# TACTiS-2: BETTER, FASTER, SIMPLER ATTENTIONAL COPULAS FOR MULTIVARIATE TIME SERIES

**Arjun Ashok**[1 2 3], *[*]**Étienne Marcotte**[1], *[*]**Valentina Zantedeschi**[1],
[†]**Nicolas Chapados**[1 2], [†]**Alexandre Drouin**[1 2]
[1]ServiceNow Research [2]Mila-Quebec AI Institute [3]Université de Montréal
Montréal, Canada
firstname.lastname@servicenow.com

## ABSTRACT

We introduce a new model for multivariate probabilistic time series prediction, designed to flexibly address a range of tasks including forecasting, interpolation, and their combinations. Building on copula theory, we propose a simplified objective for the recently-introduced *transformer-based attentional copulas* (TACTiS), wherein the number of distributional parameters now scales linearly with the number of variables instead of factorially. The new objective requires the introduction of a training curriculum, which goes hand-in-hand with necessary changes to the original architecture. We show that the resulting model has significantly better training dynamics and achieves state-of-the-art performance across diverse real-world forecasting tasks, while maintaining the flexibility of prior work, such as seamless handling of unaligned and unevenly-sampled time series. Code is made available at https://github.com/ServiceNow/TACTiS.

## 1 INTRODUCTION

Optimal decision-making involves reasoning about the evolution of quantities of interest over time, as well as the likelihood of various scenarios (Peterson, 2017). In its most general form, this problem amounts to estimating the joint distribution of a set of variables over multiple time steps, i.e., *multivariate probabilistic time series forecasting* (Gneiting & Katzfuss, 2014). Although individual aspects of this problem have been extensively studied by the statistical and machine learning communities (Hyndman et al., 2008; Box et al., 2015; Hyndman & Athanasopoulos, 2018), they have often been examined in isolation. Recently, an emerging stream of research has started to seek general-purpose models that can handle several stylized facts of real-world time series problems, namely (i) a large number of time series, (ii) arbitrarily complex data distributions, (iii) heterogeneous or irregular sampling frequencies (Shukla & Marlin, 2020), (iv) missing data (Fang & Wang, 2020), and (v) the availability of deterministic covariates for conditioning (e.g., holiday indicators), while being flexible enough to handle a variety of tasks, such as forecasting and interpolation (Drouin et al., 2022).

Classical forecasting methods (Hyndman et al., 2008; Box et al., 2015; Hyndman & Athanasopoulos, 2018) often make strong assumptions about the nature of the data distribution (e.g., parametric forms) and are thus limited in their handling of these desiderata. The advent of deep learning-based methods enabled significant progress on this front and led to models that excel at a wide range of time series prediction tasks (Torres et al., 2021; Lim & Zohren, 2021; Fang & Wang, 2020). Yet, most of these methods lack the flexibility required to meet the aforementioned requirements. Recently, some general-purpose models have been introduced (Tashiro et al., 2021; Drouin et al., 2022; Biloš et al., 2023; Alcaraz & Strodthoff, 2023), but most of them only address a subset of the desiderata. One notable exception is *Transformer-Attentional Copulas for Time Series* (TACTiS; Drouin et al., 2022), which addresses all of them, while achieving state-of-the-art predictive performance. TACTiS relies on a modular *copula-based* factorization of the predictive joint distribution (Sklar, 1959), where multivariate dependencies are modeled using *attentional copulas*. These consist of neural networks trained by solving a specialized objective that guarantees their convergence to mathematically valid copulas. While the approach of Drouin et al. (2022) is theoretically sound, it requires solving an optimization problem with a number of distributional parameters that grows factorially with the number of variables, leading to poor training dynamics and suboptimal predictions.

[*]Equal Contribution, [†]Equal Contribution

In this paper, we build on copula theory to propose a simplified training procedure for *attentional copulas*, solving a problem where the number of distributional parameters scales linearly, instead of factorially, with the number of variables. Our work results in a general-purpose approach to multivariate time series prediction that also meets all of the above desiderata, while having considerably better training dynamics, namely converging faster to better solutions (see Fig. 1).

**Contributions:**

- We show that, while nonparametric copulas require specialized learning procedures (Prop. 1), the permutation-based approach used in TACTiS (Drouin et al., 2022) is unnecessarily complex, and that valid copulas can be learned by solving a two-stage problem whose number of parameters scales linearly with the number of variables (Sec. 3.2);
- We build on these theoretical findings to propose TACTiS-2, an improved version of TACTiS with a revised architecture that is trained using a two-stage curriculum guaranteed to result in valid copulas (Sec. 4);
- We empirically show that our simplified training procedure leads to better training dynamics (e.g., faster convergence to better solutions) as well as state-of-the-art performance on a number of real-world forecasting tasks, while preserving the high flexibility of the TACTiS model (Sec. 5);

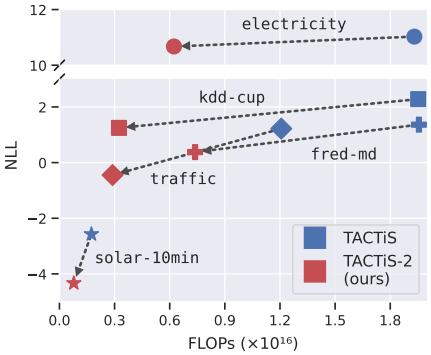



**Figure 1:** TACTiS-2 *outperforms* TACTiS *in (i) density estimation (lower validation negative log-likelihoods, NLL) and (ii) training compute (fewer floating point operations, FLOPs) in real-world forecasting tasks (see Sec. 5).*

## 2 Problem Setting

This work addresses the general problem of estimating the joint distribution of unobserved values at arbitrary time points in multivariate time series. This setting subsumes classical ones, such as forecasting, interpolation and backcasting. Let $\mathbf{X}$ be a multivariate time series comprising of $n$, possibly related, univariate time series, denoted as $\mathbf{X} \stackrel{\text{def}}{=} \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$. Each $\mathbf{X}_i \stackrel{\text{def}}{=} [X_{i1}, \ldots, X_{i,\ell_i}]$ is a random vector representing $\ell_i$ observations of some real-valued process in time. We assume that, for any realization $\mathbf{x}_i \stackrel{\text{def}}{=} [x_{i1}, \ldots, x_{i,\ell_i}]$ of $\mathbf{X}_i$, each $x_{ij}$ is paired with (i) a timestamp, $t_{ij} \in \mathbb{R}$, with $t_{ij} < t_{i,j+1}$, marking its measurement time and (ii) a vector of non-stochastic covariates $\mathbf{c}_{ij} \in \mathbb{R}^p$ that represents arbitrary additional information available at each time step.

**Tasks:** We define our learning tasks with the help of a mask $m_{ij} \in \{0, 1\}$, which determines if any $X_{ij}$ should be considered as observed ($m_{ij} = 1$) or to be inferred ($m_{ij} = 0$). For example, a task that consists in forecasting the last $k$ time steps in $\mathbf{X}$ would be defined as $m_{ij} = 0$ for all $i$ and $j$, s.t. $\ell_i - k < j \leq \ell_i$ and $m_{ij} = 1$ otherwise. Similarly, a task that consists in interpolating the values of time steps $k$ to $p$ in $\mathbf{X}$ could be defined by setting all $m_{ij} = 0$ for all $i$ and $j$ s.t. $k \leq j \leq p$ and $m_{ij} = 1$ otherwise. Arbitrary, more complex, tasks can be defined using this approach.

**General Problem:** We consider the general problem of estimating the joint distribution of *missing values* (i.e., $m_{ij} = 0$), given the observed ones ($m_{ij} = 1$), the covariates, and the timestamps:

$$P\left(\mathbf{X}^{(m)} \;\middle|\; \mathbf{X}^{(o)}, \mathbf{C}^{(m)}, \mathbf{C}^{(o)}, \mathbf{T}^{(m)}, \mathbf{T}^{(o)}\right), \tag{1}$$

where $\mathbf{X}^{(m)} = [X_{11}, \ldots, X_{1,l_1}; \ldots; X_{n1}, \ldots, X_{n,l_n} \mid m_{ij} = 0]$ is a random vector containing the $d$ random variables $X_{ij}$ corresponding to all missing values, $\mathbf{X}^{(o)}$ is the same, but for observed values, and $\mathbf{C}^{(m)}, \mathbf{C}^{(o)}, \mathbf{T}^{(m)}, \mathbf{T}^{(o)}$ correspond to identical partitionings of the covariates and timestamps, respectively. In what follows, we estimate the joint probability density function (PDF) of Eq. (1) using a *copula-based density estimator* $g_{\boldsymbol{\phi}}(x_1, \ldots, x_d)$, whose parameters $\boldsymbol{\phi}$ are conditioned on $\mathbf{X}^{(o)}, \mathbf{C}^{(m)}, \mathbf{C}^{(o)}, \mathbf{T}^{(m)}, \mathbf{T}^{(o)}$.

### 2.1 Copula-Based Density Estimators

A copula is a probabilistic object that allows capturing dependencies between $d$ random variables independently from their respective marginal distributions. More formally the joint cumulative distribution function

(CDF) for any random vector of $d$ variables $\mathbf{X} = [X_1, \ldots, X_d]$, can be written as:

$$P(X_1 \leq x_1, \ldots, X_d \leq x_d) = C\big(F_1(x_1), \ldots, F_d(x_d)\big), \tag{2}$$

where $F_i(x_i) \stackrel{\text{def}}{=} P(X_i \leq x_i)$ is the *univariate* marginal CDF of $X_i$ and $C : [0,1]^d \to [0,1]$, the *copula*, is the CDF of a *multivariate* distribution on the unit cube with *uniform* marginals (Sklar, 1959). Notably, if all $F_i$ are continuous, then this copula-based decomposition is unique.

The present work integrates into a stream of research seeking specific parametrizations of copula-based density estimators for multivariate time series (Salinas et al., 2019; Drouin et al., 2022), namely,

$$g_{\boldsymbol{\phi}}(x_1, \ldots, x_d) \stackrel{\text{def}}{=} c_{\phi_c}\Big(F_{\phi_1}(x_1), \ldots, F_{\phi_d}(x_d)\Big) \times f_{\phi_1}(x_1) \times \cdots \times f_{\phi_d}(x_d), \tag{3}$$

where $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_d; \phi_c\}$, with $\{\phi_i\}_{i=1}^d$ the parameters of the marginal distributions (with $F_{\phi_i}$ and $f_{\phi_i}$ the estimated CDF and PDF respectively) and $\phi_c$ the parameters of the copula density $c_{\phi_c}$. The choice of distributions for $F_{\phi_i}$ and $c_{\phi_c}$ is typically left to the practitioner. For example, one could take $F_{\phi_i}$ to be the CDF of a Gaussian distribution and $c_{\phi_c}$ to be a *Gaussian copula* (Nelsen, 2007). The parameters can then be estimated by minimizing the negative log-likelihood:

$$\underset{\boldsymbol{\phi}}{\arg\min} \; - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathbf{X}} \log g_{\boldsymbol{\phi}}(x_1, \ldots, x_d)\,. \tag{4}$$

## 3 Learning Nonparametric Copulas

Oftentimes, to avoid making parametric assumptions, one may take the $c_{\phi_c}$ and $F_{\phi_i}$ components of the estimator to be highly flexible neural networks (Wiese et al., 2019; Janke et al., 2021; Drouin et al., 2022). While it is easy to constrain $c_{\phi_c}$ to have a valid domain and codomain, this does not imply that its marginal distributions will be uniform. Thus, as observed by Janke et al. (2021) and Drouin et al. (2022), a key challenge is ensuring that the distribution $c_{\phi_c}$ satisfies the mathematical definition of a copula (see Sec. 2.1). We strengthen this observation by proving a new theoretical result, showing that solving Problem (4) without any additional constraints can lead to infinitely many solutions where $c_{\phi_c}$ is not a valid copula:

**Proposition 1.** *(Invalid Solutions) Assuming that all random variables $X_1, \ldots, X_d$ have continuous marginal distributions and assuming infinite expressivity for $\{F_{\phi_i}\}_{i=1}^d$ and $c_{\phi_c}$, Problem (4) has infinitely many invalid solutions wherein $c_{\phi_c}$ is not the density function of a valid copula.*

*Proof.* The proof, detailed in App. B.1, shows that one can create infinitely many instantiations of $F_{\phi_i}$ and $c_{\phi_c}$ where $p(x_1, \ldots, x_d) = g_{\boldsymbol{\phi}}(x_1, \ldots, x_d)$, but the true marginals and the copula are entangled. □

Hence, using neural networks to learn copula-based density estimators is non-trivial and requires more than a simple modular parametrization of the model.

### 3.1 Permutation-Invariant Copulas

Recently, Drouin et al. (2022) showed that valid nonparametric copulas can be learned using a permutation-based objective. Their approach considers a factorization of $c_{\phi_c}$ based on the chain rule of probability according to an arbitrary permutation of the variables $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_d] \in \Pi$, where $\Pi$ is the set of all $d!$ permutations of $\{1, \ldots, d\}$. The resulting copula density, $c_{\phi_c^{\boldsymbol{\pi}}}$, can be written as:

$$c_{\boldsymbol{\phi}_c^{\boldsymbol{\pi}}}(u_1, \ldots, u_d) \stackrel{\text{def}}{=} c_{\phi_{c,1}^{\boldsymbol{\pi}}}(u_{\pi_1}) \times c_{\phi_{c,2}^{\boldsymbol{\pi}}}(u_{\pi_2} \mid u_{\pi_1}) \times \cdots \times c_{\phi_{c,d}^{\boldsymbol{\pi}}}\big(u_{\pi_d} \mid u_{\pi_1}, \ldots, u_{\pi_{d-1}}\big)\,, \tag{5}$$

where $u_{\pi_k} = F_{\phi_{\pi_k}}(x_{\pi_k})$, with $F_{\phi_{\pi_k}}$ arbitrary marginal CDFs, and the $c_{\phi_{c,i}^{\boldsymbol{\pi}}}$ are arbitrary distributions (e.g., histograms) on the unit interval with parameters $\phi_{c,i}^{\boldsymbol{\pi}}$. There is, however, one important exception: the density of the first variable in the permutation $c_{\phi_{c,1}^{\boldsymbol{\pi}}}$ is always taken to be that of a *uniform distribution* $U_{[0,1]}$ and thus $c_{\phi_{c,1}^{\boldsymbol{\pi}}}(u_{\pi_1}) = 1$. This choice, combined with solving the following problem, guarantees that, at the minimum, all of the $c_{\phi_c^{\boldsymbol{\pi}}}$, irrespective of $\boldsymbol{\pi}$, are equivalent and correspond to valid copula densities:

$$\underset{\phi_1, \ldots, \phi_n, \phi_c^{\boldsymbol{\pi}}}{\arg\min} \; - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathbf{X}} \mathop{\mathbb{E}}_{\boldsymbol{\pi} \sim \Pi} \log c_{\phi_c^{\boldsymbol{\pi}}}\Big(F_{\phi_1}(x_1), \ldots, F_{\phi_d}(x_d)\Big) \times f_{\phi_1}(x_1) \times \cdots \times f_{\phi_d}(x_d), \tag{6}$$

where $\phi_c^{\boldsymbol{\pi}} \stackrel{\text{def}}{=} \{\phi_{c,1}^{\boldsymbol{\pi}}, \dots \phi_{c,d}^{\boldsymbol{\pi}}\}_{\boldsymbol{\pi} \in \Pi}$ are the parameters for each of the $d!$ factorizations of the copula density (see Drouin et al. (2022), Theorem 1).

