# GAUSSIAN MIXTURE COPULA PROCESSES FOR IRREGULAR MULTIVARIATE TIME SERIES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We introduce *Gaussian Mixture Copula Processes* (GMCP), a conditional copula process for irregularly sampled multivariate time series (IMTS) that is expressive and marginalization consistent at the same time. Existing conditional copula processes achieve only one of the two. Gaussian copula processes are consistent by construction but confined to elliptical dependence, whereas attentional copulas such as TACTiS-2 are far more expressive but neither preserve the marginals they are built on nor guarantee that integrating out a target point returns the joint over the remaining ones. GMCP closes this gap by inferring the parameters of a Gaussian Mixture Copula from the context in a way that guarantees consistent marginalization. This yields a non-elliptical copula process with tractable likelihood and consistency guarantees. Our experiments show that copulas equipped with MargFlow, a normalizing-flow model for the univariate marginals fitted in isolation, outperform jointly trained baselines in probabilistic IMTS forecasting. We further provide evidence that GMCP is more expressive than Gaussian copula processes, and on par with the inconsistent TACTiS-2.

## 1 INTRODUCTION

Probabilistic forecasting of irregularly sampled multivariate time series (IMTS) implies modeling a stochastic process. Given a context of past observations, a model must assign a predictive distribution to any finite set of target points, where a target is a (channel, time) pair. Predicting each target point in isolation is not enough. Many downstream tasks, such as sampling coherent trajectories, necessitate the *joint* predictive distribution over all target points. Because the target set is chosen at prediction time, a model does not merely output one joint distribution but a whole family of them, one per target set. For this family to describe a valid stochastic process, the Kolmogorov extension theorem (Øksendal, 2003) requires it to be *marginalization consistent*. That is, integrating a target point out of a larger joint must return exactly the joint over the remaining points. For example, consider a weather model asked for tomorrow's temperature and snowfall. Marginalizing snowfall out of the joint forecast must reproduce the temperature forecast the model gives when asked about temperature alone. If not, the answer to "How cold will it be?" depends on whether snow was asked about. Hence, the model's predictions cannot all come from one underlying process. It would be hard to trust such a model.

*Copula processes* are an attractive way to build marginalization consistent models. By Sklar's theorem (Sklar, 1959; Nelsen, 2006), a joint distribution factorizes into its univariate marginals

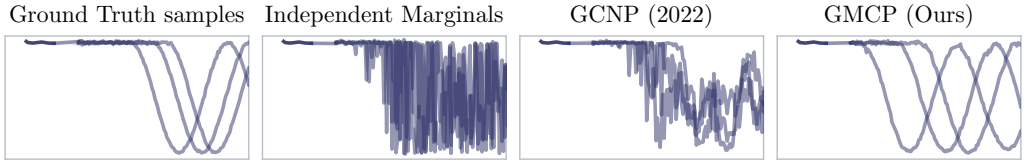


Figure 1: Three samples for the trajectory of a pendulum. Independently sampled marginals and a Gaussian Copula Neural Process (GCNP) (Markou et al., 2022) generate incoherent trajectories. In contrast, GMCP samples coherent swings.



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.

and a copula that carries the dependency structure alone. Modeling the two parts separately buys modeling freedom, which can lead to better predictive performance (Ashok et al., 2023). Additionally, univariate forecasts are explicit and unchanged by the joint model, which matters wherever univariate forecasts are calibrated or certified on their own, as in regulated financial and clinical settings. The existing conditional copula processes, however, force a choice between expressivity and consistency. Gaussian copula processes (Wilson & Ghahramani, 2010; Salinas et al., 2019; Markou et al., 2022) are marginalization consistent by construction, but their dependence structure is that of a multivariate Gaussian and hence elliptical and unimodal. TACTiS/TACTiS-2 (Drouin et al., 2022; Ashok et al., 2023) are far more expressive, but the attention-based copula is not guaranteed to be marginalization consistent and does not preserve the marginals it is built on. A copula family that offers more expressivity is the *Gaussian Mixture Copula* (GM-C) (Tewari et al., 2011). A GM-C is not a mixture of Gaussian copulas, but the copula induced by a latent Gaussian *mixture*. It therefore can represent dependency structures that lie outside the reach of Gaussian copulas, while retaining a tractable likelihood. So far, however, GM-Cs have only been studied as *static* models: one copula is fit to one fixed set of variables from repeated draws of those variables (Rajan & Bhattacharya, 2016; Bilgrau et al., 2016; Tewari, 2023). Whether a GM-C can be turned into a stochastic process, conditioned on a context and evaluated on arbitrary target sets, is unexplored.

We close this gap with *Gaussian Mixture Copula Processes* (GMCP). Rather than fitting one copula to one dataset, an encoder maps each instance's context and target set to its own copula, so that a different context induces a different dependency structure, not merely different marginals. Marginalization consistency has to hold while the copula itself varies with the context and with which points are requested. This restricts how the latent mixture may be parameterized. The encoder has to be trained by SGD backpropagating through a copula density that requires the inverse marginal CDF of a Gaussian mixture, for which automatic differentiation is expensive and numerically unstable. What we obtain is a copula process that is expressive and marginalization consistent at the same time, rather than one at the cost of the other. We demonstrate GMCP's expressivity advantage over Gaussian Copula Neural Processes (Markou et al., 2022) in Figure 1. The latter is not able to model the ground truth distribution due to the non-Gaussian dependence structure, while GMCP resembles the ground truth stochastic process. In the context of IMTS forecasting, GMCP learns to couple well-fitted marginals to attain a better joint likelihood than every jointly trained model. Our contributions can be summarized as follows:

- With GMCP, we introduce a **conditional neural copula process** whose dependency structure is a Gaussian Mixture Copula. To our knowledge it is the first copula process that is simultaneously conditioned on a context set, marginalization consistent, permutation equivariant, and able to express non-Gaussian dependence with a tractable likelihood.
- We derive a method to train GMCP as a marginalization consistent copula process for IMTS, built on a numeric inverse-CDF solver and using Tewari (2023)'s analytic gradients for GM-Cs.
- Using a synthetic dataset and four IMTS benchmarks we provide evidence that **GMCP is more expressive than Gaussian copula processes**.
- Furthermore, our experiments show that **GMCP matches TACTiS-2's performance** in probabilistic IMTS forecasting. In contrast to TACTiS-2, GMCP is constructed to guarantee marginalization consistency. We confirm this empirically by showing that TACTiS-2 changes the univariate marginal distributions, while GMCP preserves them.
- We show that training a model to explicitly predict univariate marginals yields superior marginal likelihood estimation compared to models that are optimized on the joint likelihood.

## 2 PRELIMINARIES

Throughout, uppercase letters denote cumulative distribution functions (CDFs) and the corresponding lowercase letters denote their densities. A copula is a CDF with domain $[0, 1]^N$ and uniform marginals. *Sklar's theorem* (Sklar, 1959; Nelsen, 2006) states that every joint CDF $F$ with marginals $F_n$ can be written as $F(\mathbf{y}) = \mathbf{C}(\mathbf{u})$, where $\mathbf{u} = \big(F_1(y_1), \ldots, F_N(y_N)\big)$ collects the probability integral transforms of the coordinates, and that the copula $\mathbf{C}$ is unique whenever the marginals are continuous.

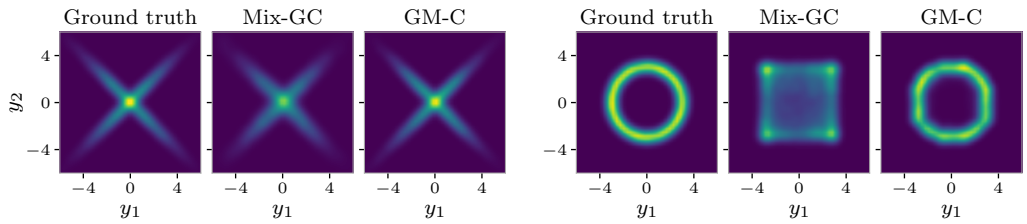


Figure 2: Comparing the expressiveness of a Mixture of Gaussian Copulas (Mix-GC) with a Gaussian Mixture Copula (GM-C) on two 2-D toy datasets. (More details are in Appendix A).

For any continuous density function $f$, its copula density can be written explicitly as:

$$\mathbf{c}_f(\mathbf{u}) \coloneqq \frac{f\big(F_1^{-1}(u_1), \ldots, F_N^{-1}(u_N)\big)}{\prod_{n=1}^{N} f_n\big(F_n^{-1}(u_n)\big)}, \qquad \mathbf{u} \in [0,1]^N. \tag{1}$$

Conversely, one can define a joint distribution by combining separate univariate density functions $g_n(y_n)$ with a copula density function: $p(\mathbf{y}) = \mathbf{c}\big(G_1(y_1), \ldots, G_N(y_N)\big) \prod_{n=1}^{N} g_n(y_n)$.

A well-established family of copula density functions is the *Gaussian copula*, obtained by applying Eq. (1) to a multivariate normal density $\mathcal{N}(\mu, \Sigma)$. As it depends neither on the mean $\mu$ nor on the scaling of the marginal variances, it is sufficient to consider multivariate normal distributions with mean $\mu = 0$ and a covariance matrix $\Sigma$ with all 1's on its diagonal, called a correlation matrix; thus the Gaussian copula is denoted simply by $\mathbf{c}_\Sigma \coloneqq \mathbf{c}_{\mathcal{N}(0,\Sigma)}$. It can only represent the dependence structure of a multivariate Gaussian. As mixtures of copula densities are also copula densities, a more expressive class of copula densities can be constructed as *mixtures of Gaussian copulas* (Mix-GC) (Marbac et al., 2017):

$$\mathbf{c}_{\pi,\Sigma_1,\ldots,\Sigma_K}(\mathbf{u}) \coloneqq \sum_{j=1}^{K} \pi_j \, \mathbf{c}_{\Sigma_j}(\mathbf{u}), \qquad \pi_j \geq 0, \quad \sum_{j=1}^{K} \pi_j = 1. \tag{2}$$

This construction is, however, fundamentally restricted. Khaled & Kohn (2023) showed that mixtures of Gaussian copulas are always radially symmetric on the unit cube, i.e. invariant under $\mathbf{u} \mapsto \mathbf{1} - \mathbf{u}$. A second obstruction concerns the center $\mathbf{u}^\star = (1/2, \ldots, 1/2)$ of the cube. Each component of Eq. (2) is standardized by its own marginals and, being a Gaussian copula, discards the mean, so $\mathbf{u}^\star$ is the componentwise median of every component alike and none of them can be placed off-center. Component $j$ therefore contributes $|\Sigma_j|^{-1/2} \geq 1$ at $\mathbf{u}^\star$ whatever its parameters, and hence $\mathbf{c}(\mathbf{u}^\star) \geq 1$ however many components are used. Hence, a mixture of Gaussian copulas can raise the density at the center above the independence value 1, but never lower it (Appendix A). Copulas with a hole at the center, such as that of the circle in Figure 2, are therefore out of reach.

**Gaussian Mixture Copulas.** A *Gaussian Mixture Copula* (GM-C) (Smillie, 2008; Tewari et al., 2011) is **not** a mixture of Gaussian copulas. A GM-C first defines a Gaussian Mixture Model (GMM):

$$f^{\mathrm{GMM}}(z) \;=\; \sum_{j=1}^{K} \pi_j \, \mathcal{N}(z; \mu_j, \Sigma_j), \qquad \pi_j \geq 0, \quad \sum_{j=1}^{K} \pi_j = 1, \tag{3}$$

and maps $u_n \in [0,1]$ to the latent space via the marginal inverse CDFs: $z_n = F_n^{\mathrm{GMM},-1}(u_n)$, where $F_n^{\mathrm{GMM}}$ denotes the marginal CDF of the Gaussian mixture $f^{\mathrm{GMM}}$. The copula is then obtained by applying Eq. (1) to $f^{\mathrm{GMM}}$, whereas Eq. (2) applies it to each component $g_j$ separately. The difference is the order of operations: $\sum_{j=1}^{K} \pi_j \, \mathbf{c}_{g_j} \;\neq\; \mathbf{c}_{\sum_{j=1}^{K} \pi_j g_j}$. Mixing after the transformation forces every component through the same standard normal marginals, whereas mixing before it lets the marginal transformations $F_n^{\mathrm{GMM},-1}$ depend on the full mixture, so that the induced copula inherits the multimodal structure of $f^{\mathrm{GMM}}$. This enables GM-Cs to represent complex, non-elliptical dependence patterns and the circle distribution in Figure 2.

## 3 Gaussian Mixture Copula Processes (GMCP)

We now introduce GMCP, a conditional copula process whose dependency structure is a Gaussian Mixture Copula. We define it in the general setting of neural processes.

**Neural Processes and Copula Processes.** A conditional neural process (Garnelo et al., 2018a) predicts outputs at freely chosen inputs, given a set of already observed input-output pairs. Let $\mathcal{X}$ denote the input space. We restrict outputs to be scalar and real-valued: $y^{\mathrm{tgt}} \in \mathbb{R}$. A *context set* $\mathcal{D}^{\mathrm{ctx}} = \{(x_m^{\mathrm{ctx}}, y_m^{\mathrm{ctx}})\}_{m=1}^{M}$ with $x_m^{\mathrm{ctx}} \in \mathcal{X},\ y_m^{\mathrm{ctx}} \in \mathbb{R}$, collects what has been observed, and a *target set* $\mathbf{X}^{\mathrm{tgt}} = (x_1^{\mathrm{tgt}}, \ldots, x_N^{\mathrm{tgt}})$ with $x_n^{\mathrm{tgt}} \in \mathcal{X}$ collects the inputs at which a prediction is requested. We write $\mathbf{y}^{\mathrm{tgt}} = (y_1^{\mathrm{tgt}}, \ldots, y_N^{\mathrm{tgt}}) \in \mathbb{R}^N$ for the corresponding target outputs. The model has to supply a joint predictive density $\hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$ for every context set and every finite target set. Because $\mathbf{X}^{\mathrm{tgt}}$ is chosen at prediction time, this is not one joint distribution but a whole family of them, one per target set. For the family to be the finite-dimensional distributions of a single stochastic process conditioned on $\mathcal{D}^{\mathrm{ctx}}$, the Kolmogorov extension theorem (Øksendal, 2003) requires it to be permutation equivariant and *marginalization consistent*: for every $k \in \{1, \ldots, N\}$,

$$\int_{\mathbb{R}} \hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})\, dy_k^{\mathrm{tgt}} \;=\; \hat{p}(\mathbf{y}_{-k}^{\mathrm{tgt}} \mid \mathbf{X}_{-k}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}), \tag{4}$$

where $\mathbf{X}_{-k}^{\mathrm{tgt}}$ and $\mathbf{y}_{-k}^{\mathrm{tgt}}$ denote $\mathbf{X}^{\mathrm{tgt}}$ and $\mathbf{y}^{\mathrm{tgt}}$ with the $k$-th entry removed. The property extends to arbitrary subsets by induction. In their original setup, conditional neural processes (Garnelo et al., 2018a) satisfy Eq. (4) in the simplest possible way: they factorize the joint across target points, $\hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) = \prod_{n=1}^{N} \hat{p}(y_n^{\mathrm{tgt}} \mid x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$, and thereby declare the target points conditionally independent, so that no dependence is modeled at all. The common remedy is an autoregressive factorization, applied either at deployment time (Bruinsma et al., 2022) or already during training (Nguyen & Grover, 2022). While this does induce dependency, it sacrifices permutation equivariance and marginalization consistency.

A *copula process* (Wilson & Ghahramani, 2010) is a neural process whose predictive density is built by combining the univariate marginal CDFs of the targets, $\mathbf{u} = \big(u_n = F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}} \mid x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})\big)_{n=1}^{N}$, with a copula density $\mathbf{c}$:

$$\hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) = \mathbf{c}(\mathbf{u} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \prod_{n=1}^{N} f_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}} \mid x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}). \tag{5}$$

**Conditioning the Gaussian Mixture Copula.** To the best of our knowledge, GMCP is the first copula process that is marginalization consistent and expresses non-Gaussian dependence with a tractable likelihood. Building a GM-C conditional on the context ($\mathcal{D}^{\mathrm{ctx}}$) and target set ($\mathbf{X}^{\mathrm{tgt}}$) necessitates inferring the parameters $\{\pi_j, \mu_j, \Sigma_j\}_{j=1}^{K}$ of this latent Gaussian mixture from $\mathcal{D}^{\mathrm{ctx}}$ and $\mathbf{X}^{\mathrm{tgt}}$.

The mixture weights are inferred from the context alone, $\pi = \phi_\pi(\mathcal{D}^{\mathrm{ctx}})$. In particular, $\pi$ is not informed about which target points are requested. This is intended: a mixture whose weights depend on the context alone stays marginalization consistent (Lemma 5, Appendix B), whereas weights that changed with $\mathbf{X}^{\mathrm{tgt}}$ would no longer commute with integrating out a target point. Each component $j$ of the Gaussian mixture has a mean vector $\mu_j \in \mathbb{R}^N$ and per-target standard deviations $\sigma_{j,n} > 0$, with one entry per target point. We infer the $n$-th entry of all $K$ components jointly, from the $n$-th target input and the context,

$$\big[\mu_{1,n}, \ldots, \mu_{K,n}\big] = \phi_\mu(x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \in \mathbb{R}^K, \qquad \big[\sigma_{1,n}, \ldots, \sigma_{K,n}\big] = \phi_\sigma(x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \in \mathbb{R}^K_{>0}. \tag{6}$$

Letting the components have their own means is what gives GMCP its expressivity beyond mixtures of Gaussian copulas: because $\mu_{j,n} \neq \mu_{j',n}$ in general, the components of the latent mixture can have spatially separated modes, and the induced copula can therefore represent more complex dependencies. The off-diagonal entries of $\Sigma_j$ encode the dependence between target points. They are inherently pairwise, yet each $\Sigma_j$ has to stay positive definite. We therefore do not infer them directly, but infer a feature vector per target point and component,

$$\big[\mathbf{v}_{1,n}, \ldots, \mathbf{v}_{K,n}\big] = \phi_V(x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \in \mathbb{R}^{K \times H}, \tag{7}$$

and read the dependence off their Gram matrix. We first normalize each feature vector, $\tilde{\mathbf{v}}_{j,n} = \frac{\mathbf{v}_{j,n}}{\sqrt{1+\|\mathbf{v}_{j,n}\|^2}}$, which places it strictly inside the unit ball. Stacking $\tilde{V}_j = [\tilde{\mathbf{v}}_{j,1}, \ldots, \tilde{\mathbf{v}}_{j,N}]^\top \in \mathbb{R}^{N \times H}$, we complete the diagonal of its Gram matrix to ones, which makes it a correlation matrix, and rescale by the inferred standard deviations,

$$R_j = \tilde{V}_j \tilde{V}_j^\top + \mathrm{diag}\big(1 - \|\tilde{\mathbf{v}}_{j,n}\|^2\big)_{n=1}^{N}, \qquad \Sigma_j = D_j R_j D_j, \tag{8}$$

with $D_j = \mathrm{diag}(\sigma_{j,1}, \ldots, \sigma_{j,N})$. The normalization keeps every diagonal entry of the completion strictly positive, so $R_j$ is a positive definite correlation matrix, and $\Sigma_j$ is symmetric positive definite and carries the inferred variances $\sigma^2_{j,n}$ on its diagonal.

**Theorem 1.** *GMCP, paired with any model that models univariate marginals independently, is marginalization consistent. Hence, for every context set $\mathcal{D}^{\mathrm{ctx}}$ and every target set $\mathbf{X}^{\mathrm{tgt}}$, the predicted joint density satisfies marginalization consistency as defined in Eq.* (4)*.*

*Proof sketch.* The mixture weights $\pi$ are inferred from $\mathcal{D}^{\mathrm{ctx}}$ alone, the $n$-th entries of $\mu_j$ and $\sigma_j$ depend only on $x_n^{\mathrm{tgt}}$ and $\mathcal{D}^{\mathrm{ctx}}$, and each entry of $\Sigma_j \in \mathbb{R}^{N\times N}$ depends only on the two target points it indexes. Removing a target point from $\mathbf{X}^{\mathrm{tgt}}$ therefore deletes exactly the corresponding entry of $\mu_j$ and the corresponding row and column of $\Sigma_j$, and marginalizing it out reduces to the standard rule for a multivariate Gaussian. Appendix B proves it formally.

**Training.** Following previous work (Yalavarthi et al., 2025; 2026), we fit GMCP by minimizing the normalized joint negative log-likelihood (njNLL). Since the predicted joint density factorizes as in Eq. (5), its logarithm splits into a copula term and a sum of marginal log-densities, and the copula term in turn unfolds via Eq. (1):

$$\mathcal{L}_{\mathrm{njNLL}} = -\frac{1}{N}\log \hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}), \tag{9}$$

$$= -\frac{1}{N}\Big(\log \mathbf{c}(\mathbf{u} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) + \sum_{n=1}^{N} \log f_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}} \mid x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})\Big), \tag{10}$$

$$\log \mathbf{c}(\mathbf{u} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) = \log f^{\mathrm{GMM}}(\mathbf{z}) - \sum_{n=1}^{N} \log f_n^{\mathrm{GMM}}(z_n). \tag{11}$$

Here, $f^{\mathrm{GMM}}(\mathbf{z}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j)$ is the latent Gaussian mixture density evaluated at $z_n = F_n^{\mathrm{GMM},-1}(u_n)$, and $f_n^{\mathrm{GMM}}$ is its $n$-th one-dimensional marginal. GMCP's parameters affect only the copula term, and they see the marginal model only through the values $u_n$ it supplies. Fitting the copula is therefore exactly minimizing the normalized copula log-density on whichever $\mathbf{u} \in (0,1)^N$ is handed to it, and no gradient has to be propagated through $F_n^{\mathrm{marg}}$.

Evaluating the copula log-density requires $z_n = F_n^{\mathrm{GMM},-1}(u_n)$, but the GMM marginal CDF cannot be inverted analytically. We obtain $z_n$ by a safeguarded Newton-bisection scheme in the forward pass (Appendix G), and compute gradients of $F_n^{\mathrm{GMM},-1}$ with respect to the GMM parameters analytically (Tewari, 2023). Proposition 2 states the resulting closed forms, and Appendix H derives them step by step; the backward pass composes them with the Jacobians of $\phi_\pi$, $\phi_\mu$ and $\phi_\sigma$.

