
Marginalization Consistent Mixture of Separable Flows for Probabilistic Irregular Time Series Forecasting

Vijaya Krishna Yalavarthi
ISMLL
University of Hildesheim
Germany
yalavarthi@ismll.de

Randolf Scholz
ISMLL
University of Hildesheim
Germany
scholz@ismll.de

Kiran Madhusudhanan
ISMLL
University of Hildesheim
Germany
madhusudhanan@ismll.de

Stefan Born
Institute of Mathematics
TU Berlin
Germany
born@math.tu-berlin.de

Lars Schmidt-Thieme
ISMLL
University of Hildesheim
Germany
schmidt-thieme@ismll.de

Abstract

Probabilistic forecasting models for joint distributions of targets in irregular time series are a heavily under-researched area in machine learning with, to the best of our knowledge, only three models researched so far: GPR, the Gaussian Process Regression model [16], TACTiS, the Transformer-Attentional Copulas for Time Series [14, 2] and ProFITi [43], a multivariate normalizing flow model based on invertible attention layers. While ProFITi, thanks to using multivariate normalizing flows, is the more expressive model with a better predictive performance, we will show that it suffers from marginalization inconsistency: it does not guarantee that the marginal distributions of a subset of variables in its predictive distributions coincide with the directly predicted distributions of these variables. Also, TACTiS does not provide any guarantees for marginalization consistency.

We develop a novel probabilistic irregular time series forecasting model, Marginalization Consistent Mixtures of Separable Flows (*moses*), that mixes several normalizing flows with (i) Gaussian Processes with full covariance matrix as source distributions and (ii) a separable invertible transformation, aiming to combine the expressivity of normalizing flows with the marginalization consistency of Gaussians. In experiments on four different datasets we show that *moses* outperform other state-of-the-art marginalization consistent models, perform on par with ProFITi, but different from ProFITi, guarantees marginalization consistency.

1 Introduction

In many real-world domains ranging from health to astronomy time-variant data is measured in an irregular fashion: different channels are measured at different times and usually not on a regular grid. Besides mere point estimates, i.e., just the **expected target values** for some channels at some future time points, one is usually interested in a probabilistic/distributional forecast of **a distribution of target values**. A full predicted distribution of target values provides way more information about the targets. For example, when bioengineers are growing bacteria in tanks they are interested in predicting the oxygen levels. The expected value is already interesting, but if additionally we can

predict more fine-grained the whole distribution of possible oxygen levels this allows them now to quantify the risk that the oxygen level falls below some critical threshold and all bacteria die; the expected value alone is not sufficient to do so. Consequently, in recent years several models have been developed for probabilistic forecasting of irregular time series [10, 11, 4, 35].

However, all these models are limited to forecasting the distribution of the **value of a single channel at a single time**. But decision makers often require information about several variables and their interaction, that is, about the predicted joint distribution of several channels at several times. In our initial example, the bioengineers will be interested not just in the forecasts of the oxygen levels and the biomass, but also in their interaction: a risk for either low biomass or low oxygen levels might be tolerable, but a risk that both occur at the same time will require their intervention.

Probabilistic forecasting models for joint distributions of targets in irregular time series are a heavily under-researched area in machine learning with, to the best of our knowledge, only three models researched so far: GPR, the Gaussian Process Regression model [16] uses a multivariate normal distribution as predictive distribution, TACTiS, the Transformer-Attentional Copulas for Time Series [14, 2] uses a copula with normalizing flows for the marginal distributions, and ProFITi [43] uses a multivariate normalizing flow model based on invertible attention layers. ProFITi, thanks to using normalizing flows, is able to learn way more expressive predictive distributions, for example, it could express multi-modal predictive distributions, while GPR being limited to multivariate Gaussians cannot do that. Consequently, it outperforms GPR by a wide margin in experiments.

Models for probabilistic forecasting of joint target distributions in irregular time series have to be able to express distributions over a varying number of variables, as by the nature of irregular time series a forecast for say the next 3 minutes for some training instances may involve 5 observations, for others 7 observations. However, this capability to provide predictive distributions for a varying number of observations/variables introduces a new issue: to get the predictive joint distribution for values of channels at some time points, we can query the model multiple ways: (i) we can just ask for the predictive distribution of those variables directly or (ii) we can ask for a predictive distribution of these variables plus some further variables and then marginalize out those additional variables again from the answer provided by the model. Using a consistent model we expect to get the same answer either way. An important special case is that the single variable marginals of all predictive distributions provided by a model should agree with the single variable predictive distributions it provides when we query it directly. We call this property **marginalization consistency**.

We will show in this paper that the currently best performing probabilistic forecasting model for joint target distributions in irregular time series, ProFITi, neither provides any guarantees to be marginalization consistent nor empirically happens to be marginalization consistent on the standard datasets used in experiments. TACTiS does not provide guarantees for marginalization consistency either. In contrast, GPR marginalization consistent, but has worse predictive performance.

From this starting point we will construct a novel model that combines the ideas of Gaussian Processes and normalizing flows in a way completely different from ProFITi and GPR, to achieve both, guaranteed marginalization consistency and high predictive accuracy (see Figure. 3).

Overall are our contributions as follows:

1. We propose a measure for the degree of marginalization consistency of models for joint distributions with varying size, the Wasserstein Distance between the (possibly numerically) marginalized predicted joint distribution of several variables and the directly predicted marginal distribution.
2. We show that the currently best performing model for probabilistic irregular time series forecasting, ProFITi, does not provide any guarantees for marginalization consistency, and in experiments we show furthermore that it actually suffers from marginal inconsistency.
3. We develop a novel probabilistic irregular time series forecasting model, Marginalization Consistent Mixtures of Conditional Flows (moses), that mixes several normalizing flows with (i) Gaussian Processes **with full covariance matrix** as source distributions (but the usual identity matrix) and (ii) a **separable** invertible transformation (i.e., for each dimension separately, instead of the usual multivariate ones), aiming to combine the expressivity of normalizing flows with the marginalization consistency of mixtures of Gaussian Processes.

4. We prove that Marginalization Consistent Mixtures of Separable Flows are guaranteed marginalization consistent.
5. In experiments on four different datasets we show that Marginalization Consistent Mixtures of Separable Flows outperform other state-of-the-art marginalization consistent models, perform on par with ProFITi, but different from ProFITi, guarantee marginalization consistency.
6. Our source code can be found at github.com.

2 Preliminaries

We make use of the triplet representation of an irregular time series for probabilistic forecasting [19]. In this representation, an **irregularly sampled time series** X is a sequence of N -many triplets:

$$X := ((t_n^{\text{OBS}}, c_n^{\text{OBS}}, v_n^{\text{OBS}}))_{n=1:N} \in \text{Seq}(\mathcal{X}) \quad (1)$$

where $t_n^{\text{OBS}} \in \mathbb{R}$ is the observation time point, and $v_n^{\text{OBS}} \in \mathbb{R}$ is the observed value in channel $c_n^{\text{OBS}} \in \{1, \dots, C\}$. A **time series query** Q is a sequence of K -many pairs:

$$Q := ((t_k^{\text{QRY}}, c_k^{\text{QRY}}))_{k=1:K} \in \text{Seq}(\mathcal{Q}) \quad (2)$$

where $t_k^{\text{QRY}} \in \mathbb{R}$ is the future time point and $c_k^{\text{QRY}} \in \{1, \dots, C\}$ is the queried channel. A **forecasting answer** y is a sequence of scalars: $y = (y_1, \dots, y_K)$, where y_k is the forecasted value in channel c_k^{QRY} at time t_k^{QRY} . Here, $\text{Seq}(\mathcal{X})$ denotes the space of finite sequences over \mathcal{X} .

For forecasting, all the query time points are after the observations: $\min_{k=1:K} t_k^{\text{QRY}} > \max_{n=1:N} t_n^{\text{OBS}}$. The problem of probabilistic irregular time series forecasting is to find a model \hat{p} that can predict the joint multivariate distribution $\hat{p}(y \mid Q, X)$ of the answers y , given the query points Q and observed series X . Both the context length $N = |X|$ and the query length $K = |Q|$ are allowed to be dynamic (R1).

