$D_{\mathrm{KL}}(p\|q) = \frac{1}{2}\left[\mathrm{tr}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) - K + \log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|}\right]. \tag{S4}$$

KL to the NUTS predictive, and the non-Gaussianity floor. Let $q^\star(\boldsymbol{y}) = J^{-1}\sum_{j=1}^{J}\mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ be the NUTS predictive mixture and let N be the Gaussian with the same mean and covariance, the **moment-matched** Gaussian. This N is the symbol every table prints; $q^\star$ always denotes the mixture itself. Adding and subtracting the cross-entropy $H(q^\star, q_\theta) = -\mathbb{E}_{q^\star}\log q_\theta$ gives the identity

$$D_{\mathrm{KL}}(q^\star\|q_\theta) = \underbrace{[H(\mathrm{N}) - H(q^\star)]}_{\text{non-Gaussianity floor}} + \underbrace{D_{\mathrm{KL}}(\mathrm{N}\|q_\theta)}_{\text{closed form}}, \tag{S5}$$

which holds because a Gaussian q_θ makes the cross-entropy depend on the other distribution only through its mean and covariance, so $H(q^\star, q_\theta) = H(\mathrm{N}, q_\theta)$.

Only $H(q^\star)$ then needs Monte Carlo, and the second term is closed form from Equation 16; estimating $D_{\mathrm{KL}}(q^\star \| q_\theta)$ directly would put sampling noise on the whole quantity. We draw $\boldsymbol{y}^{(s)} \sim q^\star$, evaluate the mixture density with log-sum-exp, and report Monte Carlo standard errors on the floor.

The floor is a diagnostic that measures how far the reference mixture is from Gaussian, which bounds how much of any measured divergence could be attributed to mixture structure against our Gaussian output family. Across the reported suites its median is 8.3×10^{-3} nats per cell, with an interquartile range of $[3.1 \times 10^{-3},\ 2.6 \times 10^{-2}]$, so it is typically an order of magnitude below the divergences reported beside it. Its maximum of 5.9×10^{-1} is not negligible, however, so a single configuration's total divergence can be dominated by mixture structure rather than by our output family. Five of 476 configurations come out slightly negative, down to -1.1×10^{-3}, which is the expected behaviour of a difference of two entropy estimates when the true value is near the Monte Carlo error. This small difference is illustrated in Table S13.

Aggregation. Every reported number is a median over configurations. The divergence columns are right-skewed, so a mean would report a few hard configurations instead of typical fidelity.

Divergence directions. Three divergences are defined on the matched cells, with q_θ the JOINT-TIMEPFN predictive of Equation 9:

- $D_{\mathrm{KL}}(\text{oracle} \| \mathrm{N})$ and $D_{\mathrm{KL}}(\text{oracle} \| q_\theta)$, the two oracle-referenced columns. As mentioned, the NUTS entry uses N rather than the mixture $q^\star$, since $D_{\mathrm{KL}}(\text{oracle} \| q^\star)$ has no closed form.
- $D_{\mathrm{KL}}(\mathrm{N} \| q_\theta)$, how closely we reproduce the Bayesian reference. Since N comes from a sampler that is given the kernel tree and ν while q_θ marginalizes over both, this mixes amortization error with the cost of that marginalization and should not be read as amortization error alone.

Which term each table reports. The columns labelled $D_{\mathrm{KL}}(\mathrm{N} \| q_\theta)$, in the main panel and in the per-band marginal tables, report the closed-form term of Equation S5. The per-band joint tables report the total $D_{\mathrm{KL}}(q^\star \| q_\theta)$ instead, so those entries exceed the panel column by the floor term: 0.312 against 0.325 on A, 0.286 against 0.301 on B, 0.504 against 0.518 on C and 0.333 against 0.350 on D. The gap is not the median floor quoted above, since a median of sums is not the sum of medians.

Total correlation and the diagonal null. For a reference covariance $\boldsymbol{\Sigma}_{\mathrm{N}}$ over K queried cells,

$$\mathrm{TC}(\mathrm{N}) = \frac{1}{2}\left(\sum_i \log \Sigma_{\mathrm{N},ii} - \log |\boldsymbol{\Sigma}_{\mathrm{N}}|\right) = D_{\mathrm{KL}}(\mathrm{N} \,\|\, \mathrm{diag}\,\mathrm{N})\,, \tag{S6}$$

zero if and only if $\boldsymbol{\Sigma}_{\mathrm{N}}$ is diagonal. **This is the diag null of the joint tables**: the divergence of a predictive with the reference's exact marginals versus no correlation at all. The reference diagonal null should be distinguished from the model's own diagonal predictive,

$$q_\theta^{\mathrm{diag}} = \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\Sigma_{11}, \ldots, \Sigma_{KK}))\,.$$

Indeed,

$$D_{\mathrm{KL}}(\mathrm{N} \| q_\theta^{\mathrm{diag}}) = \mathrm{TC}(\mathrm{N}) + \sum_{i=1}^{K} D_{\mathrm{KL}}(\mathrm{N}_i \| q_{\theta,i}).$$

A comparison with the reference diagonal therefore reflects marginal errors as well as dependence modeling. Comparing q_θ with q_θ^{diag} instead holds the model's marginals fixed. Moreover, $\mathrm{TC}(\mathrm{N})$ measures dependence in the Gaussian reference, not all possible dependence in the predictive mixture.