**Proposition 2** (Gradients of the GMM inverse CDF)**.** *Let $z_n = F_n^{\mathrm{GMM},-1}(u_n)$, and let $\Phi$ and $\varphi$ denote the standard-normal CDF and PDF. Then the gradients with respect to the GMM parameters are:*

$$\frac{\partial z_n}{\partial \pi_j} = -\frac{\Phi\left(\frac{z_n-\mu_{j,n}}{\sigma_{j,n}}\right)}{f_n^{\mathrm{GMM}}(z_n)}, \quad \frac{\partial z_n}{\partial \mu_{j,n}} = \frac{\pi_j\,\varphi\left(\frac{z_n-\mu_{j,n}}{\sigma_{j,n}}\right)}{\sigma_{j,n}\, f_n^{\mathrm{GMM}}(z_n)}, \quad \frac{\partial z_n}{\partial \sigma^2_{j,n}} = \frac{\pi_j\,\varphi\left(\frac{z_n-\mu_{j,n}}{\sigma_{j,n}}\right)\,(z_n-\mu_{j,n})}{2\,\sigma^3_{j,n}\, f_n^{\mathrm{GMM}}(z_n)}$$

These gradients agree with finite differences of the solver output, and the remaining inversion error changes the njNLL by less than $5\times10^{-5}$ on average (Appendix I). The numerical CDF inversion has little impact on GMCP's computational complexity, which is analyzed in Appendix J.

**Sampling.** Given $\mathcal{D}^{\mathrm{ctx}}$ and $\mathbf{X}^{\mathrm{tgt}}$, we first apply $\phi_\pi, \phi_\mu, \phi_\sigma, \phi_V$ to parameterize the GM-C. We then draw a sample $\hat{\mathbf{z}} \in \mathbb{R}^N$ from this mixture by first sampling a component index $j \sim \mathrm{Categorical}(\pi)$ and then $\hat{\mathbf{z}} \sim \mathcal{N}(\mu_j, \Sigma_j)$. Next, we transform $\hat{\mathbf{z}}$ to the unit cube using the GMM's own marginal CDFs $F_n^{\mathrm{GMM}}$, that is $\hat{u}_n = F_n^{\mathrm{GMM}}(\hat{z}_n)$, where each $F_n^{\mathrm{GMM}}$ is itself a one-dimensional Gaussian mixture CDF and is available in closed form. Finally, we invert the marginal model on each coordinate to obtain $\hat{y}_n^{\mathrm{tgt}} = F_n^{\mathrm{marg},-1}(\hat{u}_n)$.

**Limitations.** First, each $\Sigma_j$ has a rank-$H$ off-diagonal Gram structure, so GMCP cannot represent near-full-rank dependence when $H \ll N$. In practice this ceiling is rarely binding (real-world dependencies are rarely full-rank) and improves scaling in $N$ (see Appendices J and Q). Second, to guarantee marginalization consistency, the mixture weights $\pi$ depend solely on the context $\mathcal{D}^{\mathrm{ctx}}$, not on the target set $\mathbf{X}^{\mathrm{tgt}}$. In Appendix P.2 we show that this is only a small restriction in practice.

## 4 Application to Irregular Multivariate Time Series

In this section we instantiate the parameter functions $\phi_\pi, \phi_\mu, \phi_\sigma, \phi_V$ and the model for the univariate marginals $F_n^{\text{marg}}$, so that GMCP can be used for IMTS forecasting.

**IMTS forecasting as a neural process.** An IMTS instance over $C$ channels is a set of observations $\{(t_m, c_m, y_m^{\text{ctx}})\}_{m=1}^M$, where $t_m \in \mathbb{R}_{\geq 0}$ is a timestamp, $c_m \in \{1, \dots, C\}$ a channel index, and $y_m^{\text{ctx}} \in \mathbb{R}$ the measured value. Channels are observed asynchronously and often only a handful of times over the whole observation window. Forecasting asks for the values at a set of future (time, channel) pairs. This is precisely the neural process setup with $\mathcal{X} = \mathbb{R}_{\geq 0} \times \{1, \dots, C\}$, and $x_n^{\text{tgt}} = (t_n, c_n)$, so that the context set $\mathcal{D}^{\text{ctx}} = \{((t_m, c_m), y_m^{\text{ctx}})\}_{m=1}^M$ is the observed history and the target set $\mathbf{X}^{\text{tgt}} = (x_1^{\text{tgt}}, \dots, x_N^{\text{tgt}})$ collects the (time, channel) pairs to be forecast, with outputs $\mathbf{y}^{\text{tgt}} \in \mathbb{R}^N$. Both sets are unstructured. The channel and time points the targets refer to, and number of targets $N$, vary per instance and are revealed only at prediction time.

**Encoder and parameter heads.** We adopt the encoder from CircuITS (Klötergens et al., 2026a), the state of the art for joint IMTS forecasting. Crucially, this encoder infers the target encodings independent of each other, and is therefore *target-separable*, a crucial property to allow for marginalization consistency. More details are in Appendix E. As an intermediate step the encoder produces a per-channel summary of the history $\tilde{\mathbf{H}} \in \mathbb{R}^{C \times D}$, which we aggregate into a global summary $\bar{\mathbf{h}}$ by linear attention with a single attention-query:

$$\mathbf{e}_n = \text{Enc}\big(x_n^{\text{tgt}}, \mathcal{D}^{\text{ctx}}\big) \in \mathbb{R}^D, \qquad \bar{\mathbf{h}} = \text{Pool}\big(\tilde{\mathbf{H}}\big) \in \mathbb{R}^D. \tag{12}$$

Since $\tilde{\mathbf{H}}$ is a function of the history alone, so is $\bar{\mathbf{h}}$, as required for the mixture weights. On top of these two representations, the parameter functions of Section 3 are 2-layer MLP heads: $\phi_\pi$ reads the global summary $\bar{\mathbf{h}}$ and passes it through a softmax, while $\phi_\mu$, $\phi_\sigma$ and $\phi_V$ read the target encoding $\mathbf{e}_n$, with a softplus on $\phi_\sigma$ that enforces positive standard deviations. GMCP and the marginal model apply two fully separated instances of the encoder, with potentially different hyperparameters.

**Marginals.** Existing marginal models for IMTS (De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022) rely on Neural ODEs, which have been shown to be ineffective in forecasting (Klötergens et al., 2024), and restrict each predictive distribution to be Gaussian. We therefore take the strongest available joint model CircuITS (Klötergens et al., 2026a) and reduce it to a univariate model by dropping all components that are designed to model dependency. What is left is the encoding $\mathbf{e}_n$ of Eq. (12) with a Deep Sigmoidal Flow (DSF) head (Huang et al., 2018; Papamakarios et al., 2021),

$$F_n^{\text{marg}}(y_n^{\text{tgt}} \mid x_n^{\text{tgt}}, \mathcal{D}^{\text{ctx}}) = T_{\theta_n}(y_n^{\text{tgt}}), \qquad \theta_n = \text{MLP}_{\text{DSF}}(\mathbf{e}_n), \tag{13}$$

where $T_{\theta_n} : \mathbb{R} \to (0, 1)$ is a strictly monotone univariate normalizing flow. The flow makes the marginals non-Gaussian while keeping their CDF in closed form, and conditioning $\theta_n$ on $\mathbf{e}_n$ alone keeps them target separable and hence marginalization consistent. We refer to the resulting model as MargFlow and refer to Appendix F for more details. While GMCP and MargFlow have encoders with the same architecture, the encoders have separated weights. Figure 4 in Appendix D illustrates how MargFlow works together with GMCP.

## 5 Experiments

Our experiments are designed to test whether GMCP captures dependence beyond Gaussian copulas and how two-stage estimation compares to jointly trained models and TACTiS-2 (Ashok et al., 2023).

### 5.1 Modeling Non-Gaussian Processes

GMCP is built to express dependence that no Gaussian copula process can represent. We demonstrate that our copula model actually achieves this by comparing it with Gaussian copulas and mixtures of those on the task of modeling non-Gaussian processes. In the *pendulum* process of Figure 1, a pendulum is held at a certain height and released from rest at an unobserved time. Its context covers the stretch in which it is held, hence the model gets to know the initial height. The fact that the release time is random results in bivariate marginals with non-Gaussian dependence and the relatively poor performance of the Gaussian Copula Neural Process (GCNP) (Markou et al., 2022) or mixtures

Table 1: **njNLL** on synthetic processes; lower is better. All copulas share the same frozen MargFlow marginals, so the differences are due to the copula alone. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**.

| | Independent Marginals | GCNP | Mix-GC | **GMCP** (ours) |
|---|---|---|---|---|
| Pendulum | $0.918 \pm 0.002$ | $0.244 \pm 0.013$ | $0.085 \pm 0.006$ | $\mathbf{-0.962 \pm 0.035}$ |
| Lévy (CPP) | $0.714 \pm 0.073$ | $-0.203 \pm 0.080$ | $-0.252 \pm 0.075$ | $\mathbf{-0.411 \pm 0.054}$ |
| Lévy (Gamma) | $2.047 \pm 0.003$ | $0.895 \pm 0.003$ | $0.850 \pm 0.004$ | $\mathbf{0.552 \pm 0.004}$ |

of those. Additionally, we evaluate the copula processes on two pure-jump Lévy processes (Sato, 1999): a compound Poisson process (CPP) with unit jumps and a Gamma process (Cont & Tankov, 2003). Because the jumps land at random times after the context ends, two targets have a joint distribution with many modes spread across the unit square, whereas every component of a mixture of Gaussian copulas sits at the center of that square (Proposition 3). Appendices K.1 and K.2 give the full specification of all three processes. All models share one frozen MargFlow, so the columns of Table 1 differ in the copula alone: none at all (independent marginals), GCNP (Markou et al., 2022) (implemented by GMCP with $K = 1$), Mix-GC and GMCP, the latter two with $K = 10$ components. The results from Table 1 provide evidence that GMCP can learn to build dependence structures beyond the range of Gaussian copulas (or mixtures of those).

### 5.2 IMTS Forecasting

**Datasets.** Our evaluation uses the four benchmark IMTS datasets: `USHCN` (Menne et al., 2016), `PhysioNet'12` (Silva et al., 2012), `MIMIC-III` (Johnson et al., 2016), and `MIMIC-IV` (Johnson et al., 2023). Crucially, the well-established preprocessing, binning, and split protocol is adopted directly from prior work (Biloš et al., 2021; Yalavarthi et al., 2025; 2026; Klötergens et al., 2026a) to provide a fair comparison. This protocol trains and evaluates each model five times on different train, validation, and test splits. Importantly, every model is evaluated on the same five splits, so the runs are paired across methods. We refer to Appendix L for more details.

**Baselines.** On the joint side we use ProFITi (Yalavarthi et al., 2025), MOSES (Yalavarthi et al., 2026), and CircuITS (Klötergens et al., 2026a). As a copula baseline we use the attentional copula of TACTiS-2. Since TACTiS-2's (Ashok et al., 2023) original implementation was infeasible for sparse IMTS, we adapt it to our setting in the way most favorable to it. We keep its attentional copula and its two-stage training protocol, supply it with the same MargFlow marginals GMCP uses, and give it the same IMTS encoder. This adaptation, denoted TACTiS-2$^\dagger$, isolates the copula module (Appendix L.7). Details on hyperparameters are given in Appendix L.3. As a second copula baseline we report Mix-GC, the ablation of GMCP that replaces the Gaussian Mixture Copula by a mixture of Gaussian copulas (Appendix P.1). To measure the effect of training marginals and copula in isolation we also add GMCP-Joint to the baselines. GMCP-Joint does not change the architecture of the compound of MargFlow and GMCP, but the two components are optimized together. We report numbers from previous work (Yalavarthi et al., 2026; Klötergens et al., 2026a) whenever appropriate.

**Estimating the marginals in isolation pays off.** Table 2 reports the marginal Negative Log-Likelihood (mNLL, defined in Eq. (56), Appendix L.5); MargFlow attains the lowest mNLL on all four datasets. The comparison that isolates the effect of the two-stage training is the one with GMCP-Joint, the very same compound of MargFlow and GMCP, optimized jointly instead of in

Table 2: Comparing **mNLL** (Marginal Negative Log-Likelihood) across datasets. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**, the second best is underlined. Table 14 adds the baselines omitted here.

| | **Model** | `USHCN` | `PhysioNet'12` | `MIMIC-III` | `MIMIC-IV` |
|---|---|---|---|---|---|
| Joint | ProFITi | $-3.324 \pm 0.206$ | $-0.016 \pm 0.085$ | $0.408 \pm 0.030$ | $0.500 \pm 0.322$ |
| | MOSES | $-3.355 \pm 0.156$ | $-0.271 \pm 0.028$ | $0.163 \pm 0.026$ | $-0.634 \pm 0.017$ |
| | CircuITS | $\underline{-3.717 \pm 0.201}$ | $\underline{-0.287 \pm 0.037}$ | $\underline{0.095 \pm 0.107}$ | $\underline{-0.731 \pm 0.063}$ |
| | GMCP-Joint | $-3.663 \pm 0.210$ | $-0.176 \pm 0.470$ | $0.108 \pm 0.238$ | $-0.677 \pm 0.044$ |
| | **MargFlow** | $\mathbf{-3.948 \pm 0.294}$ | $\mathbf{-0.479 \pm 0.021}$ | $\mathbf{-0.158 \pm 0.205}$ | $\mathbf{-0.833 \pm 0.077}$ |

Table 3: Comparing **njNLL** (Normalized Joint Negative Log-Likelihood) across datasets. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**, the second best is underlined. Table 13 adds the baselines omitted here.

| | Model | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|---|
| | MargFlow (independent) | -3.969 ± 0.305 | -0.442 ± 0.021 | -0.505 ± 0.208 | -1.819 ± 0.065 |
| Joint | ProFITi | -3.226 ± 0.225 | -0.647 ± 0.078 | -0.377 ± 0.032 | -1.777 ± 0.066 |
| Joint | MOSES | -3.357 ± 0.176 | -0.491 ± 0.041 | -0.305 ± 0.027 | -1.668 ± 0.097 |
| Joint | CircuITS | -3.789 ± 0.218 | -0.550 ± 0.013 | -0.574 ± 0.080 | -2.113 ± 0.044 |
| Joint | GMCP-Joint | -3.907 ± 0.269 | -0.489 ± 0.399 | -0.508 ± 0.309 | -2.019 ± 0.049 |
| Copula | TACTiS-2† | **-4.254 ± 0.294** | -0.732 ± 0.115 | -0.780 ± 0.282 | -2.053 ± 0.170 |
| Copula | Mix-GC | -4.078 ± 0.308 | -0.728 ± 0.018 | -0.779 ± 0.216 | -2.080 ± 0.064 |
| Copula | **GMCP** | -4.135 ± 0.304 | **-0.745 ± 0.019** | **-0.835 ± 0.218** | **-2.201 ± 0.058** |

stages, and worse on every dataset. The same holds against CircuITS, whose induced marginals are a mixture of flows of exactly the form MargFlow uses and are therefore at least as expressive (Appendix F). Against the Neural-ODE based models listed in Appendix M, Table 14 the margin is larger still, as those are restricted to Gaussian predictions.

**Comparing joint density modeling.** Table 3 reports njNLL, the objective every model is trained and tuned on. GMCP surpasses all jointly trained and univariate models on all four datasets (Table 13). Because Table 3 also reports the njNLL of MargFlow with independent marginals, it shows how much of the njNLL stems from the copula alone. All copulas in this experiment share the exact same MargFlow instances, and are therefore directly comparable in njNLL. Comparing to TACTiS-2, GMCP has a lower average njNLL on three out of four datasets. Furthermore, Table 3 indicates that GMCP's expressivity advantage over Mix-GC actually leads to better dependence modeling on real-world and not only synthetic data. The combination of GMCP and MargFlow provides a consistently better likelihood estimation, when trained in separate stages, compared to joint training. We provide the mNLL and njNLL of additional but non-competitive baselines in Appendix M.

**Quantifying that GMCP is *more valid* than TACTiS-2.** To quantify the validity gap, we draw 1000 samples from MargFlow and from the full copula setup, and measure the univariate Wasserstein-1 distance between the two sample sets (see Appendix L.6 for the definition). A copula must leave the marginals untouched, so any distance beyond sampling noise is a violation. The *Control* row is not a separate copula model: it is obtained by sampling twice from the same MargFlow instance using different random seeds. Therefore, it represents the irreducible sampling error. As shown in Table 4, GMCP is either within standard deviations of the control or very close to it. The WD of TACTiS-2, however, is between five and forty times the control, indicating that it did not learn to preserve the marginal distributions. The experiment above probes only the *univariate* marginalization consistency, which is the defining property of a copula. Due to the page limit, we refer to Appendix C for a more in-depth analysis of TACTiS-2's marginalization inconsistency.

**Sample-based metrics.** Beyond likelihood, we evaluate every model on MSE, CRPS and the Energy Score (ES). However, note that proper scoring rules agree only at the ground truth and rank *imperfect* models differently (Gneiting & Raftery, 2007). The ES in particular is a weak instrument for what a copula contributes, as its power to discriminate dependence decays with the dimension while its sensitivity to the marginal means does not (Marcotte et al., 2023). The decoupled construction makes the marginal objective a free choice, so we fit MargFlow on CRPS directly and leave the copula stage

Table 4: Wasserstein distances (in units of $10^{-3}$) for 1000 samples from MargFlow (alone) and the MargFlow + copula setup, averaged over 5 splits. The Control row quantifies the sampling error calculated based on sampling twice from MargFlow with different random seeds.

| Model | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| TACTiS-2† | 27.601 ± 14.465 | 10.775 ± 2.978 | 15.724 ± 8.570 | 2.939 ± 1.133 |
| GMCP (GM-C) | 5.083 ± 2.444 | 0.535 ± 0.131 | 0.524 ± 0.079 | 0.381 ± 0.081 |
| Control | 5.485 ± 2.541 | 0.633 ± 0.142 | 0.408 ± 0.043 | 0.274 ± 0.022 |

untouched. The resulting marginal model attains the best CRPS and MSE on three of four datasets, and GMCP on top of it the best ES on two. A jointly trained model has no analogue of this. Its single objective has to cover the dependence structure as well as the marginals. Results and details are in Appendices N and O.

## 6 Related Work

**Neural processes.** Many conditional neural processes (Garnelo et al., 2018a) only model univariate marginals and model the joint distribution by autoregressively adding the prediction to the context (Bruinsma et al., 2022; Hassan et al., 2026; Mortimer et al., 2026). While expressive, this approach sacrifices permutation equivariance and marginalization consistency. Latent neural processes (Garnelo et al., 2018b; Kim et al., 2018; Foong et al., 2020) are consistent by construction but have an intractable likelihood, while Gaussian neural processes (Bruinsma et al., 2020) have an exact likelihood but a Gaussian joint. Diffusion- and flow-based models (Dutordoir et al., 2023; Kerrigan et al., 2024) are more flexible, but provide no closed-form density for a given target set.

**Copulas in machine learning.** Gaussian copula processes (Wilson & Ghahramani, 2010) and Gaussian process conditional copulas (Hernández-Lobato et al., 2013) are Bayesian models fit to a single dataset rather than amortized over context sets. Neural models that do amortize the Gaussian copula over contexts (Salinas et al., 2019; Markou et al., 2022) are limited to linear correlations. Gaussian copulas suit copula processes because they are defined for any number of target points and are closed under marginalization. Archimedean copulas (Ling et al., 2020) share these properties but are exchangeable, so all pairs of target points have the same dependence. Vine copulas (Aas et al., 2009) and flow- or diffusion-based copulas (Kamthe et al., 2021; Huk & Damoulas, 2026) are far more flexible, but they are defined for a fixed set of variables. Implicit generative copulas (Wen & Torkkola, 2019) lack a tractable density altogether. Stationary vine processes (Bladt & McNeil, 2022; Nagler et al., 2022) do extend a vine to a process over a regular time grid, fit to a single time series rather than conditioned on a context. The attentional copula of TACTiS/TACTiS-2 (Drouin et al., 2022; Ashok et al., 2023) is autoregressive over a permutation of the target points and therefore neither marginalization consistent nor guaranteed to preserve its marginals (Table 4, Appendix C).

**Probabilistic time series forecasting.** Probabilistic forecasters for regular multivariate time series, such as DeepAR (Salinas et al., 2020), conditional normalizing flows (Rasul et al., 2021b), diffusion models (Rasul et al., 2021a; Tashiro et al., 2021), and K2VAE (Wu et al., 2025), forecast on a fixed grid of future time steps and cannot be queried at arbitrary times. They thus model a conditional joint distribution over a fixed set of variables rather than a continuous-time stochastic process, and they do not directly apply to IMTS. Most IMTS forecasters predict point estimates (Yalavarthi et al., 2024; Zhang et al., 2024; Luo et al., 2025; Li et al., 2025), and early probabilistic ones predict an independent Gaussian per target point (De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022). Among joint IMTS models, multitask Gaussian processes (Dürichen et al., 2015) are consistent but Gaussian, and ProFITi (Yalavarthi et al., 2025) is expressive but not consistent (Yalavarthi et al., 2026). MOSES (Yalavarthi et al., 2026) and CircuITS (Klötergens et al., 2026a) are consistent by construction but are not copula models, so marginals and dependence have to be learned jointly. MOSES, for instance, applies univariate flows to each component of a GMM, so with more than one component its marginals depend on all components and cannot be provided by a separate model.

## 7 Conclusion

We introduced GMCP, a conditional copula process whose dependency structure is a Gaussian Mixture Copula. Inferring the latent mixture in a target-separable way makes it marginalization consistent by construction, while letting the components keep separate means carries it beyond the elliptical dependence of Gaussian copula processes. On four IMTS benchmarks GMCP attains a better likelihood than every jointly trained model. The most interesting direction is to scale GMCP to tabular foundation models (TFMs) (Hollmann et al., 2025). These are conditional neural processes already, but predict each cell in isolation. Currently, TFMs are already used to model dependencies, by applying them autoregressively (Vetter et al., 2025), which is not marginalization consistent (Klötergens et al., 2026b). Future work could train a foundational version of GMCP on a large corpus of synthetic data that learns to extract a dependency structure via in-context learning.