As the time stamp and channel id. are included in each sample, the order of the samples does not matter, and hence any model prediction should be independent of it (R2). Moreover, for a subquery Q' of Q , there are two different ways to predict the joint distribution: either by marginalizing the predicted distribution of the complete query Q , or by predicting the joint distribution of the subquery Q' directly. The two predictions should be equivalent (R3).

Requirements. A marginalization consistent probabilistic irregularly sampled time series forecasting model must satisfy the following requirements:

R1 joint multivariate prediction.

The model \hat{p} can predict the joint distribution across multiple time steps of a multivariate time series for arbitrary sizes of both the query $K = |Q|$ and context $N = |X|$. That is, it realizes a function that maps from sequence space to the space of probability distribution $\text{Prob}(\text{Seq}(\mathbb{R}))$.

$$\begin{aligned} \hat{p}: \text{Seq}(\mathcal{Q}) \times \text{Seq}(\mathcal{X}) &\longrightarrow \text{Prob}(\text{Seq}(\mathbb{R})), \\ (Q, X) &\longmapsto \hat{p}(y_1, \dots, y_K \mid Q_1, \dots, Q_K, X_1, \dots, X_N) \end{aligned} \quad (3)$$

R2 permutation invariance.

The predicted density should be invariant under permutations of both the query or context:

$$\hat{p}(y \mid Q, X) = \hat{p}(y^\pi \mid Q^\pi, X^\tau) \quad \forall \pi \in S_{|Q|}, \tau \in S_{|X|} \quad (4)$$

R3 marginalization consistency/projection invariance.

Predicting the joint density for the sub-query Q_{-k} given by removing the k -th item should yield the same result as marginalizing the k -th variable from the complete query.

$$\hat{p}(y_{-k} \mid Q_{-k}, X) = \int_{\mathbb{R}} \hat{p}(y \mid Q, X) dy_k \quad (5)$$

This generalizes to any subset $K_S \subseteq \{1, \dots, K\}$.

For a model satisfying R1-R3, we will only have to marginalize if we try to validate the marginalization consistency. For this validation we added requirement R5. Yalavarthi et al. [43] discussed R1 and R2, but did not consider R3. We argue that irregularly sampled time series is realization of a stochastic process and R3 is a fundamental property of any model that mimics it.

Theorem 1. *Any model that satisfies R1-R3 realizes an \mathbb{R} -valued stochastic process over the index set $T = \mathbb{R} \times \{1, \dots, C\}$.*

Proof. This is a direct application of Kolmogorov’s extension theorem [28]

3 Related Work

There have been multiple works that deal with point forecasting of irregular time series [1, 6, 7, 42]. In this work we deal with probabilistic forecasting of irregular time series. Models such as NeuralFlows [4], GRU-ODE [10], and CRU [35] predict only the marginal distribution for a single time stamp. Additionally, interpolation models like HetVAE [37] and Tripletformer [41] can also be applied for probabilistic forecasting. However, they also produce only marginal distributions. All the above models assume underlying distribution is Gaussian which is not the case for lots of real-world datasets. On the other hand, Gaussian Process Regression [16], TACTiS/TACTiS-2, the Transformer-Attentional Copulas for Time Series [14, 2] and ProFITi [43] can predict proper joint distributions. TACTiS is a copula model and ProFITi a conditional normalizing flow model, both can predict any arbitrary distribution. But as both use self-attention on the queries in their encoders, their parametrization of their copula and flow, respectively, depends on all queries in a complex way, and thus they cannot provide any marginalization guarantees.

There have been works on models for tractable and consistent marginals for fixed number of variables such as tabular data. Probabilistic Circuits [8] create a sum-prod network on the marginal distributions in such a way that marginals are tractable and consistent. Later, univariate normalizing flows were added to the leaf nodes of the circuit for better expressivity by Sidheekh et al. [38]. However, it is not trivial to extend such circuits to deal with sequential data of variable size. Gaussian Mixture Models (GMMs) [15] are often used only for unconditional density estimation, but can be extended to conditional density estimation. They can provide tractable and consistent marginal distributions. However, GMMs are not expressive enough and often require a very large number of components to approximate even simple distributions, as shown in Figure 1. Note that normalizing flow models such as [12, 29, 30] neither provide tractable marginals nor are applicable to varying number of variables.

Since our model is a mixture of normalizing flows, we discuss existing works in this direction as well. Pires and Figueiredo [31] and Ciobanu [9] proposed a mixture of flows with transformation functions, such as affine coupling or masked autoregressive functions for density estimation. Postels et al. [32] apply a mixture of flows for reconstruction tasks. All three models are developed for fixed length sequences and cannot accommodate dynamic size, and additionally, their marginals are intractable.

4 Constructing Marginalization Consistent Conditional Distributions

Our goal is to build a model for the conditional joint distribution $p(y_1, \dots, y_K \mid Q_1, \dots, Q_k, X)$, as in Equation (3). Since the model should satisfy R3, it follows that the marginal distribution of y_k must only depend on Q_k and X .

Separably Parametrized Gaussians. The arguably most simple model for a conditional permutation invariant distribution for variably many variables is the family of multivariate Gaussian $\mathcal{N}(y \mid \mu(x), \Sigma(x))$, whose conditional mean function $\mu(x)$ and conditional covariance function $\Sigma(x)$ are separable, i.e.:

$$\mu_k = \tilde{\mu}(Q_k, X) \quad \Sigma_{k,\ell} = \tilde{\Sigma}(Q_k, Q_\ell, X) \quad (6)$$

With mean function $\tilde{\mu}: \mathcal{Q} \times \text{Seq}(\mathcal{X}) \rightarrow \mathbb{R}$ and a covariance function $\tilde{\Sigma}: \mathcal{Q} \times \mathcal{Q} \times \text{Seq}(\mathcal{X}) \rightarrow \mathbb{R}$, a setup very well known from Gaussian processes. Such a separably parametrized multivariate Gaussian is marginalization consistent by design, as marginalizing a Normal distribution boils down to selecting the relevant rows and columns of the covariance matrix and the corresponding elements of the mean vector.

However, Gaussian Processes form a restrictive class of models, as any joint distribution of variables is Gaussian. To model more complex distributions, we need to consider more expressive models. Normalizing flows are a popular choice for this task [34, 30].

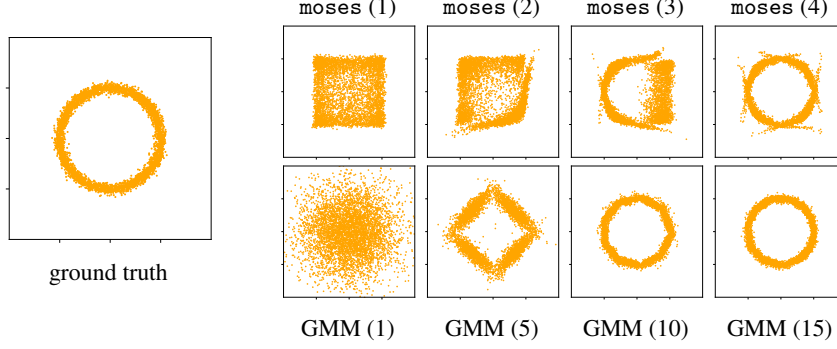


Figure 1: (Top) Importance of multiple flow components: *moses*(1) cannot represent the correct distribution, but *moses*(4) can. (Bottom) Limitation of Gaussian Mixture Models: GMM needs 15 components to match the distribution of *moses*(4).

Separable Normalizing Flows. Normalizing flows model distributions by transforming a source distribution p_Z on \mathbb{R}^K by means of an invertible transformation $f: \mathbb{R}^K \rightarrow \mathbb{R}^K$. Then the target distribution, the distribution of the image of f , can be concisely described by the transformation theorem for densities

$$p_Y(y) := p_Z(f^{-1}(y; \theta)) \cdot \left| \det \left(\frac{\partial f^{-1}(y; \theta)}{\partial y} \right) \right| \quad (7)$$

Existing approaches to normalizing flows use very simple source distributions, typically a multi-variate standard normal $p_Z(z) := \mathcal{N}(z | 0, \mathbb{I})$, and model interactions between variables by means of the transformation [34, 30]. Current approaches for conditional normalizing flows for a variadic number of variables followed the same approach and tackled the problem by engineering expressive transformations between vectors of same size, for any size [25, 3, 43]. For example, ProFITi uses an invertible attention mechanism.