Variance calibration (z^2). We evaluate variance scaling via the normalized whitened residual z^2/K, where $z^2 = (\boldsymbol{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}_\theta^{-1} (\boldsymbol{y} - \boldsymbol{\mu})$. More generally,

$$\mathbb{E}[z^2/K] = \frac{1}{K}\left[\mathrm{tr}(\boldsymbol{\Sigma}_\theta^{-1}\boldsymbol{\Sigma}_{\mathrm{true}}) + (\boldsymbol{\mu}_{\mathrm{true}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}_\theta^{-1}(\boldsymbol{\mu}_{\mathrm{true}} - \boldsymbol{\mu})\right].$$

Correct mean and covariance therefore yield $\mathbb{E}[z^2/K] = 1$. Values above one can reflect either overconfidence or mean error, while a value near one does not by itself imply a correct covariance structure.

B.4 MIMIC-IV COHORT AND PREPROCESSING

We extract a cohort from the MIMIC-IV ICU database (Johnson et al., 2023) under PhysioNet credentialing. To ensure a reproducible and clinically realistic evaluation, we apply the following preprocessing and masking pipeline:

Channel selection and filtering. We select $N = 10$ variables by how densely they exist in the dataset, deliberately mixing continuously monitored vitals with intermittent measurements to prevent an artificially complete grid. Table S7 details the variables and the physiological bounds applied to filter outliers prior to standardization.

Table S7: **MIMIC-IV variables and physiological bounds.** Values outside these ranges are dropped before standardizing the data.

Measurement Type	Variable	Source	Valid Range
Continuous	Heart Rate	Bedside monitor	20–300
	Respiratory Rate	Bedside monitor	2–80
	SpO_2	Bedside monitor	50–100
	SBP / DBP / MBP	Bedside monitor	40–300 / 20–200 / 20–250
Intermittent	Temperature	Nurse charting	30–45
	Glucose	Lab order	20–1000
	FiO_2 / PEEP	Ventilator settings	21–100 / 0–30

Binning and windowing. The raw hourly data is aggregated into $B = 3$-hour bins via a NaN-skipping mean, reflecting clinical quantities rather than isolated readings. We extract the first $T = 192$ consecutive bins (24 days) per ICU stay, yielding 1,258 valid independent windows out of 9,357 stays. Data splits are strictly partitioned by `subject_id` rather than `stay_id` to prevent physiological leakage across splits (seed 0). Table 2 scores the first 400 of these windows to keep computational hours reasonable.

Natural missingness and imposed masks. Natural missingness is preserved and tracked via a binary mask. The imposed evaluation masks (Conditions A–D, origin $t_\star = 96$) are applied on top of this natural missingness. Crucially, the scored set contains only newly masked cells that were actually charted in the ground truth. Naturally missing cells are never imputed or scored. All models are evaluated on identical masked grids.

Data use. Data is utilized strictly under the PhysioNet data-use agreement. No patient-derived values are transmitted to third-party services, and only aggregate results are released.

C BASELINE DETAILS

Unified evaluation. Every model is evaluated on identical cells using one shared, pre-generated set of masks per condition. Dashes (−) mark conditions a baseline does not report. For the forecast-only interfaces of Chronos-2 and TiRex-2, interior imputation cannot be expressed at all and for TabPFN-TS it is outside the task its public interface exposes (see below). All benchmark metrics are computed strictly from the models' native outputs without post hoc adjustments. Point errors are computed from native point forecasts. $CRPS_9$ is evaluated via trapezoidal integration over the nine deciles $(0.1, \ldots, 0.9)$, as this is the widest quantile grid natively supported by all probabilistic baselines.

TimePFN (Taga et al., 2025). We evaluate the official zero-shot checkpoint using its original 96-step context/horizon setting. Because it is a point forecaster and lacks an observation mask input, it cannot distinguish missing context values. To bypass this during the MIMIC-IV forecasting, we forward-fill its context from observed data. TimePFN is evaluated only under condition B (shared-origin forecasting) for point metrics.

TabPFN-TS (Hoo et al., 2026). Official model, predicting target channels with covariates supplied via its native `future_df` interface. MSE uses its native mean point forecast and $CRPS_9$ its output quantiles. The native TabPFN-TS forecasting interface does not expose arbitrary interior-target imputation; this condition is therefore not reported.

Table S8: Baseline applicability. "Marginal" means that the method exposes per-cell or per-horizon distributions but no tractable density over the full queried vector. A mark states what the method can express, not what we evaluate: TACTiS-2 is trained per dataset rather than zero-shot and is therefore reported in Table S10 instead of Table 2, and its decoder indexes whole timesteps, so A, C and D require the per-cell selection described below. CSDI and ApolloPFN are not evaluated here.