## AI USE STATEMENT

In this work, we used generative AI tools as general-purpose writing and coding assistants. Specifically, they were used to polish the phrasing of text that was drafted by the authors, and to assist with routine implementation tasks such as plotting, table formatting, and refactoring of experiment scripts. AI was used to draft certain appendices, especially in order to describe the experimental set-up in meticulous detail. All AI-assisted text was reviewed and edited by the authors, and all AI-assisted code was verified by the authors and tested against the reference implementations of the baselines. We take responsibility for the final content of this work.

## ETHICS STATEMENT

This work is a methodological contribution to probabilistic forecasting of irregular multivariate time series, and is evaluated exclusively on established public benchmarks. Three of the four benchmarks (`PhysioNet'12`, `MIMIC-III`, and `MIMIC-IV`) are de-identified clinical datasets that we use under their respective data use agreements; we neither collect new data from human subjects nor attempt any re-identification. The main ethical consideration is the risk of over-trusting forecasts in high-stakes clinical settings: GMCP is evaluated retrospectively and is not a clinical decision-support tool, and any deployment in such a domain would require dataset-specific validation, calibration analysis, and human oversight beyond the scope of this work. The authors have no conflicts of interest to declare.

## REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure that our results can be reproduced. The full source code, including training and evaluation scripts, is available at: `https://anonymous.4open.science/r/Gaussian-Mixture-Copula-Processes-7BC1`. The experimental setup is documented in Appendix L: dataset statistics and preprocessing in Appendix L.1, the data splits in Appendix L.2, the hyperparameter search spaces and selected values in Appendix L.3, the training setup in Appendix L.4, and the definitions of all reported metrics in Appendix L.5 and Appendix L.6. Our adaptation of the TACTiS-2 baseline to sparse IMTS is described in Appendix L.7.

## REFERENCES


Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198, April 2009. ISSN 0167-6687. doi: 10.1016/j.insmatheco.2007.02.001.

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. TACTiS-2: Better, Faster, Simpler Attentional Copulas for Multivariate Time Series. In *The Twelfth International Conference on Learning Representations*, October 2023.

Anders Ellern Bilgrau, Poul Svante Eriksen, Jakob Gulddahl Rasmussen, Hans Erik Johnsen, Karen Dybkaer, and Martin Boegsted. GMCM: Unsupervised Clustering and Meta-Analysis Using Gaussian Mixture Copula Models. *Journal of Statistical Software*, 70:1–23, April 2016. ISSN 1548-7660. doi: 10.18637/jss.v070.i02.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural Flows: Efficient Alternative to Neural ODEs. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21325–21337. Curran Associates, Inc., 2021.

Martin Bladt and Alexander J. McNeil. Time series with infinite-order partial copula dependence. *Dependence Modeling*, 10(1):87–107, January 2022. ISSN 2300-2298. doi: 10.1515/demo-2022-0105.

Wessel Bruinsma, James Requeima, Andrew Y. K. Foong, Jonathan Gordon, and Richard E. Turner. The Gaussian Neural Process. In *Third Symposium on Advances in Approximate Bayesian Inference*, November 2020.

Wessel Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom Andersson, Anna Vaughan, Anthony Buonomo, Scott Hosking, and Richard E. Turner. Autoregressive Conditional Neural Processes. In *The Eleventh International Conference on Learning Representations*, September 2022.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1):6085, April 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-24271-9.

Rama Cont and Peter Tankov. *Financial Modelling with Jump Processes*. CRC Press, December 2003. ISBN 978-1-135-43794-7.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. TACTiS: Transformer-Attentional Copulas for Time Series. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 5447–5493. PMLR, June 2022.

Robert Dürichen, Marco A. F. Pimentel, Lei Clifton, Achim Schweikard, and David A. Clifton. Multitask Gaussian Processes for Multivariate Physiological Time-Series Analysis. *IEEE Transactions on Biomedical Engineering*, 62(1):314–322, January 2015. ISSN 1558-2531. doi: 10.1109/TBME.2014.2351376.

Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural Diffusion Processes. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 8990–9012. PMLR, July 2023.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-Learning Stationary Stochastic Process Prediction with Convolutional Neural Processes. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8284–8295. Curran Associates, Inc., 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional Neural Processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713. PMLR, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural Processes, July 2018b.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459. doi: 10.1198/016214506000001437.

Conor Hassan, Nasrulloh Satrio, Cen-You Li, Daolang Huang, Paul Chang, Yang Yang, Francesco Silvestrin, Samuel Kaski, and Luigi Acerbi. Efficient Autoregressive Inference for Transformer Probabilistic Models. *International Conference on Learning Representations*, 2026:87254–87292, April 2026.

José Miguel Hernández-Lobato, James R Lloyd, and Daniel Hernández-Lobato. Gaussian Process Conditional Copulas with Applications to Financial Time Series. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural Autoregressive Flows. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2078–2087. PMLR, July 2018.

David Huk and Theodoros Damoulas. Diffusion and Flow-based Copulas: Forgetting and Remembering Dependencies. *International Conference on Learning Representations*, 2026:39167–39208, April 2026.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035, May 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.35.

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*, 10(1):1, January 2023. ISSN 2052-4463. doi: 10.1038/s41597-022-01899-x.

Sanket Kamthe, Samuel Assefa, and Marc Deisenroth. Copula Flows for Synthetic Data Generation, January 2021.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2Vec: Learning a Vector Representation of Time, July 2019. URL `https://arxiv.org/abs/1907.05321`.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional Flow Matching. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 3934–3942. PMLR, April 2024.

Mohamad A. Khaled and Robert Kohn. On approximating copulas by finite mixtures, February 2023.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive Neural Processes. In *International Conference on Learning Representations*, September 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017.

Christian Klötergens, Vijaya Krishna Yalavarthi, Randolf Scholz, Maximilian Stubbemann, Stefan Born, and Lars Schmidt-Thieme. Physiome-ODE: A Benchmark for Irregularly Sampled Multivariate Time-Series Forecasting Based on Biological ODEs. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Christian Klötergens, Vijaya Krishna Yalavarthi, and Lars Schmidt-Thieme. Probabilistic circuits for irregular multivariate time series forecasting. *arXiv preprint arXiv:2604.27814*, 2026a.

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

Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. HyperIMTS: Hypergraph Neural Network for Irregular Multivariate Time Series Forecasting. In *Forty-Second International Conference on Machine Learning*, June 2025.

Chun Kai Ling, Fei Fang, and J. Zico Kolter. Deep Archimedean Copulas. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1535–1545. Curran Associates, Inc., 2020.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019.

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-Patch: Hierarchical Patch GNN for Irregular Multivariate Time Series. In *Forty-Second International Conference on Machine Learning*, June 2025.

Matthieu Marbac, Christophe Biernacki, and Vincent Vandewalle. Model-based clustering of Gaussian copulas for mixed data. *Communications in Statistics - Theory and Methods*, 46(23):11635–11656, December 2017. ISSN 0361-0926. doi: 10.1080/03610926.2016.1277753.

Étienne Marcotte, Valentina Zantedeschi, Alexandre Drouin, and Nicolas Chapados. Regions of Reliability in the Evaluation of Multivariate Probabilistic Forecasts. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 23958–24004. PMLR, July 2023.

Stratis Markou, James Requeima, Wessel Bruinsma, Anna Vaughan, and Richard E. Turner. Practical Conditional Neural Process Via Tractable Dependent Predictions. In *International Conference on Learning Representations*, 2022.

M. J. Menne, Jr Williams, and R. S. Vose. Long-Term Daily and Monthly Climate Records from Stations Across the Contiguous United States (U.S. Historical Climatology Network). Technical Report osti:1394920; cdiac:NDP-019; doi:10.3334/CDIAC/CLI.NDP019, Environmental System Science Data Infrastructure for a Virtual Ecosystem (ESS-DIVE) (United States); CDIAC, January 2016.

Philip Mortimer, Cristiana Diaconu, Tommy Rochussen, Bruno Kacper Mlodozeniec, and Richard E. Turner. Incremental Transformer Neural Processes. In *Forty-Third International Conference on Machine Learning*, June 2026.

Thomas Nagler, Daniel Krüger, and Aleksey Min. Stationary vine copula models for multivariate time series. *Journal of Econometrics*, 227(2):305–324, April 2022. ISSN 0304-4076. doi: 10.1016/j.jeconom.2021.11.015.

Roger B. Nelsen. *An Introduction to Copulas*. Springer Series in Statistics. Springer, New York, NY, 2006. ISBN 978-0-387-28659-4. doi: 10.1007/0-387-28678-0.

Tung Nguyen and Aditya Grover. Transformer Neural Processes: Uncertainty-Aware Meta Learning Via Sequence Modeling. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16569–16594. PMLR, June 2022.

Bernt Øksendal. *Stochastic Differential Equations*. Universitext. Springer, Berlin, Heidelberg, 2003. ISBN 978-3-540-04758-2 978-3-642-14394-6. doi: 10.1007/978-3-642-14394-6.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. ISSN 1533-7928.

William H. Press. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, September 2007. ISBN 978-0-521-88068-8.

Vaibhav Rajan and Sakyajit Bhattacharya. Dependency clustering of mixed data with Gaussian mixture copulas. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 1967–1973, New York, New York, USA, July 2016. AAAI Press. ISBN 978-1-57735-770-4.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8857–8868. PMLR, July 2021a.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows, January 2021b.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, July 2020. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2019.07.001.

Ken-Iti Sato. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press, 1999. ISBN 978-0-521-55302-5.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling Irregular Time Series with Continuous Recurrent Units. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 19388–19405. PMLR, June 2022.

Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of ICU patients: The PhysioNet/Computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pp. 245–248, September 2012.

M. Sklar. Fonctions de répartition à N dimensions et leurs marges. *Annales de l'ISUP*, VIII(3):229, 1959.

Alan Smillie. *New copula models in quantitative finance*. PhD thesis, Tanaka Business Centre, Imperial College London, 2008.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816. Curran Associates, Inc., 2021.

Ashutosh Tewari. On the Estimation of Gaussian Mixture Copula Models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 34090–34104. PMLR, July 2023.

Ashutosh Tewari, Michael J. Giering, and Arvind Raghunathan. Parametric Characterization of Multimodal Distributions with Non-gaussian Modes. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pp. 286–292, December 2011. doi: 10.1109/ICDMW.2011.135.

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

Julius Vetter, Manuel Glöckler, Daniel Gedon, and Jakob H. Macke. Effortless, simulation-efficient bayesian inference using tabular foundation models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/770cabd044c4eacb6dc5924d9a686dce-Abstract-Conference.html.

Ruofeng Wen and Kari Torkkola. Deep Generative Quantile-Copula Models for Probabilistic Forecasting, July 2019.

Andrew G Wilson and Zoubin Ghahramani. Copula Processes. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. $K^2$VAE: A Koopman-Kalman Enhanced Variational AutoEncoder for Probabilistic Time Series Forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 67562–67583. PMLR, October 2025.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. GraFITi: Graphs for Forecasting Irregularly Sampled Time Series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16255–16263, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i15.29560.

Vijaya Krishna Yalavarthi, Randolf Scholz, Stefan Born, and Lars Schmidt-Thieme. Probabilistic Forecasting of Irregularly Sampled Time Series with Missing Values via Conditional Normalizing Flows. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20):21877–21885, April 2025. ISSN 2374-3468. doi: 10.1609/aaai.v39i20.35494.

Vijaya Krishna Yalavarthi, Randolf Scholz, Christian Klötergens, Kiran Madhusudhanan, Stefan Born, and Lars Schmidt-Thieme. Reliable Probabilistic Forecasting of Irregular Time Series through Marginalization-Consistent Flows. *International Conference on Learning Representations*, 2026:85403–85427, April 2026.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular Multivariate Time Series Forecasting: A Transformable Patching Graph Neural Networks Approach. In *Forty-First International Conference on Machine Learning*, June 2024.

## APPENDIX: CONTENTS


**A Expressivity: GM-C versus Mixtures of Gaussian Copulas** **16**

**B Marginalization Consistency of GMCP** **18**

**C On TACTiS-2's Marginalization Consistency** **20**

**D Model Architecture** **22**

**E The Encoder** **23**

**F MargFlow: A Conditional Neural Process with Normalizing-Flow Marginals** **25**

**G Numerical Inversion of the GMM Marginal CDF** **27**

**H Derivation of the Analytic iCDF Gradients** **28**

**I Numerical Validation of the iCDF Solver and Its Gradients** **31**

**J Computational Complexity** **32**

**K Synthetic Processes** **33**

**L Experimental Details** **36**

**M Complete Likelihood Results** **40**

**N Sampling Results** **41**

**O Training MargFlow on CRPS** **44**

**P Ablations** **45**

**Q Sensitivity Analysis** **47**

**R Learning Curves** **48**

## A EXPRESSIVITY: GM-C VERSUS MIXTURES OF GAUSSIAN COPULAS

Section 2 contrasts a Gaussian Mixture Copula (GM-C) with a *mixture of Gaussian copulas* (Mix-GC) and states that the latter can never deplete mass at the center of the unit cube, regardless of the number of components. We prove that statement here as an obstruction that holds uniformly in $K$, and then read it off the two toy distributions of Figure 2.

**Setup and notation.** Both constructions use $K$ mixture weights $\pi_j$ and $K$ positive-definite matrices. They differ in where the copula transformation Eq. (1) is applied. A Mix-GC combines $K$ Gaussian copulas *after* the transformation,

$$\mathbf{c}_{\mathrm{Mix}}(\mathbf{u}) \;=\; \sum_{j=1}^{K} \pi_j \, \mathbf{c}_{R_j}(\mathbf{u}), \qquad \mathbf{c}_{R_j}(\mathbf{u}) \;=\; \frac{\mathcal{N}\big(\Phi^{-1}(\mathbf{u}); 0, R_j\big)}{\prod_{n=1}^{N} \varphi\big(\Phi^{-1}(u_n)\big)}, \tag{14}$$

where $\Phi$ denotes the standard normal CDF, $\Phi^{-1}$ its inverse (the standard normal quantile function) and $\varphi$ the standard normal density, all applied elementwise. Each $R_j$ is a correlation matrix. A GM-C combines the same components *before* the transformation. It applies Eq. (1) to the joint mixture density $f^{\mathrm{GMM}}(\mathbf{z}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j)$ and uses that mixture's own marginal CDFs $F_n^{\mathrm{GMM}}$,

$$\mathbf{c}_{\mathrm{GM}}(\mathbf{u}) \;=\; \frac{f^{\mathrm{GMM}}\big(F^{\mathrm{GMM},-1}(\mathbf{u})\big)}{\prod_{n=1}^{N} f_n^{\mathrm{GMM}}\big(F_n^{\mathrm{GMM},-1}(u_n)\big)}, \qquad F^{\mathrm{GMM},-1}(\mathbf{u}) \;=\; \big(F_n^{\mathrm{GMM},-1}(u_n)\big)_{n=1}^{N}. \tag{15}$$

Write $\mathbf{z} = \Phi^{-1}(\mathbf{u})$ for the latent point that $\mathbf{u}$ is mapped to. Then Eq. (14) shows that the latent density a Mix-GC places on $\mathbb{R}^N$ is $\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{z}; 0, R_j)$, a Gaussian mixture whose components are all *pinned to the origin* and whose marginals are all standard normal. The free means $\mu_j$ and per-component scales of Eq. (15) have no counterpart.

**An obstruction at the center.** Recall what the *value* of a copula density means. The independence copula has density $\mathbf{c} \equiv 1$, so by Eq. (1) the number $\mathbf{c}(\mathbf{u})$ is the likelihood ratio of the joint against independence at $\mathbf{u}$. Above 1, the dependence structure concentrates mass relative to independent components. Below 1, it depletes mass. A mixture of Gaussian copulas can only concentrate mass at the center of the unit cube. It can never deplete it, whatever $K$ it is given.

**Proposition 3.** *Let $\mathbf{c}_{\mathrm{Mix}}$ be any mixture of Gaussian copulas as in Eq.* (14)*, with arbitrary $K \in \mathbb{N}$, weights $\pi_j \geq 0$ summing to* 1*, and correlation matrices $R_j$. Let $\mathbf{u}^\star = (1/2, \ldots, 1/2)$ denote the center of the unit cube. Then*

*(i)* $\mathbf{c}_{\mathrm{Mix}}(\mathbf{u}^\star) \;=\; \sum_{j=1}^{K} \pi_j \, |R_j|^{-1/2} \;\geq\; 1$*, with equality if and only if $R_j = I$ for every $j$ with $\pi_j > 0$;*

*(ii)* $\mathbf{c}_{\mathrm{Mix}}$ *is radially symmetric, i.e.* $\mathbf{c}_{\mathrm{Mix}}(\mathbf{u}) = \mathbf{c}_{\mathrm{Mix}}(\mathbf{1} - \mathbf{u})$ *for all* $\mathbf{u} \in (0,1)^N$ *(Khaled & Kohn, 2023).*

*No mixture of Gaussian copulas therefore represents a copula whose density at the center falls below* 1*, and none represents a copula that is not radially symmetric. Both statements hold for every $K$, so adding components cannot lift the restriction.*

*Proof.* The standard normal CDF $\Phi$ has median 0, so $\Phi^{-1}(1/2) = 0$ and evaluating Eq. (14) at $\mathbf{u}^\star$ gives $\mathbf{z} = 0$. Both the numerator and the denominator of $\mathbf{c}_{R_j}$ are then Gaussian densities evaluated at their own mean,

$$\mathbf{c}_{R_j}(\mathbf{u}^\star) \;=\; \frac{\mathcal{N}(0; 0, R_j)}{\prod_{n=1}^{N} \varphi(0)} \;=\; \frac{(2\pi)^{-N/2} |R_j|^{-1/2}}{(2\pi)^{-N/2}} \;=\; |R_j|^{-1/2}. \tag{16}$$

By Hadamard's inequality, $|R_j| \leq \prod_n (R_j)_{nn} = 1$ for any correlation matrix, with equality if and only if $R_j$ is diagonal, hence equal to $I$. Therefore $|R_j|^{-1/2} \geq 1$ for every $j$, and hence so is the convex combination, which proves (i). For (ii), the symmetry $\Phi(-z) = 1 - \Phi(z)$ of the standard normal CDF gives $\Phi^{-1}(\mathbf{1} - \mathbf{u}) = -\Phi^{-1}(\mathbf{u})$, and both $\mathcal{N}(\,\cdot\,; 0, R_j)$ and $\varphi$ are even, so every summand in Eq. (14) is invariant under $\mathbf{u} \mapsto \mathbf{1} - \mathbf{u}$ $(Khaled\&Kohn,\ 2023)$. □

For $N = 2$ the bound is explicit. A bivariate Gaussian copula with correlation $\rho$ has $|R| = 1 - \rho^2$, so

$$\mathbf{c}_{\mathrm{Mix}}(1/2, 1/2) \;=\; \sum_{j=1}^{K} \frac{\pi_j}{\sqrt{1-\rho_j^2}} \;\geq\; 1, \tag{17}$$

which equals 1 only in the independent case $\rho_j = 0$ and grows without bound as any $|\rho_j| \to 1$. Whatever correlations and weights are chosen, and however many components are used, the center of the unit square is at least as dense as it would be under independence.

The bound excludes target densities that have a *hole* at their center. Suppose a density $f$ on $\mathbb{R}^N$ is symmetric about the origin in each coordinate, so that $F_n(0) = 1/2$ and the origin is mapped to $\mathbf{u}^\star$. Suppose further that $f(0) \approx 0$ while the marginals $f_n(0)$ remain strictly positive. Evaluating Eq. (1) at $\mathbf{y} = 0$,

$$\mathbf{c}(\mathbf{u}^\star) \;=\; \frac{f(0)}{\prod_{n=1}^{N} f_n(0)} \;\approx\; 0, \tag{18}$$

which falls below the bound of Proposition 3 (i) by the full margin. The ring of Figure 2 is such a density. It has no mass at the origin, yet each of its marginals is strictly positive there, because the ring's mass projects onto the whole interval in every coordinate. Its copula lies outside the Mix-GC class, and no number of components brings it inside.

A GM-C obeys neither constraint of Proposition 3. Placing the component means $\mu_j$ away from the origin makes the marginals $F_n^{\mathrm{GMM}}$ multimodal. The transformation $F^{\mathrm{GMM},-1}$ in Eq. (15) then stretches the low-density valleys between the modes over a wide range of $\mathbf{u}$ and compresses the modes, so the resulting copula density comes arbitrarily close to 0 at $\mathbf{u}^\star$ and is in general not symmetric about it.

**The X distribution.** Figure 2(a) is the case a Mix-GC is built for. The X density superposes two elongated Gaussian ridges with correlations $\approx +\rho$ and $\approx -\rho$ that share the origin as their common center, so both components already sit where Eq. (14) forces them to sit. Its marginals are unimodal. Its copula density peaks at the center and is radially symmetric, so it satisfies both conclusions of Proposition 3, and Mix-GC and GM-C are visually indistinguishable from the ground truth. On dependence structures of this type the cheaper Mix-GC parameterization is an adequate model, which is also why the two variants are close on some of the real datasets in Table 3.

**The circle distribution.** Figure 2(b) is the instance of Eq. (18) discussed above, and Proposition 3 already tells us the ring is unreachable. The figure shows *how* a Mix-GC fails when asked to fit it anyway. It cannot carve out the hole, and Proposition 3 (i) penalizes the surplus density that any $R_j \neq I$ adds at the center, so the fitted model drives its components towards the equality case $R_j \approx I$, that is, towards near-independence. Under near-independence the joint reduces to the product of the marginals. Both marginals are bimodal with modes near $\pm 2.8$, so the product has four modes at the corners $(\pm 2.8, \pm 2.8)$. That is the square-with-corner-modes pattern of the Mix-GC panel. The model reproduces the correct marginals, as any copula construction must, and then has nowhere to place the mass except the four locations the marginals alone dictate. The ring's angular coupling is inexpressible in the class.