All these models in general will not have a guarantee for marginalization consistency. To the best of our knowledge, there is no simple condition on the transform that would provide such a guarantee.

We therefore propose a drastic change, reversing the standard approach for normalizing flows: to combine (i) simple, separable transforms with (ii) a richer source distribution, namely a Gaussian Process with full covariance matrix. This way interactions between variables cannot be represented by the transformation anymore, but they can be represented by the covariance of the source distribution.

Lemma 1. A conditional normalizing flow model over \mathbb{R}^K or $\text{Seq}(\mathbb{R})$ is called *separable*, if it can be expressed in the form

$$f(z | Q, X) = (\phi(z_1 | Q_1, X), \dots, \phi(z_K | Q_K, X)) \quad (8)$$

for some univariate function $\phi: \mathbb{R} \times \mathcal{Q} \times \text{Seq}(\mathcal{X}) \rightarrow \mathbb{R}$, that is invertible in the first argument. Any model that consists of such a separable flow transformation, combined with a marginalization consistent model for the source distribution, is itself marginalization consistent. (Proof: see Appendix A.1)

Conditional Mixtures of Flows. When using separably parametrized Gaussians as source distributions in Lemma 1, and expressive univariate transformations, we can model any kind of marginal as well as rich interactions between variables. However, the model is still restricted in its expressiveness, allowing for variable-wise separable transformations of a unimodal (Gaussian) distribution only. We therefore resort to the most simple way to further increase the expressiveness of the model: we combine several of such separable flows into a mixture. Figure 1 shows that even just a few components can lead to a much more expressive model, in particular comparable to a simple GMM without flow transformations.

Lemma 2. Given probabilistic models $(\hat{p}_d)_{d=1:D}$ that satisfy R1-R3, then a mixture model

$$\hat{p}(y | Q, X) = \sum_{d=1}^D w_d(X) \hat{p}_d(y | Q, X) \quad (9)$$

with permutation invariant weight function $w: \text{Seq}(\mathcal{X}) \rightarrow \Delta^D$, where Δ^D denotes probability simplex in D variables: $\Delta^D := \{w \in \mathbb{R}^D \mid w_d \geq 0, \sum_d w_d = 1\}$, also satisfies R1-R3. Proof in Appendix A.2.

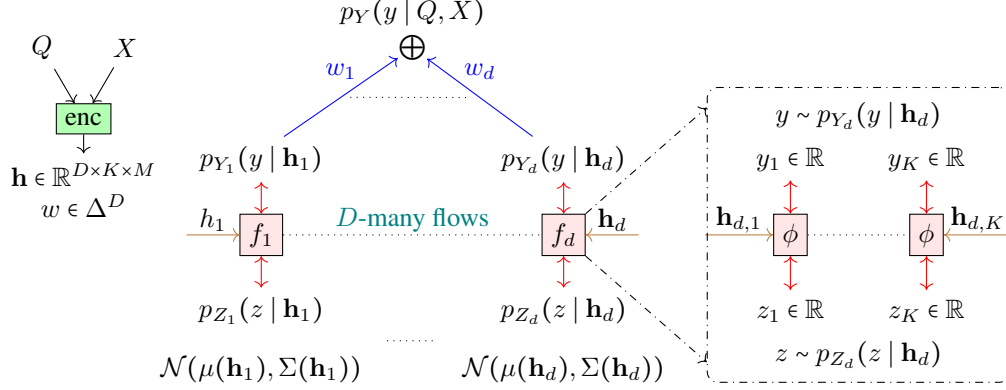


Figure 2: Illustration of proposed mooses. D -many flows (fixed). K -many variables (variable). Encoder (enc) takes X, Q (observed series and query timepoint-channel ids.) as input, and outputs an embedding \mathbf{h} (depends on both X , and Q) and w (depends on X only). μ, Σ of p_{Z_d} are parametrized by \mathbf{h}_d . Flow transformation of p_{Z_d} is parametrized by \mathbf{h}_d . Transformation layer consists of K -many univariate transformations ϕ that transforms z_k of $z \sim p_{Z_d}(z | \mathbf{h}_D)$ to y_k of $y \sim p_d^{\text{LOW}}(y | \mathbf{h}_d)$.

5 Mixtures of Separable Flows (mooses)

Based on the constructions from the last section, we propose to build a marginalization consistent model for forecasting irregular time series in four components (see Figure 2):

1. A separable encoder, consisting of
 - (i) A shared encoding $\mathbf{h}^{\text{OBS}} := \text{enc}^{\text{OBS}}(X; \theta^{\text{OBS}})$ of the observations, used for all queries.
 - (ii) D -many encodings $\mathbf{h}_{d,k} := \text{enc}^{\text{QRY}}(Q_k, X; \theta_d^{\text{QRY}})$ of each query and entire context.
2. D -many Gaussian Processes $p_{Z_d}(z | \mu_d, \Sigma_d)$, each separably parametrized according to (6), by the encoder for queries \mathbf{h}_d .
3. D -many separable normalizing flows \hat{p}_d^{LOW} , one on top of each of the source distributions, whose transformations f_d are also separably parametrized by the encoded queries \mathbf{h}_d .
4. A mixture of the D -many normalizing flows with mixing weights $w := w(\mathbf{h}^{\text{OBS}})$, depending only on the encoded observations \mathbf{h}^{OBS} , but not the queries.

1. Separable Encoder. To encode both the observations $X = ((t_n^{\text{OBS}}, c_n^{\text{OBS}}, v_n^{\text{OBS}}))_{n=1:N}$ and queries $Q = ((t_k^{\text{QRY}}, c_k^{\text{QRY}}))_{k=1:K}$, we apply a positional embedding with learnable parameters $(a_f, b_f)_{f=1:F}$ to the time component [22].

$$\text{pos_embed}(t)_f := \begin{cases} a_f t + b_f & \text{if } f = 1 \\ \sin(a_f t + b_f) & \text{else} \end{cases} \quad (10)$$

And one-hot encodings for the channel component. The value is simply passed through.

$$\mathbf{x} := [\text{pos_embed}(t_n^{\text{OBS}}), \text{one-hot}(c_n^{\text{OBS}}), v_n^{\text{OBS}}]_{n=1:N} \quad (\in \mathbb{R}^{N \times (F+C+1)}) \quad (11a)$$

$$\mathbf{q} := [\text{pos_embed}(t_k^{\text{QRY}}), \text{one-hot}(c_k^{\text{QRY}})]_{k=1:K} \quad (\in \mathbb{R}^{K \times (F+C)}) \quad (11b)$$

The observations are further encoded via self-attention and the queries via cross-attention w.r.t. the encoded observations:

$$\mathbf{h}^{\text{OBS}} := \text{MHA}(\mathbf{x}, \mathbf{x}, \mathbf{x}; \theta^{\text{OBS}}) \quad (\in \mathbb{R}^{N \times M}) \quad (12a)$$

$$\tilde{\mathbf{h}} := \text{MHA}(\mathbf{q}, \mathbf{h}^{\text{OBS}}, \mathbf{h}^{\text{OBS}}; \theta^{\text{QRY}}) \quad (\in \mathbb{R}^{K \times D \cdot M}) \quad (12b)$$

$$\mathbf{h} := \text{reshape}(\tilde{\mathbf{h}}) \quad (\in \mathbb{R}^{D \times K \times M}) \quad (12c)$$

where MHA denotes multihead attention. For the encoding of the queries we use an encoding dimension $D \cdot M$ and reshape each \mathbf{h}_k into D encodings $\mathbf{h}_{d,k}$ of dimension M .