**Limitations:** Obtaining a valid copula using this approach requires solving an optimization problem with $O(d!)$ parameters. This is extremely prohibitive for large $d$, which are common in multivariate time series prediction (e.g., $d = 8880$ for the common `electricity` benchmark; Marcotte et al. (2023)). The approach proposed in Drouin et al. (2022), consists of parametrizing a single neural network to output $\boldsymbol{\phi}^{\Pi}$ and using a Monte Carlo approximation of the expectation on $\Pi$. However, such an approach has several caveats, e.g., (i) the neural network must have sufficient capacity to produce $O(d!)$ distinct values and (ii) due to the sheer size of $\Pi$, only a minuscule fraction of all permutations can be observed in one training batch, resulting in slow convergence rates. This is supported by empirical observations in Sec. 5.

### 3.2 Two-Stage Copulas

In this work, we take a different approach to learning nonparametric copulas, one that does not rely on permutations, alleviating the aforementioned limitations. Our approach builds on the following two-stage optimization problem, whose properties have previously been studied in the context of parametric (Joe & Xu, 1996; Andersen, 2005; Joe, 2005) and semi-parametric estimators (Andersen, 2005), and where the number of parameters scales with $O(d)$ instead of $O(d!)$:

$$\arg\min_{\phi_c} \quad - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathbf{X}} \log c_{\phi_c}\left(F_{\phi_1^\star}(x_1), \dots, F_{\phi_d^\star}(x_d)\right) \tag{7}$$

$$s.t. \qquad (\phi_1^\star, \dots, \phi_d^\star) \in \arg\min_{\phi_1, \dots, \phi_d} - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathbf{X}} \log \prod_{i=1}^{d} f_{\phi_i}(x_i). \tag{8}$$

Hence, the optimization proceeds in two stages:

- » **Stage 1:** Learn the marginal parameters, irrespective of multivariate dependencies (Eq. (8));
- » **Stage 2:** Learn the copula parameters, given the optimal marginals (Eq. (7)).

Beyond obtaining an optimization problem that is considerably simpler, we show that any nonparametric copula $c_{\phi_c}$ learned using this approach is valid, i.e., that it satisfies the mathematical definition of a copula:

**Proposition 2.** *(Validity) Assuming that all random variables $X_1, \dots, X_d$ have continuous marginal distributions and assuming infinite expressivity for $\{F_{\phi_i}\}_{i=1}^d$ and $c_{\phi_c}$, solving Problem (7) yields a solution to Problem (4) where $c_{\phi_c}$ is a valid copula.*

*Proof.* The proof builds on results from Sklar (1959) and is provided in App. B.2. □

## 4 The TACTiS-2 Model

Building on Sec. 3.2, we propose TACTiS-2, a model for multivariate probabilistic time series prediction that inherits the flexibility of the Drouin et al. (2022) model while benefiting from a considerably *simpler* training procedure, with *faster* convergence to *better* solutions. Figure 2 provides an overview of its architecture. In essence, the main difference with TACTiS lies in the choice of the copula-based density estimator (Sec. 3): rather than being *permutation-invariant* (Sec. 3.1), it is of the proposed *two-stage type* (Sec. 3.2). This difference mandates changes to the architecture and the loss of the model, as well as the introduction of a training curriculum, which we outline below. In what follows, we use $\boldsymbol{\theta}$ to denote parameters of neural networks, which are not to be confused with distributional parameters, denoted by $\boldsymbol{\phi}$.

**Dual Encoder:** Whereas TACTiS relies on a *single encoder* to produce all parameters of the output density, TACTiS-2 relies on *two distinct encoders* ($\text{Enc}_{\theta_M}$ and $\text{Enc}_{\theta_C}$) whose representations are used to parametrize the marginal CDFs ($F_{\phi_i}$) and the copula distribution ($c_{\phi_c}$), respectively. As in TACTiS, these are transformer encoders that embed realizations of both observed $\mathbf{X}^{(o)}$ and missing $\mathbf{X}^{(m)}$ values, with the missing values masked as in *masked language models* (Devlin et al., 2018)). Let $x_i$ refer to the realization of a generic random variable (missing or observed) and $m_i$, $\mathbf{c}_i$, and $t_i$ its corresponding mask, covariates, and timestamp. The representations are obtained as:

$$\mathbf{z}_i^M = \text{Enc}_{\theta_M}(x_i \cdot m_i, \mathbf{c}_i, m_i, \mathbf{p}_i)\,, \qquad \mathbf{z}_i^C = \text{Enc}_{\theta_C}(x_i \cdot m_i, \mathbf{c}_i, m_i, \mathbf{p}_i)\,, \tag{9}$$

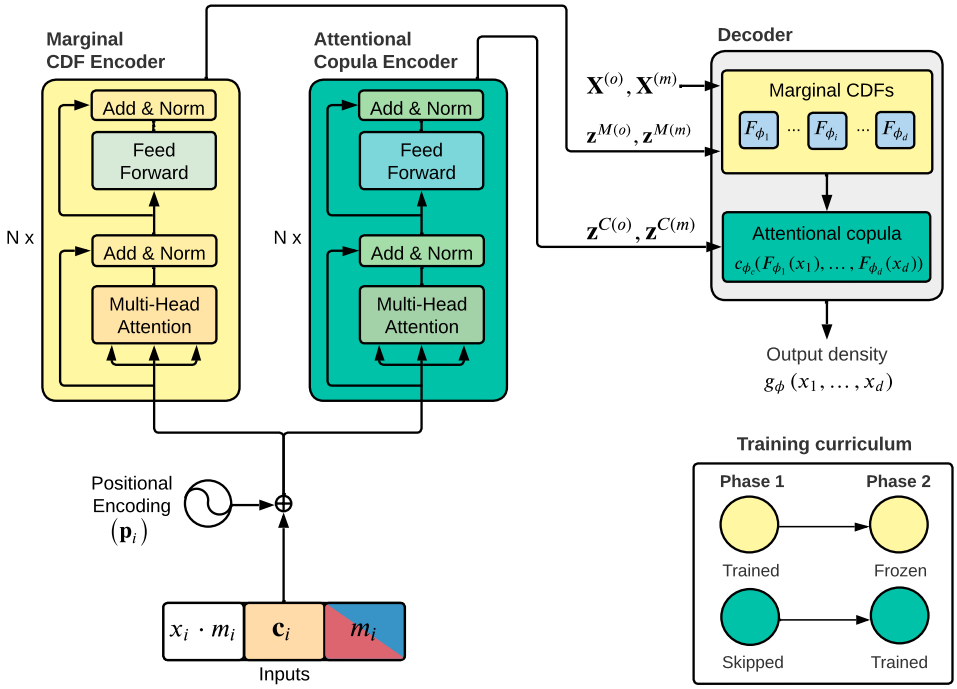



**Figure 2:** *The* TACTiS-2 *architecture with the dual encoder and the decoder. The training curriculum (bottom right) shows the proposed two-stage approach.*

where information about the timestamp $t_i$ is incorporated in the process via an additive positional encoding $\mathbf{p}_i$, which we take to be sinusoidal features as in Vaswani et al. (2017).

**Decoder:** As in TACTiS, the decoder estimates the conditional density of missing values $\mathbf{X}^{(m)}$ (Eq. (1)) with a copula-based estimator structured as in Eq. (3). It consists of two modules, which are tasked with producing the distributional parameters for (i) the marginal CDFs $F_{\phi_i}$, and (ii) the copula $c_{\phi_c}$, respectively. The modules are akin to those used in TACTiS but differ in that they only make use of their respective encodings: $\mathbf{z}^M$ and $\mathbf{z}^C$. For completeness, we outline them below.

**(Marginals)** The first module is a Hyper-Network ($\text{HN}_{\theta_M}$) producing the parameters $\phi_i$ of *Deep Sigmoidal Flows* (DSF) (Huang et al., 2018)[1] that estimate each $F_{\phi_i}$. The value $u_i$, corresponding to the probability integral transform of $x_i$, is then obtained as

$$\phi_i = \text{HN}_{\theta_M}(\mathbf{z}_i^M), \qquad u_i = \text{DSF}_{\phi_i}(x_i). \tag{10}$$

**(Copula)** The second module parametrizes a construct termed *attentional copula*. This corresponds to a factorization of the copula density, as in Eq. (5), where each conditional is parametrized using a causal attention mechanism (Vaswani et al., 2017). More formally, consider an arbitrary ordering of the missing variables $\mathbf{X}^{(m)}$ and let $c_{\phi_{c,i}}(u_i \mid u_1, \ldots, u_{i-1})$ be the $i$-th term in the factorization. Parameters $\phi_{c,i}$ are produced by attending to the $\mathbf{z}_j^C$ (see Eq. (9)) and $u_j$ (see Eq. (10)) for all observed tokens, denoted by $\mathbf{z}^{C(o)}$ and $\mathbf{u}^{(o)}$, and for all missing variables that precede in the ordering, denoted by $\mathbf{z}_{1:i-1}^{C(m)}$ and $\mathbf{u}_{1:i-1}^{(m)}$:

$$\mathbf{K}_i = \text{Key}_{\theta_C}(\mathbf{z}^{C(o)}, \mathbf{z}_{1:i-1}^{C(m)}, \mathbf{u}^{(o)}, \mathbf{u}_{1:i-1}^{(m)}) \qquad \mathbf{V}_i = \text{Value}_{\theta_C}(\mathbf{z}^{C(o)}, \mathbf{z}_{1:i-1}^{C(m)}, \mathbf{u}^{(o)}, \mathbf{u}_{1:i-1}^{(m)})$$

$$\mathbf{q}_i = \text{Query}_{\theta_C}(\mathbf{z}_i^{C(m)}) \qquad \phi_{c,i} = \text{Attn}_{\theta_C}(\mathbf{q}_i, \mathbf{K}_i, \mathbf{V}_i)$$

where $\text{Attn}_{\theta_C}$ is an attention mechanism that attends to the keys $\mathbf{K}_i$ and values $\mathbf{V}_i$ using query $\mathbf{q}_i$, and applies a non-linear transformation to the output. As in Drouin et al. (2022), we take each $c_{\phi_{c,i}}$ to be a histogram distribution with support in $[0, 1]$, but other choices are compatible with this approach.

**Curriculum Learning:** A crucial difference between TACTiS and TACTiS-2 lies in their training procedure. TACTiS optimizes the cumbersome, permutation-based, Problem (6). In contrast, TACTiS-2 is trained by maximum likelihood using a training curriculum, enabled by the use of a *dual encoder*, which we show is equivalent to solving the two-phase Problem (7). See Fig. 2 for an illustration.

[1]The original Deep Sigmoidal Flow is modified such that it outputs values in $[0, 1]$.

In the first phase, only the parameters $\theta_M$ for the marginal components are trained, while those for the copula, $\theta_C$, are skipped. This boils down to optimizing Problem (4) using a trivial copula where all variables are independent (which we denote by $c_I$):

$$\underset{\theta_M}{\arg\min} \; - \underset{\mathbf{x}\sim\mathbf{X}}{\mathbb{E}} \log \left[ c_I \left( F_{\phi_1}(x_1^{(m)}), \ldots, F_{\phi_{n_m}}(x_d^{(m)}) \right) \cdot \prod_{i=1}^{d} f_{\phi_i}(x_i^{(m)}) \right], \tag{11}$$

where each $f_{\phi_i}$ is obtained by differentiating $F_{\phi_i}$ w.r.t. $x_i$, an operation that is efficient for DSFs. Since by definition $c_I(\ldots) \equiv 1$, this problem reduces to Problem (8).

In the second phase, the parameters learned for the marginal components $\theta_M$ are frozen and the parameters for the copula components $\theta_C$ are trained until convergence. The optimization problem is hence given by:

$$\underset{\theta_C}{\arg\min} \; - \underset{\mathbf{x}\sim\mathbf{X}}{\mathbb{E}} \log \left[ c_{\phi_c}(F_{\phi_1^\star}(x_1^{(m)}), \ldots, F_{\phi_d^\star}(x_d^{(m)})) \cdot \prod_{i=1}^{d} f_{\phi_i^\star}(x_i^{(m)}) \right], \tag{12}$$

which reduces to Problem (7), since $\{f_{\phi_i^\star}(\ldots)\}_{i=1}^n$ are constant. Hence, by Prop. 2, we have that, given sufficient capacity, the *attentional copulas* learned by TACTiS-2 will be valid. We stress the importance of the proposed learning curriculum, since by Prop. 1, we know that simply maximizing the likelihood w.r.t. $\theta_M$ and $\theta_C$ is highly unlikely to result in valid copulas.

**Sampling:** Inference proceeds as in TACTiS: (i) sampling according to the copula density and (ii) applying the inverse CDFs to obtain samples from Eq. (1). We defer to Drouin et al. (2022) for details.

## 5 Experiments

We start by empirically validating the two-stage approach to learning attentional copulas (Sec. 5.1). Then, we show that TACTiS-2 achieves state-of-the-art performance in a forecasting benchmark and that it can perform highly accurate interpolation (Sec. 5.2). Finally, we show that TACTiS-2 outperforms TACTiS in all aspects, namely accuracy and training dynamics, while preserving its high flexibility.

### 5.1 Empirical Validation of Two-Stage Attentional Copulas

According to Prop. 2, TACTiS-2 should learn valid copulas. We empirically validate this claim in a setting where the sample size, training time, and capacity are finite. As in Drouin et al. (2022), we rely on an experiment in which data is drawn from a bivariate distribution with a known copula structure. The results in Fig. 3 show that, in this setting, TACTiS-2 recovers a valid copula that closely matches the ground truth. See App. B.3 for experimental details and additional results.