The GM-C recovers the ring. Its latent components are not restricted to share the same center. The transformation $F^{\mathrm{GMM},-1}$ then maps the center of the unit cube into the low-density valley between the modes and opens the hole. The residual squareness is a finite-$K$ effect.

## B MARGINALIZATION CONSISTENCY OF GMCP

We prove here that GMCP is marginalization consistent by construction. The argument reuses the framework of separable flows and mixtures of separable flows introduced by Yalavarthi et al. (2026). We first restate the two lemmas of that paper that we will invoke, adapted to the notation of the present work, and then apply them to GMCP.

We adopt the notation of Section 3, in which $\mathcal{D}^{\mathrm{ctx}}$ denotes the context set, $\mathbf{X}^{\mathrm{tgt}} = (x_1^{\mathrm{tgt}}, \ldots, x_N^{\mathrm{tgt}})$ the target set, $x_n^{\mathrm{tgt}}$ the $n$-th target input, $F_n^{\mathrm{marg}}$ the marginal CDF at target point $n$, and $F_n^{\mathrm{GMM}}$ the $n$-th one-dimensional marginal of the latent Gaussian mixture density $f^{\mathrm{GMM}}$. A conditional density $\hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$ on $\mathbb{R}^N$ is *marginalization consistent* if it satisfies Eq. (4) for every $k \in \{1, \ldots, N\}$.

### B.1 CITED LEMMAS FROM YALAVARTHI ET AL. (2026)

**Lemma 4** (Separable flows preserve marginalization consistency; Yalavarthi et al., 2026, Lemma 3.1)**.** *Let $T(\cdot \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) : \mathbb{R}^N \to \mathbb{R}^N$ be a conditional flow that is separable, i.e. of the form*

$$T(\mathbf{z} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \;=\; \big(\tau(z_1 \mid x_1^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}), \ldots, \tau(z_N \mid x_N^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})\big) \tag{19}$$

*for some univariate function $\tau$ that is invertible in its first argument. If the base density $\hat{p}_Z(\mathbf{z} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$ is marginalization consistent, then the pushforward density*

$$\hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \;=\; \hat{p}_Z\big(T^{-1}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \,\big|\, \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}\big) \cdot \left|\det \frac{\partial T^{-1}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})}{\partial \mathbf{y}^{\mathrm{tgt}}}\right| \tag{20}$$

*is also marginalization consistent.*

**Lemma 5** (Mixtures with target-independent weights preserve marginalization consistency; Yalavarthi et al., 2026, Lemma 3.2)**.** *Let $\hat{p}_1, \ldots, \hat{p}_D$ be conditional densities that are each marginalization consistent, and let $w$ be a weight function mapping a context set to the simplex $\Delta^D$, i.e. depending only on $\mathcal{D}^{\mathrm{ctx}}$. Then the mixture*

$$\hat{p}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \;=\; \sum_{d=1}^{D} w_d(\mathcal{D}^{\mathrm{ctx}})\, \hat{p}_d(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \tag{21}$$

*is marginalization consistent.*

For completeness, we recall the proof idea of Lemma 4: separability makes the Jacobian of $T^{-1}$ diagonal, so its determinant factors across coordinates, the integral over $y_k^{\mathrm{tgt}}$ collapses to an integral over $z_k$ via the change-of-variables theorem, and marginalization consistency of $\hat{p}_Z$ then yields the claim. Lemma 5 follows by interchanging the integral over $y_k^{\mathrm{tgt}}$ with the finite sum over mixture components, which is permitted because $w_d$ does not depend on $\mathbf{X}^{\mathrm{tgt}}$. Full proofs are given in Appendices A.1 and A.2 of Yalavarthi et al. (2026).

### B.2 MAIN RESULT

We restate the result of Section 3 and prove it.

**Theorem** (Theorem 1, restated)**.** *GMCP, paired with any model that models univariate marginals independently, is marginalization consistent. Hence, for every context set $\mathcal{D}^{\mathrm{ctx}}$ and every target set $\mathbf{X}^{\mathrm{tgt}}$, the predicted joint density satisfies marginalization consistency as defined in Eq.* (4)*.*

*Proof of Theorem 1.* The proof proceeds in three steps.

**Step 1: GMCP is a target-separable transformation of a latent base density.** The generative process of GMCP maps a latent vector $\mathbf{z}$, drawn from the latent Gaussian mixture base density $f^{\mathrm{GMM}}(\mathbf{z} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$, to the output domain via the per-coordinate transformation

$$y_n^{\mathrm{tgt}} \;=\; F_n^{\mathrm{marg},-1}\big(F_n^{\mathrm{GMM}}(z_n)\big). \tag{22}$$

Both $F_n^{\mathrm{marg}}$ and $F_n^{\mathrm{GMM}}$ are parameterized by quantities that, by target separability, depend only on $\mathcal{D}^{\mathrm{ctx}}$ and on the single target input $x_n^{\mathrm{tgt}}$. Defining $\tau(z_n \mid x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \coloneqq F_n^{\mathrm{marg},-1}(F_n^{\mathrm{GMM}}(z_n))$, the joint transformation has the separable form required by Lemma 4. By that lemma, marginalization consistency of GMCP reduces to marginalization consistency of the latent base density $f^{\mathrm{GMM}}$.

**Step 2: Each Gaussian component of the latent mixture is marginalization consistent.** The latent base density is the Gaussian mixture

$$f^{\mathrm{GMM}}(\mathbf{z} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \;=\; \sum_{j=1}^{K} \pi_j \, \mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j). \tag{23}$$

As established in Section 3, the entry $[\mu_j]_n$ depends only on $x_n^{\mathrm{tgt}}$ and $\mathcal{D}^{\mathrm{ctx}}$ (see Eq. (6)), and the entry $\Sigma_{j,nm}$ only on $x_n^{\mathrm{tgt}}$, $x_m^{\mathrm{tgt}}$ and $\mathcal{D}^{\mathrm{ctx}}$ (see Eqs. (6) to (8)). Removing a subset of target points from $\mathbf{X}^{\mathrm{tgt}}$ therefore deletes the corresponding entries of $\mu_j$ and the corresponding rows and columns of $\Sigma_j$, leaving the remaining entries unchanged. This coincides with the standard marginalization rule for a multivariate Gaussian, so each component $\mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j)$ is marginalization consistent in the sense of Eq. (4).

**Step 3: The mixture weights are target-independent, so the latent GMM is marginalization consistent.** The mixture weights $\pi = (\pi_1, \ldots, \pi_K)$ are inferred from the context set $\mathcal{D}^{\mathrm{ctx}}$ alone and are uninformed of the target set $\mathbf{X}^{\mathrm{tgt}}$. Together with Step 2, the two hypotheses of Lemma 5 are satisfied, so $f^{\mathrm{GMM}}$ is marginalization consistent.

Combining Steps 1 to 3, the latent base density of GMCP is marginalization consistent and is pushed forward to the output domain by a target-separable transformation. Lemma 4 then yields that the full joint density predicted by GMCP is marginalization consistent. □

The IMTS instantiation of Section 4 satisfies the hypotheses. The CircuITS encoder is target separable, and MargFlow conditions its flow parameters on $\mathbf{e}_n$ alone (Appendix F).

Step 3 is the only place where GMCP gives up a degree of freedom it could otherwise have taken. Appendix P.2 ablates that step, letting the mixture weights read the target set, and reports what the guarantee costs in likelihood.

## C On TACTiS-2's Marginalization Consistency

The Wasserstein gap of Table 4 probes only the *univariate* aspect of marginalization consistency, whether the univariate marginals of the model's samples still match MargFlow's. We put this in the main paper because uniform univariate marginals are the defining property of a copula, and any deviation already violates the Sklar decomposition the model is built on. The full form is stronger. It asks that the model produce the *same* marginal on *any* subset of dimensions, regardless of which other dimensions are requested alongside. GMCP satisfies that condition by construction (Appendix B). TACTiS-2's attentional copula guarantees neither the univariate nor the multivariate version, since every target subset induces a different autoregressive factorization and attention conditioning. Below we isolate the multivariate failure on a controlled 3-d toy distribution. TACTiS-2's bivariate marginal requested directly disagrees visibly with the same bivariate marginal obtained by marginalizing its 3-d joint, while GMCP's two routes coincide up to sampling noise.

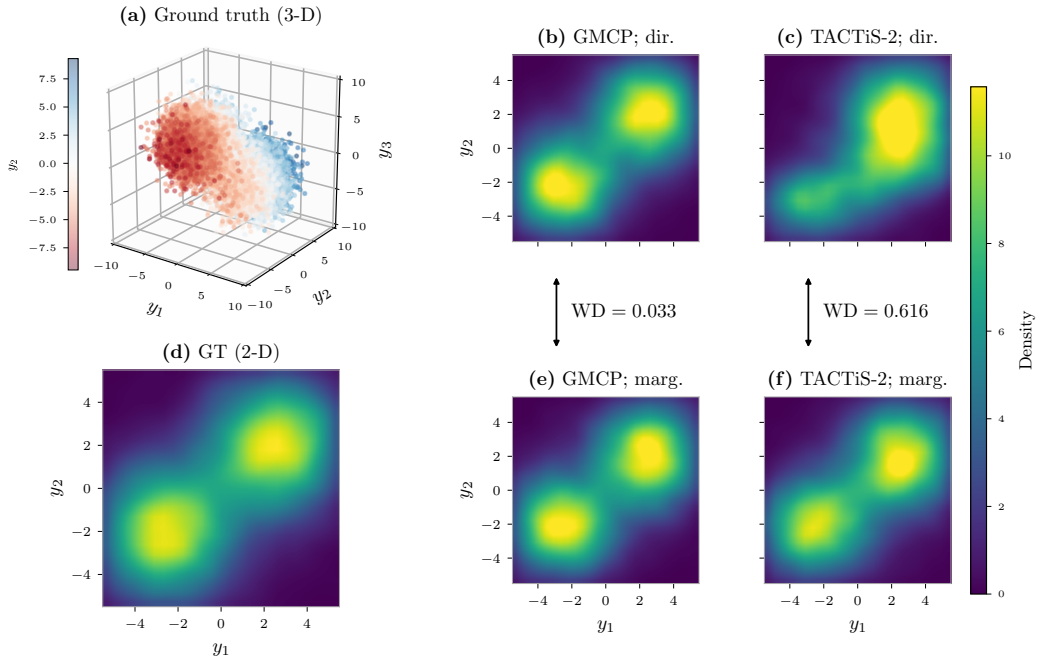


Figure 3: The Attentional Copula fails to be marginalization consistent. (a) shows the ground truth 3-d distribution. Panels (b) and (e) show GMCP's direct and marginalized views of $p(y_1, y_2)$, which match by construction. Panels (c) and (f) show TACTiS-2's direct and marginalized views, which disagree visibly; the arrows report the corresponding WD.

**Setup.** We construct a 3-dimensional toy distribution with two symmetric Gaussian clusters indexed by a quasi-discrete cluster variable $y_2$. Cluster A is centered at $(y_1, y_2, y_3) \approx (-2.5, -2, +2.5)$, and cluster B at $(y_1, y_2, y_3) \approx (+2.5, +2, -2.5)$. Knowing $y_2$ localizes $y_1$, so the true 2-d marginal $p(y_1, y_2)$ is unimodal per cluster, while the 2-d marginal $p(y_1, y_3)$ is a strongly bimodal anti-diagonal mixture. We train both models on the full 3-d distribution. Both models share the same pre-trained 1-d marginals $p(y_i)$, so the only component that differs between them is the copula. Any disagreement in the predicted joints is therefore attributable to the copula alone.

For each model, we obtain the 2-d marginals $p(y_1, y_2)$ by sampling in two different ways:

1. *Direct:* request only the dimensions $(y_1, y_2)$, with the third dimension absent from the target set.
2. *Marginalized:* request the full 3-d distribution $(y_1, y_2, y_3)$ and marginalize $y_3$ out.

A marginalization consistent copula must produce the same distribution under both routes. We quantify the gap with the Wasserstein distance (WD) between the two sample sets, indicated by the arrows between the two rows of Figure 3.

**Why TACTiS-2 fails.** The attentional copula factorizes the joint as an autoregressive product of conditional CDFs over the target dimensions, with attention conditioned on the specific index set $S$ being modeled,

$$\hat{p}(\mathbf{y}_S) \;=\; \prod_{i \in S} \hat{p}\big(y_i \,\big|\, y_{<i}, S\big). \tag{24}$$

Both the autoregressive ordering and the attention weights depend on $S$, so each target set instantiates a distinct factorization of the same network. During training on the full 3-d distribution the model only ever observes $S = \{1, 2, 3\}$, so the only factors that receive a gradient signal are those induced by that target set, namely $\hat{p}(y_3)$, $\hat{p}(y_2 \mid y_3)$, and $\hat{p}(y_1 \mid y_2, y_3)$. Evaluating the bivariate marginal under the direct route asks the same network for a different set of factors, $\hat{p}(y_2)$ and $\hat{p}(y_1 \mid y_2)$, which are never instantiated during training. The attentional copula is free to map this out-of-distribution conditioning to an arbitrary distribution, and no architectural mechanism ties its output back to the marginal of the trained 3-d joint.

**Multivariate consistency on the benchmark datasets.** The experiment above isolates the failure on a toy distribution. We now check that it also occurs on the four IMTS datasets, again comparing a direct and a marginalized route, but now on the multivariate marginal $\hat{p}(\mathbf{y}^{\mathrm{tgt}}_{-k} \mid \mathbf{X}^{\mathrm{tgt}}_{-k}, \mathcal{D}^{\mathrm{ctx}})$ of Eq. (4) instead of a univariate one. For each test instance we remove one target point $k$ from the target set. The *direct* route samples the model on the reduced target set $\mathbf{X}^{\mathrm{tgt}}_{-k}$. The *marginalized* route samples it on the full target set $\mathbf{X}^{\mathrm{tgt}}$ and discards the $k$-th coordinate. Since the two sample sets live on $\mathbb{R}^{N-1}$, we compare them with the energy distance, which for distributions $P$ and $Q$ with independent draws $\mathbf{a}, \mathbf{a}' \sim P$ and $\mathbf{b}, \mathbf{b}' \sim Q$ is

$$D_E(P, Q) \;=\; 2\,\mathbb{E}\,\|\mathbf{a} - \mathbf{b}\| \;-\; \mathbb{E}\,\|\mathbf{a} - \mathbf{a}'\| \;-\; \mathbb{E}\,\|\mathbf{b} - \mathbf{b}'\|, \tag{25}$$

estimated with sample averages. As in Table 4, a finite sample never yields a distance of exactly zero, so we normalize by a control: the energy distance between two sample sets of the direct route drawn under different random seeds. The resulting ratio

$$\rho \;=\; \frac{D_E\big(\text{marginalized, direct}\big)}{D_E\big(\text{direct (seed 1), direct (seed 2)}\big)} \tag{26}$$

expresses the inconsistency in units of the sampling error. A marginalization consistent model has $\rho \approx 1$, since removing a target point then changes its prediction no more than reseeding does. Table 5 confirms that GMCP stays close to 1 on all datasets, whereas TACTiS-2 exceeds the sampling error by a factor of 11 to 70.

Table 5: Multivariate marginalization consistency. Ratio $\rho$ of the energy distance between the marginalized and the direct route, with one target point removed, to the energy distance between two seeds of the direct route. $\rho \approx 1$ means the inconsistency is at the level of the sampling error. †Our adaptation of TACTiS-2 to sparse IMTS (Appendix L.7).

| **Model** | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| TACTiS-2† | 28.963 | 53.588 | 70.500 | 11.101 |
| GMCP (GM-C) | 1.176 | 1.035 | 1.069 | 1.060 |

## D Model Architecture

Figure 4 shows how MargFlow (Appendix F) and GMCP (Section 3) work together to model joint distributions of IMTS forecasting targets. The encoder of Appendix E supplies the target encodings $\mathbf{e}_n$ and the context summary $\bar{\mathbf{h}}$ to both.

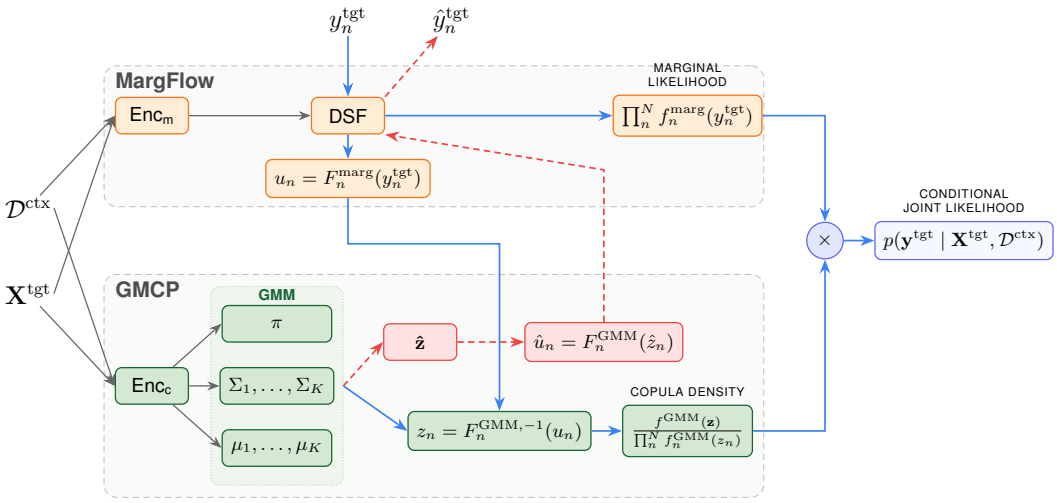


Figure 4: GMCP architecture, instantiated for IMTS with MargFlow as the marginal model. $\rightarrow$ is used to indicate the computation of the joint likelihood. In contrast, we use $\dashrightarrow$ to indicate the sampling procedure. $\rightarrow$ represents the conditioning via the encoder, which is performed during likelihood estimation and sampling.

## E THE ENCODER

Both GMCP and MargFlow draw their target encodings $\mathbf{e}_n$ and the global context summary $\bar{\mathbf{h}}$ from the encoder of CircuITS (Klötergens et al., 2026a), which we adopt unchanged. We restate it here so that the present paper is self-contained, and because the property it supplies, *target separability*, is the hypothesis on which Theorem 1 rests. The single addition we make, the pooling step of Eq. (31), is marked as such.

Throughout, $C$ is the number of channels, $D$ the latent dimension, $M$ the number of context observations, and $\mathrm{MHA}(Q, K, V)$ a multi-head attention layer (Vaswani et al., 2017) with queries $Q$, keys $K$ and values $V$. We write $\psi(\cdot)$ for a sinusoidal time embedding (Kazemi et al., 2019) and $[\cdot \,\|\, \cdot]$ for concatenation. To avoid collision with the parameter functions $\phi_\pi, \phi_\mu, \phi_\sigma, \phi_V$ of Section 3, the time embedding is denoted $\psi$ here rather than $\phi$.

**Step 1: Observation embeddings.** The context set $\mathcal{D}^{\mathrm{ctx}} = \{((t_m, c_m), y_m^{\mathrm{ctx}})\}_{m=1}^{M}$ is an unordered collection of triplets, each consisting of a continuous timestamp, a channel index and a scalar value. Every triplet is embedded independently,

$$\mathbf{h}_m \;=\; \big[\psi(t_m) \,\|\, \mathbf{E}^{\mathrm{chan}}_{c_m} \,\|\, y_m^{\mathrm{ctx}}\big] \;\in\; \mathbb{R}^D, \tag{27}$$

where $\mathbf{E}^{\mathrm{chan}} \in \mathbb{R}^{C \times D'}$ is a learnable channel-embedding matrix and the three parts are sized so that the concatenation has dimension $D$. Stacking the $M$ embeddings gives $\mathbf{H}^{\mathrm{obs}} \in \mathbb{R}^{M \times D}$. No positional information enters beyond $\psi(t_m)$, so the representation is permutation invariant in the observations, as the set-valued nature of an IMTS requires.

**Step 2: Intra-channel aggregation.** A channel may carry anywhere between zero and several hundred observations, so the per-channel summaries are formed by attention rather than by a fixed-size reduction. A learnable prototype matrix $\mathbf{E}^{\mathrm{proto}} \in \mathbb{R}^{C \times D}$, holding one row per channel, supplies the queries of a masked cross-attention layer over the observation embeddings,

$$\mathbf{H}^{\mathrm{chan}} \;=\; \mathrm{MHA}\big(\mathbf{E}^{\mathrm{proto}}, \mathbf{H}^{\mathrm{obs}}, \mathbf{H}^{\mathrm{obs}}; \mathbf{M}\big) \;\in\; \mathbb{R}^{C \times D}, \tag{28}$$

with a binary mask $\mathbf{M}$ whose entry $\mathbf{M}_{c,m}$ is 1 exactly when observation $m$ belongs to channel $c$, and 0 otherwise. The mask makes row $c$ of $\mathbf{H}^{\mathrm{chan}}$ a summary of the observations of channel $c$ alone. The channel structure of the IMTS thus stays present in the latent state instead of being flattened into a single set of triplets, which is why we prefer this encoder to a generic set transformer in the sparse regime.

**Step 3: Inter-channel interaction.** Channels are not independent, and a channel observed only a handful of times, or not at all, must be able to borrow information from the others. A residual self-attention layer over the channel summaries provides that,

$$\tilde{\mathbf{H}} \;=\; \mathbf{H}^{\mathrm{chan}} + \mathrm{MHA}\big(\mathbf{H}^{\mathrm{chan}}, \mathbf{H}^{\mathrm{chan}}, \mathbf{H}^{\mathrm{chan}}\big) \;\in\; \mathbb{R}^{C \times D}. \tag{29}$$

If channel $c$ is entirely unobserved its mask row in Eq. (28) is empty, and $\tilde{\mathbf{h}}_c$ is then determined by its prototype together with its interactions with the remaining channels.