Method	A	B	C	D	Probabilistic object
Exact/NUTS MOGP (Álvarez & Lawrence, 2011; Hoffman & Gelman, 2014)	✓	✓	✓	✓	explicit joint
TimePFN (Taga et al., 2025)	✗	✓	✗	✗	point forecast
TabPFN-TS (Hoo et al., 2026)	▲	✓	▲	✗	marginal target PPD
ApolloPFN (Potapczynski et al., 2026)	✓	✓	▲	✗	marginal target PPD
Chronos-2 (Ansari et al., 2025)	✓	✓	▲	✗	forecast quantiles
TiRex-2 (Podest et al., 2026)	✓	✓	▲	✗	forecast quantiles
TACTiS-2 (Ashok et al., 2024)	✓	✓	✓	✓	attentional-copula joint
CSDI (Tashiro et al., 2021)	▲	▲	▲	✓	conditional samples
TabImpute (Feitelberg et al., 2026)	✓	✓	✓	✓	Riemann-bar marginals
JointTimePFN (this work)	✓	✓	✓	✓	low-rank Gaussian joint

Chronos-2 (Ansari et al., 2025). Official zero-shot checkpoint through `predict_quantiles`. Under the shared-origin forecast the queried channels are passed as a single multivariate target from the mask's own origin. Under known-future conditioning and the ragged edge, each queried channel is a univariate target and every other channel enters as a covariate carrying `NaN` at steps whose value is not yet known, so a partially observed future contributes its known prefix rather than being discarded. The interface accepts `NaN` in the context and in a covariate without error, and returns finite quantiles for channels that carry data. Interior imputation is not expressible through this forecasting interface and is dashed. CRPS_9 comes from its native quantiles.

Note: Chronos-2's condition-B average is dominated by a small set of windows in which a queried channel carries no observed value anywhere in the context, so the multivariate target it receives is an all-`NaN` series. This occurs in 9 of the 400 windows and affects 11 (window, channel) pairs, or 317 of the scored cells (0.098%), and those cells carry 88.6% of the arm's condition-B CRPS_9 sum: the returned quantiles reach order 10^6 while the standardized targets have median magnitude below one. We verified from the saved predictions that the effect is a property of the returned quantiles and not of our harness: the adapter applies no rescaling of any kind, never reorders quantiles, reproduces the stored targets in the evaluation's own cell order, and scores every arm with the same imported CRPS_9 routine. The installed interface neither documents nor rejects a target history with no observed value. Excluding those windows the two Chronos-2 arms average 0.414 and 0.418 under condition B, against 0.412 for TiRex-2; we report the unfiltered averages in Table 2, since no computational error is involved and no other arm receives such a correction.

TiRex-2 (Podest et al., 2026). We use the zero-shot multivariate checkpoint. TiRex-2 natively supports known-future covariates. Its input arrays allow `NaN` for missing context values without disrupting temporal alignment. Like Chronos-2, it is restricted to horizon-based masks. It reports deciles only, so it supplies no 0.025 and 0.975 quantiles and its coverage column is empty.

TACTiS-2 (Ashok et al., 2024). We use the released configuration and its two-stage schedule, marginals first and then the attentional copula with the marginal weights frozen. Training uses a patient-level split of this cohort: no patient contributing to the 400 evaluation windows appears in the 150 training or 163 validation windows, which requires splitting on subject rather than stay identifiers. Its decoder indexes whole timesteps, so scoring an arbitrary per-cell mask requires replacing that indexing with a per-cell selection. The architecture, the weights and the likelihood itself are unchanged, and the two implementations coincide only on the mask family both express, a shared forecast origin over a fully observed window.

TabImpute (Feitelberg et al., 2026). We use the official TabImpute model (ImputePFN). As a tabular foundation model, TabImpute treats time-steps as unordered rows. It is the only zero-shot external baseline here that expresses interior missingness (Condition D) under arbitrary mask geometries. Its metrics are evaluated over its native Riemann bar distribution.

JOINTTIMEPFN (marginal). The same architecture and pretraining prior as JOINTTIMEPFN, trained on the marginal score of Equation 12 alone, so it is the ablation that prices the joint head. Dropping the factor head leaves 11.0M parameters against 20.2M (Full JOINTTIMEPFN).

Autoregressive (AR) extraction. AR is a teacher-forced chain-rule factorization of our own marginals. For a window we fix a raster order, i.e. every queried cell at the first timestep, then every queried cell at the next, and so on. This is done over a random subset of at most 60 queried cells and then evaluated

$$\log q_\pi(\boldsymbol{y}_\mathcal{Q} \mid \boldsymbol{Y}_\mathcal{O}) = \sum_{k=1}^{K} \log q\big(y_{\pi_k} \mid \boldsymbol{Y}_\mathcal{O},\, y_{\pi_{<k}}\big)\,, \tag{S7}$$

proceeding cell by cell: the model is re-run on the current partially patterned grid, the next cell's predicted marginal is evaluated at its true value, and that true value is then written into the input with its mask flipped to observed. This needs K sequential forward passes. Because the density is evaluated, not sampled, the AR entry carries no Monte Carlo error and needs no fitted covariance. Each factor is the model's own Gaussian conditional, while their product is not necessarily Gaussian. Our model is scored on the same cells through $\boldsymbol{\Sigma}_S = [\boldsymbol{\Sigma}]_{S,S}$. This restriction is exact, i.e. any sub-block of a joint Gaussian's covariance is the covariance of the corresponding marginal, so the two constructions differ only in how the joint is formed. The same 60 query cell cap and the same construction are used in the fidelity experiments and on MIMIC-IV.

D ADDITIONAL RESULTS

D.1 MIMIC-IV SUPPLEMENTARY TABLES

Table S9: **Bootstrap intervals and coverage for Table 2** on the same 400 evaluation windows. Intervals are 95% percentile intervals from 4000 resamples of the windows, which are the resampling unit because cells within a window are dependent. These are per-arm intervals, so an overlap or a gap between two rows is not a significance statement. A parenthesised count marks an arm scored on fewer than all 400 windows. TiRex-2 reports deciles only, so it has no coverage entry.