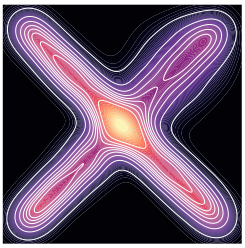


**Figure 3:** *The density of the learned copula (contours) closely matches that of the ground truth (colors).*

### 5.2 Evaluation of Predictive Performance

We now evaluate the forecasting and interpolation abilities of TACTiS-2 in a benchmark of five common real-world datasets from the *Monash Time Series Forecasting Repository* (Godahewa et al., 2021): `electricity`, `fred-md`, `kdd-cup`, `solar-10min`, and `traffic`. These were chosen due to their diverse dimensionality ($n \in [107, 826]$), sampling frequencies (monthly, hourly, and 10 min.), and prediction lengths ($\ell_i \in [12, 72]$). All datasets are detailed in App. C.1.

**Evaluation Protocol:** The evaluation follows the protocol of Drouin et al. (2022), which consists in a backtesting procedure that combines rolling-window evaluation with periodic retraining. The estimated distributions are scored using the Continuous Ranked Probability Score (CRPS; Matheson & Winkler, 1976), the CRPS-Sum (Salinas et al., 2019)—a multivariate generalization of the CRPS—and the Energy score (Gneiting & Raftery, 2007), as is standard in the literature. We further compare some models using the negative log-likelihood (NLL), which was found to be more effective in detecting errors in modeling multivariate dependencies (Marcotte et al., 2023). Experimental details are provided in App. C.

**Forecasting Benchmark:** We compare TACTiS-2 with the following state-of-the-art multivariate probabilistic forecasting methods: GPVar (Salinas et al., 2019), an LSTM-based method that parametrizes a Gaussian copula; TempFlow (Rasul et al., 2021b), a method that parametrizes normalizing flows using

**Table 1:** *Mean CRPS-Sum values for the forecasting experiments. Standard errors are calculated using the Newey-West (1987; 1994) estimator. Average ranks report the average ranking of methods across all evaluation windows and datasets. Lower is better and the best results are in bold.*

| Model | electricity | fred-md | kdd-cup | solar-10min | traffic | Avg. Rank |
|---|---|---|---|---|---|---|
| Auto-ARIMA | $0.077 \pm 0.016$ | $0.043 \pm 0.005$ | $0.625 \pm 0.066$ | $0.994 \pm 0.216$ | $0.222 \pm 0.005$ | $6.2 \pm 0.3$ |
| ETS | $0.059 \pm 0.011$ | $0.037 \pm 0.010$ | $0.408 \pm 0.030$ | $0.678 \pm 0.097$ | $0.353 \pm 0.011$ | $6.0 \pm 0.2$ |
| TempFlow | $0.075 \pm 0.024$ | $0.095 \pm 0.004$ | $0.250 \pm 0.010$ | $0.507 \pm 0.034$ | $0.242 \pm 0.020$ | $5.4 \pm 0.2$ |
| SPD | $0.062 \pm 0.016$ | $0.048 \pm 0.011$ | $0.319 \pm 0.013$ | $0.568 \pm 0.061$ | $0.228 \pm 0.013$ | $5.2 \pm 0.3$ |
| TimeGrad | $0.067 \pm 0.028$ | $0.094 \pm 0.030$ | $0.326 \pm 0.024$ | $0.540 \pm 0.044$ | $0.126 \pm 0.019$ | $5.0 \pm 0.2$ |
| GPVar | $0.035 \pm 0.011$ | $0.067 \pm 0.008$ | $0.290 \pm 0.005$ | $0.254 \pm 0.028$ | $0.145 \pm 0.010$ | $3.8 \pm 0.2$ |
| TACTiS | $0.021 \pm 0.005$ | $0.042 \pm 0.009$ | $0.237 \pm 0.013$ | $0.311 \pm 0.061$ | $\mathbf{0.071 \pm 0.008}$ | $2.4 \pm 0.2$ |
| TACTiS-2 | $\mathbf{0.020 \pm 0.005}$ | $\mathbf{0.035 \pm 0.005}$ | $\mathbf{0.234 \pm 0.011}$ | $\mathbf{0.240 \pm 0.027}$ | $0.078 \pm 0.008$ | $\mathbf{1.9 \pm 0.2}$ |

**Table 2:** *Mean NLL values for forecasting experiments and training FLOP counts. Standard errors are calculated using the Newey-West (1987; 1994) estimator. Lower is better. Best results in bold. .*

| Model | | electricity | fred-md | kdd-cup | solar-10min | traffic |
|---|---|---|---|---|---|---|
| TACTiS | NLL | $11.028 \pm 3.616$ | $1.364 \pm 0.253$ | $2.281 \pm 0.770$ | $-2.572 \pm 0.093$ | $1.249 \pm 0.080$ |
| | FLOPs ($\times 10^{16}$) | $1.931 \pm 0.182$ | $1.956 \pm 0.192$ | $1.952 \pm 0.208$ | $0.174 \pm 0.018$ | $1.207 \pm 0.517$ |
| TACTiS-2 | NLL | $\mathbf{10.674 \pm 2.867}$ | $\mathbf{0.378 \pm 0.076}$ | $\mathbf{1.055 \pm 0.713}$ | $\mathbf{-4.333 \pm 0.181}$ | $\mathbf{-0.358 \pm 0.077}$ |
| | FLOPs ($\times 10^{16}$) | $\mathbf{0.623 \pm 0.018}$ | $\mathbf{0.738 \pm 0.022}$ | $\mathbf{0.324 \pm 0.014}$ | $\mathbf{0.078 \pm 0.005}$ | $\mathbf{0.289 \pm 0.061}$ |

transformers; TimeGrad (Rasul et al., 2021a), an autoregressive model based on denoising diffusion; and Stochastic Process Diffusion (SPD) (Biloš et al., 2023), the only other general-purpose approach which models time series as continuous functions using stochastic processes as noise sources for diffusion. Moreover, we include the following classical forecasting methods, which tend to be strong baselines (Makridakis et al., 2018a;b; 2022): ARIMA (Box et al., 2015) and ETS exponential smoothing (Hyndman et al., 2008).

The CRPS-Sum results are reported in Tab. 1. Clearly, TACTiS-2 shows state-of-the-art performance, achieving the lowest values on 4 out of 5 datasets, while being slightly outperformed by TACTiS on traffic. Looking at the average rankings, which are a good indicator of performance as a general-purpose forecasting tool, we see that TACTiS-2 outperforms all baselines, with TACTiS being its closest competitor. To further contrast these two, we perform a comparison of NLL and report the results in Tab. 2 and Fig. 1. We observe that TACTiS-2 outperforms TACTiS, additionally with no overlap in the confidence intervals for 3 out of 5 datasets, including traffic. This is strong evidence that TACTiS-2 better captures multivariate dependencies (Marcotte et al., 2023), which is plausible given the proposed improvements to the attentional copula. The results for all other metrics are in line with the above findings and are reported in App. A.2.

**Training Dynamics:** As discussed in Sec. 3, the optimization problem solved by TACTiS-2 is considerably simpler than the one solved by TACTiS. We quantify this by measuring the number of floating-point operations (FLOPs) required to train until convergence in the forecasting benchmark (see Tab. 2). From these results, it is clear that TACTiS-2 achieves greater accuracies while using much less compute than TACTiS. As additional evidence of improved training dynamics, we report training curves that compare the NLL, on a validation set, with respect to training FLOPs for TACTiS-2, TACTiS, and an ablation of TACTiS-2 that does not use the two-stage curriculum. Results for the kdd-cup dataset are reported in Fig. 4 and those for other datasets are available in App. A.3. From these, we reach two conclusions: (i) TACTiS-2 converges much faster to better solutions, (ii) the two-stage curriculum is crucial TACTiS-2's success.

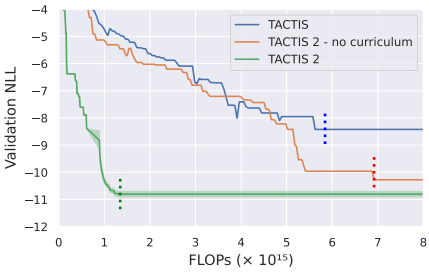
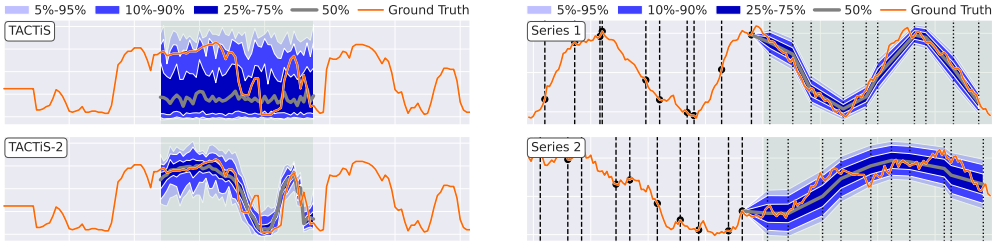



**Figure 4:** TACTiS-2 *converges to better NLLs using fewer FLOPs than* TACTiS*, as well as an ablation that trains all parameters jointly without the two-stage curriculum. Vertical bars indicate the latest convergence point over 5 runs with a maximum duration of three days.*

**Interpolation Performance:** While both TACTiS and TACTiS-2 are capable of interpolation, the results reported in Tab. 3 indicate that TACTiS-2 is much better at this task. This is illustrated in Fig. 5a, which

**Table 3:** *Mean NLL values for the interpolation experiments. Standard errors are calculated using the Newey-West (1987; 1994) estimator. Lower is better and the best results are in bold.*

| Model | electricity | fred-md | kdd-cup | solar-10min | traffic |
|---|---|---|---|---|---|
| TACTiS | $-0.159 \pm 0.007$ | $-0.119 \pm 0.021$ | $1.050 \pm 0.045$ | $-0.948 \pm 0.089$ | $0.825 \pm 0.149$ |
| TACTiS-2 | $\mathbf{-0.189 \pm 0.034}$ | $\mathbf{-0.250 \pm 0.017}$ | $\mathbf{0.138 \pm 0.043}$ | $\mathbf{-3.262 \pm 0.186}$ | $\mathbf{-0.431 \pm 0.063}$ |




**(a)** *Interpolation:* TACTiS-2 *(bottom) produces more accurate interpolation distributions than* TACTiS *(top) – example from* kdd-cup.

**(b)** TACTiS-2 *correctly forecasts two unaligned/unevenly-sampled series on the noisy sines dataset. Dashed bars indicate measurements.*

**Figure 5:** *An illustration of the flexibility of* TACTiS-2.

shows an example in which TACTiS-2 produces a much more plausible interpolation distribution than TACTiS. Additional examples are available in App. A.5.

**Model Flexibility:** Like TACTiS, the TACTiS-2 architecture supports heterogenous datasets composed of unaligned series with uneven sampling frequencies. We illustrate this by replicating an experiment of Drouin et al. (2022) which consists in forecasting a bivariate noisy sine process with irregularly-spaced observations. The results in Fig. 5b show that TACTiS-2 faithfully performs forecasting in this setting, preserving the flexiblity of TACTiS. Additional results on real-world datasets are available in App. D.1.

## 6 Related Work

**Deep Learning for Probabilistic Time Series Prediction:** The majority of prior work in this field has been geared toward forecasting tasks. Early work explored the application of recurrent and convolutional neural networks to estimating univariate forecast distributions (Rangapuram et al., 2018; Shih et al., 2019; Chen et al., 2020; de Bézenac et al., 2020; Yanchenko & Mukherjee, 2020). More relevant to the present work are methods for *multivariate* forecasting, which we review in light of the desiderata outlined in Sec. 1. Many such works lack the flexibility to model arbitrary joint distributions. For instance, DeepAR (Salinas et al., 2020) uses an RNN to parametrize an autoregressive factorization of the joint density using a chosen parametric form (e.g., Gaussian). Similarly, GPVar (Salinas et al., 2019) uses an RNN to parametrize a low-rank Gaussian copula approximation of the joint distribution. Wu et al. (2020) rely on an adversarial sparse transformer to estimate conditional quantiles of the predictive distribution but do not model the full joint density. Other approaches relax such limitations, by enabling the estimation of arbitrary distributions. For instance, TempFlow (Rasul et al., 2021b) uses RNNs and transformers (Vaswani et al., 2017) to parametrize a multivariate normalizing flow (Papamakarios et al., 2021). TimeGrad (Rasul et al., 2021a) models the one-step-ahead predictive joint distribution with a denoising diffusion model (Ho et al., 2020; Sohl-Dickstein et al., 2015) conditioned on an RNN. However, all of the above lack the flexibility to handle unaligned/unevenly-sampled series, missing data, and are limited to forecasting.

Recently, multiple works have made progress toward our desiderata. Among those, TACTiS (Drouin et al., 2022), which we have extensively described, and SPD (Biloš et al., 2023), which uses stochastic processes as noise sources during diffusion, are the only ones that satisfy all of them. An alternative approach, CSDI (Tashiro et al., 2021), which relies on a conditional score-based diffusion model trained using a self-supervised interpolation objective, supports both forecasting and interpolation. However, it cannot learn from unaligned time series. Similarly, SSSD (Alcaraz & Strodthoff, 2023), a conditional diffusion model that relies on structured state-space models (Gu et al., 2022) as internal layers, can perform both forecasting and interpolation but does not support unaligned and unevenly-sampled time series.

**Copulas for Probabilistic Forecasting:** Copula-based models for multivariate forecasting have been extensively studied in economics and finance (Patton, 2012; Größer & Okhrin, 2021), with applications such as modeling financial returns and volatility (Bouyé & Salmon, 2009; Bouyé et al., 2008; Wang & Tao,

2020). A common semiparametric approach consists of combining a parametric copula, e.g., Archimedean, with nonparametric empirical estimates of the marginal CDFs (ECDFs). One notable example is the work of Salinas et al. (2019), which combines ECDFs with an RNN that dynamically parametrizes a Gaussian copula. A key limitation of such approaches is that ECDF estimates are only valid for stationary processes. Moving away from this assumption, Wen & Torkkola (2019) propose to use neural networks to model the CDFs in addition to a Gaussian copula. However, the choice of a Gaussian copula is still a strong parametric assumption which we seek to avoid. Closer to our work are methods that relax such parametric assumptions, such as Krupskii & Joe (2020); Mayer & Wied (2021); Toubeau et al. (2019); Drouin et al. (2022). Toubeau et al. (2019) use the historical data to estimate a histogram-based nonparametric copula. A key caveat to this approach is that it assumes that the historical data is independent and identically distributed. Finally, the work most similar to ours is the nonparametric approach of Drouin et al. (2022), which, as extensively described, we significantly simplify and improve upon.