**Step 4: Target encodings.** A target input is a pair $x_n^{\mathrm{tgt}} = (t_n, c_n)$. Its encoding retrieves the row of $\tilde{\mathbf{H}}$ belonging to its channel, concatenates the embedded query time, and projects back to $\mathbb{R}^D$ with a single linear layer,

$$\mathbf{e}_n \;=\; \mathrm{Enc}\big(x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}\big) \;=\; \mathrm{Proj}^{\mathrm{QRY}}\big(\big[\psi(t_n) \,\|\, \tilde{\mathbf{h}}_{c_n}\big]\big) \;\in\; \mathbb{R}^D. \tag{30}$$

This is the vector that Section 4 feeds to $\phi_\mu$, $\phi_\sigma$ and $\phi_V$, and that MargFlow feeds to $\mathrm{MLP}_{\mathrm{DSF}}$.

**Step 5: Global summary.** GMCP additionally requires a representation of the context that carries no target information, because the mixture weights $\pi = \phi_\pi(\mathcal{D}^{\mathrm{ctx}})$ of Section 3 must not see $\mathbf{X}^{\mathrm{tgt}}$. We obtain it by pooling $\tilde{\mathbf{H}}$ with a single learnable attention query $\mathbf{q} \in \mathbb{R}^D$,

$$\bar{\mathbf{h}} \;=\; \mathrm{Pool}\big(\tilde{\mathbf{H}}\big) \;=\; \sum_{c=1}^{C} \alpha_c \, \tilde{\mathbf{h}}_c, \qquad \alpha = \mathrm{softmax}\big(\tilde{\mathbf{H}}\mathbf{q}\big) \in \Delta^C, \tag{31}$$

which is the one step that is ours rather than adopted. CircuITS instead derives its circuit weights by cross-attention with a separate set of learnable embeddings. Everything up to and including Eq. (30) is taken over unchanged. Appendix P.2 ablates this pooling step by replacing $\tilde{\mathbf{H}}$ with the target encodings, which is what target separability forbids.

**Why the encoder is target separable.** Theorem 1 requires that dropping a target point from $\mathbf{X}^{\mathrm{tgt}}$ leave the encodings of the remaining target points untouched. The encoder satisfies this because no step lets one target point see another. Eqs. (27) to (29) read the context set only, so $\tilde{\mathbf{H}}$, and with it $\bar{\mathbf{h}}$, is a function of $\mathcal{D}^{\mathrm{ctx}}$ alone. Eq. (30) then combines $\tilde{\mathbf{H}}$ with a single target input. Hence

$$\mathbf{e}_n \text{ depends on } \mathbf{X}^{\mathrm{tgt}} \text{ only through } x_n^{\mathrm{tgt}}, \qquad \bar{\mathbf{h}} \text{ does not depend on } \mathbf{X}^{\mathrm{tgt}} \text{ at all,} \tag{32}$$

which are the two hypotheses invoked in Steps 1 and 3 of the proof of Theorem 1. A Transformer Neural Process (Nguyen & Grover, 2022) attends over the target set, and so does the attentional copula of TACTiS-2 (Ashok et al., 2023). That destroys the property. It is why MargFlow performs no attention among target points (Appendix F), and why TACTiS-2 forfeits marginalization consistency (Appendix C).

# F MargFlow: A Conditional Neural Process with Normalizing-Flow Marginals

A copula-based model needs a dedicated model for the univariate marginals, and the quality of that model turns out to matter as much as the copula on top of it (Section 5). Existing marginal models for IMTS (De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022) rely on Neural ODEs, which have been shown to be ineffective in forecasting (Klötergens et al., 2024), and restrict the predictive distribution at each target point to be Gaussian. MargFlow replaces them. It pairs the target-separable encoder of Section 4 with Deep Sigmoidal Flows (DSF) (Huang et al., 2018) as the per-target output head. Both parts and their combination are adopted from CircuITS (Klötergens et al., 2026a). Ours is the use of that combination as a model on its own, which the paragraph *Relation to CircuITS* below makes precise.

**Relation to neural processes.** Structurally, MargFlow is an instance of the Conditional Neural Process recipe (Garnelo et al., 2018a). A permutation-invariant encoder maps the context, here the observed history $\mathcal{D}^{\mathrm{ctx}}$, to a representation, and a decoder turns that representation together with a target location into a predictive distribution at that location. Instantiating that encoder with attention places MargFlow close to the Transformer Neural Process family (Nguyen & Grover, 2022). Two things are specialized relative to that recipe, both of them already so in CircuITS:

- **The encoder.** Instead of a generic set transformer over context points, we use the encoder of CircuITS (Klötergens et al., 2026a), which is built for the sparse IMTS regime in which a channel may be observed only a handful of times over the whole observation window, and which attends over the channel–time structure of $\mathcal{D}^{\mathrm{ctx}}$ rather than over a flat set. It is adopted unchanged, and restated in full in Appendix E.
- **The output head.** The usual NP decoder emits the mean and variance of a Gaussian. We replace it by a one-dimensional normalizing flow. That makes the per-target predictive distribution non-Gaussian and keeps its CDF available in closed form, which the copula construction of Section 3 depends on, since it consumes $u_n = F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}})$ directly.

**Relation to CircuITS.** Pairing this encoder with DSF outputs is not itself a new construction. The leaves of CircuITS (Klötergens et al., 2026a) are built from the same encoder and from normalizing-flow marginals of the same form, and MargFlow is such a leaf with a single mixture component and the leaf copula removed. CircuITS never uses that construction on its own. Its leaves exist only inside the circuit and are fitted under the joint objective, and none of them is ever evaluated as a marginal forecaster. Our claim here concerns the isolation rather than the architecture, and the numbers reward the isolation. Table 2 puts MargFlow next to CircuITS, whose induced univariate marginals are a mixture of such flows and whose marginal family is therefore at least as rich as MargFlow's. Only the objective and the surrounding model differ, and MargFlow attains the lower mNLL on all four datasets.

**Normalizing flows for the marginal CDFs.** We use the target encodings $\mathbf{e}_n$ of Eq. (12) to model the respective marginal CDFs $F_n^{\mathrm{marg}}$ via a DSF (Huang et al., 2018). A DSF is a strictly monotone scalar transformation $T_{\theta_n} : \mathbb{R} \to (0,1)$ parameterized by $\theta_n = \{a^{(\ell)}, b^{(\ell)}, w^{(\ell)}\}_{\ell=1}^{L}$, defined as a composition of $L$ sigmoidal blocks. A single block of width $W$ acts on a scalar input $x$ as

$$s(x; a, b, w) = \sigma^{-1}\left(\sum_{i=1}^{W} w_i \, \sigma(a_i x + b_i)\right) \tag{33}$$

Here $\sigma$ is the logistic sigmoid, $a_i > 0$, $b_i \in \mathbb{R}$, and $w_i \geq 0$ with $\sum_i w_i = 1$. Positivity of $a_i$ and the convex combination over sigmoids ensure that the inner sum is a strictly increasing function valued in $(0,1)$, so applying $\sigma^{-1}$ yields a strictly increasing map $\mathbb{R} \to \mathbb{R}$. Stacking $L$ such blocks gives a strictly increasing map $\mathbb{R} \to \mathbb{R}$, and a final logistic sigmoid squashes the output to $(0,1)$, producing a valid CDF. The conditional CDF at target point $n$ is then $F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}}) = T_{\theta_n}(y_n^{\mathrm{tgt}})$, where the block parameters $\theta_n$ are produced from the target encoding $\mathbf{e}_n$ via a small MLP,

$$\theta_n \;=\; \mathrm{MLP}_{\mathrm{DSF}}(\mathbf{e}_n). \tag{34}$$

The positivity of $a^{(\ell)}$ is enforced by a softplus activation and the simplex constraint on $w^{(\ell)}$ is enforced by a softmax. Because the map is strictly monotonic, $T_{\theta_n}$ is a one-dimensional normalizing

flow that pushes the target $y_n^{\mathrm{tgt}}$ forward to $u_n = F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}}) \in (0,1)$, distributed uniformly under the model. The likelihood of a normalizing flow is obtained by the change of variable formula,

$$f_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}}) \;=\; \underbrace{p_U\big(T_{\theta_n}(y_n^{\mathrm{tgt}})\big)}_{=1} \;\cdot\; \left|\frac{\partial T_{\theta_n}}{\partial y_n^{\mathrm{tgt}}}(y_n^{\mathrm{tgt}})\right|, \tag{35}$$

where the first term equals one because the base distribution $p_U$ is uniform on $(0,1)$, so the likelihood reduces to the Jacobian of $T_{\theta_n}$. Conditioning $\theta_n$ only on $\mathbf{e}_n$ is what makes MargFlow target-separable, and thus marginalization consistent.

**Training.** MargFlow is trained on its own, by minimizing the marginal negative log-likelihood $\sum_n -\log f_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}})$, which is the first stage of the two-stage scheme of Appendix L.4. Its weights are then frozen and the copula is trained on top of them. Appendix R contrasts this with training all weights jointly under njNLL, and Table 10 lists the hyperparameter search space.

## G Numerical Inversion of the GMM Marginal CDF

The forward pass of the copula training stage (Section 3) evaluates $z_n = F_n^{\mathrm{GMM},-1}(u_n)$. The marginal CDF of a one-dimensional Gaussian mixture,

$$F_n^{\mathrm{GMM}}(z) \;=\; \sum_{j=1}^{K} \pi_j \, \Phi\Big(\tfrac{z-\mu_{j,n}}{\sigma_{j,n}}\Big), \tag{36}$$

is strictly increasing and smooth in $z$, but its inverse has no closed form. We therefore solve $F_n^{\mathrm{GMM}}(z) = u$ for $z$ numerically. The key requirements are (i) batched evaluation across all target points and mini-batch elements simultaneously on GPU, (ii) robustness in the extreme tails where the PDF is near zero, and (iii) sufficient accuracy so that the analytic gradients of Proposition 2, which are evaluated at the returned $z$, are reliable. A pure Newton iteration meets (i) and (iii) when initialized well. It can also overshoot or stall in the tails, where a pure bisection stays robust but converges only linearly. We combine the two into a *safeguarded Newton-bisection* solver, which is a standard choice for monotone scalar root finding (Press, 2007).

**Initial bracket.** Because each component is Gaussian, the probability mass of $F_n^{\mathrm{GMM}}$ outside the interval

$$\big[\; \min_j \mu_{j,n} - 10 \, \max_j \sigma_{j,n},\; \max_j \mu_{j,n} + 10 \, \max_j \sigma_{j,n} \;\big] \tag{37}$$

is below $\Phi(-10) \approx 7.6 \times 10^{-24}$, well under floating-point precision. We use Eq. (37) as the initial bracket $[\textsc{low}, \textsc{high}]$ for every $u \in (0,1)$ and initialize the iterate at its midpoint $x \leftarrow (\textsc{low} + \textsc{high})/2$. Throughout the iteration we maintain the invariant $F_n^{\mathrm{GMM}}(\textsc{low}) \le u \le F_n^{\mathrm{GMM}}(\textsc{high})$, so the true root is always contained in the bracket.

**Iteration.** At each step we evaluate the CDF and PDF at the current iterate $x$, both available in closed form from the GMM parameters, and perform the following updates, all batched along the target and mini-batch axes:

1. *Bracket update.* For each element, if $F_n^{\mathrm{GMM}}(x) < u$ then the root lies above $x$, so we set $\textsc{low} \leftarrow x$; otherwise we set $\textsc{high} \leftarrow x$. This preserves the bracketing invariant.
2. *Newton step.* Using $F'(z) = f_n^{\mathrm{GMM}}(z)$, the Newton update is
$$x_{\text{new}} \;=\; x \;-\; \frac{F_n^{\mathrm{GMM}}(x) - u}{f_n^{\mathrm{GMM}}(x)}.$$
The denominator is clamped from below by $10^{-8}$ to avoid division by zero in regions where the mixture density is numerically negligible.
3. *Safeguard.* We accept the Newton step only if it falls inside the current bracket, $x_{\text{new}} \in [\textsc{low}, \textsc{high}]$. Otherwise (the Newton step overshot, or the local gradient was too flat for Newton to be informative), we fall back to a bisection step $x \leftarrow (\textsc{low} + \textsc{high})/2$ for that element.

**Convergence.** Inside the bracket, Newton's method on a smooth, strictly increasing $F$ converges quadratically, while the bisection fallback contracts the bracket by a factor of two whenever Newton would have left it. The number of iterations is fixed to 10, with no early exit (the same budget for every element of the batch keeps the kernel launches static and the whole loop compilable), and Appendix I shows that this budget is enough both for the downstream gradients and for the reported njNLL.

The gradients of $F_n^{\mathrm{GMM},-1}$ are obtained *analytically* via implicit differentiation (Appendix H), so we never backpropagate through the iterations. The loop is wrapped in a `no_grad` context, and only the converged $z$ is fed into the closed-form expressions of Proposition 2. Treating the solver as a black box that returns $z$, and pairing it with the analytic gradients of Proposition 2, buys three things. The memory footprint of the backward pass is independent of the number of Newton iterations, since none of the intermediate iterates are retained on the autograd tape. The safeguarded update keeps the solver from diverging in the tails, where the GMM density is small and a naive Newton step would overshoot far outside the support of the marginals. And because the bracket Eq. (37) is built directly from the predicted GMM parameters, no dataset-dependent hyperparameters are introduced.

## H Derivation of the Analytic iCDF Gradients

In this appendix we show how to compute the gradients of the marginal inverse CDF $z_n = F_n^{\mathrm{GMM},-1}(u_n)$ with respect to the mixture parameters $(\pi_j, \mu_{j,n}, \sigma_{j,n}^2)$. While Tewari (2023) already gave these gradients, we add a detailed derivation. We do this because the inverse CDF has *no closed form*, so at training time $z_n$ is obtained numerically, via Newton's method. Naively back-propagating through that solver would require unrolling all of its iterations, which is expensive and numerically fragile. The better alternative is *implicit differentiation*. It expresses the gradient of $z_n$ directly in terms of quantities we already have at the converged solution, and never enters the solver. The result is a handful of simple algebraic expressions, given in Proposition 2 of Section 3. The derivation below shows where each of those expressions comes from.

**What does the inverse CDF do?** At a high level, $F_n^{\mathrm{GMM},-1}$ takes a number $u \in (0,1)$, interpreted as a probability, and returns the value $z$ at which a Gaussian-mixture random variable accumulates exactly that much probability mass below it. There is no algebraic formula that solves this equation for a general mixture, so we use Newton's method to find $z$ numerically. ***But*** once $z$ is found, the equation $F_n^{\mathrm{GMM}}(z) = u$ is an *identity* (up to the solver tolerance) that one can differentiate.

**Which partials do we need?** The probability integral transform $u_n$ is treated as an input to the inverse CDF and is upstream of the mixture parameters. Therefore, we do need a gradient with respect to it in order to train GMCP. The mixture parameters $(\pi_j, \mu_{j,n}, \sigma_{j,n}^2)$ are predicted by the copula parameter functions $\phi_\pi$, $\phi_\mu$ and $\phi_\sigma$ of Section 3, so the partial derivatives we actually need are $\partial z_n/\partial \pi_j,\ \partial z_n/\partial \mu_{j,n},\ \partial z_n/\partial \sigma_{j,n}^2$. This restriction makes the derivation cleaner, since $u$ is treated as a constant whenever we differentiate the defining identity.

Throughout the derivation $n$ is fixed and we drop the subscript $n$ wherever doing so does not cause confusion: write $z = z_n$, $u = u_n$, $\mu_j = \mu_{j,n}$, $\sigma_j = \sigma_{j,n}$, $\sigma_j^2 = \sigma_{j,n}^2$. The marginal CDF and PDF are then:

$$F_n^{\mathrm{GMM}}(z) \;=\; \sum_{j=1}^{K} \pi_j\, \Phi\big(t_j(z)\big), \qquad t_j(z) \;:=\; \frac{z-\mu_j}{\sigma_j}, \tag{38}$$

$$f_n^{\mathrm{GMM}}(z) \;=\; \frac{dF_n^{\mathrm{GMM}}}{dz}(z) \;=\; \sum_{j=1}^{K} \frac{\pi_j}{\sigma_j}\, \varphi\big(t_j(z)\big), \tag{39}$$

where $\Phi$ and $\varphi$ are the standard-normal CDF and PDF. The variable $t_j(z) = (z-\mu_j)/\sigma_j$ is the standardized offset of $z$ from the mean of component $j$.

**A note on the simplex constraint.** The mixture weights satisfy $\pi_j > 0$ and $\sum_j \pi_j = 1$, but **we do not enforce the simplex constraint inside this derivation**. We treat each $\pi_j$ as a free positive scalar in Eq. (38) and take partial derivatives with the other $\pi_{j'}$ held fixed. The softmax (or whichever projection produces $\pi$) handles the constraint outside. This is what the code does. PyTorch's autograd composes the softmax Jacobian with $\partial z/\partial \pi_j$ during back-propagation, so working with unconstrained $\pi_j$ here gives the correct end-to-end gradient once it is composed with the constraint-respecting layer.

### H.1 Setup: The Implicit Equation and Its Differential

The forward pass solves

$$F_n^{\mathrm{GMM}}(z) \;=\; u \tag{40}$$

for $z = z(u)$, which also depends on the mixture parameters. Eq. (40) is the *defining identity*. At the converged Newton iterate it holds as a true equality, up to the residual left by the fixed iteration budget, which Appendix I shows to be negligible. This is the only place in the derivation where we use that Newton's method returned a true root, since Eq. (40) has to be an identity before we can differentiate it.

The trick of implicit differentiation is to take the derivative of *both* sides of an identity and read off the gradient of the implicit quantity (here $z$) without ever solving the equation symbolically. Concretely,

we differentiate Eq. (40) with respect to a generic mixture parameter $\xi \in \{\pi_j, \mu_j, \sigma_j^2\}$. Since $u$ does not depend on $\xi$, the right-hand side is zero:

$$\frac{\partial}{\partial \xi} F_n^{\mathrm{GMM}}\big(z(\xi)\big) \;=\; \underbrace{\frac{\partial F_n^{\mathrm{GMM}}}{\partial z}(z)}_{f_n^{\mathrm{GMM}}(z)} \frac{\partial z}{\partial \xi} \;+\; \left.\frac{\partial F_n^{\mathrm{GMM}}}{\partial \xi}\right|_{z \text{ fixed}} \;=\; 0. \tag{41}$$

The middle expression is the chain rule. The variable $\xi$ enters $F_n^{\mathrm{GMM}}(z)$ in two ways: through the *explicit* dependence of $F_n^{\mathrm{GMM}}$ on its parameters *at a fixed argument* $z$, and through the *implicit* dependence of the argument $z$ itself on $\xi$ (because changing $\xi$ changes the root of Eq. (40)). The first term in Eq. (41) collects the implicit channel and the second the explicit one. Solving Eq. (41) for $\partial z/\partial \xi$ gives

$$\boxed{\frac{\partial z}{\partial \xi} \;=\; -\,\frac{1}{f_n^{\mathrm{GMM}}(z)} \left.\frac{\partial F_n^{\mathrm{GMM}}}{\partial \xi}\right|_{z \text{ fixed}}} \tag{42}$$

Two remarks make this formula useful in practice. First, the denominator $f_n^{\mathrm{GMM}}(z) > 0$ everywhere, because each Gaussian component has full support on the real line, so the formula is well-defined and never blows up in finite precision provided the $\sigma_j$ are bounded away from zero (which is enforced by the softplus parameterization in our network). Second, every term on the right-hand side is something we already know how to compute at the converged $z$, the value of the mixture PDF and the explicit partial of the mixture CDF, both available in closed form. So this single line replaces the backward pass through the Newton solver.

### H.2 Specializing for Each Parameter

We now plug each of the three mixture parameters into Eq. (42). The pattern is always the same. Compute the explicit partial of $F_n^{\mathrm{GMM}}$ at fixed $z$, then divide by $-f_n^{\mathrm{GMM}}(z)$.

**(a) $\xi = \pi_j$ (gradient with respect to a mixture weight).** Looking at Eq. (38), only the $j$-th term in the sum $\sum_{j'} \pi_{j'} \Phi(t_{j'}(z))$ contains $\pi_j$, and it is linear in $\pi_j$. The explicit partial of $F_n^{\mathrm{GMM}}$ at fixed $z$ therefore reads off immediately:

$$\left.\frac{\partial F_n^{\mathrm{GMM}}}{\partial \pi_j}\right|_{z \text{ fixed}} \;=\; \Phi\big(t_j(z)\big). \tag{43}$$

Substituting Eq. (43) into Eq. (42),

$$\frac{\partial z}{\partial \pi_j} \;=\; -\,\frac{\Phi\big(t_j(z)\big)}{f_n^{\mathrm{GMM}}(z)}. \tag{44}$$

$\Phi(t_j(z)) \in (0,1)$ is the contribution of component $j$ to the total probability mass at $z$, so increasing $\pi_j$ by a small amount pushes more probability to the left of $z$. To keep $F_n^{\mathrm{GMM}}(z) = u$ unchanged we have to compensate by sliding $z$ leftwards, hence the negative sign. The magnitude of the response is set by how peaked the overall density is near $z$, through the $1/f_n^{\mathrm{GMM}}(z)$ factor. In flat regions a small change in mass requires a large displacement of $z$, in peaked regions a tiny one.

**(b) $\xi = \mu_j$ (gradient with respect to a mean).** From Eq. (38), only the $j$-th term depends on $\mu_j$, and $\partial t_j(z)/\partial \mu_j = -1/\sigma_j$ (with $z$ held fixed, because $t_j$ depends on $\mu_j$ only through the numerator $z - \mu_j$). The chain rule, together with $\Phi' = \varphi$, gives

$$\left.\frac{\partial F_n^{\mathrm{GMM}}}{\partial \mu_j}\right|_{z \text{ fixed}} \;=\; \pi_j\, \varphi\big(t_j(z)\big)\, \frac{\partial t_j(z)}{\partial \mu_j} \;=\; -\,\frac{\pi_j}{\sigma_j}\, \varphi\big(t_j(z)\big). \tag{45}$$

Plugging Eq. (45) into Eq. (42) cancels its minus sign and yields

$$\frac{\partial z}{\partial \mu_j} \;=\; \frac{\pi_j\, \varphi\big(t_j(z)\big)/\sigma_j}{f_n^{\mathrm{GMM}}(z)}. \tag{46}$$

Intuitively, shifting the $j$-th component to the right (increasing $\mu_j$) drags its probability mass to the right, so the cumulative mass below any fixed $z$ *decreases*. To restore $F_n^{\mathrm{GMM}}(z) = u$ we have to move $z$ rightwards as well, matching the positive sign of the gradient. The size of the response is governed by how much component $j$ contributes *at* $z$, namely $\pi_j \varphi(t_j(z))/\sigma_j$, divided by the total mixture density at $z$. Components that are far from $z$ exert almost no influence, because $\varphi(t_j(z))$ is exponentially small.