2. D separably parametrized Gaussian source distributions $p_{Z_d}(z | \mu_d, \Sigma_d)$. We model means and covariances simply by a linear and a quadratic function in the encoded queries \mathbf{h}_d :

$$\mu(\mathbf{h}_d) = \mathbf{h}_d \theta^{\text{MEAN}} \implies \mu(\mathbf{h}_d)_k = \mathbf{h}_{d,k} \theta^{\text{MEAN}} \quad (13a)$$

$$\Sigma(\mathbf{h}_d) = \mathbb{I}_K + \frac{(\mathbf{h}_d \theta^{\text{COV}})(\mathbf{h}_d \theta^{\text{COV}})^T}{\sqrt{M'}} \implies \Sigma(\mathbf{h}_d)_{k,l} = \delta_{kl} + \frac{(\mathbf{h}_{d,k} \theta^{\text{COV}})(\mathbf{h}_{d,l} \theta^{\text{COV}})^T}{\sqrt{M'}} \quad (13b)$$

where $\theta^{\text{MEAN}} \in \mathbb{R}^{M \times 1}$ and $\theta^{\text{COV}} \in \mathbb{R}^{M \times M'}$ are trainable weights shared across all D mixture components. \mathbb{I}_K is an identity matrix of size K , and $\delta_{kl} = 1$ if $k = l$ else 0 denotes the Kronecker delta. In Eq. (13b), we divide the inner product with $\sqrt{M'}$ for stable learning, as done in [40]. Since $\Sigma(\mathbf{h}_d)$ is the sum of a positive semi-definite matrix and positive definite matrix, it is guaranteed to be positive definite itself. Note that, \mathbf{h}_d , which is encoded from both context X and queries Q , takes the role of X and Q together in (6).

3. D separable normalizing flows \hat{p}_d^{FLOW} . To achieve separable invertible transformations, any univariate bijective functions can be applied on each variable separately. Spline based functions attracted interest due to their expressive and generalization capabilities [17, 13]. We employ computationally efficient Linear Rational Spline (LRS) transformations [13]. For a conditional LRS $\phi(z_k; \mathbf{h}_{d,k}, \theta^{\text{FLOW}})$, the function parameters such as width and height of each bin, the derivatives at the knots, and λ are computed from the conditioning input $\mathbf{h}_{d,k}$ and some model parameters θ^{FLOW} . θ^{FLOW} helps to project $\mathbf{h}_{d,k}$ to the function parameters, and is common to all the variables $z_{1:K}$ so that the transformation ϕ can be applied for varying number of variables K . Note that we also share the same θ^{FLOW} across all the D -many mixture components as well. For details, see Appendix A.4.

4. Mixture Model. We model the mixture weights via cross attention, using trainable parameters $\beta \in \mathbb{R}^{D \times M}$ as attention queries, and a softmax to ensure the weights to sum to 1:

$$w := \text{softmax}(\text{MHA}(\beta, \mathbf{h}^{\text{OBS}}, \mathbf{h}^{\text{OBS}}; \theta^{\text{MIX}})) \quad (14)$$

Theorem 2. *Our model, mooses, satisfies R1-R3 and hence realizes a stochastic process via Kolmogorov’s Extension Theorem (see Theorem 1). Proof. See Appendix A.3.*

Training. Given a batch of training instances \mathcal{B} , where for each instance, we have Q , X and y , we minimize the normalized joint negative log-likelihood (njNLL) [43]:

$$\mathcal{L}^{\text{njNLL}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{(Q, X, y) \in \mathcal{B}} -\frac{1}{|y|} \log \hat{p}(y | Q, X) \quad (15)$$

where $\theta := (\theta^{\text{OBS}}, \theta^{\text{QRY}}, \theta^{\text{MIX}}, \theta^{\text{MEAN}}, \theta^{\text{COV}}, \theta^{\text{FLOW}})$. njNLL generalizes negative log-likelihood to varying number of variables.

6 Experiments

Toy experiment. We demonstrate the marginalization consistency of mooses using two simple bivariate distributions. The equations to generate these distributions are provided in Appendix B. The task is unconditional density estimation. mooses predicts both joints and marginals correctly, showing marginalization consistency. ProFITi predicts correct joints for the blast dataset and close to correct joints for the circle but is inconsistent in the marginal distribution of the second variable in both examples. This inconsistency arises because ProFITi uses invertible triangular attention to learn the covariance between variables, causing the second variable to always depend on the first. In contrast, the Gaussian Process Regression model is marginalization consistent but cannot predict correct distributions.

Main experiment. We use four real-world datasets: the climate dataset USHCN and three physiological datasets: PhysioNet2012, MIMIC-III, and MIMIC-IV. Following previous work [43], we observe the first 36h and predict the next 3 time steps for the physiological datasets, and observe 3 years and forecast the next 3 time steps for the USHCN dataset. Basic statistics of the datasets are provided in Appendix B. Although we predict the next 3 time steps, the number of observations N and queries K vary (see Table 3). We split the data into Train, Validation, and Test sets in a 70:10:20 ratio. We search for hyperparameters, including the number of mixture components $D \in \{1, 2, 5, 7, 10\}$, attention heads $\in \{1, 2, 4\}$, and latent embedding $M, F \in \{16, 32, 64, 128\}$. All models are implemented in PyTorch and run on GeForce RTX-3090 and 1080i GPUs.

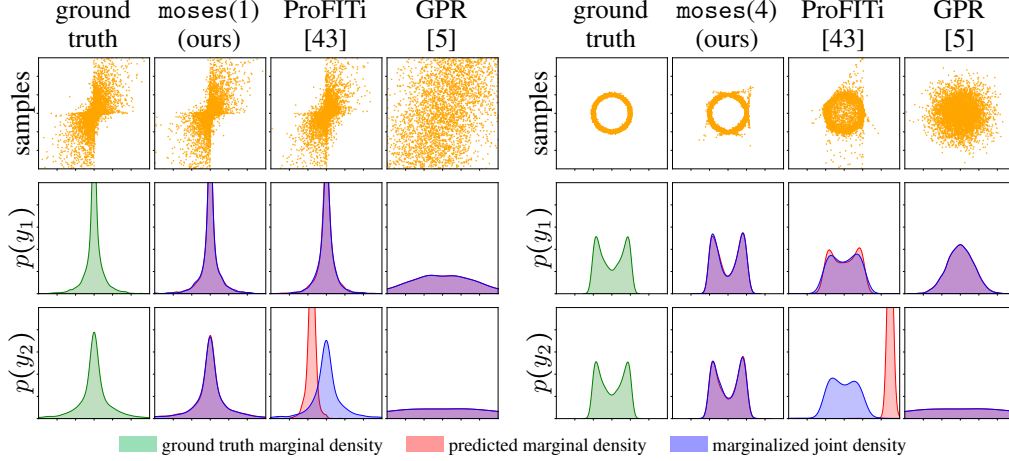


Figure 3: Demonstration of marginal consistency for moses (ours), ProFITi [43], and Gaussian Process Regression [5] on two toy datasets: blast (left) and circle (right). ProFITi is inconsistent w.r.t. the marginals of the second variable y_2 whereas moses is consistent with respect to marginals of both y_1 and y_2 . moses(D) indicates D -many mixture components. Gaussian Process Regression (GPR) is consistent w.r.t. the marginals but predicted wrong distributions.

Baselines. As baseline models, we use NeuralFlows [4], GRU-ODE [10], CRU [35], GPR [16], ProFITi [43]. Our encoder is similar to Tripletformer, predicting marginal distributions. Hence, we predict the mean and variance of a Gaussian distribution for $\tilde{\mathbf{h}}_k$, calling the model Tripletformer+. NeuralFlows, GRU-ODE, CRU, and Tripletformer+ predict only marginals and are marginalization consistent, as their joint distribution is the product of marginals. GPR is also marginalization consistent. We compare with Gaussian Mixture Model to highlight the advantage of flows in moses. We do not compare with TACTiS/TACTiS-2 [14, 2] as they do not provide results on our datasets and are not marginalization consistent like ProFITi.