**Copulas in Machine Learning:** Beyond time series, copulas have been applied to various machine learning problems, such as domain adaptation (Lopez-Paz et al., 2012), variational inference (Tran et al., 2015; Hirt et al., 2019), learning disentangled representations (Wieser et al., 2018), dependency-seeking clustering (Rey & Roth, 2012), and generative modeling (Sexton et al., 2022; Tagasovska et al., 2019; Wang & Wang, 2019). Hence, we emphasize that our proposed transformer-based nonparametric copulas are applicable beyond time series and review two closely related works. First, Janke et al. (2021) propose an approach to learning nonparametric copula that uses a generative adversarial network (Goodfellow et al., 2014) to learn a latent distribution on the unit cube. This distribution is then transformed into a valid copula distribution using its ECDFs. As opposed to our approach, which is fully differentiable, their reliance on ECDFs leads to a non-differentiable objective that must be approximated during training. Second, Wiese et al. (2019) use normalizing flows to parametrize both the marginals and the copulas, resulting in an approach that is fully differentiable. However, they focus on the bivariate case and rely on vine copulas to extend their approach to multivariate copulas, something that our approach does not require.

## 7 Discussion

This work introduces TACTiS-2, a general-purpose model for multivariate probabilistic time series prediction, combining the flexibility of transformers with a new approach to learning attention-based nonparametric copulas. TACTiS-2 establishes itself as the new state-of-the-art model for forecasting on several real-world datasets, while showing better training dynamics than its predecessor, TACTiS. This superior performance is mainly due to its simplified optimization procedure, which ultimately allows it to reach better solutions, in particular better copulas. TACTiS-2's performance is further enhanced by the use of a dual-encoder that learns representations specialized for each distributional component.

There are multiple ways in which TACTiS-2 could be improved. For example, it would be interesting to incorporate inductive biases specific to time series data into the architecture, e.g., using Fourier features to learn high-frequency patterns (Woo et al., 2022), and auto-correlation mechanisms in the attention layers (Wu et al., 2021). Next, TACTiS-2 is limited to working with continuous data due to its usage of normalizing flows for the marginal distributions. This limitation could be addressed by building on prior work (Tran et al., 2019; Ziegler & Rush, 2019) to adapt the normalizing flows to work with discrete distributions. Finally, it is important to note that, as for all related work (Wiese et al., 2019; Drouin et al., 2022; Janke et al., 2021), the nonparametric copulas learned by TACTiS-2 are valid in the limit of infinite data and capacity. As future work, it would be interesting to study the convergence properties of these approaches to valid copulas in settings with finite samples and non-convex optimization landscapes.

Beyond the scope of this work, there are interesting settings in which the capabilities of TACTiS-2 could be studied further. The proposed form of fully decoupling the marginal distributions and the dependency structure could be especially useful in handling distribution shifts, which are common in real-world time series (Yao et al., 2022; Gagnon-Audet et al., 2023). Such a decoupling allows changes in either the marginal distributions or the dependency structure or both these factors to be understood separately in scenarios of distribution shift, allowing to adapt the model appropriately. Next, it would be interesting to study large-scale training of TACTiS-2 on several related time series from a specific domain, where marginal distributions can be trained for each series, while the attentional copula component can be shared. Finally, it would be interesting to exploit the flexibility of TACTiS-2 for multitask pretraining, where the model is jointly trained on multiple probabilistic prediction tasks such as forecasting, interpolation and imputation, and can be used as a general-purpose model downstream. Such extensions to TACTiS-2 constitute exciting directions towards foundation models (Bommasani et al., 2021) for time series.

## ACKNOWLEDGEMENTS

The authors are grateful to Putra Manggala, Sébastien Paquet and Sokhna Diarra Mbacke for their valuable feedback and suggestions. The authors are grateful to Juan Rodriguez and Daniel Tremblay for their help with the experiments. This research was supported by a Mitacs Accelerate Grant.

## REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=hHiIbk7ApW.

Elisabeth Wreford Andersen. Two-stage estimation in copula models used in family studies. *Lifetime Data Analysis*, 11(3):333–350, 2005. ISSN 1380-7870. doi: 10.1007/s10985-005-2966-7.

Marin Biloš. personal communication, 2023.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2452–2470. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/bilos23a.html.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Eric Bouyé and Mark Salmon. Dynamic copula quantile regressions and tail area dynamic dependence in forex markets. *The European Journal of Finance*, 15(7-8):721–750, 2009. URL https://EconPapers.repec.org/RePEc:taf:eurjfi:v:15:y:2009:i:7-8:p:721-750.

Eric Bouyé, Mark Salmon, and Nicolas Gaussel. Investing dynamic dependence using copulae. *SSRN Electronic Journal*, 09 2008. doi: 10.2139/ssrn.1272353.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, fifth edition, 2015.

Yitian Chen, Yanfei Kang, Yixiong Chen, and Zizhuo Wang. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing*, 399:491–501, 2020.

Emmanuel de Bézenac, Syama Sundar Rangapuram, Konstantinos Benidis, Michael Bohlke-Schneider, Richard Kurle, Lorenzo Stella, Hilaf Hasson, Patrick Gallinari, and Tim Januschowski. Normalizing Kalman filters for multivariate time series analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2995–3007. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1f47cef5e38c952f94c5d61726027439-Paper.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. TACTiS: Transformer-attentional copulas for time series. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5447–5493. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/drouin22a.html.

Chenguang Fang and Chen Wang. Time series data imputation: A survey on deep learning approaches. *ArXiv*, abs/2011.11347, 2020. URL https://arxiv.org/abs/2011.11347.

Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad Javad Darvishi Bayazi, Pooneh Mousavi, Guillaume Dumas, and Irina Rish. WOODS: Benchmarks for out-of-distribution generalization in time series. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=mvftzofTYQ. Featured Certification.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014. doi: 10.1146/annurev-statistics-062713-085831. URL https://doi.org/10.1146/annurev-statistics-062713-085831.

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

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Joshua Größer and Ostap Okhrin. Copulae: An overview and recent developments. *WIREs Computational Statistics*, n/a(n/a):e1557, 2021. doi: https://doi.org/10.1002/wics.1557. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1557.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Marcel Hirt, Petros Dellaportas, and Alain Durmus. Copula-like variational inference. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/e721a54a8cf18c8543d44782d9ef681f-Paper.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2078–2087. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/huang18d.html.

Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Australia, 2nd edition, 2018.

Tim Janke, Mohamed Ghanmi, and Florian Steinke. Implicit generative copulas. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 26028–26039. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/dac4a67bdc4a800113b0f1ad67ed696f-Paper.pdf.

Harry Joe. Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of Multivariate Analysis*, 94(2):401–419, 2005. URL https://EconPapers.repec.org/RePEc:eee:jmvana:v:94:y:2005:i:2:p:401-419.

Harry Joe and James Jianmeng Xu. The estimation method of inference functions for margins for multivariate models, Oct 1996. URL https://open.library.ubc.ca/collections/facultyresearchandpublications/52383/items/1.0225985.

Pavel Krupskii and Harry Joe. Flexible copula models with dynamic dependence and application to financial data. *Econometrics and Statistics*, 16:148–167, 2020. ISSN 2452-3062. doi: https://doi.org/10.1016/j.ecosta.2020.01.005. URL https://www.sciencedirect.com/science/article/pii/S2452306220300216.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194): 20200209, 2021. doi: 10.1098/rsta.2020.0209. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0209.

David Lopez-Paz, Jose Hernández-lobato, and Bernhard Schölkopf. Semi-supervised domain adaptation with non-parametric copulas. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/8e98d81f8217304975ccb23337bb5761-Paper.pdf.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: Results, findings, conclusion and way forward. *International Journal of forecasting*, 34(4):802–808, 2018a.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13(3):e0194889–26, 2018b.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 2022. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.11.013. URL https://www.sciencedirect.com/science/article/pii/S0169207021001874.

Étienne Marcotte, Valentina Zantedeschi, Alexandre Drouin, and Nicolas Chapados. Regions of reliability in the evaluation of multivariate probabilistic forecasts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23958–24004. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/marcotte23a.html.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Alexander Mayer and Dominik Wied. Estimation and inference in factor copula models with exogenous covariates, 2021. URL https://arxiv.org/abs/2107.03366.

Roger B. Nelsen. *An introduction to copulas*. Springer Science & Business Media, second edition, 2007.

Whitney K. Newey and Kenneth D. West. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3):703–708, 1987. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913610.

Whitney K. Newey and Kenneth D. West. Automatic Lag Selection in Covariance Matrix Estimation. *The Review of Economic Studies*, 61(4):631–653, 10 1994. ISSN 0034-6527. doi: 10.2307/2297912. URL https://doi.org/10.2307/2297912.

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. URL http://jmlr.org/papers/v22/19-1028.html.

Andrew J. Patton. A review of copula models for economic time series. *Journal of Multivariate Analysis*, 110:4–18, 2012.

Martin Peterson. *An Introduction to Decision Theory*. Cambridge Introductions to Philosophy. Cambridge University Press, second edition, 2017. doi: 10.1017/9781316585061.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/rasul21a.html.

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*, 2021b. URL https://openreview.net/forum?id=WiGQBFuVRv.

Mélanie Rey and Volker Roth. Copula mixture model for dependency-seeking clustering. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 275–282, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank Gaussian copula processes. *Advances in Neural Information Processing Systems*, 32:6827–6837, 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, 2020.

Jean-Thomas Sexton, Michael Morin, and Jonathan Gaudreault. Vine copula based data generation for machine learning with an application to industrial processes. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022. URL https://openreview.net/forum?id=EIPGOyAgIh5.

Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8-9):1421–1441, 2019.

Satya Narayan Shukla and Benjamin M Marlin. A survey on principles, models and methods for learning from irregularly sampled time series: From discretization to attention and invariance. *ArXiv*, abs/2012.00168, 2020. URL https://arxiv.org/abs/2012.00168.

Abe Sklar. Fonctions de répartition à $n$ dimensions et leurs marges. *Publications de l'Institut Statistique de l'Université de Paris*, 8:229–231, 1959.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. Copulas as high-dimensional generative models: Vine copula autoencoders. *Advances in neural information processing systems*, 32, 2019.

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, 2021.

José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: A survey. *Big Data*, 9(1):3–21, 2021. doi: 10.1089/big.2020.0159. URL https://doi.org/10.1089/big.2020.0159. PMID: 33275484.

Jean-François Toubeau, Jérémie Bottieau, François Vallée, and Zacharie De Grève. Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets. *IEEE Transactions on Power Systems*, 34(2):1203–1215, 2019. doi: 10.1109/TPWRS.2018.2870041.

Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/e4dd5528f7596dcdf871aa55cfccc53c-Paper.pdf.

Dustin Tran, Keyon Vafa, Kumar Agrawal, Laurent Dinh, and Ben Poole. Discrete flows: Invertible generative models of discrete data. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/e046ede63264b10130007afca077877f-Paper.pdf.

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*, pp. 5998–6008, 2017.

Prince Zizhuang Wang and William Yang Wang. Neural Gaussian copula for variational autoencoder. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:202538985.

Wenjing Wang and Minjing Tao. Forecasting realized volatility matrix with copula-based models, 2020.

Ruofeng Wen and Kari Torkkola. Deep generative quantile-copula models for probabilistic forecasting, 2019.

Magnus Wiese, Robert Knobloch, and Ralf Korn. Copula & marginal flows: Disentangling the marginal from its joint. *arXiv preprint arXiv:1907.03361*, 2019.

Mario Wieser, Aleksander Wieczorek, Damian Murezzan, and Volker Roth. Learning sparse latent representations with the deep copula information bottleneck. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk0wHx-RW.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=I55UqU-M11y.

Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17105–17115. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c6b8c8d762da15fa8dbbdfb6baf9e260-Paper.pdf.

Anna K. Yanchenko and Sayan Mukherjee. Stanza: A nonlinear state space model for probabilistic inference in non-stationary time series. *arXiv*, pp. 2006.06553v1, 2020.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=F9ENmZABB0.

Achim Zeileis, Susanne Köll, and Nathaniel Graham. Various versatile variances: An object-oriented implementation of clustered covariances in R. *Journal of Statistical Software*, 95(1):1–36, 2020. doi: 10.18637/jss.v095.i01.

Zachary Ziegler and Alexander Rush. Latent normalizing flows for discrete sequences. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7673–7682. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ziegler19a.html.

# Appendix

## Table of Contents


**A Metrics and Additional Results** **15**
- A.1 Metric Definitions . . . 15
- A.2 Results on Alternative Forecasting Metrics . . . 16
- A.3 Training Dynamics . . . 16
- A.4 Qualitative Results on Forecasting . . . 18
- A.5 Qualitative Results on Interpolation . . . 19

**B Theory and Proofs** **21**
- B.1 Proof of Proposition 1 . . . 21
- B.2 Proof of Proposition 2 . . . 21
- B.3 Example of Valid and Invalid Decompositions . . . 22

**C Experimental Details** **25**
- C.1 Datasets . . . 25
- C.2 Training Procedure . . . 25
- C.3 Hyperparameter Search Protocol . . . 25
- C.4 Selected Hyperparameters . . . 26

**D Additional Results** **29**
- D.1 Demonstration of Model Flexibility in Real-World Datasets . . . 29


## A Metrics and Additional Results

### A.1 Metric Definitions

**CRPS:** The Continuous Ranked Probability Score (Matheson & Winkler, 1976; Gneiting & Raftery, 2007) is a univariate metric used to assess the accuracy of a forecast distribution $X$, given a ground-truth realization $x^*$:

$$\mathrm{CRPS}(x^*, X) \stackrel{\text{def}}{=} 2 \int_0^1 \left( \mathbf{1}\big[F^{-1}(q) - x^*\big] - q \right) \left( F^{-1}(q) - x^* \right) dq, \tag{13}$$

where $F(x)$ is the cumulative distribution function (CDF) of the forecast distribution, and $\mathbf{1}[x]$ is the Heaviside function. We follow previous authors in normalizing the CRPS results using the mean absolute value of the ground-truth realizations of each series, before taking the average over all series and time steps. Since the CRPS is a univariate metric, it fully ignores the quality of the multivariate dependencies in the forecasts being assessed.

**CRPS-Sum:** The CRPS-Sum (Salinas et al., 2019) is an adaptation of the CRPS to take into account some of the multivariate dependencies in the forecasts. The CRPS-Sum is the result of summing both the forecast and the ground-truth values over all series, before computing the CRPS over the resulting sums.

**Energy Score:** The Energy Score (ES) (Gneiting & Raftery, 2007) is a multivariate metric. For a multivariate forecast distribution $\mathbf{X}$ and ground-truth realisation $\mathbf{x}^*$, it is computed as:

$$\mathrm{ES}(\mathbf{x}^*, \mathbf{X}) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathbf{X}} |\mathbf{x} - \mathbf{x}^*|_2^\beta - \frac{1}{2} \mathop{\mathbb{E}}_{\substack{\mathbf{x} \sim \mathbf{X} \\ \mathbf{x}' \sim \mathbf{X}}} |\mathbf{x} - \mathbf{x}'|_2^\beta, \tag{14}$$

where $|\mathbf{x}|_2$ is the Euclidean norm and $0 < \beta \leq 2$ is a parameter we set to 1.