(a) CRPS_9

Model	A	B	C	D
TimePFN	–	–	–	–
TabPFN-TS	0.3210 [0.3135, 0.3284]	0.4288 [0.4196, 0.4380] (391)	0.3619 [0.3547, 0.3689] (399)	–
TabImpute	0.3526 [0.3447, 0.3608]	0.4322 [0.4229, 0.4414]	0.3875 [0.3796, 0.3956]	0.3440 [0.3390, 0.3492]
Chronos-2 (120M)	0.9885 [0.4556, 1.7997]	3.6180 [1.0354, 7.8151]	0.5427 [0.3828, 0.7605]	–
Chronos-2 (28M)	0.6232 [0.3920, 0.9099]	2.1598 [0.7438, 4.4989]	0.4358 [0.3580, 0.5460]	–
TiRex-2	0.3382 [0.3308, 0.3459]	0.4121 [0.4035, 0.4207]	0.3522 [0.3453, 0.3590]	–
JOINTTIMEPFN (marginal)	0.3798 [0.3721, 0.3875]	0.4264 [0.4184, 0.4343]	0.3788 [0.3720, 0.3856]	0.2733 [0.2698, 0.2769]
JOINTTIMEPFN (joint)	0.3647 [0.3572, 0.3724]	0.4292 [0.4212, 0.4373]	0.3749 [0.3681, 0.3816]	0.2698 [0.2663, 0.2733]

(b) MSE

Model	A	B	C	D
TimePFN	–	0.7835 [0.7518, 0.8174]	–	–
TabPFN-TS	0.5339 [0.5066, 0.5640]	0.8098 [0.7681, 0.8556] (391)	0.6109 [0.5879, 0.6333] (399)	–
TabImpute	0.5650 [0.5396, 0.5917]	0.7558 [0.7233, 0.7884]	0.6399 [0.6133, 0.6663]	0.5316 [0.5157, 0.5491]
Chronos-2 (120M)	0.5247 [0.5009, 0.5497]	0.7179 [0.6884, 0.7475]	0.5715 [0.5498, 0.5931]	–
Chronos-2 (28M)	0.5340 [0.5102, 0.5591]	0.7351 [0.7035, 0.7681]	0.5815 [0.5595, 0.6035]	–
TiRex-2	0.5420 [0.5187, 0.5663]	0.7241 [0.6935, 0.7553]	0.5647 [0.5433, 0.5857]	–
JOINTTIMEPFN (marginal)	0.6168 [0.5921, 0.6422]	0.7213 [0.6918, 0.7514]	0.6083 [0.5860, 0.6300]	0.3487 [0.3395, 0.3580]
JOINTTIMEPFN (joint)	0.5778 [0.5538, 0.6028]	0.7123 [0.6826, 0.7421]	0.5920 [0.5704, 0.6129]	0.3420 [0.3330, 0.3510]

(c) Cov95, target 0.95

Model	A	B	C	D
TimePFN	–	–	–	–
TabPFN-TS	0.922	0.924 (391)	0.932 (399)	–
TabImpute	0.958	0.943	0.949	0.960
Chronos-2 (120M)	0.970	0.969	0.958	–
Chronos-2 (28M)	0.959	0.967	0.955	–
TiRex-2	–	–	–	–
JOINTTIMEPFN (marginal)	0.941	0.959	0.946	0.930
JOINTTIMEPFN (joint)	0.947	0.963	0.950	0.929

Table S10: **Joint negative log-likelihood of four constructions on MIMIC-IV** (nats per queried cell, **lower is better**), on the same 400 evaluation windows and one shared subset of at most 60 queried cells per group, with the window-level K-weighted aggregation used throughout. The cap is the autoregressive cost of one forward pass per cell. The subset is drawn once and every row is scored on it, so these numbers are not comparable to Table 2(b), which uses all queried cells. JOINTTIMEPFN (diagonal) is the same model's own joint with the off-diagonal set to zero. **TACTiS-2 is not zero-shot**: it is trained on a patient-disjoint split of 150 windows from this cohort and evaluated through a masked-likelihood adaptation of its decoder, whereas JOINTTIMEPFN has never seen MIMIC-IV.

Model	A	B	C	D
JOINTTIMEPFN (joint)	1.6500	2.9579	2.3826	1.3972
JOINTTIMEPFN (diagonal)	2.6490	3.2519	2.8665	1.8562
Autoregressive extraction (teacher-forced)	0.8532	0.5063	0.5763	0.6273
TACTiS-2 (trained on MIMIC-IV, 150 windows)	0.5680	0.7630	0.7083	0.7088

D.2 ADDITIONAL DATASETS

With external datasets (Table S11) we show that the performance remains competitive while the joint gain varies. It is positive in every condition on ETTh1, but negative under conditions B and C on exchange rate. These results show that the joint head does not consistently improve likelihood over its diagonal counterpart across datasets and thus that is left for future work to solve. Additionally, as shown with MIMIC-IV table in the main text, here it is also visible that especially condition A is harder for our model than to other models. In electronic health records domain, with PhysioNet, the same pattern holds under the forecasting conditions, while under interior imputation TabImpute and our model are within 0.03 CRPS_9 of each other. The joint gain is positive under A, B and C there, and negative under D. TabPFN-TS scored 36 and 81 of the 100 groups under B and C, so only its A entry rests on every group. The B and C subsets are selected by whether its feature regression converged, which happens on the better-conditioned contexts, so those two entries are marked and should not be read against the arms that scored every group. Additionally, TimePFN is omitted here: it has no mask input, and PhysioNet's 36 channels are measured sparsely within a 24-hour context, so its fixed 96-step context would have had to be almost entirely for example forward-filled.