Evaluation metric. We use normalized joint negative log-likelihood (njNLL) [43] as the evaluation metric (15). njNLL generalizes the joint negative log-likelihood of varying number of variables. We note that CRPS, widely used metric for probabilistic univariate forecasting, cannot be applied for multivariate distributions. Also, sampling based metrics for multivariate distributions like Energy Score not only suffers from the curse of dimensionality but also cannot evaluate the forecasts properly [26]. On the other hand, another metric CRPS-sum [33] was shown to provide misleading indication of the performance [23]. In their study CRPS-sum cannot distinguish the predictions of a state-of-the-art model from random noise. Hence, we evaluate using njNLL in our experiments.

Evaluating Marginal Consistency Violation. To evaluate the amount of marginalization Inconsistency of a probabilistic model, we propose the marginalization inconsistency metric (MI). It is

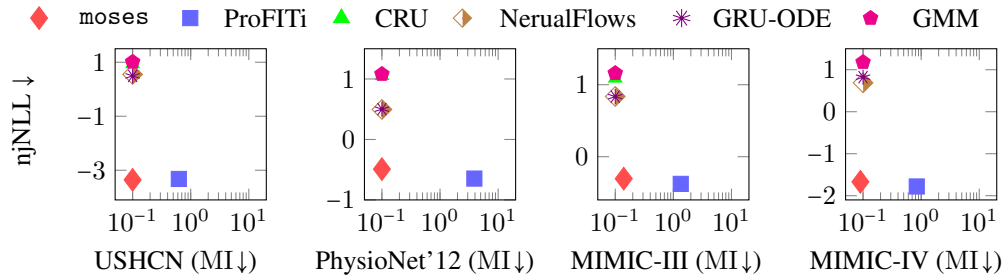


Figure 4: njNLL vs. MI. The Marginalization Inconsistency (16) is computed between the predicted marginal distributions of an individual query and the numerically integrated joint distribution. moses is marginalization consistent up to sampling error.

Table 1: Comparing njNLL values for probabilistic forecasting of irregularly sampled time series. Lower the better, best results in bold, second best in italics.

	Model	USHCN	PhysioNet2012	MIMIC-III	MIMIC-IV
inconsistent	ProFITi [43]	-3.226±0.225	-0.647±0.078	-0.377±0.032	-1.777±0.066
consistent univariate	GRU-ODE [10]	0.766±0.159	0.501±0.001	0.961±0.064	0.823±0.318
	NeuralFlows [4]	0.775±0.152	0.496±0.001	0.998±0.177	0.689±0.087
	CRU [35]	0.761±0.191	1.057±0.007	1.234±0.076	OOM
	Tripletformer+ [41]	4.632±8.179	0.519±0.112	1.051±0.141	0.686±0.115
consistent multivariate	GPR [16]	2.011±1.376	1.367±0.074	3.146±0.359	2.789±0.057
	GMM	1.050±0.031	1.063±0.002	1.160±0.020	1.076±0.003
	moses (ours)	-3.357±0.176	<i>-0.491±0.041</i>	<i>-0.305±0.027</i>	<i>-1.668±0.097</i>

the average of the Wasserstein Distance (W_r ; also known as earth movers distance) between univariate marginals that are directly predicted $\hat{p}_k(y_k | Q_k, X)$ and univariate marginals $\hat{p}_k^{\text{int}}(y | Q, X)$ computed by integrating the joint $\hat{p}(y | Q, X)$ (refer to Appendix A.5 for more details):

$$\text{MI}_r(\hat{p}(y)) := \frac{1}{K} \sum_{k=1}^K W_r(\hat{p}(y_k | Q_k, X), \hat{p}^{\text{int}}(y_k | Q_k, X)) \quad (16)$$

Results. We compare the results of *moses* with the published results from [43] in Table 1. It can be seen that *moses* performs better than all the marginalization consistent models and comparable to non-marginalization consistent model ProFITi. In Figure 3, we demonstrate njNLL vs marginal inconsistency (MI). *moses* not only achieves similar likelihoods as ProFITi, its MI is close to 0 where ProFITi is up to an order of magnitude larger. Smaller values of MI for *moses* is due to sampling. ProFITi has smallest MI of 0.6 for USHCN because the dataset does not have strong covariances.

Ablation study. To see the contributions of each model components we perform ablation study using PhysioNet2012 dataset (Table 2). Performance drops drastically by removing the flows (*moses* – *f*) which is same as GMM. It is expected that normalizing flows are more expressive compared to simple mixture of Gaussians. On the other hand, by using only isotropic Gaussian as the base distribution (*moses* – COV) model performance decreased. Similarly, parameterizing the components weights have a slight advantage over fixing them to $1/D$ with D being the number of components. One interesting observation is even using single component (*moses*(1)) gives similar results compared to mixture of such components. This could be because the dataset we have may not require multiple components. We note that we have $D = 1$ in our hyperparameter search space, and we select the best D based on validation dataset.

Table 2: Ablation study on PhysioNet2012

Model	njNLL (\downarrow)
<i>moses</i>	-0.491±0.041
<i>moses</i> – <i>f</i>	1.063±0.002
<i>moses</i> –COV	-0.308±0.024
<i>moses</i> – <i>w</i>	-0.451±0.038
<i>moses</i> (1)	-0.493±0.029

7 Limitations

We parametrize Σ to ensure marginalization consistency. However, in practice we require $\Sigma^{-\frac{1}{2}}$ to compute the density. Computing $\Sigma^{-\frac{1}{2}}$ from Σ requires a complexity of $\mathcal{O}(K^3)$ with K being the number of variables. It is in general, not a problem for reasonable number of queries which is the case in our experiments. However, if there is a situation that requires large K , *moses* may be memory inefficient. In the future, we will work on the model that can scale to very long sequences.

Conclusions

In this work, we propose, *moses*: marginalization consistent mixture of separable flows, a novel model for probabilistic forecasting of irregular time series. We showed how to parametrize various components of *moses* such that it is decomposable and marginalization consistent. Our experimental results on 4 real-world irregularly sampled time series datasets show that *moses* not only performs similar to state-of-the-art ProFITi model but unlike ProFITi *moses* is marginalization consistent.

References

- [1] Abdul Fatir Ansari, Alvin Heng, Andre Lim, and Harold Soh. Neural continuous-discrete state space models for irregularly-sampled time series. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 926–951. PMLR, 2023-07-23/2023-07-29.
- [2] 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*, 2024.
- [3] Marin Biloš and Stephan Günnemann. Normalizing flows for permutation invariant densities. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 957–967. PMLR, 2021-07-18/2021-07-24.
- [4] Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural ODEs. *Advances in Neural Information Processing Systems*, 34:21325–21337, 2021.
- [5] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20, 2007.
- [6] 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):1–12, 2018. doi: 10.1038/s41598-018-24271-9.
- [7] Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. ContiFormer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.
- [8] YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. October 2020.
- [9] Sebastian Ciobanu. Mixtures of normalizing flows. In *Proceedings of ISCA 34th International Conference on Computer Applications in Industry and Engineering, EPIc Series in Computing*, volume 79, pages 82–90, 2021.
- [10] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. *Advances in Neural Information Processing Systems*, 32, 2019.
- [11] Ruizhi Deng, Bo Chang, Marcus A Brubaker, Greg Mori, and Andreas Lehrmann. Modeling continuous stochastic processes with dynamic normalizing flows. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7805–7815. Curran Associates, Inc., 2020.
- [12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.
- [13] Hadi Mohaghegh Dolatabadi, Sarah Erfani, and Christopher Leckie. Invertible generative modeling using linear rational splines. In *International Conference on Artificial Intelligence and Statistics*, pages 4236–4246. PMLR, 2020.
- [14] Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. Tactis: Transformer-attentional copulas for time series. In *International Conference on Machine Learning*, pages 5447–5493. PMLR, 2022.
- [15] Richard O. Duda and Peter E. Hart. Pattern classification and scene analysis. In *A Wiley-Interscience Publication*, 1974.
- [16] 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, 2015. doi: 10.1109/TBME.2014.2351376.