**NLL:** The Negative Log-Likelihood (NLL) is a metric that directly uses the forecast probability distribution function (PDF) instead of sampling from it. In the case of a continuous distribution with PDF $p(\mathbf{x})$ and a ground-truth realisation $\mathbf{x}^*$, the NLL is as follow:

$$\text{NLL}(\mathbf{x}^*, p(\mathbf{x})) \stackrel{\text{def}}{=} -\log p(\mathbf{x}^*). \tag{15}$$

We normalize the NLL results by dividing it by the number of dimensions of the forecast.

**Standard Error Computation:** In all of our metric results, the standard errors (indicated by $\pm$) are computed using the Newey-West (1987; 1994) estimator. This estimator takes into account the autocorrelation and heteroscedasticity that are inherent to the sequential nature of our backtesting procedure, leading to wider standard errors than those computed using the assumption that the metric values for each backtesting period are independent. In particular, we use the implementation from the R `sandwich` package (Zeileis et al., 2020), with the Bartlett kernel weights, 3 lags, and automatic bandwidth selection.

### A.2 Results on Alternative Forecasting Metrics

**Table 4:** *CRPS means: Averaged across all backtest sets. Standard errors are calculated using the Newey-West (1987; 1994) estimator. Average ranks report the average ranking of methods across all evaluation windows and datasets. Lower is better and the best results are in bold.*

| **Model** | `electricity` | `fred-md` | `kdd-cup` | `solar-10min` | `traffic` | **Avg. Rank** |
|---|---|---|---|---|---|---|
| ETS | $0.094 \pm 0.014$ | $0.050 \pm 0.011$ | $0.560 \pm 0.028$ | $0.844 \pm 0.119$ | $0.437 \pm 0.012$ | $6.5 \pm 0.2$ |
| Auto-ARIMA | $0.129 \pm 0.015$ | $0.052 \pm 0.005$ | $0.477 \pm 0.015$ | $0.636 \pm 0.060$ | $0.310 \pm 0.004$ | $5.7 \pm 0.2$ |
| TempFlow | $0.109 \pm 0.024$ | $0.110 \pm 0.003$ | $0.451 \pm 0.005$ | $0.547 \pm 0.036$ | $0.320 \pm 0.015$ | $5.6 \pm 0.2$ |
| TimeGrad | $0.101 \pm 0.027$ | $0.142 \pm 0.058$ | $0.495 \pm 0.023$ | $0.560 \pm 0.047$ | $0.217 \pm 0.015$ | $5.2 \pm 0.3$ |
| SPD | $0.099 \pm 0.016$ | $0.058 \pm 0.011$ | $0.465 \pm 0.005$ | $0.585 \pm 0.050$ | $0.283 \pm 0.007$ | $5.1 \pm 0.3$ |
| GPVar | $0.067 \pm 0.010$ | $0.086 \pm 0.009$ | $0.459 \pm 0.009$ | $0.298 \pm 0.034$ | $0.213 \pm 0.009$ | $4.1 \pm 0.1$ |
| TACTiS | $0.052 \pm 0.006$ | $0.048 \pm 0.010$ | $0.420 \pm 0.007$ | $0.326 \pm 0.049$ | $\mathbf{0.161 \pm 0.009}$ | $2.2 \pm 0.1$ |
| TACTiS-2 | $\mathbf{0.049 \pm 0.006}$ | $\mathbf{0.043 \pm 0.006}$ | $\mathbf{0.413 \pm 0.007}$ | $\mathbf{0.256 \pm 0.029}$ | $0.162 \pm 0.010$ | $\mathbf{1.6 \pm 0.2}$ |

**Table 5:** *Energy score means: Averaged across all backtest sets. Standard errors are calculated using the Newey-West (1987; 1994) estimator. Average ranks report the average ranking of methods across all evaluation windows and datasets. Lower is better and the best results are in bold.*

| **Model** | `electricity` $\times 10^4$ | `fred-md` $\times 10^5$ | `kdd-cup` $\times 10^3$ | `solar-10min` $\times 10^2$ | `traffic` $\times 10^0$ | **Avg. Rank** |
|---|---|---|---|---|---|---|
| Auto-ARIMA | $44.59 \pm 8.56$ | $8.72 \pm 0.81$ | $18.76 \pm 3.31$ | $19.42 \pm 3.37$ | $4.10 \pm 0.05$ | $6.7 \pm 0.2$ |
| TempFlow | $10.25 \pm 2.03$ | $20.16 \pm 0.74$ | $3.30 \pm 0.28$ | $4.25 \pm 0.16$ | $4.59 \pm 0.25$ | $6.1 \pm 0.3$ |
| ETS | $7.94 \pm 0.93$ | $7.90 \pm 1.88$ | $3.60 \pm 0.24$ | $4.74 \pm 0.17$ | $4.98 \pm 0.07$ | $5.7 \pm 0.2$ |
| TimeGrad | $9.69 \pm 2.62$ | $19.87 \pm 7.23$ | $3.30 \pm 0.19$ | $4.31 \pm 0.23$ | $3.38 \pm 0.11$ | $4.7 \pm 0.3$ |
| SPD | $8.90 \pm 1.18$ | $8.94 \pm 1.82$ | $3.13 \pm 0.24$ | $3.68 \pm 0.31$ | $3.94 \pm 0.10$ | $4.4 \pm 0.3$ |
| GPVar | $6.80 \pm 0.62$ | $11.43 \pm 1.60$ | $3.18 \pm 0.20$ | $2.60 \pm 0.10$ | $3.57 \pm 0.10$ | $3.9 \pm 0.2$ |
| TACTiS | $5.42 \pm 0.57$ | $8.18 \pm 1.83$ | $2.93 \pm 0.22$ | $2.88 \pm 0.23$ | $\mathbf{3.10 \pm 0.13}$ | $2.7 \pm 0.3$ |
| TACTiS-2 | $\mathbf{4.91 \pm 0.52}$ | $\mathbf{6.72 \pm 0.10}$ | $\mathbf{2.81 \pm 0.19}$ | $\mathbf{2.37 \pm 0.14}$ | $3.36 \pm 0.27$ | $\mathbf{1.9 \pm 0.2}$ |

### A.3 Training Dynamics

We now look deeper into the training dynamics of TACTiS-2. Figs. 6 and 7 compare the validation NLL with respect to the number of FLOPs performed during training on the `kdd-cup` and `solar-10min` datasets respectively. We present such an analysis for TACTiS, TACTiS-2, and we further consider an ablation of TACTiS-2 where the proposed two-stage curriculum (see Sec. 4) is not used. Instead, all the model parameters are trained jointly to minimize the NLL, in a single stage. In doing so, the resulting attentional copulas are absolutely not guaranteed to be valid (Prop. 1). Such an ablation study allows us to understand if the increased performance and efficiency come from the architecture of TACTiS-2 or the two-stage curriculum itself.

From the results, it is clear that TACTiS-2 converges in much fewer FLOPs and to better likelihoods, compared to TACTiS. Notably, TACTiS-2 achieves better performance at any given FLOP budget. TACTiS-2 without the curriculum goes to slightly better negative-log-likelihoods than TACTiS, but requires more FLOPs to reach convergence. This suggests that, in addition to producing valid copulas, the modular architecture of TACTiS-2 and its two-stage curriculum, help it achieve better performance and efficiency.

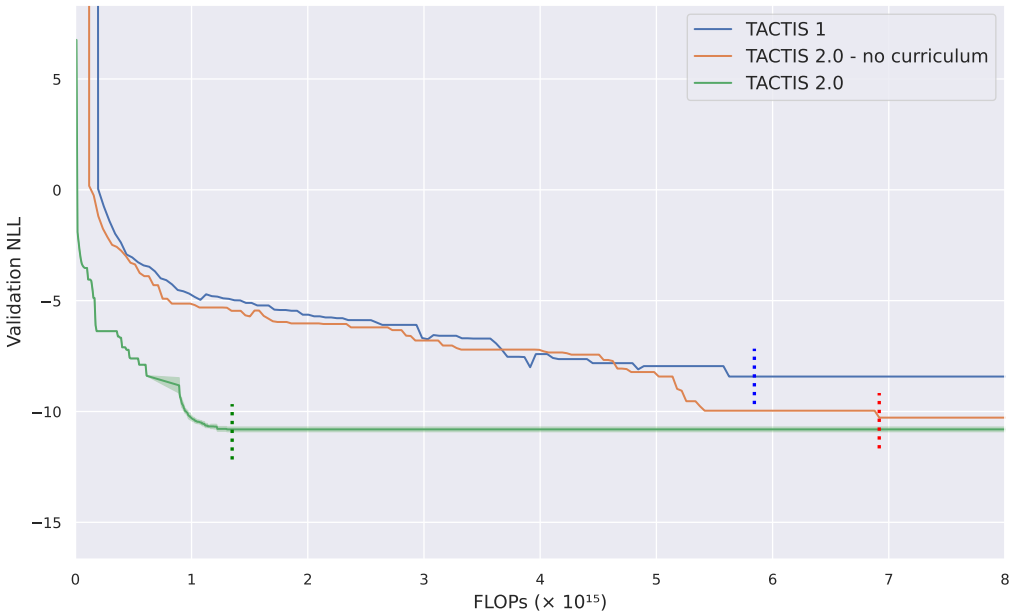



**Figure 6:** *FLOPs consumed vs validation negative log-likelihood on the* `kdd-cup` *dataset. The results are means of 5 seeds; the shaded region shows the standard error between the seeds. The dashed vertical lines on each curve indicate the latest point of convergence over the 5 seeds with a maximum training duration of three days.*

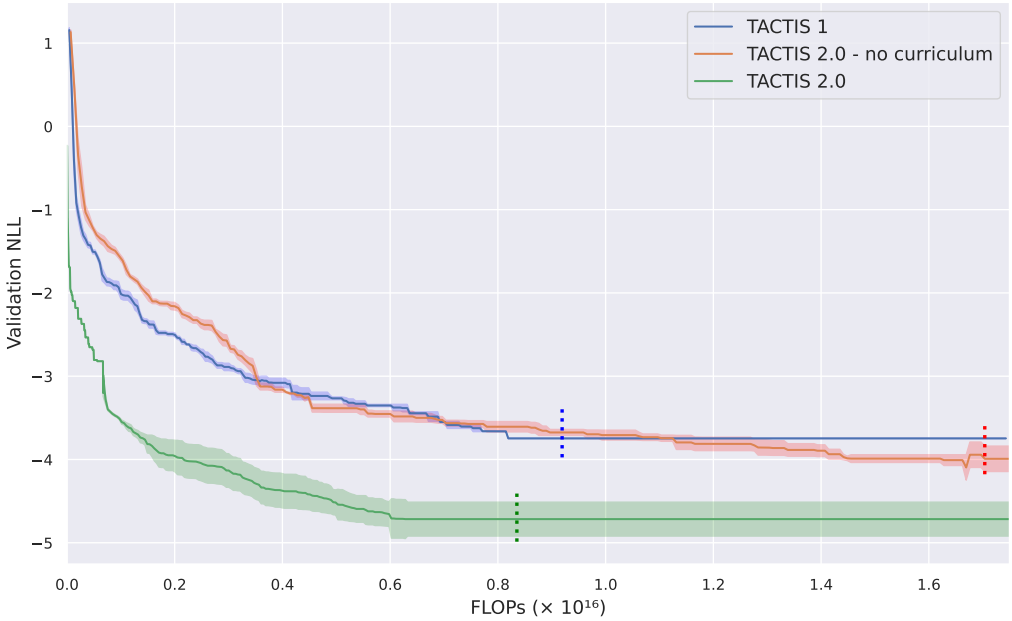



**Figure 7:** *FLOPs consumed vs validation negative log-likelihood on the* `solar-10min` *dataset. The results are means of 5 seeds; the shaded region shows the standard error between the 5 seeds. The dashed vertical lines on each curve indicate the latest point of convergence over the 5 seeds with a maximum training duration of three days.*

### A.4 QUALITATIVE RESULTS ON FORECASTING

Figs. 8 and 9 show a few examples forecasts by TACTiS-2 on the `solar-10min` and `electricity` datasets, respectively. These were picked as examples of particularly good forecasts.