Table S11: **ETTh1, exchange rate and PhysioNet 2012**, the four conditions of Table 2, per queried cell, **lower is better** except Δ. ETTh1 has $N = 7$ channels and all 90 non-overlapping windows of the series, exchange rate $N = 8$ and all 39. These two channel counts lie inside the $N \in [2, 20]$ range seen in pretraining; PhysioNet has $N = 36$ and is the one dataset here above that range. Its records are 48 hours long, so they are padded to the 192-step grid our model expects, with the real hours at $t = 144..191$ and the forecast origin at $t = 168$, giving 24 hours of context and a 24-hour horizon. Our model reads that padded grid, while the external models read the same rows unpadded, since a context that is 86% padding would measure input handling rather than forecast skill. Padded cells are neither evidence nor targets for any model. † marks TabPFN-TS under B and C, where its own feature regression fails to converge on these short contexts. Every other model scored every group of every condition it supports.

(a) ETTh1, hourly, $N = 7$, 90 windows

	$\text{CRPS}_9 \downarrow$				MSE↓			
Model	A	B	C	D	A	B	C	D
TimePFN	–	–	–	–	–	0.4757	–	–
TabPFN-TS	0.1353	0.3146	0.2295	–	0.1302	0.4960	0.3156	–
TabImpute	0.3631	0.4190	0.3631	0.3261	0.6009	0.6956	0.5822	0.4933
Chronos-2 (120M)	0.1426	0.2893	0.2056	–	0.1330	0.4386	0.2427	–
Chronos-2 (28M)	0.1489	0.2928	0.2147	–	0.1459	0.4498	0.2586	–
TiRex-2	0.1686	0.2956	0.2233	–	0.1820	0.4517	0.2805	–
JOINTTIMEPFN (marginal)	0.2482	0.3076	0.2531	0.1777	0.3226	0.4593	0.3130	0.1608
JOINTTIMEPFN (joint)	0.2268	0.3054	0.2528	0.1732	0.2687	0.4493	0.3116	0.1534

Cond	NLL_{diag}	$\text{NLL}_{\text{joint}}$	$\Delta \uparrow$
A	0.6924	0.0603	0.6321
B	1.0625	0.7303	0.3322
C	0.7595	0.3942	0.3653
D	0.2720	0.2353	0.0368

(b) Exchange rate, daily, $N = 8$, 39 windows

	$\text{CRPS}_9 \downarrow$				MSE↓			
Model	A	B	C	D	A	B	C	D
TimePFN	–	–	–	–	–	0.1177	–	–
TabPFN-TS	0.1014	0.1282	0.0941	–	0.0575	0.0938	0.0566	–
TabImpute	0.1491	0.1641	0.1393	0.0672	0.1110	0.1344	0.1047	0.0243
Chronos-2 (120M)	0.0972	0.1310	0.0876	–	0.0532	0.0978	0.0504	–
Chronos-2 (28M)	0.0959	0.1354	0.0880	–	0.0523	0.1081	0.0507	–
TiRex-2	0.0946	0.1288	0.0857	–	0.0493	0.0959	0.0476	–
JOINTTIMEPFN (marginal)	0.1013	0.1340	0.0887	0.0228	0.0584	0.1021	0.0493	0.0037
JOINTTIMEPFN (joint)	0.0991	0.1345	0.0895	0.0222	0.0558	0.1044	0.0507	0.0034

Cond	NLL_{diag}	$\text{NLL}_{\text{joint}}$	$\Delta \uparrow$
A	1.0729	-0.0581	1.1310
B	0.6743	3.0733	-2.3991
C	2.2370	6.0012	-3.7642
D	-0.9151	-1.2080	0.2929

(c) PhysioNet 2012, hourly, $N = 36$, 48-hour records

	$\text{CRPS}_9 \downarrow$				MSE↓			
Model	A	B	C	D	A	B	C	D
TabPFN-TS	0.2412	$0.3719^{\dagger}$	$0.2972^{\dagger}$	–	0.3893	$0.6852^{\dagger}$	$0.5162^{\dagger}$	–
TabImpute	0.3054	0.3753	0.3323	0.2522	0.4588	0.6187	0.5402	0.3371
Chronos-2 (120M)	0.3249	0.4034	0.3345	–	0.4321	0.6180	0.5186	–
Chronos-2 (28M)	0.3027	0.3756	0.3181	–	0.4164	0.6162	0.4960	–
TiRex-2	0.2687	0.3564	0.2998	–	0.3872	0.6197	0.4744	–
JOINTTIMEPFN (marginal)	0.3372	0.3643	0.3397	0.2763	0.5153	0.5962	0.5342	0.3478
JOINTTIMEPFN (joint)	0.3335	0.3649	0.3397	0.2775	0.5028	0.5898	0.5237	0.3402

Cond	NLL_{diag}	$\text{NLL}_{\text{joint}}$	$\Delta \uparrow$
A	3.7450	2.2650	1.4800
B	3.4227	3.0259	0.3968
C	1.9737	1.7914	0.1822
D	2.6690	3.1349	-0.4659

D.3 FULL FIDELITY TABLES

Table 1(a) in the main text carries the joint divergences alone. Tables S12 and S13 give the same rows as well as marginal divergences with point error, coverage, and the calibration statistics. These are all on the same matched cells and with the same median aggregation. Every entry is a median over the 30 configurations of Appendix B.2 that each condition and band contributes, with none screened out; divergences are in nats per queried cell. All columns are computed on matched cells ($N \leq 6$).