- [17] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.
- [18] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889. PMLR, 2015.
- [19] Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *International Conference on Machine Learning*, pages 4353–4363. PMLR, 2020.
- [20] A Johnson, L Bulgarelli, T Pollard, S Horng, and LA Celi. Mark. *R. MIMIC-IV (version 1.0)*. *PhysioNet*, 2021.
- [21] Alistair EW 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):1–9, 2016.
- [22] 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. September 2019.
- [23] Alireza Koochali, Peter Schichtel, Andreas Dengel, and Sheraz Ahmed. Random noise vs. state-of-the-art probabilistic forecasting methods: A case study on CRPS-Sum discrimination ability. *Applied Sciences*, 12(10):5104, 2022. doi: 10.3390/app12105104.
- [24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NIPS)*, 30, 2017.
- [25] Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. *Advances in Neural Information Processing Systems*, 32, 2019.
- [26] Étienne Marcotte, Valentina Zantedeschi, Alexandre Drouin, and Nicolas Chapados. Regions of reliability in the evaluation of multivariate probabilistic forecasts. In *International Conference on Machine Learning*, pages 23958–24004. PMLR, 2023.
- [27] Matthew J Menne, CN Williams Jr, and Russell S Vose. United States historical climatology network daily temperature, precipitation, and snow data. *Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, Oak Ridge, Tennessee*, 2015.
- [28] 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.
- [29] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [30] 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(1), January 2021. ISSN 1532-4435.
- [31] Guilherme G. P. Freitas Pires and Mário A. T. Figueiredo. Variational mixture of normalizing flows. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2020, Bruges, Belgium, October 2-4, 2020*, pages 205–210, 2020.
- [32] Janis Postels, Mengya Liu, Riccardo Spezialetti, Luc Van Gool, and Federico Tombari. Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction. In *2021 International Conference on 3D Vision (3DV)*, pages 1249–1258. IEEE, 2021. doi: 10.1109/3DV53792.2021.00132.

- [33] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021.
- [34] Danilo Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, June 2015.
- [35] 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*, volume 162, pages 19388–19405. PMLR, 2022-07-17/2022-07-23.
- [36] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *International Conference on Learning Representations*, 2021.
- [37] Satya Narayan Shukla and Benjamin Marlin. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *International Conference on Learning Representations*, 2022.
- [38] Sahil Sidheekh, Kristian Kersting, and Sriraam Natarajan. Probabilistic Flow Circuits: Towards Unified Deep Models for Tractable Probabilistic Inference. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 1964–1973. PMLR, July 2023.
- [39] 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*, pages 245–248. IEEE, 2012.
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [41] Vijaya Krishna Yalavarthi, Johannes Burchert, and Lars Schmidt-Thieme. Tripletformer for Probabilistic Interpolation of Irregularly sampled Time Series. *2023 IEEE International Conference on Big Data (Big Data)*, 2023.
- [42] Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolph Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. GraFITi: Graphs for Forecasting Irregularly Sampled Time Series. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 16255–16263. AAAI Press, 2024. doi: 10.1609/AAAI.V38I15.29560.
- [43] Vijaya Krishna Yalavarthi, Randolph Scholz, Stefan Born, and Lars Schmidt-Thieme. Probabilistic forecasting of irregular time series via conditional flows. *CoRR*, abs/2402.06293, 2024. doi: 10.48550/ARXIV.2402.06293.

A Theory

A.1 Proof of Lemma 1

Proof. Since X is a common conditional to all the marginals, we can ignore it. So, assume that f is a separable transformation:

$$f(z \mid Q) = (\phi(z_1 \mid Q_1), \dots, \phi(z_K \mid Q_K))$$

and that $\hat{p}_Z(z \mid Q)$ is marginalization consistent model. Then, the predictive distribution is

$$\hat{p}(y \mid Q) = \hat{p}_Z(f^{-1}(y \mid Q) \mid Q) \cdot \left| \det \frac{df^{-1}(y \mid Q)}{dy} \right|$$

Since f is separable, it follows that the Jacobian is diagonal:

$$\begin{aligned}\frac{df^{-1}(y | Q)}{dy} &= \frac{d(\phi^{-1}(y_1 | Q_1), \dots, \phi^{-1}(y_K | Q_K))}{d(y_1, \dots, y_K)} \\ &= \text{diag}\left(\frac{d\phi^{-1}(y_1 | Q_1)}{dy_1}, \dots, \frac{d\phi^{-1}(y_K | Q_K)}{dy_K}\right)\end{aligned}$$

Hence, the determinant of the Jacobian is the product of the diagonal elements:

$$\left| \det \frac{df^{-1}(y | Q)}{dy} \right| = \prod_{k=1:K} \left| \det \frac{d\phi^{-1}(y_k | Q_k)}{dy_k} \right| = \prod_{k=1:K} \left| \frac{d\phi^{-1}(y_k | Q_k)}{dy_k} \right| \quad (17)$$

Using this fact, we can integrate the joint density over y_k to get the marginal density:

$$\begin{aligned}\int \hat{p}(y | Q) dy_k &= \int \hat{p}_Z(f^{-1}(y | Q) | Q) \cdot \left| \det \frac{df^{-1}(y | Q)}{dy} \right| dy_k \quad \triangleright (7) \\ &= \int \hat{p}_Z(f^{-1}(y | Q) | Q) \cdot \prod_{k=1:K} \left| \frac{d\phi^{-1}(y_k | Q_k)}{dy_k} \right| dy_k \quad \triangleright (17) \\ &= \left(\prod_{l \neq k} \left| \frac{d\phi^{-1}(y_l | Q_l)}{dy_l} \right| \right) \cdot \int \hat{p}_Z(f^{-1}(y | Q) | Q) \cdot \left| \frac{d\phi^{-1}(y_k | Q_k)}{dy_k} \right| dy_k \\ &= \left(\prod_{l \neq k} \left| \frac{d\phi^{-1}(y_l | Q_l)}{dy_l} \right| \right) \cdot \int \hat{p}_Z(z | Q) dz_k \quad \triangleright \text{transf.-thm} \\ &= \left(\prod_{l \neq k} \left| \frac{d\phi^{-1}(y_l | Q_l)}{dy_l} \right| \right) \hat{p}_Z(z_{-k} | Q_{-k}) \quad \triangleright (5) \\ &= \hat{p}_Z(z_{-k} | Q_{-k}) \left| \det \frac{df^{-1}(y_{-k} | Q_{-k})}{dy_{-k}} \right| \quad \triangleright (17) \\ &= \hat{p}(y_{-k} | Q_{-k}) \quad \triangleright (7)\end{aligned}$$

□

A.2 Proof of Lemma 2

Proof. Consider a mixture model of the form

$$\hat{p}(y | Q, X) := \sum_{d=1}^D w_d(X) \hat{p}_d(y | Q, X) \quad (18)$$

satisfying the conditions from Lemma 2, i.e. the component models \hat{p}_d satisfy the requirements R1-R3 and the weight function $w: \text{Seq } \mathcal{X} \rightarrow \Delta^D$ is permutation invariant.

1. \hat{p} satisfies R1: By construction of the mixture model, it has the same domain and codomain as the component models.
2. \hat{p} satisfies R2: Let $\pi \in S_{|Q|}$ and $\tau \in S_{|X|}$, then

$$\begin{aligned}\hat{p}(y | Q^\pi, X^\tau) &= \sum_{d=1}^D w_d(X^\tau) \hat{p}_d(y | Q^\pi, X^\tau) \\ &= \sum_{d=1}^D w_d(X) \hat{p}_d(y | Q, X) \quad \triangleright \text{permutation invariance of } w \text{ and } \hat{p}_d \\ &= \hat{p}(y | Q, X)\end{aligned}$$

3. \hat{p} satisfies R3:

$$\begin{aligned}
\int \hat{p}(y \mid Q, X) dy_k &= \int \sum_{d=1}^D w_d(X) \hat{p}_d(y \mid Q, X) dy_k \\
&= \sum_{d=1}^D w_d(X) \int p_d(y \mid Q, X) dy_k \\
&= \sum_{d=1}^D w_d(X) p_d(y_{-k} \mid Q_{-k}, X) \quad \triangleright \hat{p}_d \text{ is marginalization consistent} \\
&= p_Y(y_{-k} \mid Q_{-k}, X)
\end{aligned}$$