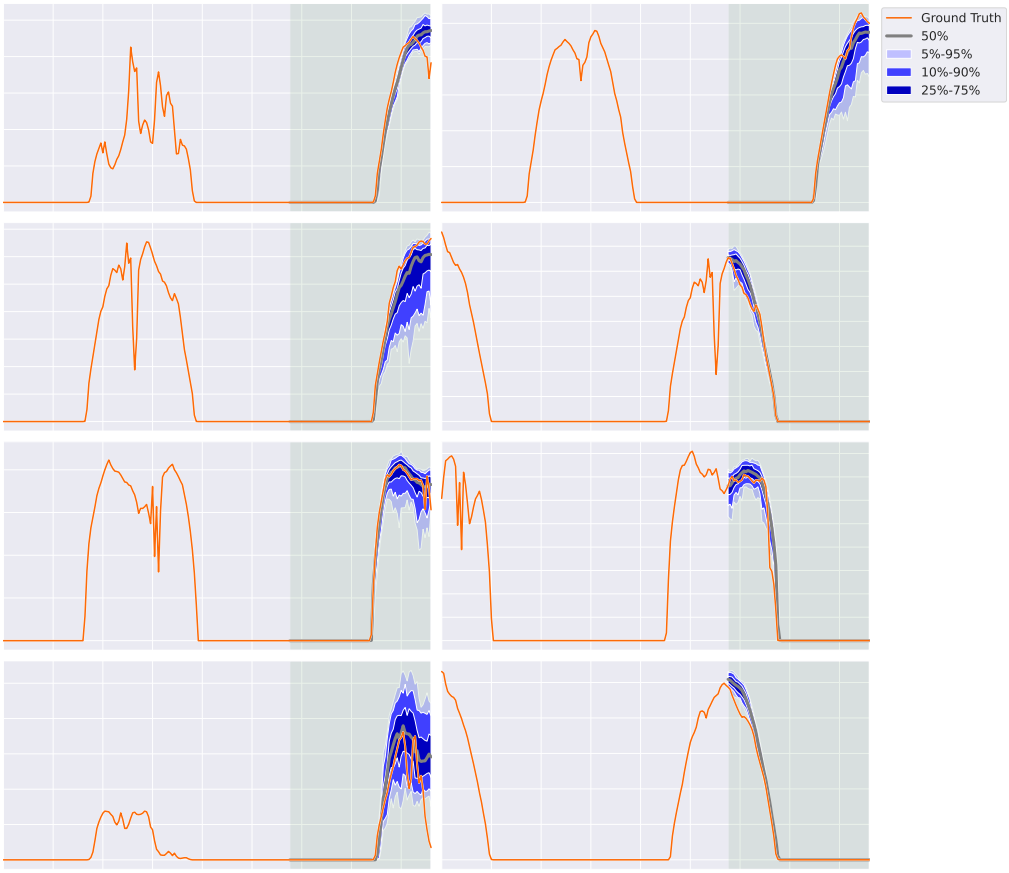



**Figure 8:** *Example forecasts by* TACTiS-2 *on the* `solar-10min` *dataset, along with the historical ground truth.*

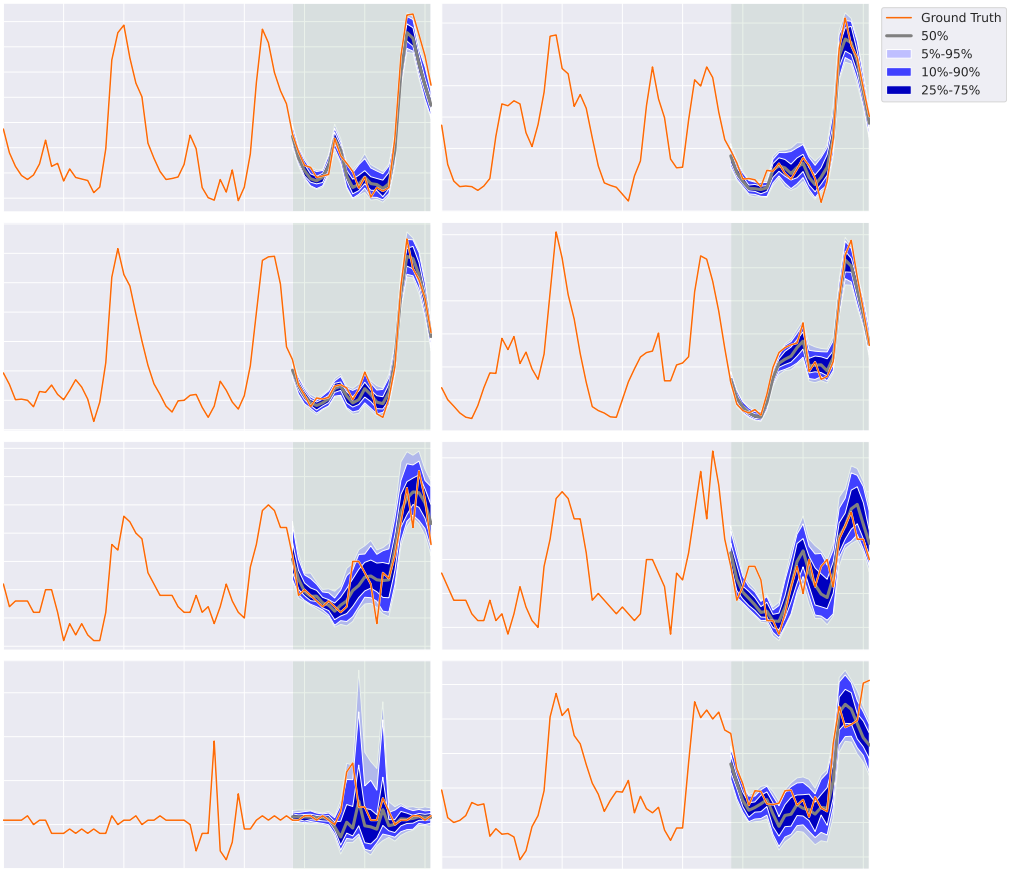



**Figure 9:** *Example forecasts by* TACT*i*S-2 *on the* `electricity` *dataset, along with the historical ground truth.*

### A.5 Qualitative Results on Interpolation

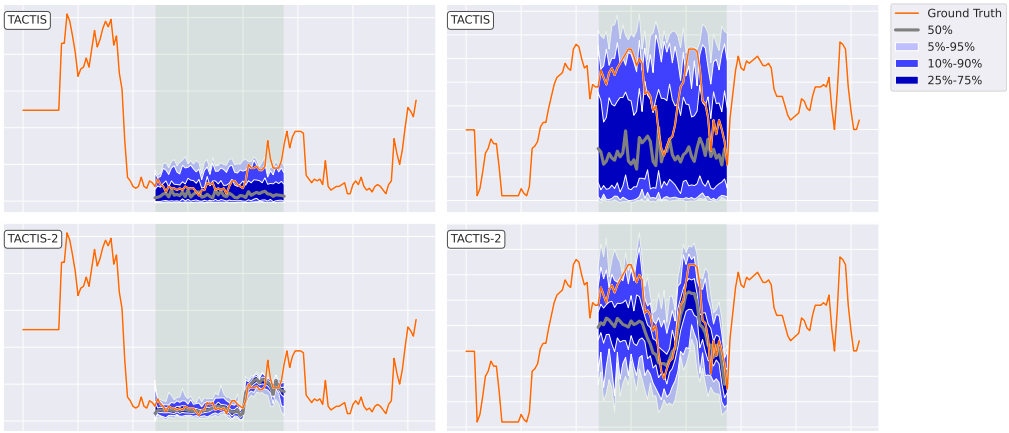



**Figure 10:** *Example interpolations by* TACT*i*S *(top) and* TACT*i*S-2 *(bottom) on the* `kdd-cup` *dataset, along with the context provided to the model.*

Fig. 10 and Fig. 11 show a few examples of interpolations for the `kdd-cup` dataset, comparing TACTiS and TACTiS-2. Fig. 12 and Fig. 13 show a few examples of interpolations for the `solar-10min` dataset, comparing TACTiS and TACTiS-2. All these examples were picked randomly.

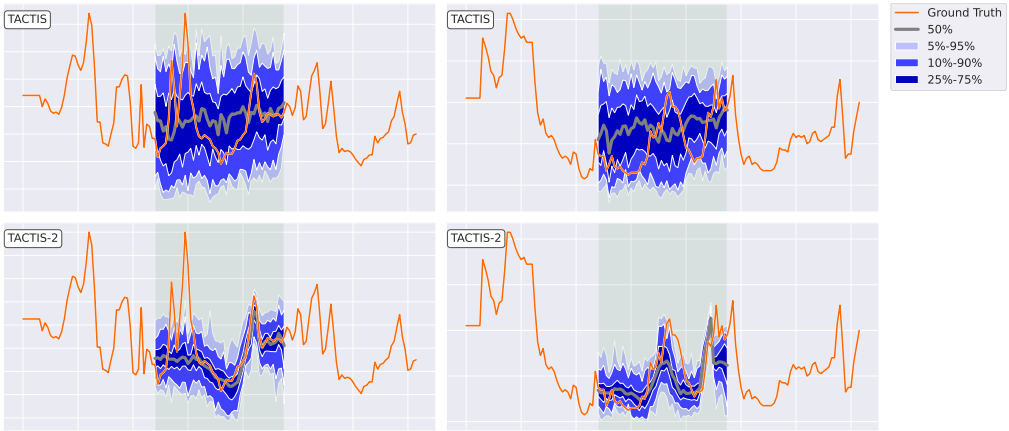



**Figure 11:** *More example interpolations by* TACT*i*S *(top) and* TACT*i*S-2 *(bottom) on the* `kdd-cup` *dataset, along with the context provided to the model.*

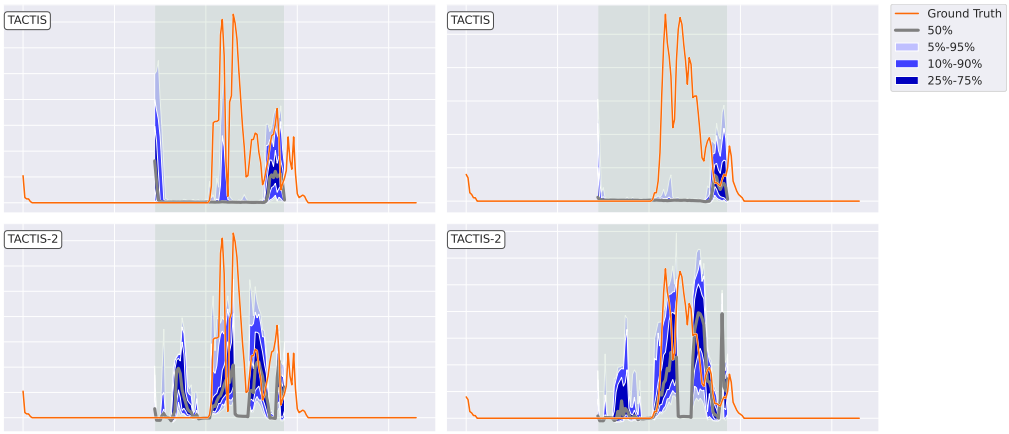



**Figure 12:** *Example interpolations by* TACT*i*S *(top) and* TACT*i*S-2 *(bottom) on the* `solar-10min` *dataset, along with the context provided to the model.*

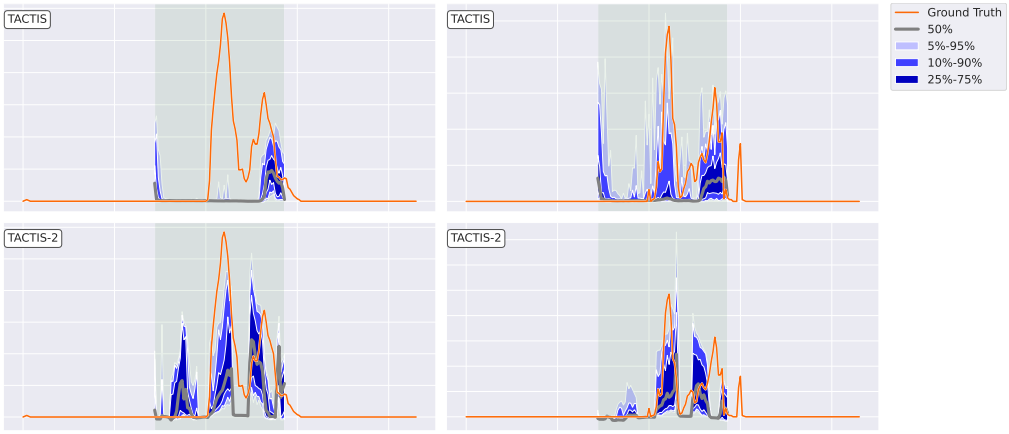



**Figure 13:** *More example interpolations by* TACT*i*S *(top) and* TACT*i*S-2 *(bottom) on the* `solar-10min` *dataset, along with the context provided to the model.*

# B THEORY AND PROOFS

## B.1 PROOF OF PROPOSITION 1

**Proposition 1**. (Invalid Solutions) Assuming that all random variables $X_1, \ldots, X_d$ have continuous marginal distributions and assuming infinite expressivity for $\{F_{\phi_i}\}_{i=1}^d$ and $c_{\phi_c}$, Problem (4) has infinitely many invalid solutions wherein $c_{\phi_c}$ is not the density function of a valid copula.

*Proof.* Consider a joint distribution over $n$ continuous random variables $X_1, \ldots, X_n$ with CDF $P(x_1, \ldots, x_n)$. According to Sklar's theorem (Sklar, 1959), we know that the joint CDF factorizes as:

$$P(x_1, \ldots, x_n) = C^\star \left(F_1^\star(x_1), \ldots, F_n^\star(x_n)\right),$$

where $C^\star$ and $F_i^\star$ are the true underlying copula and marginal distributions of $P$, respectively. Further, since all $X_i$ are continuous, this factorization is unique.

Now, consider Problem (4). Without loss of generality, the proof considers the joint CDF instead of the PDF that appears in the main text. This problem can be solved by finding $\phi_1, \ldots, \phi_d, \phi_c$, such that $P(X_1, \ldots, X_d) = C_{\phi_c}\left(F_{\phi_1}(x_1) \ldots, F_{\phi_n}(x_n)\right)$. In what follows, we show that, assuming infinite expressivity for $C_{\phi_c}$ and the $F_{\phi_i}$, there are infinitely such solutions where $C_{\phi_c}$ is not a valid copula.

Let $F_{\phi_i} : \mathbb{R} \to [0, 1]$ be an arbitrary strictly monotonic increasing function from the real line to the unit interval. The ground truth CDFs $F_i^\star$ are an example of such functions, but infinitely many alternatives exist, e.g., $F_{\phi_i}(x_i) = \text{sigmoid}(\alpha \cdot x_i)$, where $\alpha \in \mathbb{R}^{>0}$. Then, given any such parametrization of $F_{\phi_i}$ and assuming infinite expressivity for $C_{\phi_c}$, take

$$C_{\phi_c}(u_1, \ldots, u_n) = C^\star \left(F_1^\star(F_{\phi_1}^{-1}(u_1)), \ldots, F_n^\star(F_{\phi_n}^{-1}(u_n))\right),$$

Due to the invertibility of strictly monotonic functions, this yields a valid solution for Problem (4):

$$C_{\phi_c}\left(F_{\phi_1}(x_1) \ldots, F_{\phi_n}(x_n)\right) = C^\star \left(F_1^\star(x_1), \ldots, F_n^\star(x_n)\right) = P(x_1, \ldots, x_n).$$

Let us now proceed by contradiction to show that, in general, such a $C_{\phi_c}$ is not a valid copula. Assume that $C_{\phi_c}$ is a valid copula and, without loss of generality, that $F_{\phi_1} \neq F_1^\star$. Then, by definition, all univariate marginals of $C_{\phi_c}$ should be standard uniform distributions and thus,

$$C_{\phi_c}\left(1, \ldots, u_i, \ldots, 1\right) = u_i.$$

Therefore, we have that:

$$u_1 = C_{\phi_c}\left(u_1, 1, \ldots, 1\right) = C^\star \left(F_1^\star(F_{\phi_1}^{-1}(u_1)), 1, \ldots, 1\right) = F_1^\star(F_{\phi_1}^{-1}(u_1)),$$

since $C^\star$ is also a valid copula. However, we know that $F_{\phi_1} \neq F_1^\star$ and thus we arrive at a contradiction.

Hence, without further constraints on $F_{\phi_i}$ and $C_{\phi_c}$, such as the approaches presented in Secs. 3.1 and 3.2, one can obtain arbitrarily many solutions to Problem (4), where $C_{\phi_c}$ is not a valid copula.

□

## B.2 PROOF OF PROPOSITION 2

**Proposition 2**. (Validity) Assuming that all random variables $X_1, \ldots, X_d$ have continuous marginal distributions and assuming infinite expressivity for $\{F_{\phi_i}\}_{i=1}^d$ and $c_{\phi_c}$, solving Problem (7) yields a solution to Problem (4) where $c_{\phi_c}$ is a valid copula.