Table S12: **Marginal fidelity, full columns.** Off-prior, the oracle's own error is essentially the generating noise variance, so MSE ratios compress toward one and the excess is what to read. Note that the off-prior noise makes the oracle's own MSE relatively large as well.

(a) In-prior

Cond	MSE	MSE_{NUTS}	MSE_{orc}	Cov95	$\text{Cov95}_{\text{orc}}$	$D_{\text{KL}}(\text{orc}\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(\text{N}\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(\text{orc}\|\text{N})$
A	0.045	0.020	0.011	0.973	0.958	0.716 [0.360, 1.019]	0.622 [0.257, 0.965]	0.081
B	0.243	0.295	0.182	0.927	0.953	0.311 [0.202, 0.523]	0.248 [0.159, 1.402]	0.156
C	0.080	0.073	0.044	0.944	0.957	0.468 [0.222, 0.789]	0.644 [0.236, 1.492]	0.116
D	0.021	0.018	0.010	0.941	0.952	0.282 [0.112, 0.792]	0.262 [0.096, 0.987]	0.030

(b) Off-prior (shifted band)

Cond	MSE	MSE_{NUTS}	MSE_{orc}	Cov95	$\text{Cov95}_{\text{orc}}$	$D_{\text{KL}}(\text{orc}\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(\text{N}\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(\text{orc}\|\text{N})$
A	1.515	1.433	1.204	0.963	0.954	0.129 [0.088, 0.172]	0.483 [0.197, 0.632]	0.573
B	2.523	2.297	2.205	0.973	0.949	0.119 [0.090, 0.162]	0.158 [0.113, 0.190]	0.113
C	2.133	2.052	1.695	0.964	0.950	0.089 [0.066, 0.120]	0.308 [0.168, 0.411]	0.268
D	1.419	1.539	1.228	0.943	0.951	0.080 [0.060, 0.103]	0.335 [0.198, 0.443]	0.498

Table S13: **Joint fidelity, full columns.** z^2/K uses the full covariance, z^2/K_{diag} the implied marginals alone, and z^2/K_{orc} the exact oracle; all three target 1, and none of them detects missing dependence. TC(orc)$= D_{\text{KL}}(\text{orc} \,\|\, \text{diag}\,\text{orc})$ is the dependence present at known ϕ (Table S15 compares it with the amount our head captures), while *diag null* also carries the part induced by hyperparameter uncertainty. Under condition A the two differ by an order of magnitude (0.067 against 0.007), so most of that condition's dependence is ϕ-induced. The divergence column is against the mixture $q^\star$, not the moment-matched N of Table 1(a). The comparison of the tables show that in-prior the two differ by 0.013–0.017 nats, less than $D_{\text{KL}}(\text{orc}\|\text{N})$ in every condition.

(a) In-prior

Cond	diag null	$D_{\text{KL}}(\text{orc}\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(q^\star\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(\text{orc}\|\text{N})$	TC(orc)	z^2/K	z^2/K_{diag}	z^2/K_{orc}
A	0.067 [0.017, 0.246]	0.192 [0.109, 0.498]	0.325 [0.150, 0.524]	0.029	0.007	0.93	1.06	0.99
B	0.799 [0.372, 1.164]	0.263 [0.153, 0.396]	0.301 [0.148, 1.387]	0.076	0.724	1.02	1.10	0.98
C	0.312 [0.204, 0.499]	0.294 [0.113, 0.637]	0.518 [0.151, 1.135]	0.031	0.320	1.01	1.05	0.99
D	0.034 [0.008, 0.064]	0.347 [0.112, 0.774]	0.350 [0.107, 1.011]	0.023	0.017	1.11	1.07	1.00

(b) Off-prior (shifted band)

Cond	diag null	$D_{\text{KL}}(\text{orc}\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(q^\star\|\text{JOINTTIMEPFN}) \downarrow$	$D_{\text{KL}}(\text{orc}\|\text{N})$	TC(orc)	z^2/K	z^2/K_{diag}	z^2/K_{orc}
A	0.074 [0.018, 0.220]	0.077 [0.061, 0.100]	0.417 [0.232, 0.512]	0.622	0.004	1.01	0.86	0.98
B	0.270 [0.181, 0.490]	0.071 [0.055, 0.087]	0.256 [0.156, 0.324]	0.260	0.073	0.95	0.70	0.99
C	0.249 [0.147, 0.367]	0.069 [0.055, 0.106]	0.380 [0.242, 0.598]	0.440	0.053	0.96	0.83	0.98
D	0.062 [0.022, 0.119]	0.078 [0.064, 0.103]	0.383 [0.241, 0.525]	0.586	0.005	1.08	1.06	1.00

D.4 AUTOREGRESSIVE COHERENCE PROBES

Autoregressive extraction, defined in Appendix C, is a strong competitor as it can express skew and multimodality that Equation 9 cannot. We evaluate our model against this on the same queried cells. Unlike the direct joint head, autoregressive extraction sequentially conditions on previously revealed target values, which can change both its conditional marginals and its dependence structure.