□

A.3 Proof of Theorem 2

Proof. Due to Lemma 1, it is sufficient to show that all the component models satisfy the requirements R1-R3. Since we use Gaussian Processes as the base distribution, Lemma ?? ensures that each component model is marginalization consistent. Requirement R1 is by construction. Finally, permutation invariance can be seen as follows:

First, note that, by Equation (12), it follows that if \mathbf{h}^{OBS} is permutation equivariant with respect to X , and $\tilde{\mathbf{h}}$ and \mathbf{h} are both permutation equivariant with respect to Q and permutation invariant with respect to X . Now, let $\pi \in S_{|Q|}$ and $\tau \in S_{|X|}$, then, for the d -th component model $\hat{p}_{Y_d}(y \mid Q, X)$. In particular, the flow satisfies $f_d^{-1}(y^\pi, Q^\pi, X^\tau) = f^{-1}(y^\pi, \mathbf{h}_d^\pi) = z^\pi$. Therefore:

$$\begin{aligned}
\hat{p}_{Y_d}(y^\pi \mid Q^\pi, X^\tau) &= \hat{p}_{Z_d}(f^{-1}(y^\pi, Q^\pi, X^\tau) \mid Q^\pi, X^\tau) \cdot \left| \det \frac{df^{-1}(y^\pi, Q^\pi, X^\tau)}{dy^\pi} \right| \\
&= \mathcal{N}(f^{-1}(y^\pi, \mathbf{h}_d^\pi) \mid \mu(\mathbf{h}_d^\pi), \Sigma(\mathbf{h}_d^\pi)) \cdot \left| \det \frac{df^{-1}(y^\pi, \mathbf{h}_d^\pi)}{dy^\pi} \right| \quad \triangleright \text{by remark above} \\
&= \mathcal{N}(z^\pi \mid \mu(\mathbf{h}_d^\pi), \Sigma(\mathbf{h}_d^\pi)) \cdot \left| \det \frac{df^{-1}(y^\pi, \mathbf{h}_d^\pi)}{dy^\pi} \right| \\
&= \mathcal{N}(z \mid \mu, \Sigma) \cdot \left| \det \frac{df^{-1}(y^\pi, \mathbf{h}_d^\pi)}{dy^\pi} \right| \quad \triangleright \text{by permutation invariance of GP} \\
&= \mathcal{N}(z \mid \mu, \Sigma) \cdot \left| \det \frac{df^{-1}(y, \mathbf{h})}{dy} \right| \quad \triangleright \text{by (17)} \\
&= \hat{p}_{Y_d}(y \mid Q, X)
\end{aligned}$$

□

A.4 Linear Rational Splines

Linear Rational Splines (LRS) are computationally efficient spline functions [13]. Formally, given a set of monotonically increasing points $\{(u_m, v_m)\}_{m=1:M}$ called knots, that is $u_m < u_{m+1}$ and $v_m < v_{m+1}$, along with their corresponding derivatives $\{\Delta_m > 0\}_{m=1:M}$, then the LRS transformation $\phi(u)$ within a bin $u \in [u_m, u_{m+1}]$ is:

$$\phi(u) = \begin{cases} \frac{\alpha_m v_m (\lambda_m - \tilde{u}) + \bar{\alpha}_m \bar{v}_m \tilde{u}}{\alpha_m (\lambda_m - \tilde{u}) + \bar{\alpha}_m \tilde{u}} & : 0 \leq \tilde{u} \leq \lambda_m \\ \frac{\bar{\alpha}_m \bar{v}_m (1 - \tilde{u}) + \alpha_{m+1} v_{m+1} (\tilde{u} - \lambda_m)}{\bar{\alpha}_m (1 - \tilde{u}) + \alpha_{m+1} (\tilde{u} - \lambda_m)} & : \lambda_m \leq \tilde{u} \leq 1 \end{cases} \quad \text{where} \quad \tilde{u} = \frac{u - u_m}{u_{m+1} - u_m} \in [0, 1] \quad (19)$$

Here, $\lambda_m \in (0, 1)$ signifies the location of automatically inserted virtual knot between u_m and u_{m+1} with value \bar{v}_m . The values of λ_m , α_m , $\bar{\alpha}_m$ and \bar{v}_m are all automatically derived from the original knots and their derivatives [13]. For a conditional LRS $\phi(z_k; \mathbf{h}_{d,k}, \theta)$, the function parameters such as width and height of each bin, the derivatives at the knots, and λ are computed from the conditioning input $\mathbf{h}_{d,k}$ and some model parameters θ . θ helps to project $\mathbf{h}_{d,k}$ to the function parameters, and is common to all the variables $z_{1:K}$ so that the transformation ϕ can be applied for varying number of

Table 3: Statistics of the datasets used in our experiments. Sparsity means the percentage of missing observations in the time series. N is the total number of observations and K is the number of queries in our experiments in Section 6.

name	#samples	#channels	sparsity	N	K
USHCN	1100	5	77.9%	8 – 322	3 – 6
PhysioNet’12	12,000	37	85.7%	3 – 519	1 – 53
MIMIC-III	21,000	96	94.2%	4 – 709	1 – 85
MIMIC-IV	18,000	102	97.8%	1 – 1382	1 – 79

variables K . Additionally, we set θ common to all the components as well. Since, each component has separate embedding for a variable z_k ($\mathbf{h}_{d,k}$), we achieve different transformations in different components for same variable.

In summary, the conditional flow model is separable across the query size $f = f_1 \times \dots \times f_K$ with

$$f_d(y) := f(y | \mathbf{h}_d) = (\phi(y_1, \mathbf{h}_{d,1}), \dots, \phi(y_K, \mathbf{h}_{d,K})) \quad (20)$$

A.5 Measuring marginalization Consistency Violation

To quantify the marginal inconsistency of a probabilistic model, we use Wasserstein distance (also known as earth movers distance) between the computed univariate marginals from joints (by integrating) and predicted univariate marginals. Wasserstein distance between two univariate distributions p_U and p_V is computed from samples. Let $(u_1 \dots u_I) \sim p_U$ and $(v_1, \dots, v_I) \sim p_V$, then the distance is a function of the order statistics:

$$W_r(p_U, p_V) := \left(\sum_{i=1}^I \|u_{(i)} - v_{(i)}\|^r \right)^{1/r} \quad (21)$$

For models \hat{p} that can predict both univariate marginals ($\hat{p}(y_1 | Q_1, X), \dots, \hat{p}(y_K | Q_K, X)$) and their joint distributions $\hat{p}(y | Q, X)$, we can evaluate the inconsistency w.r.t. to the corresponding marginals ($\hat{p}(y_k | Q_k, X), \hat{p}^{\text{marg}}(y_k | Q_k, X)$) where $\hat{p}^{\text{marg}}(y_k | Q_k, X) := \int \hat{p}(y | Q, X) dy_{-k}$ through sampling. While sampling $y_k \sim \hat{p}(y_k | Q_k, X)$ is easy, sampling from $\hat{p}^{\text{marg}}(y_k | Q_k, X)$ is tricky. For this we sample $y^{\text{joint}} \sim \hat{p}(y | Q, X)$ and then y_k^{marg} is considered as sample from $\hat{p}^{\text{marg}}(y_k | Q_k, X)$. Now the marginal inconsistency (MI_r) is measured using mean W_r :

$$\text{MI}_r(\hat{p}(y | X, Q)) := \frac{1}{K} \sum_{k=1}^K W_r(\hat{p}(y_k | Q_k, X), \hat{p}^{\text{int}}(y_k | Q_k, X))$$

In our experiments we set $r = 2$ and $I = 1000$.

B Datasets

4 real-world datasets are used in the experiments.

USHCN [27]. is a climate dataset. It consists of 5 climate variables such as daily temperatures, precipitation and snow measured over 150 years at 1218 meteorological stations in the USA. Following [10, 43], we selected 1114 stations and an observation window of 4 years from 1996 until 2000.

PhysioNet2012 [39]. is a physiological dataset consists of the medical records of 12,000 patients who are admitted into ICU. 37 vitals are recorded for 48 hrs. Following the protocol of [42, 6], dataset consists of hourly observations in each series.