**(c) $\xi = \sigma_j^2$ (gradient with respect to a variance).** The more delicate piece is $\partial t_j(z)/\partial \sigma_j^2$. Differentiating $t_j(z) = (z - \mu_j)\,\sigma_j^{-1}$ at fixed $z$ and using the chain rule for $\sigma_j^{-1}$ as a function of $\sigma_j^2$,

$$\frac{\partial t_j(z)}{\partial \sigma_j^2} \;=\; (z-\mu_j)\,\frac{d\sigma_j^{-1}}{d\sigma_j^2} \;=\; (z-\mu_j)\,\Big(-\tfrac{1}{2}\,\sigma_j^{-3}\Big) \;=\; -\,\frac{z-\mu_j}{2\,\sigma_j^3} \;=\; -\,\frac{t_j(z)}{2\,\sigma_j^2}, \tag{47}$$

where in the last step we used $(z - \mu_j)/\sigma_j = t_j(z)$ to fold the prefactor back into $t_j$ and keep the formula compact. Hence

$$\begin{aligned}\left.\frac{\partial F_n^{\mathrm{GMM}}}{\partial \sigma_j^2}\right|_{z\ \text{fixed}} &= \pi_j\,\varphi\big(t_j(z)\big)\,\frac{\partial t_j(z)}{\partial \sigma_j^2} \\ &= -\,\frac{\pi_j\,\varphi\big(t_j(z)\big)\,t_j(z)}{2\,\sigma_j^2}.\end{aligned} \tag{48}$$

Substituting Eq. (48) into Eq. (42), the minus sign once more cancels and we get

$$\frac{\partial z}{\partial \sigma_j^2} \;=\; \frac{\pi_j\,\varphi\big(t_j(z)\big)\,t_j(z)}{2\,\sigma_j^2\,f_n^{\mathrm{GMM}}(z)} \;=\; \frac{\pi_j\,\varphi\big(t_j(z)\big)\,(z-\mu_j)/\sigma_j}{2\,\sigma_j^2\,f_n^{\mathrm{GMM}}(z)}. \tag{49}$$

# I Numerical Validation of the iCDF Solver and Its Gradients

We validate the solver of Appendix G together with the analytic gradients of Proposition 2, derived in Appendix H.

**Tails.** The only safeguard on the input is a clip of $u$ to $[\epsilon, 1-\epsilon]$, where $\epsilon$ is the machine epsilon of the data type (`torch.finfo(u.dtype).eps`), i.e. $\epsilon = 1.19 \times 10^{-7}$ in float32. Since the initial bracket Eq. (37) satisfies $F_n^{\mathrm{GMM}}(\textsc{low}) \leq \Phi(-10)$ and $F_n^{\mathrm{GMM}}(\textsc{high}) \geq 1 - \Phi(-10)$, and $\epsilon$ exceeds $\Phi(-10)$ by many orders of magnitude, the clip also guarantees that the bracket contains the root. The clip was never active in our runs. Extreme values of $u$ are also hard to produce. $u_n = F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}} \mid x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$ is the CDF value of the frozen MargFlow, which is trained by maximum likelihood, so $u_n < \epsilon$ would mean that MargFlow places less than $\epsilon$ of its probability mass below an observed target, which its own training objective penalizes heavily. The same argument applies to $u_n > 1 - \epsilon$ and the mass above the target.

**Gradient check.** The analytic gradients of Proposition 2 agree with centered finite differences of the solver output in float64. `torch.autograd.gradcheck` passes for $K \in \{1, 3, 5\}$, with maximum relative deviations of $8.2 \times 10^{-11}$ for the means, $7.5 \times 10^{-11}$ for the variances, and $3.2 \times 10^{-9}$ for the mixture weights. Because the finite differences are taken through the solver itself, the check covers both the closed forms and the assumption of Appendix H that the defining identity $F_n^{\mathrm{GMM}}(z_n) = u_n$ holds at the returned $z_n$.

**Accuracy and effect on the reported numbers.** To measure how much the remaining inversion error changes the njNLL, we use GM-Cs with random parameters, namely means $\mu_{j,n} \sim \mathcal{N}(0, 1)$, component scales $\sigma_{j,n} \sim \mathcal{U}(0.5, 1.5)$, and random correlation factors $V_j$. For each mixture size $K$ and number of target points $N$ we draw four parameter sets with 128 random queries $\mathbf{u} \in (0, 1)^N$ each, invert the $u$ values twice, and compute the njNLL from both inversions. The first inversion uses our solver. The second is a float64 reference that runs 120 bisection steps on the bracket $\left[\min_j q_{j,n}(u),\, \max_j q_{j,n}(u)\right]$, where $q_{j,n}(u) = \mu_{j,n} + \sigma_{j,n}\Phi^{-1}(u)$ is the $u$-quantile of component $j$. Because $F_n^{\mathrm{GMM}}$ is a convex combination of the component CDFs, its $u$-quantile lies between the smallest and the largest component quantile, so this bracket contains the root. Each bisection step halves the bracket, so the bracket narrows to adjacent float64 numbers after roughly 60 steps; with 120 steps the reference is therefore accurate to float64 precision. Table 6 reports the mean absolute difference between the two njNLL values, which isolates the inversion error because the marginal term is identical in both runs.

Table 6: Effect of the inversion error on the njNLL: mean absolute difference $|\Delta\,\mathrm{njNLL}|$ between our solver and a float64 bisection reference, on random GM-Cs with $K$ components and $N$ target points.

| | Target points $N$ | | | |
|---|---|---|---|---|
| $K$ | 16 | 32 | 64 | 128 |
| 1 | $1.8 \times 10^{-7}$ | $3.9 \times 10^{-7}$ | $5.1 \times 10^{-7}$ | $5.6 \times 10^{-7}$ |
| 3 | $6.3 \times 10^{-6}$ | $1.1 \times 10^{-5}$ | $2.3 \times 10^{-5}$ | $2.7 \times 10^{-5}$ |
| 5 | $3.7 \times 10^{-6}$ | $1.6 \times 10^{-5}$ | $2.2 \times 10^{-5}$ | $3.0 \times 10^{-5}$ |
| 7 | $8.6 \times 10^{-6}$ | $2.3 \times 10^{-5}$ | $3.4 \times 10^{-5}$ | $3.8 \times 10^{-5}$ |
| 10 | $9.0 \times 10^{-6}$ | $1.9 \times 10^{-5}$ | $3.3 \times 10^{-5}$ | $4.7 \times 10^{-5}$ |

The difference grows with $N$ in every row and is more than an order of magnitude smaller for $K = 1$, where each marginal is a single Gaussian, than for $K \geq 3$. In every configuration it stays below $5 \times 10^{-5}$. The grid covers the mixture and target-set sizes of our experiments. $K$ ranges over the full search space of Table 11, and no instance in our datasets has more than 85 target points (Table 9). The inversion error is therefore more than an order of magnitude below the last digit reported in Table 3, and more than two orders of magnitude below the smallest standard deviation of GMCP.

## J Computational Complexity

Evaluating the GMCP log-density at a target set of size $N$ is dominated by the $K$ multivariate Gaussian densities $\mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j)$. Each one requires a linear solve in $\Sigma_j$ and a $\log\det\Sigma_j$, which by Cholesky costs $\mathcal{O}(N^3)$ apiece and $\mathcal{O}(KN^3)$ in total. That worst case is never realized. By construction (Section 3), $\Sigma_j = D_j R_j D_j$ with $R_j = \tilde{V}_j\tilde{V}_j^\top + \Lambda_j$, where $\tilde{V}_j \in \mathbb{R}^{N\times H}$ stacks the normalized feature vectors $\tilde{\mathbf{v}}_{j,n}$, and $D_j$ and $\Lambda_j = \operatorname{diag}\big(1 - \|\tilde{\mathbf{v}}_{j,n}\|^2\big)_{n=1}^{N} \succ 0$ are diagonal. Each $R_j$ is therefore a rank-$H$ update of a positive diagonal matrix, so the Sherman-Morrison-Woodbury identity and the matrix determinant lemma give

$$R_j^{-1} = \Lambda_j^{-1} - \Lambda_j^{-1}\tilde{V}_j\big(I + \tilde{V}_j^\top\Lambda_j^{-1}\tilde{V}_j\big)^{-1}\tilde{V}_j^\top\Lambda_j^{-1}, \tag{50}$$

$$\log\det R_j = \log\det\big(I + \tilde{V}_j^\top\Lambda_j^{-1}\tilde{V}_j\big) + \log\det\Lambda_j, \tag{51}$$

which need only $\mathcal{O}(N)$ diagonal work plus operations on the $H \times H$ matrix $I + \tilde{V}_j^\top\Lambda_j^{-1}\tilde{V}_j = I + V_j^\top V_j$. One component thus costs $\mathcal{O}(NH^2 + H^3)$ and the full log-density $\mathcal{O}\big(K(NH^2 + H^3)\big)$. For $H \ll N$ the cubic dependence on $N$ becomes linear. The surrounding stages are closed-form and negligible against this, at $\mathcal{O}(MD^2 + ND^2)$ for the encoder, $\mathcal{O}((N+1)D^2)$ for the parameter MLPs and $\mathcal{O}(NLW)$ for the marginal flow with $L$ DSF blocks of width $W$.

**Cost of the numerical inversion.** The inversion $z_n = F_n^{\mathrm{GMM},-1}(u_n)$ (Appendix G), with the implicit-function gradients of Appendix H, is the one operation GMCP performs that a Gaussian-copula model does not. The solve is $\mathcal{O}(TKN)$ for $T$ Newton steps, and the analytic backward one further $\mathcal{O}(KN)$ pass, which Table 7 counts against the rest of the copula stage. The inversion stays under $2\%$ throughout, and its share *falls* as $N$ grows. It is $\mathcal{O}(N)$, like the per-target parameter MLPs that dominate at $N = 16$, while the $\mathcal{O}(KN^2H)$ Gram construction and $\mathcal{O}(KN^3)$ factorization grow faster. The picture is unchanged at $K = 10$ and under the structured evaluation above ($1.3\%$).

Table 7: Analytic FLOP counts (MFLOP) for one forward and backward evaluation of the GMCP copula log-density ($K = 5$, $H = 32$, $D = 64$, batch $64$, $T = 10$). Ranges span charging a transcendental at 1 and at 10 flops; only the inversion contains any. Counts are analytic: matmul-based profilers report this elementwise solve as zero.

| $N$ | GMM inversion | Parameter MLPs | Gram | Cholesky | Share of total |
|---|---|---|---|---|---|
| 16 | 1.0–2.0 | 118.6 | 5.2 | 0.4 | 0.78–1.58% |
| 64 | 3.9–8.0 | 474.5 | 83.9 | 28.0 | 0.66–1.33% |
| 128 | 7.8–15.9 | 949.0 | 335.5 | 223.7 | 0.51–1.04% |
| 256 | 15.6–31.9 | 1897.9 | 1342.2 | 1789.6 | 0.31–0.62% |

On an RTX 4060 the ten-iteration solve takes a flat $1.8\,\mathrm{ms}$ for all $N \le 256$, a *single scalar* root-find still costs $1.76\,\mathrm{ms}$, and replaying the identical iterations as one captured CUDA graph recovers $79\%$ of it.

**Empirical training time.** When we compare epoch times, GMCP trains faster than TACTiS-2 on all four datasets, by up to $3.2\times$ (Table 8).

Table 8: Average wall-clock training time per epoch (seconds), on one NVIDIA A100, averaged over 5 splits. †Our adaptation of TACTiS-2 to sparse IMTS (Appendix L.7).

| Model | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| TACTiS-2† | 0.28 | 6.1 | 4.2 | 11.3 |
| GMCP (GM-C) | 0.22 | 2.2 | 4.0 | 3.5 |

# K SYNTHETIC PROCESSES

Section 5.1 uses three synthetic processes, the released pendulum (Appendix K.1) and the two Lévy jump processes (Appendix K.2). We specify them here. All three are built so that their conditional copulas lie outside the class of mixtures of Gaussian copulas for *any* number of components. They violate the two conclusions of Proposition 3 by different mechanisms. The pendulum comes with ring-shaped bivariate marginals (Appendix A). The Lévy processes put mass on modes that a Mix-GC cannot move a component to.

**Models.** All models share one MargFlow marginal model, which is trained first on the mNLL and then frozen, so the models differ only in the copula. The MargFlow column of Table 1 uses no copula, i.e. independent marginals. GCNP is a conditional Gaussian copula (Markou et al., 2022), which we implement as a Mix-GC with a single component. Mix-GC and GMCP use $K = 10$ components, and all three copulas use the same encoder architecture.

**Training.** Each training step draws a fresh batch of $64$ instances from the process, so every step sees unseen data and there is no train set to overfit. We use Adam at a learning rate of $3 \cdot 10^{-4}$ and select the checkpoint with the lowest loss on a fixed held-out batch of $1{,}024$ instances, which also serves for early stopping with a patience of $4{,}000$ steps. Every process gets the same budget of $20{,}000$ steps for the marginal model and $40{,}000$ for each copula, set by the slowest of the three to fit.

**Evaluation.** Every model is trained $5$ times under different seeds, and every run is evaluated on the same $10$ freshly drawn test sets of $1{,}000$ instances each, with $16$ targets per instance, so the runs are paired across columns. Table 1 reports the njNLL as mean and standard deviation across the $5$ training runs.

## K.1 THE RELEASED PENDULUM

The pendulum process is built so that (i) the context changes the dependence structure of the targets, and not only their marginals, and (ii) its bivariate marginals lie outside the class of mixtures of Gaussian copulas for any number of components (Proposition 3).

**Generating process.** Every instance is a single-channel series of a pendulum that is held at a height $A \sim \mathcal{U}[3, 4]$ and released from rest at an unobserved time $S$,

$$Y(t) \;=\; \begin{cases} A, & t \le S, \\ A\cos\big(2\pi\,(t-S)/T(A)\big), & t > S, \end{cases} \qquad T(A) \;=\; T_0\sqrt{A/\bar{A}}. \tag{52}$$

At a fixed release angle the height is proportional to the arm length, and the period of a pendulum grows with the square root of its arm length. A single latent therefore sets both the amplitude and the period of the swing. We use $T_0 = 6$ and $\bar{A} = 3.5$, the midpoint of the height range. The release time is drawn as $S \sim \mathcal{U}[s_0, s_0 + T(A)]$ with $s_0 = 5$, i.e. uniformly over one period, so that the phase of the swing is uniform on $(0, 2\pi)$ at every height. All observations carry Gaussian noise with standard deviation $0.05$. The context consists of $6$ observations at uniformly drawn times in $[0, 4.5]$, all of which fall before the earliest possible release. The context therefore determines $A$, and with it $T(A)$, but carries no information about $S$. Each instance has $16$ targets at uniformly drawn times in $[4.5, 15]$, a window that contains the release.

**The bivariate marginals are not Gaussian.** Conditioned on the context, consider two targets $t_1 < t_2$ after the latest possible release, with lag $\Delta = t_2 - t_1$ and $\delta = 2\pi\Delta/T(A)$. With $\varphi \sim \mathcal{U}(0, 2\pi)$ the phase at $t_1$, the noise-free pair is $(y_1, y_2) = (A\cos\varphi,\ A\cos(\varphi + \delta))$, which lies on the ellipse

$$y_1^2 - 2\,y_1 y_2 \cos\delta + y_2^2 \;=\; A^2 \sin^2\delta, \tag{53}$$

a circle if $\Delta = T(A)/4$. Each coordinate follows the arcsine law on $[-A, A]$, which is bimodal and symmetric around $0$, so the origin is mapped to the center $\mathbf{u}^\star = (1/2, 1/2)$ of the unit square. The ellipse keeps a distance of at least $A\sqrt{1 - |\cos\delta|}$ from the origin. Unless $\Delta$ is a multiple of $T(A)/2$, the copula density at $\mathbf{u}^\star$ is therefore close to $0$ by Eq. (18), whereas every mixture of Gaussian copulas has density at least $1$ there by Proposition 3. The bivariate marginals of the pendulum are

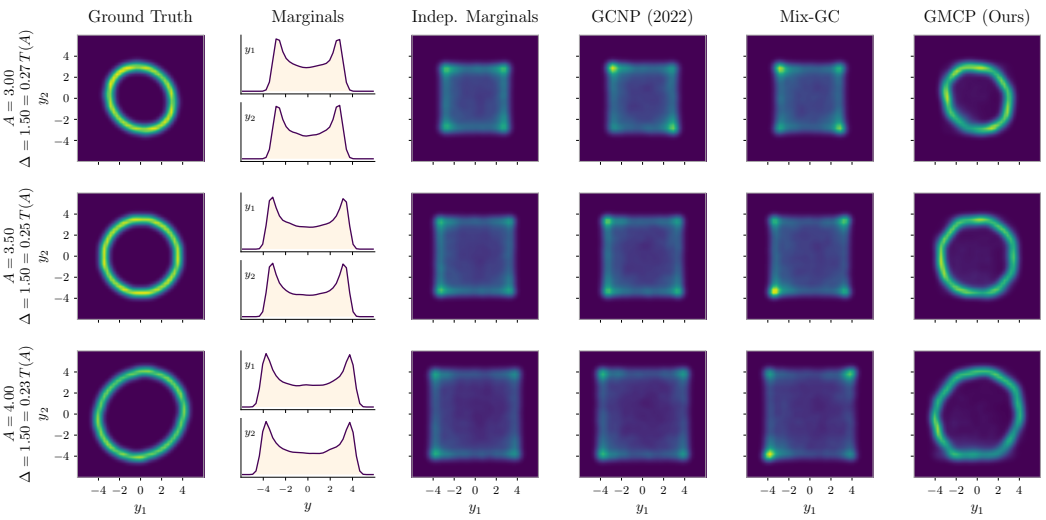


Figure 5: Bivariate marginals of the pendulum process for two targets after the release at a fixed lag $\Delta = 1.5$, conditioned on contexts with held heights $A = 3, 3.5, 4$ (rows, top to bottom). Columns show the ground-truth density, its univariate marginals, and smoothed histograms of samples from independent MargFlow marginals, GCNP, Mix-GC ($K = 10$), and GMCP ($K = 10$), all sharing the same marginals.

thus a conditional version of the *circle* distribution of Figure 2. Since $\delta$ depends on $A$ through $T(A)$, their shape also changes with the context at a fixed lag, from an ellipse tilted one way through a circle to an ellipse tilted the other way (Figure 5). A copula that reads the lag but not the context has to pick one of them and be wrong about the rest.

**Where the copulas differ.** The gap is widest on the bivariate marginals themselves, on pairs of targets drawn after the latest release. Figure 5 shows why. GCNP and Mix-GC cannot place a hole at the center, so they stay close to independence. The independent density is the product of the two bimodal marginals and has four corner modes. GMCP recovers the ring and tilts it with the context. Over the 5 training runs of Table 1 the copula is worth 1.88 njNLL to GMCP but only 0.83 to Mix-GC and 0.67 to GCNP, so GMCP leads Mix-GC by 1.05. The anchored pairs these panels show, two targets after the latest release with the lag drawn uniformly from $[0, 2.59]$, separate the models further. There, independent marginals reach an njNLL of 1.850, which GCNP improves by 0.15 and Mix-GC by only 0.09, while GMCP improves it by 0.86.

### K.2 Lévy Jump Processes

The pendulum violates Proposition 3 (i) by putting 0 density at the center of the unit square. The Lévy processes use the other side of the same proposition. They put mass on modes far away from the center, which a Mix-GC cannot reach.

**Generating process.** Every instance is a single-channel *pure-jump* series,

$$Y(t) \;=\; A\,L(t), \tag{54}$$

with jump scale $A = 3$ and $L$ a Lévy process with independent, stationary increments. We evaluate two choices:

- *compound Poisson* (CPP), $L(t) = N_\lambda(t)$, with unit jumps at rate $\lambda = 1$. It has finitely many jumps and the sharpest lattice of the three toys.
- *Gamma*, $L(t) = G(t)$, a Gamma subordinator with variance rate $\nu = 0.3$. It has infinite activity, and its lattice is already partly smeared into a continuum.

Both are subordinators, so every jump is upward. The context consists of 12 observations at uniformly drawn times in $[0, 4]$ and each instance has 16 targets at uniformly drawn times in $(4, 7]$. Observations carry Gaussian noise with standard deviation $0.05$ on the compound Poisson process, matching the pendulum, and $0.02$ on the Gamma one. Either way the noise is far below the jump scale $A = 3$, so a jump is unambiguous in the context. The compound Poisson process needs the wider noise because of its sharp lattice.

**What the copula has to express.** The jump scale $A$ is fixed rather than drawn, so the hidden latent is the jump *times* in the target window, which fall strictly after the context closes. Anchor a pair at $a > 4$ and lag it by $\Delta$. Conditional on the context, which fixes the level $Y(4)$, the noise-free pair sits on

$$y_1 \;=\; Y(4) + A\,j, \qquad y_2 \;=\; y_1 + A\,k, \tag{55}$$

with $j$ and $k$ independent non-negative jump counts. The support is a lattice occupying a quarter-plane, since $y_2 - y_1$ is bounded below by $0$ and unbounded above. Pushed through the marginals, it becomes a grid of six to ten well-separated modes scattered across the unit square, most of them far from the center. A Mix-GC can draw overlapping ellipses through the center but cannot place a component on any of the modes. The free component means of a GM-C are the freedom to put one component per mode.

# L Experimental Details

This appendix covers how the data are split and how the models are trained and tuned. We use the four benchmark datasets (USHCN, PhysioNet'12, MIMIC-III, MIMIC-IV) following the preprocessing protocols described by Yalavarthi et al. (2026), including the binning intervals, observation horizons, and channel selections reported there. We refer the reader to that work for the full dataset characteristics. Here, we focus on the protocol that governs how we train and evaluate our model and its baselines.