*Proof.* Let $p(x_1, \ldots, x_d)$ be the joint density of $\mathbf{X} = [X_1, \ldots, X_d]$. According to Sklar (1959), we know that the density can be written as:

$$p(x_1, \ldots, x_d) = c_{\phi_c^\star}\left(F_{\phi_1^\star}(x_1), \ldots, F_{\phi_d^\star}(x_d)\right) \times f_{\phi_1^\star}(x_1) \times \cdots \times f_{\phi_d^\star}(x_d), \tag{16}$$

where $F_{\phi_i^\star}$ and $f_{\phi_i^\star}$ are the ground truth marginal CDF and PDF of $X_i$, respectively, and $c_{\phi_c^\star}$ is the PDF of the ground truth copula. Further, since all $F_{\phi_i^\star}$ are continuous, we know that $c_{\phi_c^\star}$ is unique.

The proof proceeds in two parts: (i) showing that the marginals learned by solving Problem (8) will match the ground truth ($F_{\phi_i^\star}$, $f_{\phi_i^\star}$), and (ii) showing that, given those true marginals, solving Problem (7) will lead to a copula that matches $c_{\phi_c^\star}$, which is valid by definition.

**Part 1 (Marginals):** Let us recall the nature of Problem (8):

$$\underset{\phi_1,\ldots,\phi_d}{\arg\min} \; - \underset{\mathbf{x}\sim\mathbf{X}}{\mathbb{E}} \log \prod_{i=1}^{d} f_{\phi_i}(x_i).$$

Now, notice that this problem can be rewritten as $d$ independent optimization problems:

$$\underset{\phi_i}{\arg\min} \; - \underset{\mathbf{x}\sim\mathbf{X}}{\mathbb{E}} \log f_{\phi_i}(x_i),$$

which each consists of minimizing the expected marginal negative log-likelihood for one of the $d$ random variables. Given that the NLL is a strictly proper scoring rule (Gneiting & Raftery, 2007), we know that it will be minimized if and only if $f_{\phi_i} = f_{\phi_i^\star}$. Since we assume infinite expressivity for $f_{\phi_i}$, we know that such a solution can be learned and thus, solving Problem (8) will recover the ground truth marginals.

**Part 2 (Copula):** Let us now recall the nature of Problem (7), which assumes that the true marginal parameters $\phi_1^\star, \ldots, \phi_d^\star$ are known:

$$\underset{\phi_c}{\arg\min} \quad - \underset{\mathbf{x}\sim\mathbf{X}}{\mathbb{E}} \log c_{\phi_c}\left(F_{\phi_1^\star}(x_1), \ldots, F_{\phi_d^\star}(x_d)\right).$$

Let $\mathbf{U} = [U_1, \ldots, U_d]$ be the random vector obtained by applying the probability integral transform to the $X_i$, i.e., $U_i = F_{\phi_i^\star}(X_i)$. By construction, the marginal distribution of each $U_i$ will be uniform. Problem (7) can be rewritten as:

$$\underset{\phi_c}{\arg\min} \quad - \underset{\mathbf{u}\sim\mathbf{U}}{\mathbb{E}} \log c_{\phi_c}\left(u_1, \ldots, u_d\right),$$

which corresponds to minimizing the expected joint negative log-likelihood. Again, since the NLL is a strictly proper scoring rule, it will be minimized if and only if $c_{\phi_c} = c_{\phi_c^\star}$. Since we assume infinite expressivity for $c_{\phi_c}$, we know that such a solution can be learned and thus, solving Problem (7) will recover the ground truth copula, which is valid by definition.

Finally, since solving Problem (8) yields the true marginals, and solving Problem (7) based on the solution of Problem (8) recovers the true copula, we have that the combination of both solutions yields $p(x_1, \ldots, x_d)$ (see Eq. (16)) and thus they constitute a valid solution to Problem (4).

□

### B.3 Example of Valid and Invalid Decompositions

To illustrate Props. 1 and 2, we use the TACTiS decoder[2] to fit a two-dimensional hand-crafted distribution. The target distribution is built from a copula which is an equal mixture of two Clayton copulas, with parameters $\theta = 9.75$ and $\theta = -0.99$; and from marginals which are a Gamma distribution with parameter $\alpha = 1.99$ and a Double Weibull distribution with parameter $c = 3$.

As was previously shown in Fig. 3, Fig. 14 shows that the attentional copula and the flows can accurately reproduce the target distribution copula and marginals, when trained using the procedure outlined in Sec. 3.2. This is the expected result according to Prop. 2.

Fig. 15 shows the opposite situation: an attentional copula and flows that together accurately reproduce the target distribution, but individually are very far from the target copula and marginals. This is an example of an invalid copula as described in Prop. 1.

[2]We slightly modified the attentional copula architecture by removing the forced $U_{[0,1]}$ distribution from the first variable being sampled. This was to allow the attentional copula to be able to learn any distribution on the unit cube.

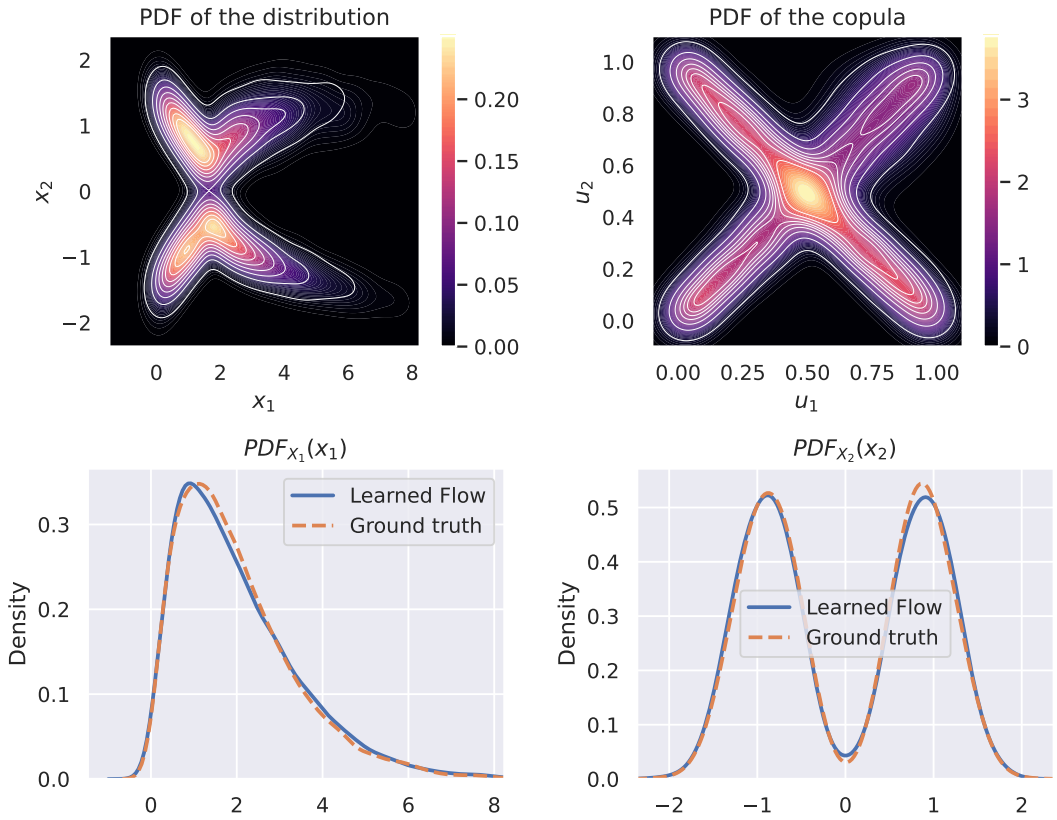



**Figure 14:** *Example of a two-dimensional distribution being accurately split into its constituent copula and marginals using an attentional copula and flows. (top left) Probability Densities of the target distribution (colors) and of the reconstructed distribution (white contours). (top right) Probability Density of the target distribution copula (colors) and of the attentional copula (white contours). (bottom) Probability Densities of each variable marginal distributions (dashed lines) and of the flows (solid lines).*

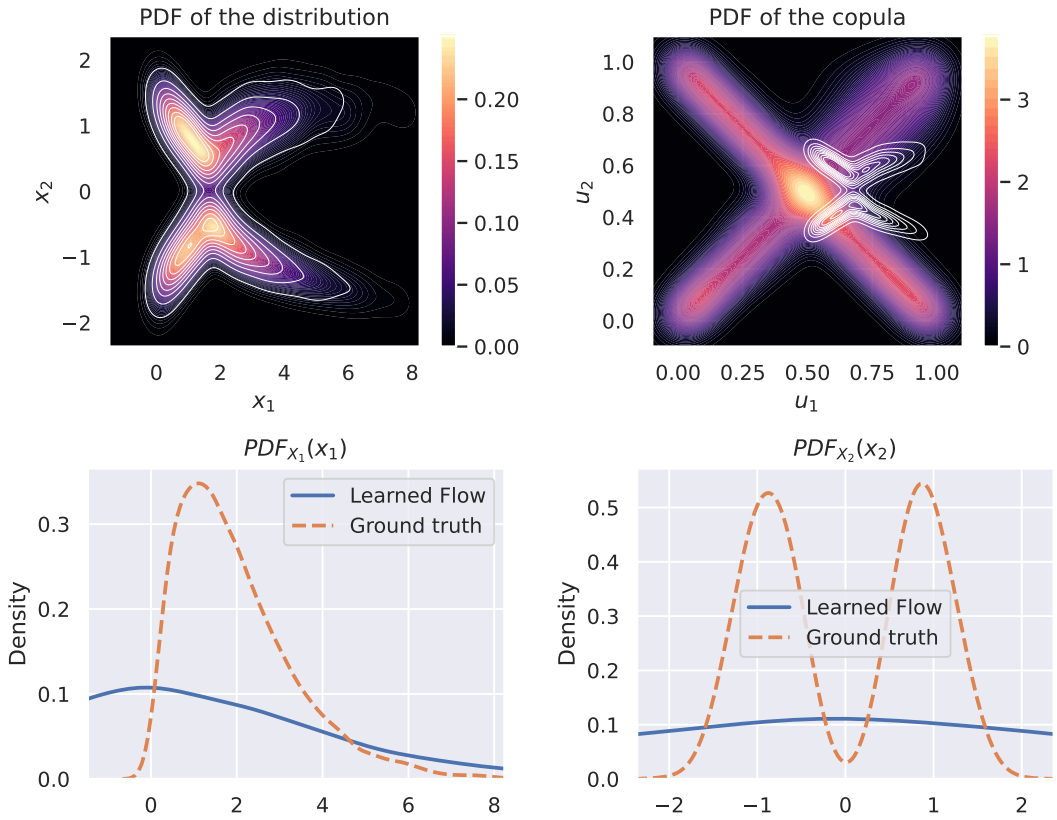



**Figure 15:** *Example of a two-dimensional distribution being inaccurately split into its constituent copula and marginals using an attentional copula and flows. (top left) Probability Densities of the target distribution (colors) and of the reconstructed distribution (white contours). (top right) Probability Density of the target distribution copula (colors) and of the attentional copula (white contours). (bottom) Probability Densities of each variable marginal distributions (dashed lines) and of the flows (solid lines).*

# C EXPERIMENTAL DETAILS

## C.1 DATASETS

**Table 6:** *Datasets used in the paper. Table reproduced from Drouin et al. (2022) with permission.*

| Short name | Monash name | Frequency | Number of series | Prediction length |
|---|---|---|---|---|
| `electricity` | Electricity Hourly Dataset | 1 hour | 321 | 24 |
| `fred-md` | FRED-MD Dataset | 1 month | 107 | 12 |
| `kdd-cup` | KDD Cup Dataset (without Missing Values) | 1 hour | 270 | 48 |
| `solar-10min` | Solar Dataset (10 Minutes Observations) | 10 minutes | 137 | 72 |
| `traffic` | Traffic Hourly Dataset | 1 hour | 862 | 24 |

We reuse the same datasets used by (Drouin et al., 2022) in their forecasting benchmark. Tab. 6 describes the datasets used in the paper for reference.

## C.2 TRAINING PROCEDURE

**Compute used** All models in the paper are trained in a Docker container with access to a Nvidia Tesla-P100 GPU (12 GB of memory), 2 CPU cores, and 32 GB of RAM. Due to sampling and NLL evaluation requiring more memory due to being done without bagging, they were done with computing resources with more available memory as needed.

**Batch size** The batch size was selected as the largest power of 2 between 1 and 256 such that the GPU memory requirement from the training loop did not go above the available memory.

**Training loop** Each epoch of the training loop consists of training the model using 512 batches from the training set, followed by computing the NLL on the validation set. We stop the training when any of these conditions are reached:

- We have reached 72 hours of training for a model without the two-stage curriculum,
- We have reached 36 hours of training for a single stage of the curriculum,
- Or we did not observe any improvement in the best value of the NLL on the validation set for 50 epochs.

We then return the version of the model that reached the best value of the NLL on the validation set.

For SPD, following Biloš (2023), in the absence of NLL, we use the loss function of the model i.e. the squared difference between the predicted and true noise.

**Validation set** During hyperparameter search, we reserve from the end of the training set a number of timesteps equal to 7 times the prediction length. The validation set is then built from all prediction windows that fit in this reserved data. During backtesting, we also remove this amount of data from the training set (except for `fred-md` where only remove a number equal to the prediction length), but the validation set is built from the 7 (or single for `fred-md`) non-overlapping prediction window we can get from this reserved data.

**Interpolation** For the interpolation experiments, we always use a centered interpolation window: we keep an equal amount of observed timesteps immediately before and immediately after the prediction window. Since we do not have access to the data that happens after the last backtesting prediction window, we instead shift all backtesting prediction windows by the length of the posterior observed data length. All parts of the training set which overlap these backtesting prediction windows are also removed, to prevent any information leakage during training.

## C.3 HYPERPARAMETER SEARCH PROTOCOL

The metric values for Auto-ARIMA, ETS, GPVar, TempFlow, and TimeGrad are taken from Drouin et al. (2022). We refer the reader to Drouin et al. (2022) for details on how the protocol used for hyperparameter

search for these models, but one important detail is that it used CRPS-Sum as the target to minimize. In theory, this should bias the comparison based on CRPS-Sum (Tab. 1) in their favor.

For TACTiS, and TACTiS-2, we use Optuna (Akiba et al., 2019) to minimize the NLL on the validation set. Note that we use the same validation set for the NLL evaluation and the early stopping during hyperparameter search. For each dataset and model, we let Optuna run for 6 days, using 50 training runs in parallel.

For SPD, following Biloš (2023), in the absence of NLL, we used the same protocol as for TACTiS and TACTiS-2 but we minimized the loss function of the model, the squared difference between the predicted and true noise.