MIMIC-III [21]. is also a physiological dataset. It is a collection of readings of the vitals of the patients admitted to ICU at Beth Israeli Hospital. Dataset consists of 18,000 instances and 96 variables are measured for 48 hours. Following [10, 4, 43] observations are rounded to 30 minute intervals.

Table 4: Trained for njNLL and evaluate for Marginal Negative Log-likelihood (mNLL), lower the better. Demonstrates the advantage of marginalizing consistency.

	USHCN	PhysioNet'12	MIMIC-III	MIMIC-IV
GRU-ODE	0.776±0.172	0.504±0.061	0.839±0.030	0.876±0.589
Neural-Flows	0.775±0.180	0.492±0.029	0.866±0.097	0.796±0.053
CRU	0.762±0.180	0.931±0.019	1.209±0.044	OOM
ProFITi-marg	<u>-3.324±0.206</u>	<u>-0.016±0.085</u>	<u>0.408±0.030</u>	<u>0.500±0.322</u>
moses (ours)	-3.355±0.156	-0.271±0.028	0.163±0.026	-0.634±0.017

Table 5: Comparing models w.r.t. CRPS score on marginals. Lower the better.

	USHCN	PhysioNet'12	MIMIC-III	MIMIC-IV
GRU-ODE	0.313±0.012	0.278±0.001	0.308±0.005	0.281±0.004
Neural-flows	0.306±0.028	0.277±0.003	0.308±0.004	0.281±0.004
CRU	0.247±0.010	0.363±0.002	0.410±0.005	OOM
ProFITi	0.183±0.009	<u>0.268±0.002</u>	0.295±0.002	0.226±0.002
ProFITi-marg	<u>0.182±0.007</u>	0.271±0.003	0.319±0.003	0.279±0.012
moses (ours)	0.220±0.019	0.260±0.002	<u>0.296±0.005</u>	<u>0.245±0.010</u>

MIMIC-IV [20]. is extension of MIMIC-III dataset. Here, 102 variables from patients admitted to ICU at a tertiary academic medical center in Boston are measured for 48 hours. Following [10, 4, 43] observations, are rounded to 1 minute intervals.

Blast distribution (toy dataset). Blast distribution is a bivariate distribution which is created as follows:

$$z \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1, 1 \\ 1, 2 \end{bmatrix}\right)$$

$$y = \text{sign}(z) \odot z^2$$

Circle (toy dataset). Circle is also a bi-variate distribution.

$$z \sim \mathcal{N}(0, \mathbb{I}_2)$$

$$y = \frac{z}{\|z\|_2} + 0.05 * \mathcal{N}(0, \mathbb{I}_2)$$

C Additional experiments

C.1 Comparing for marginals, trained for njNLL

Here, we compare the models for predicting marginal Negative Log-Likelihood (mNLL) which was used as evaluation metric by many of our baseline models [4, 10, 35]. Given a test data $\mathcal{D}^{\text{TEST}}$, mNLL is the average of the NLL of all the future values (only marginals):

$$\ell^{\text{mNLL}}(\hat{p}; \mathcal{D}^{\text{TEST}}) := - \frac{\sum_{(X, Q, y) \in \mathcal{D}^{\text{TEST}}} \sum_{k=1}^{|y|} \log(\hat{p}(y_k | X, Q_k))}{\sum_{(X, Q, y) \in \mathcal{D}^{\text{TEST}}} |y|} \quad (22)$$

This experiment shows the importance of marginalization consistency. Results are presented in Table 4. ProFITi-marg is the model where ProFITi is asked for one query only ($k = 1$) in a single pass. If a series have $K = 5$, we need to run ProFITi for 5 times. It can be seen that, moses outperforms all the baseline models including ProFITi-marg in terms of mNLL. It is expected that marginalization consistent model that have better njNLL also have better mNLL. However, although ProFITi has better njNLL, ProFITi-marg performs worse than moses because, ProFITi suffers from marginalization inconsistency.

Table 6: Comparing models w.r.t. MSE. Lower the better.

	USHCN	PhysioNet'12	MIMIC-III	MIMIC-IV
GRU-ODE	0.410±0.106	0.329±0.004	0.479±0.044	0.365±0.012
Neural-Flows	0.424±0.110	0.331±0.006	0.479±0.045	0.374±0.017
CRU	<i>0.290±0.060</i>	0.475±0.015	0.725±0.037	OOM
GraFITi	0.256±0.027	0.286±0.001	0.401±0.028	0.233±0.005
Tripletformer+	0.349±0.131	0.293±0.018	0.547±0.068	0.369±0.030
ProFITi	0.321±0.041	<i>0.299±0.007</i>	<i>0.495±0.075</i>	<i>0.268±0.007</i>
ProFITi-marg	0.308±0.061	0.305±0.007	0.548±0.063	0.389±0.015
moses (ours)	0.411±0.099	0.307±0.006	0.517±0.057	0.342±0.028

Table 7: Comparing NLL values for density estimation of tabular data. Lower the better, best results in bold, second best in italics.

Model	power	gas	miniboone	hepmass
MADE [18]	3.08±0.03	-3.56±0.04	15.59±0.50	20.98±0.02
Real NVP [12]	0.02±0.01	-4.78±1.80	<i>13.55±0.49</i>	<i>19.62±0.02</i>
MAF [29]	-0.14±0.01	-9.07±0.02	11.75±0.44	17.70±0.02
EinsumNet [38]	<i>-0.20±0.01</i>	-3.57±0.08	35.93±0.06	22.79±0.05
Einsum+LRS [38]	-0.36±0.01	-4.79±0.04	34.21±0.01	22.46±0.01
moses	-0.10±0.01	-5.87±0.27	22.66±2.08	23.95±0.20

We also compare with CRPS score in Table 5, a widely used evaluation metric in time series forecasting. We see that *moses* outperforms all the consistent models. It performs better than ProFITi-marg in 3 out of 4 dataset. For ProFITi, ProFITi-marg and *moses*, we sampled 1000 instances and computed the CRPS. Due to marginal inconsistency in ProFITi, the CRPS scores of ProFITi are vastly different from that of ProFITi-marg in PhysioNet'12, MIMIC-III and MIMIC-IV datasets.

C.2 Comparing for point forecasting

Here, we would like to see how the models compare with point forecasting. We use mean squared error as the evaluation metric. Here, we also compare with GraFITi [42], state-of-the-art point forecasting model for irregularly sampled time series. Results are presented in Table 6. While GraFITi continues to be the best, ProFITi is the second best. *moses* has similar performance as ProFITi in PhysioNet'12 and MIMIC-III datasets. Both GraFITi and ProFITi are inconsistent models.

Interesting to observe that *moses* is significantly better than Tripletformer+ in MIMIC-III and MIMIC-IV datasets and have similar results in USHCN and PhysioNet'12 datasets. Also, it is better than existing consistent models GRU-ODE [10], Neural Flows [4] and CRU [35]. Note that, it is often observed in the literature that models for uncertainty quantification quite often suffer from somewhat worse point forecasts [24, 36].

C.3 Results for unconditional density estimation on tabular data

moses can be considered as instance of a Probabilistic Circuit [8]. Hence, we compare *moses* with the existing circuit models: EinsumNet [8] and EinsumNet+LRS [38] on fixed length variables. We also compare the results with the well known density estimation models.

We compare the models on the published results from [38] for 4 tabular datasets: power, gas, miniboone and hepmass. We follow the same protocol for the dataset splits as shown in [29, 38]. Since, it is not a conditional density estimation, the univariate transformations, mixture weights w , and the parameters of the base distribution (μ, Σ) are trainable parameters. Again, since the number of variables are fixed, we use NLL as the evaluation measure.

In Table 7 we present the NLL results for all the models. We see that MAF is the best performing model. However, *moses* beats Real NVP in 2 out of 4 datasets. Also, *moses* performs better than the best decomposable model Einsum+LRS in 2 out of 4 datasets. Note that, unlike *moses* all the above models are limited to fixed number of variables.