Table 1(b) reports this comparison on the $Q = 2$ pool. Off-prior the four constructions fall within 0.18 nats per cell of one another, so the ladder does not separate them there and the panel reports the in-prior band only. Table S16 scores the same four constructions on the $Q = N$ pool, where the ordering is the same.

Table S14: **Joint fidelity with as many latent functions as channels** (nats per cell, **lower is better**), columns as in Table 1(a). These runs set $Q = N$, so the reference posterior is not restricted to a rank-Q dependence structure. Thus these rows are comparable within this table but not against the panels of Table 1. Table S16 scores the same pool. Medians over 30 configurations per cell.

	In-prior				Off-prior (shifted band)			
Cond	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{N})$	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{Ours})$	$D_{\mathrm{KL}}(\mathrm{N}\|\mathrm{Ours})$	diag null	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{N})$	$D_{\mathrm{KL}}(\mathrm{orc}\|\mathrm{Ours})$	$D_{\mathrm{KL}}(\mathrm{N}\|\mathrm{Ours})$	diag null
A	0.059	0.242	0.319	0.345	0.448	0.062	0.357	0.151
B	0.210	0.219	0.374	0.490	0.124	0.071	0.128	0.155
C	0.222	0.318	0.548	0.404	0.246	0.054	0.204	0.172
D	0.240	0.367	0.322	0.044	0.279	0.062	0.207	0.030

Table S15: **Total correlation of the exact oracle against the head's own**, in-prior, nats per cell. *Available* is $D_{\mathrm{KL}}(\mathrm{orc}\|\,\mathrm{diag}\,\mathrm{orc})$, the dependence present at known ϕ; *captured* is the same quantity for our joint, so the head under-couples below available and asserts absent dependence above it. Left: the pool of Table 1, where a rank-2 structure has to describe up to six channels. Right: $Q = N$, where both cover $N \in \{2, 4, 6\}$ with medians over 30 configurations. At $Q = 2$ the oracle carries almost no dependence under A or D, so B and C are the informative rows there.

	$Q = 2$, $N \in \{2, 4, 6\}$		$Q = N$, $N \in \{2, 4, 6\}$	
Cond	available	captured	available	captured
A	0.0067	0.4592	0.1252	0.3904
B	0.7236	0.5627	0.4075	0.3599
C	0.3205	0.3965	0.3632	0.2999
D	0.0167	0.0227	0.0367	0.0136

Table S16: **Joint negative log-likelihood of four constructions** (in-prior, nats per cell, **lower is better**), on identical cells. The oracle is closed-form at the true hyperparameters and is the optimum a model marginalizing over ϕ should not reach in LMC domain. The AR column is the exact chain-rule density of our own conditionals, under one fill order, so this table reports no spread across orders. These runs share the $Q = N$ pool of Table S14, so rows are again comparable within this table but not against the panels of Table 1.

Cond	Exact oracle	Ours	AR extraction	Independent marginals
A	-1.20	-0.45	-0.39	-0.02
B	-0.25	0.15	0.20	0.55
C	-0.78	-0.16	-0.13	0.19
D	-0.86	-0.11	-0.14	-0.11

Table S17: **Joint KL divergence from the exact oracle ($D_{\mathrm{KL}}(\mathrm{orc}\|\cdot)$) with external prior-fitted baselines** (nats per queried cell, **lower is better**). Results use the same $N \in \{2, 4, 6\}$, ten seeds per band, 60-cell joint query cap, and configuration-level aggregation as Table 1. The masks are independently redrawn for this experiment so results differ slightly from our model's performance in Table 1. TabImpute and TabPFN-TS are evaluated as diagonal Gaussians matched to their native predictive means and variances. This emphasises the meaningfulness of our model's joint head.

	In-prior				Off-prior (shifted band)			
Cond	Ours	Ours (diag)	TabImpute	TabPFN-TS	Ours	Ours (diag)	TabImpute	TabPFN-TS
A	0.194	0.856	2.334	1.227	0.077	0.151	0.483	0.182
B	0.251	1.088	2.282	1.768	0.074	0.205	0.481	0.182
C	0.294	0.872	2.131	1.231	0.072	0.146	0.439	0.166
D	0.322	0.338	1.444	–	0.085	0.089	0.265	–

D.5 Ablations

Results are in Table S18. We ablate: (i) absolute positional encodings versus RoPE; (ii) the broadened prior, against TimePFN's fixed kernel bank and against unsigned loadings; (iii) joint rank $R \in \{4, 8, 16, 32, 64, 128\}$; (iv) marginal loss weight $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$; and (v) β-NLL at $\beta = 0.5$ (Seitzer et al., 2022), versus the strict log score, for the marginal baseline. Every variant is evaluated on the full MIMIC-IV data, a larger evaluation than the 400 windows of Table 2, so ablation rows are comparable with one another rather than with that table.

The broadened prior matters most: TimePFN's fixed kernel bank raises the joint NLL from 1.945 to 8.273 and is worse on every other column, and unsigned loadings cost 0.053 nats. RoPE is worth 0.160, when evaluating on MIMIC-IV.

Dropping the marginal term ($\lambda = 0$) gives the best marginal NLL in the table, 2.342 against 2.716 at $\lambda = 0.5$. The joint term already constrains the marginals through the diagonal of $\mathbf{\Sigma}$, so $\lambda = 0$ does not leave them unsupervised; it removes the term that penalizes a per-cell variance for being too wide, which shows up as over-coverage at 0.963 against a target of 0.95. It is also worse on MSE, on CRPS_9 and on the joint NLL we select against, so we keep the composite loss.