For the interpolation experiments, we do not perform a separate hyperparameter search and instead reuse the hyperparameters found for forecasting.

### C.4 Selected Hyperparameters

In this section, we present the possible values for the hyperparameters during the hyperparameter search for TACTiS, TACTiS-2, and SPD, together with their optimal values as found by Optuna. Tabs. 7 to 9 present the hyperparameters which are restricted to finite sets of values for TACTiS, TACTiS-2, and SPD, respectively. Tab. 10 shows the optimal values for the learning rate parameters, which were allowed to be in the $[10^{-6}, 10^{-2}]$ continuous range.

**Table 7:** *Possible hyperparameters for* TACT*i*S. $^{\mathbf{e}}$, $^{\mathbf{f}}$, $^{\mathbf{k}}$, $^{\mathbf{s}}$, *and* $^{\mathbf{t}}$ *respectively indicate the optimal hyperparameters for* `electricity`, `fred-md`, `kdd-cup`, `solar-10min`, *and* `traffic`.

| | Hyperparameter | Possible values |
|---|---|---|
| Model | Encoder transformer embedding size (per head) and feed-forward network size | $4, 8, 16, 32^{\mathbf{et}}, 64^{\mathbf{fs}}, 128, 256^{\mathbf{k}}, 512$ |
| | Encoder transformer number of heads | $1^{\mathbf{k}}, 2^{\mathbf{f}}, 3^{\mathbf{s}}, 4, 5, 6^{\mathbf{t}}, 7^{\mathbf{e}}$ |
| | Encoder number of transformer layers pairs | $1^{\mathbf{k}}, 2, 3^{\mathbf{s}}, 4, 5^{\mathbf{eft}}, 6, 7$ |
| | Encoder input embedding dimensions | $1^{\mathbf{t}}, 2^{\mathbf{e}}, 3, 4^{\mathbf{s}}, 5, 6, 7^{\mathbf{fk}}$ |
| | Encoder time series embedding dimensions | $5^{\mathbf{t}}, 8, 16^{\mathbf{e}}, 32^{\mathbf{k}}, 48^{\mathbf{s}}, 64, 128, 256^{\mathbf{f}}, 512$ |
| | Decoder DSF number of layers | $1^{\mathbf{t}}, 2, 3^{\mathbf{f}}, 4, 5^{\mathbf{es}}, 6, 7^{\mathbf{k}}$ |
| | Decoder DSF hidden dimensions | $4, 8, 16^{\mathbf{fs}}, 32, 48^{\mathbf{e}}, 64, 128, 256, 512^{\mathbf{kt}}$ |
| | Decoder MLP number of layers | $1^{\mathbf{e}}, 2^{\mathbf{ft}}, 3, 4, 5, 6, 7^{\mathbf{ks}}$ |
| | Decoder MLP hidden dimensions | $4, 8^{\mathbf{k}}, 16, 32^{\mathbf{e}}, 48, 64^{\mathbf{f}}, 128^{\mathbf{t}}, 256, 512^{\mathbf{s}}$ |
| | Decoder transformer number of layers | $1^{\mathbf{k}}, 2^{\mathbf{f}}, 3^{\mathbf{e}}, 4, 5, 6^{\mathbf{t}}, 7^{\mathbf{e}}$ |
| | Decoder transformer embedding size (per head) | $4^{\mathbf{s}}, 8, 16, 32, 48^{\mathbf{f}}, 64^{\mathbf{ek}}, 128^{\mathbf{t}}, 256, 512$ |
| | Decoder number transformer heads | $1^{\mathbf{e}}, 2, 3, 4^{\mathbf{ks}}, 5, 6^{\mathbf{ft}}, 7$ |
| | Decoder number of bins in conditional distribution | $10, 20^{\mathbf{ft}}, 50^{\mathbf{s}}, 100^{\mathbf{e}}, 200^{\mathbf{k}}, 500$ |
| Data | Normalization | Standardization$^{\mathbf{efkst}}$ |
| | History length to prediction length ratio | $1, 2^{\mathbf{ks}}, 3^{\mathbf{eft}}$ |
| Training | Optimizer | Adam$^{\mathbf{efkst}}$ |
| | Weight decay | $0^{\mathbf{efks}}, (10^{-5}), (10^{-4}), (10^{-3})^{\mathbf{t}}$ |
| | Gradient clipping | $(10^{3})^{\mathbf{efkst}}, (10^{4})$ |

**Table 8:** *Possible hyperparameters for* TACTiS-2. $^{e}$, $^{f}$, $^{k}$, $^{s}$, *and* $^{t}$ *respectively indicate the optimal hyperparameters for* `electricity`, `fred-md`, `kdd-cup`, `solar-10min`, *and* `traffic`.

| | Hyperparameter | Possible values |
|---|---|---|
| Model | Marginal CDF Encoder transformer embedding size (per head) and feed-forward network size | $4, 8^{\mathbf{st}}, 16, 32^{\mathbf{k}}, 64, 128^{\mathbf{e}}, 256, 512^{\mathbf{f}}$ |
| | Marginal CDF Encoder transformer number of heads | $1, 2, 3^{\mathbf{e}}, 4^{\mathbf{ft}}, 5^{\mathbf{s}}, 6^{\mathbf{k}}, 7$ |
| | Marginal CDF Encoder number of transformer layers pairs | $1^{\mathbf{f}}, 2^{\mathbf{s}}, 3^{\mathbf{e}}, 4^{\mathbf{kt}}, 5, 6, 7$ |
| | Marginal CDF Encoder input encoder layers | $1, 2, 3, 4^{\mathbf{fk}}, 5, 6^{\mathbf{s}}, 7^{\mathbf{et}}$ |
| | Marginal CDF Encoder time series embedding dimensions | $5^{\mathbf{fks}}, 8^{\mathbf{e}}, 16^{\mathbf{t}}, 32, 48, 64, 128, 256, 512$ |
| | Attentional Copula Encoder transformer embedding size (per head) and feed-forward network size | $4^{\mathbf{k}}, 8^{\mathbf{fst}}, 16^{\mathbf{e}}, 32, 64, 128, 256, 512$ |
| | Attentional Copula Encoder transformer number of heads | $1, 2^{\mathbf{e}}, 3^{\mathbf{f}}, 4^{\mathbf{t}}, 5^{\mathbf{s}}, 6^{\mathbf{k}}, 7$ |
| | Attentional Copula Encoder number of transformer layers pairs | $1, 2^{\mathbf{s}}, 3^{\mathbf{k}}, 4^{\mathbf{t}}, 5^{\mathbf{ef}}, 6, 7$ |
| | Attentional Copula Encoder input encoder layers | $1^{\mathbf{s}}, 2^{\mathbf{k}}, 3, 4, 5^{\mathbf{t}}, 6, 7^{\mathbf{ef}}$ |
| | Attentional Copula Encoder time series embedding dimensions | $5, 8^{\mathbf{fk}}, 16, 32, 48^{\mathbf{s}}, 64^{\mathbf{t}}, 128, 256^{\mathbf{e}}, 512$ |
| | Decoder DSF number of layers | $1, 2^{\mathbf{s}}, 3, 4^{\mathbf{kt}}, 5, 6^{\mathbf{f}}, 7^{\mathbf{e}}$ |
| | Decoder DSF hidden dimensions | $4, 8, 16^{\mathbf{t}}, 32, 48^{\mathbf{ef}}, 64, 128^{\mathbf{k}}, 256^{\mathbf{s}}, 512$ |
| | Decoder MLP number of layers | $1, 2^{\mathbf{ef}}, 3^{\mathbf{ks}}, 4, 5, 6^{\mathbf{t}}, 7$ |
| | Decoder MLP hidden dimensions | $4, 8, 16^{\mathbf{kst}}, 32, 48^{\mathbf{f}}, 64, 128^{\mathbf{e}}, 256, 512$ |
| | Decoder transformer number of layers | $1, 2^{\mathbf{t}}, 3^{\mathbf{s}}, 4^{\mathbf{k}}, 5, 6, 7^{\mathbf{ef}}$ |
| | Decoder transformer embedding size (per head) | $4, 8, 16^{\mathbf{s}}, 32^{\mathbf{e}}, 48^{\mathbf{kt}}, 64^{\mathbf{f}}, 128, 256, 512$ |
| | Decoder transformer number of heads | $1^{\mathbf{k}}, 2, 3, 4^{\mathbf{tf}}, 5, 6^{\mathbf{s}}, 7^{\mathbf{e}}$ |
| | Decoder number of bins in conditional distribution | $10, 20^{\mathbf{e}}, 50^{\mathbf{fks}}, 100^{\mathbf{t}}, 200, 500$ |
| Data | Normalization | Standardization$^{\mathbf{efkst}}$ |
| | History length to prediction length ratio | $1^{\mathbf{t}}, 2^{\mathbf{es}}, 3^{\mathbf{fk}}$ |
| Training Phase 1 | Optimizer | Adam$^{\mathbf{efkst}}$ |
| | Weight decay | $0^{\mathbf{est}}, (10^{-5})^{\mathbf{k}}, (10^{-4}), (10^{-3})^{\mathbf{f}}$ |
| | Gradient clipping | $(10^{3})^{\mathbf{efst}}, (10^{4})^{\mathbf{k}}$ |
| Training Phase 2 | Optimizer | Adam$^{\mathbf{efkst}}$ |
| | Weight decay | $0^{\mathbf{skt}}, (10^{-5}), (10^{-4})^{\mathbf{ef}}, (10^{-3})$ |
| | Gradient clipping | $(10^{3})^{\mathbf{es}}, (10^{4})^{\mathbf{fkt}}$ |

**Table 9:** *Possible hyperparameters for SPD.* $^{e}$, $^{f}$, $^{k}$, $^{s}$, *and* $^{t}$ *respectively indicate the optimal hyperparameters for* `electricity`, `fred-md`, `kdd-cup`, `solar-10min`, *and* `traffic`. *The choice of possible hyperparameters was discussed with the authors (Biloš, 2023).*

| | Hyperparameter | Possible values |
|---|---|---|
| Model | `type` | `'discrete'`$^{\mathbf{efkst}}$, `'continuous'` |
| | `noise` | `'normal'`$^{\mathbf{efks}}$, `'ou'`, `'gp'`$^{\mathbf{t}}$ |
| | `num_layers` | $1, 2^{\mathbf{efs}}, 3^{\mathbf{kt}}$ |
| | `num_cells` | $20^{\mathbf{fkt}}, 40^{\mathbf{s}}, 60^{\mathbf{e}}$ |
| | `dropout_rate` | $0^{\mathbf{k}}, 0.001^{\mathbf{tf}}, 0.01^{\mathbf{es}}, 0.1$ |
| | `diff_steps` | $25, 50, 100^{\mathbf{efkst}}$ |
| | `beta_schedule` | `'linear'`, `'quad'`$^{\mathbf{efkst}}$ |
| | `residual_layers` | $4, 8^{\mathbf{e}}, 16^{\mathbf{fkst}}$ |
| | `residual_channels` | $4, 8, 16^{\mathbf{efkst}}$ |
| | `scaling` | `False`$^{\mathbf{ks}}$, `True`$^{\mathbf{eft}}$ |
| Data | History length to prediction length ratio | $1^{\mathbf{efkst}}$ |
| Training | Optimizer | Adam$^{\mathbf{efkst}}$ |
| | Weight decay | $0^{\mathbf{s}}, (10^{-5})^{\mathbf{ek}}, (10^{-4})^{\mathbf{ft}}, (10^{-3})$ |
| | Gradient clipping | $(10^{1}), (10^{2}), (10^{3})^{\mathbf{k}}, (10^{4})^{\mathbf{efst}}$ |

**Table 10:** *Optimal learning rate as obtained by our hyperparameter search procedure.*

| **Model** | | electricity | fred-md | kdd-cup | solar-10min | traffic |
|---|---|---|---|---|---|---|
| TACTiS | | $7.4 \times 10^{-5}$ | $2.7 \times 10^{-4}$ | $9.2 \times 10^{-5}$ | $8.0 \times 10^{-5}$ | $2.8 \times 10^{-3}$ |
| TACTiS-2 | Phase 1 | $5.0 \times 10^{-5}$ | $1.7 \times 10^{-3}$ | $2.3 \times 10^{-5}$ | $1.8 \times 10^{-3}$ | $2.5 \times 10^{-3}$ |
| TACTiS-2 | Phase 2 | $2.2 \times 10^{-4}$ | $1.9 \times 10^{-3}$ | $9.4 \times 10^{-4}$ | $7.0 \times 10^{-4}$ | $6.7 \times 10^{-4}$ |
| SPD | | $8.9 \times 10^{-4}$ | $5.6 \times 10^{-3}$ | $2.7 \times 10^{-3}$ | $3.1 \times 10^{-3}$ | $4.4 \times 10^{-3}$ |

**Table 11:** *Mean NLL values for the forecasting experiments on uneven data. Results are over 5 seeds. Lower is better and the best results are in bold.*

| **Model** | kdd-cup | solar-10min |
|---|---|---|
| TACTiS | $2.858 \pm 0.476$ | $1.001 \pm 0.783$ |
| TACTiS-2 | $\mathbf{1.866 \pm 0.230}$ | $\mathbf{0.261 \pm 0.011}$ |

**Table 12:** *Mean NLL values for the forecasting experiments on unaligned data. Results are over 5 seeds. Lower is better and the best results are in bold.*

| **Model** | kdd-cup | solar-10min |
|---|---|---|
| TACTiS | $1.527 \pm 0.072$ | $-0.038 \pm 0.220$ |
| TACTiS-2 | $\mathbf{0.722 \pm 0.163}$ | $\mathbf{-3.218 \pm 0.249}$ |

# D ADDITIONAL RESULTS

## D.1 DEMONSTRATION OF MODEL FLEXIBILITY IN REAL-WORLD DATASETS

We present empirical results on real-world datasets verifying the ability of TACTiS-2 to work with uneven and unaligned data. The datasets used in these experiments are derived from real-world datasets with aligned and evenly sampled observations. To create uneven sampling, for each series we independently follow a process where if we keep the observation at timestep $t$, then the next kept observation will be at timestep $t + \delta$, with $\delta$ randomly chosen from 1, 2, or 3. To create unalignment, we randomly choose the sampling frequency for each series from the original frequency, half the original frequency and a quarter of the original frequency in the dataset. The datasets resulting from such corruptions are faithful to real-world scenarios. We follow the same protocol as described in Sec. 5, except that we use a single backtest timestamp for evaluation. Tab. 11 and Tab. 12 compare the ability of TACTiS and TACTiS-2 to perform forecasting with uneven and unaligned data respectively on the two datasets with the largest forecast horizon. It can be seen that the ability of TACTiS-2 to work with uneven and unaligned data is much more pronounced than that of TACTiS.