## L.1 Dataset Statistics

For convenience we summarize the four datasets in Table 9, including the per-instance target-count statistics (minimum, average, maximum). USHCN (Menne et al., 2016) is a climatological dataset of daily measurements from 1,100 U.S. weather stations over four years (1996–2000), tracking five variables (snow precipitation, rain precipitation, snow depth, minimum temperature, maximum temperature). Following De Brouwer et al. (2019), we transform USHCN into an IMTS by randomly dropping $95\%$ of the recorded observations. PhysioNet'12 (Silva et al., 2012) contains ICU records of $12{,}000$ patients monitored over a 48-hour period across 37 vital signs, binned into hourly intervals (Che et al., 2018). MIMIC-III (Johnson et al., 2016) provides ICU stays from Beth Israel Deaconess Medical Center, with 96 variables aggregated into 30-minute bins (De Brouwer et al., 2019). MIMIC-IV (Johnson et al., 2023) records ICU stays from an academic medical center in Boston with 102 variables binned at the 1-minute resolution (Biloš et al., 2021). This fine-grained binning yields the largest number of observations and queries among the four datasets.

Table 9: Summary of dataset characteristics and preprocessing. The last three columns report the minimum, average, and maximum number of forecasting target points $N$ per instance.

| | | | | | Target points $N$ | | |
|---|---|---|---|---|---|---|---|
| Dataset | Samples | Channels | Duration | Binning | min | $\bar{N}$ | max |
| USHCN | 1,100 (Stations) | 5 | 4 Years | 1 Week | 3 | 3.3 | 6 |
| PhysioNet'12 | 12,000 (Patients) | 37 | 48 Hours | 1 Hour | 1 | 19.8 | 53 |
| MIMIC-III | 21,000 (Patients) | 96 | 48 Hours | 30 Mins | 1 | 10.3 | 85 |
| MIMIC-IV | 18,000 (Patients) | 102 | 48 Hours | 1 Min | 1 | 7.8 | 79 |

## L.2 Data Splits

We adopt the splitting protocol of Yalavarthi et al. (2026) without modification, both to ensure a fair head-to-head comparison with previously reported numbers and to remove any degree of freedom that could be tuned in our favor. Each dataset is partitioned into training, validation, and test sets in a 70:20:10 ratio. We repeat this partitioning under five distinct random seeds. We use the same seeds as experiments done in previous work (Yalavarthi et al., 2026; Klötergens et al., 2026a). The test seed is tied to the fold index (e.g., Split 1 uses Seed 1), so the same five test sets are used by every model in our experiments. All reported numbers are means and standard deviations across these five folds.

**Interpreting the standard deviation.** The five folds differ in which forecasting windows they expose to the model, and some folds contain inherently harder forecasting horizons than others (e.g. target points with a higher aleatoric uncertainty). Consequently, the per-fold scores of *every* model on a given dataset shift up or down together as the fold becomes harder or easier, which inflates the across-fold standard deviation relative to the much smaller within-fold gap between models. The absolute magnitude of either model's standard deviation hence cannot rule out a significant difference. As a rule of thumb, when a model outperforms a baseline by about one reported standard deviation in this protocol, the improvement is consistent across folds and statistically meaningful.

## L.3 Hyperparameter Selection

For each model and dataset, we draw 10 hyperparameter configurations from the search space below using random search, train each configuration on the first fold, and select the configuration with the

best validation njNLL. The selected configuration is then retrained from scratch on all five folds; the test numbers we report come from this retrained configuration.

Tables 10 to 12 list the search spaces for MargFlow, GMCP, and our re-implemented TACTiS-2 baseline (see Appendix L.7). GMCP and TACTiS-2 share the same encoder, so the encoder-side hyperparameters (`copula_n_heads`, `copula_hidden_dim`) are drawn from identical grids; the remaining hyperparameters are specific to each copula module.

Table 10: Hyperparameter search space for MargFlow

| Hyperparameter | Description | Values |
|---|---|---|
| `marg_weight_decay` | Weight decay of the marginal encoder | $\{10^{-4}, 10^{-3}\}$ |
| `marg_n_heads` | Attention heads of the marginal encoder | $\{1, 2, 4\}$ |
| `marg_hidden_dim` | Hidden dim of the marginal encoder | $\{32, 64, 128\}$ |
| `flow_mlp_layers` | MLP layers of the conditioner network | $\{1, 2, 3\}$ |
| `flow_mlp_dim` | MLP hidden dim of the conditioner network | $\{32, 64, 128\}$ |
| `flow_layers` | Number of normalizing-flow layers | $\{1, 2, 3\}$ |
| `flow_hid_dim` | Hidden dim of each normalizing-flow layer | $\{10, 20\}$ |

Table 11: Hyperparameter search space for GMCP and variants.

| Hyperparameter | Symbol | Description | Values |
|---|---|---|---|
| `copula_components` | $K$ | Number of mixture components | $\{1, 3, 5, 7, 10\}$ |
| `copula_n_heads` | — | Attention heads of the encoder | $\{1, 2, 4\}$ |
| `copula_hidden_dim` | $D$ | Hidden dim of the encoder | $\{32, 64, 128\}$ |
| `corr_net_hidden_dim` | $H$ | Hidden dim of the MLPs producing $\mu$, $\sigma$, and the Gram-matrix vectors | $\{16, 32, 64, 128\}$ |

Table 12: Hyperparameter search space for our re-implemented TACTiS-2 baseline. The copula-side ranges match those reported by Ashok et al. (2023); the encoder-side ranges (`copula_hidden_dim`, `copula_n_heads`) match GMCP.

| Hyperparameter | Description | Values |
|---|---|---|
| `copula_attn_heads` | Attention heads of the attentional copula | $\{2, 4, 8\}$ |
| `copula_attn_layers` | Attention layers of the attentional copula | $\{1, 2, 3, 4\}$ |
| `copula_attn_dim` | Attention dim of the attentional copula | $\{8, 16, 32\}$ |
| `copula_mlp_layers` | MLP layers of the attentional copula | $\{1, 2, 3\}$ |
| `copula_mlp_dim` | MLP hidden dim of the attentional copula | $\{32, 64, 128\}$ |
| `copula_resolution` | CDF discretization resolution | $\{8, 16, 32\}$ |
| `copula_dropout` | Dropout rate of the attentional copula | $\{0.0, 0.1, 0.2\}$ |
| `copula_hidden_dim` | Hidden dim of the encoder | $\{32, 64, 128\}$ |
| `copula_n_heads` | Attention heads of the encoder | $\{1, 2, 4\}$ |

### L.4 Training Setup

Unless stated otherwise, we train GMCP in the *two-stage* scheme of Ashok et al. (2023). We first fit the marginal model MargFlow on its own, by treating all target points as independent, and then train the copula on top of the frozen marginals. As discussed in Section 3, this is an optimization choice rather than a property of GMCP. GMCP-Joint, reported in Section 5, trains the same architecture jointly under njNLL.

All models are trained with the AdamW optimizer (Kingma & Ba, 2017; Loshchilov & Hutter, 2019) using an initial learning rate of $10^{-3}$, weight decay of $10^{-3}$, and a batch size of $64$. We schedule the learning rate with reduce-on-plateau (factor $0.5$, patience of $5$ validation epochs without

improvement) and train for at most 2000 epochs with early stopping (patience of 30 epochs on the validation njNLL). All experiments use float32 precision. For the MIMIC-IV dataset we reduce the batch size to 32 when memory pressure requires it; we apply this reduction uniformly across all models on that dataset. We train on a single NVIDIA A100 GPU per fold.

### L.5 Evaluation Metrics

We evaluate every model on the held-out test split of each fold using the normalized joint negative log-likelihood (njNLL) and the marginal negative log-likelihood (mNLL). The njNLL is defined in Eq. (9); the mNLL evaluates the predictive density at each target coordinate independently and averages over target points,

$$\mathrm{mNLL}(\mathbf{y}^{\mathrm{tgt}} \mid \mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}) \;=\; \frac{1}{N}\sum_{n=1}^{N} -\log p\big(y_n^{\mathrm{tgt}} \,\big|\, x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}\big). \tag{56}$$

For a factorized model such as MargFlow, the njNLL reduces to an average of marginal negative log-likelihoods, yet its njNLL and mNLL differ in absolute value. The difference is due only to how the two metrics are aggregated over the test set. The njNLL first averages over the $N$ target points of each instance and then over instances, so every instance carries equal weight. The mNLL averages once, over all target points of the test set pooled together, so instances with more target points carry more weight.

### L.6 Univariate Wasserstein-1 Distance

The Wasserstein distances reported in Table 4 compare two *sample sets* of the same model, the samples of MargFlow alone and the samples of the full MargFlow+copula setup, either GMCP or TACTiS-2. The metric asks whether attaching the copula leaves the univariate marginals where MargFlow put them, which is what the definition of a copula requires. GMCP guarantees it by construction (Appendix B) and TACTiS-2's attentional copula does not. The Control row quantifies the irreducible sampling error. It is computed by drawing two independent sets of $S = 1000$ samples from the same MargFlow under different random seeds and applying the same estimator to those two sample sets. For two probability measures $P$ and $Q$ on $\mathbb{R}$ with cumulative distribution functions $F_P$ and $F_Q$, the Wasserstein-1 distance admits the closed form

$$W_1(P, Q) \;=\; \int_{\mathbb{R}} \big|F_P(t) - F_Q(t)\big|\, dt. \tag{57}$$

Given two equal-sized samples $\{a_i\}_{i=1}^{S}$ and $\{b_i\}_{i=1}^{S}$ drawn from $P$ and $Q$, the empirical Wasserstein-1 distance reduces to the average absolute gap between order statistics,

$$\widehat{W}_1\big(\{a_i\}, \{b_i\}\big) \;=\; \frac{1}{S}\sum_{i=1}^{S} \big|a_{(i)} - b_{(i)}\big|, \tag{58}$$

where $a_{(1)} \le \cdots \le a_{(S)}$ and $b_{(1)} \le \cdots \le b_{(S)}$ denote the sorted samples. At every target point of a test fold we draw $S = 1000$ samples from each of the two setups and pool them, across target points, forecasting dimensions and test instances, into two empirical distributions of equal size, to which the estimator above is applied. This yields one distance per fold, and we report its mean and standard deviation over the five splits. Note that pooling before comparing measures the marginal distribution of a test fold as a whole rather than that of an individual target point; Appendix C examines marginalization consistency in the sharper, per-target-set form.

### L.7 Adapting TACTiS-2 to Sparse IMTS

The attentional copula is introduced in TACTiS (Drouin et al., 2022) and reused in TACTiS-2 (Ashok et al., 2023). The two methods share the same attentional copula module and differ in the training scheme. In the original TACTiS, a single shared encoder feeds both the marginals and the copula, and all components are trained jointly under the joint negative log-likelihood. In TACTiS-2, the marginals and the copula are produced by two separate encoders and trained in two stages. The marginals are trained first and then frozen, and the copula is trained on top of them. The two methods also differ in

how the autoregressive copula orders the target points: TACTiS draws a random permutation of the targets at every training step, whereas TACTiS-2 does not permute them. Ashok et al. (2023) report that this two-stage protocol outperforms the joint TACTiS training on all of their tasks. It is also the protocol used by GMCP, so comparing against the TACTiS-2 form of the attentional copula gives the most direct copula-vs-copula comparison.

One component of the original TACTiS-2 does not transfer to our setting unchanged, namely its encoder. It was designed for moderately irregular but still relatively dense multivariate time series, in which every channel is observed at most timestamps. The IMTS benchmarks we use are far sparser, with channels that may be observed only a handful of times over the whole observation window, and the original TACTiS-2 encoder underperforms badly in this regime.

To isolate the contribution of the copula module itself, we therefore re-implement the attentional copula on top of the same encoder that GMCP uses (described in Section 4). Concretely, our TACTiS-2 baseline keeps the original attentional copula, the original two-stage training protocol and the unpermuted target ordering of TACTiS-2, and only swaps the encoder for one that is suited to sparse IMTS. This is also the configuration in which it performs best on our benchmarks, so we use it everywhere we refer to TACTiS-2 in the main paper.

## M Complete Likelihood Results

Tables 2 and 3 in the main paper restrict the comparison to the models that are within reach of GMCP and MargFlow, namely the jointly trained IMTS forecasters and, for njNLL, the copula baselines. We report additional baselines here. These are the univariate neural-ODE models GRU-ODE-Bayes (De Brouwer et al., 2019), NeuralFlows (Biloš et al., 2021) and CRU (Schirmer et al., 2022), together with multitask Gaussian Process Regression (GPR) (Dürichen et al., 2015). They were moved out of the main paper for space reasons. As Tables 13 and 14 show, none of them is competitive on any dataset under either metric, and all of them are separated from the models in the main tables by a margin far larger than the spread across folds. The three neural-ODE models emit an independent Gaussian per target point, so their joint score carries no information about dependence beyond what their marginal score already says. GPR is consistent, but restricted to multivariate Gaussians. Where a baseline number is available from previous work under the identical protocol, we report it directly (Yalavarthi et al., 2026; Klötergens et al., 2026a). CRU runs out of memory on `MIMIC-IV`, which we mark as OOM. Table 13 additionally reports GMCP with $K = 1$, to which we refer as Gaussian Copula Neural Process (GCNP) (Markou et al., 2022). It improves on independent marginals everywhere, but trails Mix-GC and GMCP with $K > 1$ on all four datasets. Its Energy Score is in Table 16; Tables 15 and 17 omit it because a copula leaves the MargFlow marginals untouched (Appendix B).

Table 13: Complete version of Table 3: **njNLL** (Normalized Joint Negative Log-Likelihood) across datasets, including the univariate and Gaussian-process baselines and the GCNP copula that Table 3 omits. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**. †Our adaptation of TACTiS-2 to sparse IMTS (Appendix L.7).

| | Model | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|---|
| Univariate | GRU-ODE-Bayes | 0.766 ± 0.159 | 0.501 ± 0.001 | 0.961 ± 0.064 | 0.823 ± 0.318 |
| | NeuralFlows | 0.775 ± 0.152 | 0.496 ± 0.000 | 0.998 ± 0.111 | 0.689 ± 0.087 |
| | CRU | 0.761 ± 0.191 | 1.057 ± 0.007 | 1.234 ± 0.076 | OOM |
| | MargFlow | -3.969 ± 0.305 | -0.442 ± 0.021 | -0.505 ± 0.208 | -1.819 ± 0.065 |
| Joint | ProFITi | -3.226 ± 0.225 | -0.647 ± 0.078 | -0.377 ± 0.032 | -1.777 ± 0.066 |
| | GPR | 2.011 ± 1.376 | 1.367 ± 0.074 | 3.146 ± 0.359 | 2.789 ± 0.057 |
| | MOSES | -3.357 ± 0.176 | -0.491 ± 0.041 | -0.305 ± 0.027 | -1.668 ± 0.097 |
| | CircuITS | -3.789 ± 0.218 | -0.550 ± 0.013 | -0.574 ± 0.080 | -2.113 ± 0.044 |
| | GMCP-Joint | -3.907 ± 0.269 | -0.489 ± 0.399 | -0.508 ± 0.309 | -2.019 ± 0.049 |
| Copula | TACTiS-2† | **-4.254 ± 0.294** | -0.732 ± 0.115 | -0.780 ± 0.282 | -2.053 ± 0.170 |
| | GCNP | -3.998 ± 0.325 | -0.680 ± 0.019 | -0.642 ± 0.210 | -1.873 ± 0.067 |
| | Mix-GC | -4.078 ± 0.308 | -0.728 ± 0.018 | -0.779 ± 0.216 | -2.080 ± 0.064 |
| | **GMCP** | -4.135 ± 0.304 | **-0.745 ± 0.019** | **-0.835 ± 0.218** | **-2.201 ± 0.058** |

Table 14: Complete version of Table 2: **mNLL** (Marginal Negative Log-Likelihood) across datasets, including the univariate and Gaussian-process baselines that Table 2 omits. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**.

| | Model | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|---|
| Joint | ProFITi | -3.324 ± 0.206 | -0.016 ± 0.085 | 0.408 ± 0.030 | 0.500 ± 0.322 |
| | GPR | 1.235 ± 0.096 | 1.161 ± 0.065 | 1.341 ± 0.009 | 1.161 ± 0.010 |
| | MOSES | -3.355 ± 0.156 | -0.271 ± 0.028 | 0.163 ± 0.026 | -0.634 ± 0.017 |
| | CircuITS | -3.717 ± 0.201 | -0.287 ± 0.037 | 0.095 ± 0.107 | -0.731 ± 0.063 |
| | GMCP-Joint | -3.663 ± 0.210 | -0.176 ± 0.470 | 0.108 ± 0.238 | -0.677 ± 0.044 |
| Univariate | GRU-ODE-Bayes | 0.776 ± 0.172 | 0.504 ± 0.061 | 0.839 ± 0.030 | 0.876 ± 0.589 |
| | NeuralFlows | 0.775 ± 0.180 | 0.492 ± 0.029 | 0.866 ± 0.097 | 1.796 ± 0.050 |
| | CRU | 0.762 ± 0.180 | 0.931 ± 0.019 | 1.209 ± 0.044 | OOM |
| | **MargFlow** | **-3.948 ± 0.294** | **-0.479 ± 0.021** | **-0.158 ± 0.205** | **-0.833 ± 0.077** |

## N SAMPLING RESULTS

We complement the likelihood-based evaluation in the main paper with three sample-based metrics, namely Mean Squared Error (MSE) for point forecasts (Table 15), Energy Score (ES) for multivariate distributional accuracy (Table 16) and Continuous Ranked Probability Score (CRPS) for marginal distributional accuracy (Table 17). For each test instance we draw $S = 1000$ samples $\hat{\mathbf{y}}^{(s)} \in \mathbb{R}^N$ from the predictive distribution conditioned on $(\mathbf{X}^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}})$, and compare them against the ground-truth target outputs $\mathbf{y}^{\mathrm{tgt}} \in \mathbb{R}^N$. All three metrics are reported as means across the test set and across the five folds.

**MSE.** We use the sample mean $\bar{\hat{y}}_n = \frac{1}{S}\sum_{s=1}^{S} \hat{y}_n^{(s)}$ as the point forecast and report the per-coordinate squared error averaged over target points,

$$\mathrm{MSE}(\hat{\mathbf{y}}, \mathbf{y}) \;=\; \frac{1}{N}\sum_{n=1}^{N} (\bar{\hat{y}}_n - y_n)^2. \tag{59}$$

**Energy Score.** The Energy Score (Gneiting & Raftery, 2007) is a strictly proper scoring rule for multivariate distributions. For a predictive distribution $\hat{F}$ and target $\mathbf{y} \in \mathbb{R}^N$ it is defined as

$$\mathrm{ES}(\hat{F}, \mathbf{y}) \;=\; \mathbb{E}_{\mathbf{Y}\sim\hat{F}} \|\mathbf{Y} - \mathbf{y}\|_2 \;-\; \tfrac{1}{2}\,\mathbb{E}_{\mathbf{Y},\mathbf{Y}'\sim\hat{F}} \|\mathbf{Y} - \mathbf{Y}'\|_2, \tag{60}$$

where $\mathbf{Y}, \mathbf{Y}'$ are independent draws from $\hat{F}$. Given $S$ samples, we use the standard unbiased estimator

$$\widehat{\mathrm{ES}} \;=\; \frac{1}{S}\sum_{s=1}^{S} \|\hat{\mathbf{y}}^{(s)} - \mathbf{y}\|_2 \;-\; \frac{1}{2\,S\,(S-1)} \sum_{s\neq s'} \|\hat{\mathbf{y}}^{(s)} - \hat{\mathbf{y}}^{(s')}\|_2. \tag{61}$$

**CRPS.** The Continuous Ranked Probability Score (Gneiting & Raftery, 2007) is the univariate counterpart of the Energy Score. For a one-dimensional predictive CDF $\hat{F}_n$ and target $y_n \in \mathbb{R}$,

$$\mathrm{CRPS}(\hat{F}_n, y_n) \;=\; \int_{\mathbb{R}} \big(\hat{F}_n(z) - \mathbb{1}\{z \geq y_n\}\big)^2 \,\mathrm{d}z \;=\; \mathbb{E}\,|Y_n - y_n| \;-\; \tfrac{1}{2}\,\mathbb{E}\,|Y_n - Y_n'|, \tag{62}$$

with $Y_n, Y_n' \sim \hat{F}_n$ independent. We report the average across target points, estimated from the per-coordinate samples $\hat{y}_n^{(s)}$ as

$$\widehat{\mathrm{CRPS}} \;=\; \frac{1}{N}\sum_{n=1}^{N} \left[\frac{1}{S}\sum_{s=1}^{S} |\hat{y}_n^{(s)} - y_n| \;-\; \frac{1}{2\,S\,(S-1)} \sum_{s\neq s'} |\hat{y}_n^{(s)} - \hat{y}_n^{(s')}|\right]. \tag{63}$$

Lower values are better for all three metrics.