Because the joint head analytically induces marginals via $\Sigma_{ii} = d_i + \|\boldsymbol{w}_i\|^2$, the marginal-only checkpoints sit in the same table on identical cells, which shows whether the joint head buys its likelihood by degrading the marginals it is meant to reproduce.

Table S18: Core ablations on MIMIC-IV, averaged over the four conditions (MSE, CRPS_9, marginal NLL and joint NLL lower is better; Cov95 targets 0.95). Marginal columns for the joint models use the implied marginals $\Sigma_{ii} = d_i + \|\boldsymbol{w}_i\|^2$, so every row is comparable with the marginal-only checkpoints on the same cells. The joint NLL is the selection criterion because it evaluates the rank-dependent covariance structure. Marginal metrics are also reported, since changing the rank and retraining the model can affect its implied marginal predictions. Rank ablation is done with $\lambda = 0.5$ and λ ablation with $R = 32$; the prior and RoPE ablations use both. Run on the full cohort, 1256 windows on A–C and 1258 on D.

Variant	MSE	CRPS_9	Marg. NLL	Joint NLL	Cov95
Full JointTimePFN ($\lambda = 0.5$, $R = 32$)	0.5707	0.3653	2.716	1.945	0.947
No RoPE	0.5912	0.3703	2.919	2.105	0.915
TimePFN's fixed kernel bank	0.6506	0.4048	4.681	8.273	0.969
Unsigned loadings	0.5915	0.3714	2.904	1.998	0.944
$R = 4$	0.5784	0.3664	3.117	2.140	0.946
$R = 8$	0.5769	0.3665	3.056	2.162	0.947
$R = 16$	0.5704	0.3639	2.613	2.025	0.947
$R = 32$	0.5707	0.3653	2.716	1.945	0.947
$R = 64$	0.5766	0.3659	2.781	1.989	0.944
$R = 128$	0.5834	0.3678	3.337	2.043	0.946
$\lambda = 0$	0.5751	0.3866	2.342	1.984	0.963
$\lambda = 0.25$	0.5752	0.3680	2.813	2.015	0.951
$\lambda = 0.5$	0.5707	0.3653	2.716	1.945	0.947
$\lambda = 0.75$	0.5737	0.3648	2.908	1.979	0.946
$\lambda = 1.0$	0.5764	0.3648	2.772	2.018	0.944
Marginal only, log score	0.5867	0.3698	3.279	–	0.944
Marginal only, β-NLL	0.5859	0.3704	2.939	–	0.943

E HYPERPARAMETERS, TRAINING, AND COMPUTE

Compute. Pretraining the reported checkpoint took 18 hours on a single NVIDIA V100 (16 GB) with 10 CPU cores and 128 GB of system memory. The main-text results come from that single checkpoint; each variant in Table S18 required its own pretraining run of comparable length on the same hardware, and evaluation adds no training.

Table S19: JOINTTIMEPFN hyperparameters. Architecture and numerics are as implemented; the optimizer and schedule entries are read from the training log of the reported checkpoint.

Quantity	Value
Architecture	
Window length T	192 (`seq_len` $= 96$, `pred_len` $= 96$); fixed, since each head emits exactly T positions per channel
Patch size / patch stride	32 / 16 (hardcoded `patch_size`$/2$ in the unfold), giving $P = 11$ patches
Local convolutions	three scales, kernels $5/9/15$ over 2 input features (value and mask), 3 output channels each; magnitude-max pooling $3/5/7$
Convolution padding	symmetric, $(2,2)/(4,4)/(7,7)$, against TimePFN's causal $(4,0)/(8,0)/(14,0)$
Token embedding	inner dimension $32 \cdot 11$, projected to $d_{\text{embed}} = 256$
Transformer layers / width / heads	8 / $d = 256$, $d_{\text{ff}} = 512$ / 8
Prediction-head hidden width	$d_{\text{model}} = 1024$
Positional encoding	2-D *fixed sinusoidal* over (channel, patch), no learned position parameters, plus rotary on the patch axis, base 10
Joint rank R	32
Parameters	20.2M; 11.0M for the marginal-only checkpoint.
Factor head initialization	$\mathcal{N}(0, 10^{-4})$ (std 10^{-2}), zero bias
Objective and numerics	
λ	0.5 (Appendix D.5)
Log-variance clamp	$[-10, 10]$
Variance floor $d_{\min}$	10^{-4}
Ridge on $\boldsymbol{d}$	10^{-6}
Normalization ε	10^{-4}, with $s_n^{\text{obs}} = \sqrt{v_n^{\text{obs}} + \varepsilon}$
Loss normalization	per sample, so the declared mask-mixture weights are the realized ones in both training and validation
Queried-cell cap per batch	none; at $N = 20$, $TN = 3840$ cells fit without capping
Pretraining	
Prior	broadened LMC of Appendix A.1; mixture weights in Table S2
Generated series	length 1024, $C = 20$ channels, window stride 8, split $80/10/10$
Series count / channel subsampling	60000
Optimizer / learning rate	Adam, one-cycle / 3×10^{-4}; $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, no weight decay
Batch size	16
Epochs / validation interval / patience	1 / every 2000 steps / 200
Initialization	from scratch, no warm start, encoder and mean path trainable throughout