Table 15: Comparing models w.r.t. **MSE** (Mean Squared Error) across datasets. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**, the second best is underlined. Rows marked with * use a MargFlow that is optimized for CRPS instead of likelihood.

| **Model** | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| GRU-ODE | $0.410 \pm 0.106$ | $0.329 \pm 0.004$ | $0.479 \pm 0.044$ | $0.365 \pm 0.012$ |
| NeuralFlows | $0.424 \pm 0.110$ | $0.331 \pm 0.006$ | $0.479 \pm 0.045$ | $0.374 \pm 0.017$ |
| CRU | $\mathbf{0.290 \pm 0.060}$ | $0.475 \pm 0.015$ | $0.725 \pm 0.037$ | OOM |
| ProFITi | $0.308 \pm 0.061$ | $0.305 \pm 0.007$ | $0.548 \pm 0.063$ | $0.389 \pm 0.015$ |
| GPR | $0.597 \pm 0.110$ | $0.575 \pm 0.059$ | $0.862 \pm 0.016$ | $0.609 \pm 0.014$ |
| MOSES | $0.411 \pm 0.099$ | $0.307 \pm 0.006$ | $0.517 \pm 0.057$ | $0.342 \pm 0.028$ |
| CircuITS | $\underline{0.306 \pm 0.030}$ | $\underline{0.292 \pm 0.000}$ | $\underline{0.475 \pm 0.050}$ | $0.285 \pm 0.001$ |
| MargFlow | $0.389 \pm 0.154$ | $0.310 \pm 0.016$ | $0.480 \pm 0.047$ | $\underline{0.278 \pm 0.003}$ |
| MargFlow* | $0.316 \pm 0.082$ | $\mathbf{0.290 \pm 0.001}$ | $\mathbf{0.425 \pm 0.031}$ | $\mathbf{0.251 \pm 0.001}$ |

Table 16: Comparing models w.r.t. **Energy Score** (multivariate) across datasets. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**, the second best is underlined. Rows marked with * use a MargFlow that is optimized for CRPS instead of likelihood. GCNP is the single-component ($K = 1$) instance of GMCP, i.e. a conditional Gaussian copula (Markou et al., 2022); it shares the frozen MargFlow marginals of the row above it. †Our adaptation of TACTiS-2 to sparse IMTS (Appendix L.7).

| **Model** | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| NeuralFlows | $0.661 \pm 0.059$ | $1.691 \pm 0.001$ | $1.381 \pm 0.033$ | $0.982 \pm 0.009$ |
| ProFITi | $\mathbf{0.452 \pm 0.044}$ | $\mathbf{0.879 \pm 0.303}$ | $1.606 \pm 0.168$ | $0.808 \pm 0.003$ |
| MOSES | $0.552 \pm 0.044$ | $1.599 \pm 0.013$ | $1.353 \pm 0.033$ | $0.906 \pm 0.029$ |
| CircuITS | $0.468 \pm 0.020$ | $1.613 \pm 0.000$ | $1.489 \pm 0.040$ | $0.927 \pm 0.050$ |
| TACTiS-2† | $0.477 \pm 0.052$ | $1.613 \pm 0.018$ | $1.305 \pm 0.031$ | $0.825 \pm 0.007$ |
| MargFlow | $0.483 \pm 0.054$ | $1.606 \pm 0.012$ | $1.349 \pm 0.033$ | $0.826 \pm 0.005$ |
| GCNP | $0.482 \pm 0.055$ | $1.594 \pm 0.011$ | $1.341 \pm 0.033$ | $0.824 \pm 0.005$ |
| GMCP | $0.482 \pm 0.054$ | $1.593 \pm 0.010$ | $1.340 \pm 0.033$ | $0.826 \pm 0.005$ |
| MargFlow* | $0.472 \pm 0.043$ | $1.585 \pm 0.002$ | $1.277 \pm 0.030$ | $0.789 \pm 0.002$ |
| GMCP* | $0.472 \pm 0.043$ | $1.576 \pm 0.002$ | $\mathbf{1.270 \pm 0.029}$ | $\mathbf{0.788 \pm 0.002}$ |

Table 17: Comparing models w.r.t. **CRPS** (Continuous Ranked Probability Score) on marginals across datasets. Lower values indicate better performance. We present the mean and standard deviation of 5 runs. The best model is marked in **bold**, the second best is underlined. Rows marked with * use a MargFlow that is optimized for CRPS instead of likelihood.

| **Model** | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| NeuralFlows | $0.306 \pm 0.028$ | $0.277 \pm 0.003$ | $0.308 \pm 0.004$ | $0.281 \pm 0.004$ |
| ProFITi | $\mathbf{0.182 \pm 0.007}$ | $0.271 \pm 0.003$ | $0.319 \pm 0.003$ | $0.279 \pm 0.012$ |
| MOSES | $0.220 \pm 0.019$ | $0.260 \pm 0.002$ | $0.296 \pm 0.005$ | $0.245 \pm 0.010$ |
| CircuITS | $\mathbf{0.182 \pm 0.010}$ | $0.252 \pm 0.001$ | $0.286 \pm 0.010$ | $0.221 \pm 0.001$ |
| MargFlow | $0.187 \pm 0.022$ | $0.251 \pm 0.001$ | $0.289 \pm 0.005$ | $0.217 \pm 0.001$ |
| MargFlow* | $0.189 \pm 0.016$ | $\mathbf{0.250 \pm 0.000}$ | $\mathbf{0.271 \pm 0.004}$ | $\mathbf{0.208 \pm 0.001}$ |

**What the Energy Score Does and Does Not Resolve** The Energy Score is strictly proper, meaning its expected value under the data-generating distribution is minimized by that exact distribution, and by no other (for distributions with finite first moment; Gneiting & Raftery, 2007). The true predictive distribution is therefore its unique minimizer. Strict propriety is a statement about that minimizer alone. It says nothing about how steeply the score rises as a model is perturbed away from the truth, and nothing about whether it rises at the same rate for a perturbation of the marginals as for one of the dependence structure. Those rates decide whether a difference in ES between two models is readable at all, and they have been measured. Marcotte et al. (2023) map out the *regions of reliability* of ES-based comparisons as a function of dimension and of the number of samples used in the estimator Eq. (61), and find the reliable region to be considerably smaller than common practice assumes. Our forecasting targets reach $N = 85$ points per instance (Table 9), so this is the regime we are evaluating in.

Table 18: What the Energy Score resolves, decomposed from Tables 3 and 16. The *copula* column is the improvement obtained by attaching GMCP to MargFlow, which alters the dependence structure and provably leaves the marginals untouched; the *marginals* column is the improvement obtained by fitting MargFlow on CRPS instead of likelihood (Appendix O), which alters the marginals and leaves the copula stage untouched. Positive values are improvements. The copula is worth $0.17$–$0.38$ njNLL and next to nothing in ES, while a change that touches no dependence structure at all moves the ES several times further.

| | $\Delta$ Energy Score | | $\Delta$ njNLL |
|---|---|---|---|
| **Dataset** | copula | marginals | copula |
| USHCN | 0.001 | 0.011 | 0.166 |
| PhysioNet'12 | 0.013 | 0.021 | 0.303 |
| MIMIC-III | 0.009 | 0.072 | 0.330 |
| MIMIC-IV | 0.000 | 0.037 | 0.382 |

Table 18 makes the effect visible in our own results, by decomposing the numbers of Table 16 into the two changes that produce them. Attaching GMCP to MargFlow changes the dependence structure and nothing else, since marginalization consistency leaves the marginals where MargFlow put them (Appendix B), which Table 4 confirms empirically. Switching MargFlow to the CRPS objective of Appendix O changes the marginals and leaves the copula stage untouched. The first change is worth between $0.17$ and $0.38$ njNLL, a gap that separates GMCP from every jointly trained baseline in Table 3. In ES it is worth between $0.000$ and $0.013$, at or below the fold-to-fold standard deviations reported in Table 16. The second change improves no dependence structure at all and is worth up to $0.072$ of ES (on MIMIC-III). This is an 8 times higher increase compared to adding the copula.

On these datasets the ES column therefore ranks marginal fits, and a model's position in it says little about the quality of its copula. GMCP with the CRPS-trained marginals of Appendix O attains the best ES on MIMIC-III and MIMIC-IV. On PhysioNet'12, it comes in second after ProFITi, a model that does not guarantee marginalization consistency and has a very strong encoder in GraFITi (Yalavarthi et al., 2024), which we cannot use as it does not separate targets.

## O TRAINING MARGFLOW ON CRPS

The log-likelihood is one proper scoring rule among many, and which rule a model is fitted with decides what the fit is good at (the *optimum score estimation* view of Gneiting & Raftery (2007)). Our own numbers show the gap. MargFlow attains the lowest mNLL on all four datasets (Table 2), yet on CRPS it is only on par with CircuITS (Table 17), which it beats comfortably in likelihood. The marginals of any copula process (incl. GMCP) are a separate module with their own objective, so we can act on that. Fitting MargFlow with CRPS directly leaves the rest of the construction untouched. The rows marked $^*$ in Tables 15 to 17 are produced this way.

**CRPS is available in closed form.** We train on the integral form of Eq. (62), the quadratic distance between the predictive CDF and the step function at the observation, taken at $\hat{F}_n = F_n^{\mathrm{marg}}$. The DSF head of MargFlow *is* that conditional CDF (Appendix F), and a single forward pass returns $F_n^{\mathrm{marg}}(z)$ for any $z$. Evaluating it therefore requires neither sampling from the model nor inverting the flow, and it is differentiable in the flow parameters $\theta_n$ by ordinary backpropagation through that forward pass. That is cheap enough to train on.

**Computing the integral.** We approximate the integral numerically. We evaluate the CDF on a grid of points around the observation and apply the trapezoid rule. One detail matters. The integrand is $F_n^{\mathrm{marg}}(z)^2$ to the left of the observation and $\big(1 - F_n^{\mathrm{marg}}(z)\big)^2$ to the right of it, so it *jumps* at $z = y_n^{\mathrm{tgt}}$. A grid that runs across this jump is markedly less accurate. We therefore integrate the two sides separately, each up to a distance $w$ from the observation,

$$\widetilde{\mathrm{CRPS}}(F_n^{\mathrm{marg}}, y_n^{\mathrm{tgt}}) \;=\; \int_0^w F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}} - s)^2 \,\mathrm{d}s \;+\; \int_0^w \big(1 - F_n^{\mathrm{marg}}(y_n^{\mathrm{tgt}} + s)\big)^2 \,\mathrm{d}s, \qquad (64)$$

and compute each on $J + 1$ equally spaced points. Since the targets are normalized to zero mean and unit variance (Appendix L), we take $w = 10$ (ten standard deviations) and $J = 200$, i.e. 401 CDF evaluations per target point.

**Why a jointly trained model has no analogue of this.** The freedom exercised above belongs to the decoupled setup and does not transfer. A jointly trained model has a single objective covering marginals and dependence at once. CRPS cannot take its place, because it is a univariate score, so substituting it would delete every term that concerns the dependence structure and the model would stop being a joint model at all. Nor can the two simply be added, since a log-density and a CRPS differ in units and scale and combining them introduces a weight that has no principled value. The multivariate analogue, the Energy Score, is not available in closed form for the models at hand, and its sample estimator Eq. (61) requires differentiable sampling from the joint predictive distribution.

# P ABLATIONS

We describe two ablations of GMCP. The first, Mix-GC, replaces the Gaussian Mixture Copula by a mixture of Gaussian copulas and is reported alongside the baselines in the main tables. The second lifts the one restriction that marginalization consistency imposes on GMCP, namely that the mixture weights must not see the target set.

## P.1 THE MIX-GC ABLATION

Recall from Section 2 that a GM-C is induced by a single multimodal base density $f^{\mathrm{GMM}}(\mathbf{z}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j)$ on $\mathbb{R}^N$ via Eq. (1) using the GMM's own marginal CDFs. A Mix-GC is instead a convex combination of $K$ independent Gaussian copulas,

$$\mathbf{c}(\mathbf{u}) \;=\; \sum_{j=1}^{K} \pi_j \, \mathbf{c}_{R_j}(\mathbf{u}), \tag{65}$$

where $\mathbf{c}_{R_j}$ is the Gaussian copula induced by a correlation matrix $R_j$. Appendix A discusses what this ablation gives up in expressivity and illustrates it on two toy densities. Each Gaussian copula component is centered at the origin in latent space and uses standard-normal marginals. The per-component means $\mu_j$ and standard deviations $\sigma_{j,n}$ of the GM-C construction are therefore no longer present. Only the off-diagonal correlation structure remains.

**Architectural changes.** The GMCP-Mix-GC ablation reuses every component of GMCP described in Section 3 unchanged, except for the parts that produce the per-component means and standard deviations. Concretely:

- The encoder, the global summary $\bar{\mathbf{h}}$, and the per-target embeddings $\mathbf{e}_n$ are identical to GMCP.
- The mixture-weight head $\pi = \mathrm{softmax}(\mathrm{MLP}_\pi(\bar{\mathbf{h}}))$ is identical to GMCP.
- The mean function $\phi_\mu$ and the standard-deviation function $\phi_\sigma$ of Eq. (6) are *removed*. Each Gaussian copula component is centered at zero with unit variance by definition, so $\mu_{j,n}$ and $\sigma_{j,n}$ are not free parameters.
- The correlation network that produces the per-component feature matrices $V_j \in \mathbb{R}^{N \times H}$ is unchanged. We normalize the feature vectors $\mathbf{v}_{j,n}$ as in Section 3 and form the correlation matrices $R_j = \tilde{V}_j \tilde{V}_j^\top + \mathrm{diag}\big(1 - \|\tilde{\mathbf{v}}_{j,n}\|^2\big)_{n=1}^{N}$ as in GMCP. The construction then stops, since $R_j$ now carries the full per-component covariance and no scale parameters multiply it.

The hyperparameter search space therefore matches that of GMCP (Table 11) with the `corr_net_hidden_dim` entry reinterpreted as the hidden dimension of the only remaining MLP.

## P.2 TARGET-DEPENDENT MIXTURE WEIGHTS

GMCP imposes one restriction in order to be marginalization consistent. The mixture weights $\pi$ are inferred from the context alone, $\pi = \phi_\pi(\mathcal{D}^{\mathrm{ctx}})$, and never see the target set $\mathbf{X}^{\mathrm{tgt}}$ (Section 3, Step 3 of the proof of Theorem 1). Every other parameter function is free to read its own target input. We measure what the restriction costs by lifting it.

**The ablation.** Recall from Eq. (31) that GMCP obtains the global summary $\bar{\mathbf{h}}$ by pooling the per-channel states $\tilde{\mathbf{H}}$, which are a function of $\mathcal{D}^{\mathrm{ctx}}$ alone, with a single learnable attention query $\mathbf{q}$. The ablation uses mean-pooling on the *target encodings* instead,

$$\bar{\mathbf{e}} \;=\; \frac{1}{N}\sum_{n=1}^{N} \mathbf{e}_n \;=\; \frac{1}{N}\sum_{n=1}^{N} \mathrm{Enc}\big(x_n^{\mathrm{tgt}}, \mathcal{D}^{\mathrm{ctx}}\big) \;\in\; \mathbb{R}^D, \qquad \pi \;=\; \mathrm{softmax}\big(\mathrm{MLP}_\pi(\bar{\mathbf{e}})\big) \;\in\; \Delta^K. \tag{66}$$

Nothing else changes. The encoder is identical up to and including the target encodings of Eq. (30), the mixture-weight head $\mathrm{MLP}_\pi$ keeps its architecture, and $\phi_\mu$, $\phi_\sigma$ and $\phi_V$ are untouched, so $\mu_j$, $\sigma_j$ and $\Sigma_j$ are formed as in Section 3.

Table 19: Test **njNLL** (lower is better) of GMCP, whose mixture weights see the context only, against the target-dependent-weight ablation of Eq. (66), which is marginalization inconsistent. Mean and standard deviation over the five splits; both arms share the same frozen MargFlow marginals. Neither row is marked best.

| **Mixture weights** | USHCN | PhysioNet'12 | MIMIC-III | MIMIC-IV |
|---|---|---|---|---|
| GMCP: $\pi = \phi_\pi(\mathcal{D}^{\mathrm{ctx}})$ | -4.135 $\pm$ 0.304 | -0.745 $\pm$ 0.019 | -0.835 $\pm$ 0.218 | -2.201 $\pm$ 0.058 |
| Ablation: $\pi = \phi_\pi(\mathcal{D}^{\mathrm{ctx}}, \mathbf{X}^{\mathrm{tgt}})$ | -4.143 $\pm$ 0.331 | -0.740 $\pm$ 0.022 | -0.846 $\pm$ 0.217 | -2.207 $\pm$ 0.058 |
| Difference (ablation $-$ GMCP) | $-0.008$ | $+0.005$ | $-0.011$ | $-0.006$ |

**Results.** Table 19 shows that conditioning the mixture weights on the target set moves the njNLL very little. The two approaches differ by at most $0.011$ njNLL, on MIMIC-III, and the sign of the difference is not uniform. The ablation is nominally ahead on USHCN, MIMIC-III and MIMIC-IV, and behind on PhysioNet'12.

## Q SENSITIVITY ANALYSIS

We study the sensitivity of GMCP to the two hyperparameters that are specific to the GM-C copula module, namely the number of mixture components $K$ and the per-component correlation-network hidden dimension $H$ (i.e. the width of the MLPs that emit $\mu_j$, $\sigma_j$, and the rows of the Gram-matrix factors $V_j \in \mathbb{R}^{N \times H}$, see Table 11). For each sweep, we vary the analyzed hyperparameter while fixing all remaining hyperparameters to the values selected by the random search described in Appendix L.3. Each configuration is trained from scratch on all five splits, and we report the mean test njNLL across the five splits.

**Number of mixture components $K$.** Figure 6 sweeps $K \in \{1, 3, 5, 7, 10\}$ on all four datasets. The case $K = 1$ collapses the GM-C to a Gaussian copula and is inferior on every dataset. Moving from $K = 1$ to $K = 3$ produces a large drop in njNLL on all four benchmarks, and the curve flattens beyond $K \geq 3$.

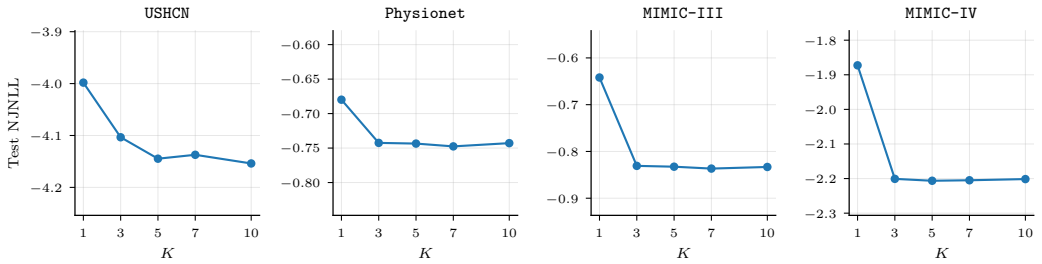


Figure 6: Sensitivity of GMCP to the number of mixture components $K$. Test njNLL (lower is better) averaged over the five splits, with all remaining hyperparameters fixed to the values selected by the random search of Appendix L.3.

**Correlation-network hidden dimension $H$.** Figure 7 sweeps $H \in \{16, 32, 64, 128\}$. Across all four datasets the test njNLL is flat in $H$. The only visible deviation is a small uptick on USHCN at $H = 128$, which we attribute to mild overfitting on the smallest dataset.

$H = 16$ suffices on MIMIC-III and MIMIC-IV even though many test instances have $N > H$ in those datasets (the maximum target count $N$ is 85 for MIMIC-III and 79 for MIMIC-IV; see Table 9). In this regime, each per-component covariance $\Sigma_j$ is built from the rank-$H$ Gram matrix of $\tilde{V}_j \in \mathbb{R}^{N \times H}$ in Eq. (8), so its off-diagonal block has rank at most $H$. That is the limitation we flag in the main text, that GMCP cannot represent full-rank or near-perfect dependence when $H \ll N$. The flat sensitivity curves show that the rank ceiling does not bind in practice. It may even help, since not all variables depend on each other.

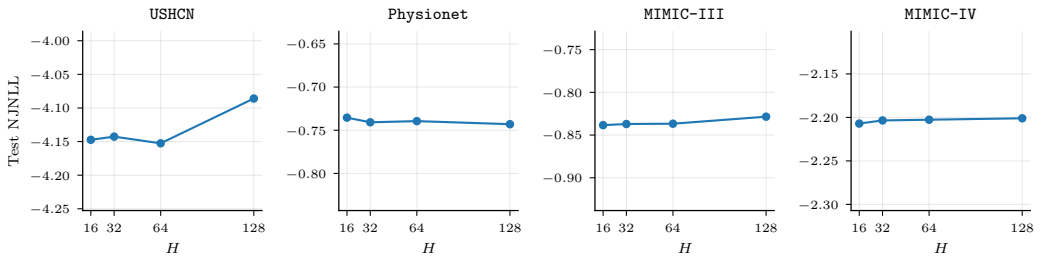


Figure 7: Sensitivity of GMCP to the correlation-network hidden dimension $H$.

## R Learning Curves

Figure 8 gives the validation mNLL learning curves of GMCP-Joint and of the decoupled MargFlow + GMCP setup on MIMIC-III, as a complement to the marginal-quality analysis in Section 5. GMCP-Joint shares the architecture of GMCP and trains all weights jointly under njNLL. The decoupled setup fits MargFlow in isolation first, then trains the copula on top of the frozen marginals. All models were trained until the validation njNLL stopped improving for 30 epochs.

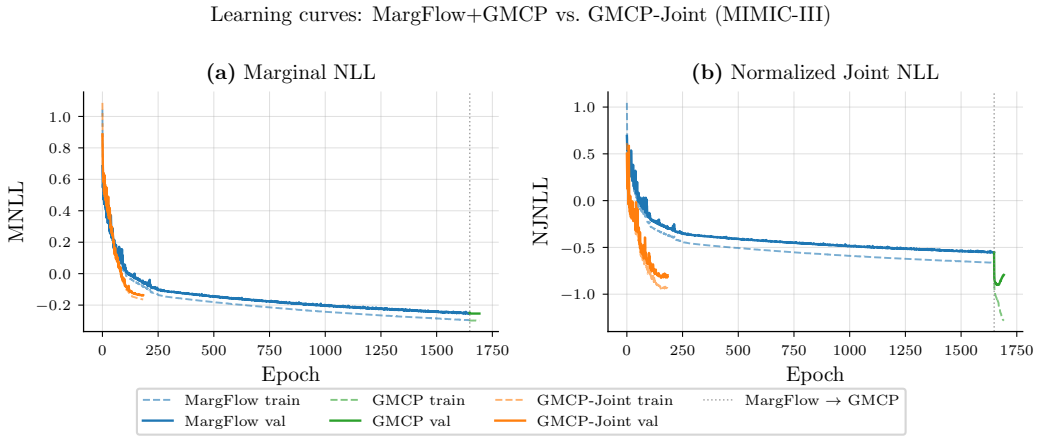


Figure 8: Validation learning curves of GMCP-Joint and the decoupled MargFlow + GMCP setup on MIMIC-III. Training the marginals in isolation reaches a lower marginal NLL than training all weights jointly under njNLL.