Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

PERCEIVING COPULAS FOR MULTIMODAL TIME SERIES FORECASTING

Cat P. Le, Chris Cannella, Ali Hasan, Yuting Ng, and Vahid Tarokh

Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA

ABSTRACT

Transformers have demonstrated remarkable efficacy in forecasting time series. However, their dependence on self-attention mechanisms demands significant computational resources, thereby limiting their applicability across diverse tasks. Here, we propose the perceiver-CDF for modeling cumulative distribution functions (CDF) of time series. Our model combines the perceiver architecture with copula-based attention for multimodal time series prediction. By leveraging the perceiver, our model transforms multimodal data into a compact latent space, thereby significantly reducing computational demands. We implement copula-based attention to construct the joint distribution of missing data for future prediction. To mitigate error propagation and enhance efficiency, we introduce output variance testing and midpoint inference for the local attention mechanism. This enables the model to efficiently capture dependencies within nearby imputed samples without considering all previous samples. The experiments on the various benchmarks demonstrate a consistent improvement over other methods while utilizing only half of the resources.

1 INTRODUCTION

Time-series prediction remains an enduring challenge since it requires effectively capturing global patterns (e.g., annual trends) and localized details (e.g., abrupt disruptions). This challenge becomes particularly pronounced when dealing with non-synchronized, incomplete, high-dimensional, or multimodal input data. Consider a time series consisting of N regularly-sampled and synchronously-measured values, where measurements are taken at the same time at intervals of length T. If the time-step is unobserved at rate r, then there are $(1-r)NT$ observed values that are relevant for inference. Consider an asynchronously-measured time series, where input variables are observed at different times, resulting in each time-step having only $1/N$ of its variables observed. In this scenario, only $(1-r)T$ values remain relevant for inference within the time series. Consequently, employing a synchronous model to address non-synchronized time series results in a missingness rate of $(N-1)/N$. This missingness rate grows rapidly as the number of variables increases, reaching 95% with just 20 variables in the time series. When designing an architecture to handle missing data, it is crucial to utilize techniques for approximating missing values while ensuring the computational overhead does not exceed the effort required to extract valuable insights from the observed data. To this end, a transformer model (Drouin et al. 2022) with attention-based mechanism (Vaswani et al. 2017) is tailored for time series. This model tokenizes input variables and utilizes a transformer-based encoding and decoding approach, making it a suitable choice for modeling non-synchronized time series data. Tokenization also offers significant advantages for missing data, as unobserved data can be seamlessly excluded from the token stream. Additionally, this model utilizes a copula-based structure (Nelsen 2006) to represent the sequence distribution to further enhance the prediction performance. Particularly, it learns the joint distribution with a non-parametric copula, which is a product of conditional probabilities. To ensure that the product results in a valid copula, it considers permutations of the margins during training such that a level of permutation invariance occurs. However, this process yields an exchangeable class of copulas in the limit of infinite permutations, diminishing the utility of the non-parametric copula. Furthermore, the transformer architecture poses significant computational demands related to the self-attention mechanism.

In this paper, we introduce the perceiver-CDF model, which utilizes the perceiver architecture (Jaegle et al. 2021) and the attention-based copulas (Nelsen 2006), to enhance multimodal time series modeling and address computational efficiency challenges. Particularly, our model consists of the perceiver-based encoder and the copula-based decoder, enabling the incorporation of a more general class of copulas that are not exchangeable. The class of copulas used in our approach are the *factor copulas*, which are conditionally exchangeable based on the factor. Initially, the perceiver-CDF model transforms the input variables into temporal embeddings through a combination of input embedding and positional encoding procedures. In this phase, the observed and the missing data points are encoded, with the value of missing data points masked. Subsequently, our proposed model utilizes a *latent* attention-based mechanism (Jaegle et al. 2021) from the perceiver to efficiently map the input embeddings to a lower-dimensional latent space. Since all subsequent computations are performed within this compact latent space, it helps reduce the complexity from a quadratic to a sub-quadratic level. Lastly, the copula-based decoder formulates the joint distribution of missing data using latent embeddings. This distribution undergoes a sampling process to yield the predicted outcomes. Our model can effectively handle synchronized, non-synchronized, and multimodal data, expanding its applicability to diverse domains. Next, we conduct extensive experiments on the unimodal and multimodal time series datasets. We also conduct memory consumption scaling experiments using random walk data to demonstrate the memory efficiency of our approach. The results demonstrate the competitive performance of our model compared to the state-of-the-art methods, including TACTiS (Drouin et al. 2022), GPVar (Salinas et al. 2019), SSAE-LSTM (Zhu et al. 2021), and AR (Kalliovirta et al. 2015) while utilizing as little as one-tenth of the computational resources.

2 RELATED WORKS

Neural networks for time series forecasting have undergone extensive research and delivered impressive results when compared to classical statistical methods (Box et al. 2015). Notably, both convolutional (Chen et al. 2020) and recurrent neural networks (Shih et al. 2019) have demonstrated the power of deep neural networks in learning historical patterns and leveraging this knowledge for precise predictions of future data points. Subsequently, various deep learning techniques have been proposed to address the modeling of regularly-sampled time series data (Lim and Zohren 2021; Benidis et al. 2022). Most recently, the transformer architecture, initially designed for sequence modeling tasks, has been adopted extensively for time series forecasting (Lim et al. 2021; Müller et al. 2021). Using the properties of the attention mechanism, these models excel at capturing long-term dependencies within the data, achieving remarkable results. In addition to these developments, score-based diffusion models (Tashiro et al. 2021) achieved competitive performance in forecasting tasks. However, it is worth noting that the majority of these approaches are tailored for handling regularly sampled and synchronized time series data. Consequently, they may not be optimal when applied to non-synchronized datasets. In financial forecasting, the copula emerges as a formidable tool for estimating multivariate distributions (Krupskii and Joe 2020; Größer and Okhrin 2022; Mayer and Wied 2023). Its computational efficiency has led to its use in the domain adaptation contexts (Lopez-Paz et al. 2012). Moreover, the copula structure has found utility in time series prediction when coupled with neural architectures like LSTMs (Lopez-Paz et al. 2012) and the transformer (Drouin et al. 2022), enabling the modeling of irregularly sampled time series data. While previous research has explored non-synchronized methods (Shukla and Marlin 2021), their practicality often falters due to computational challenges. For instance, the transformer architecture with copulas (Drouin et al. 2022; Ashok et al. 2024) is proposed and applicable to both synchronized and non-synchronized datasets. Nonetheless, the inherent computational overhead associated with the self-attention mechanism poses limitations, particularly when applied to high-dimensional inputs such as multimodal data. To mitigate the computational complexity, we utilize the perceiver (Jaegle et al. 2021) as the encoder, with a copula-based decoder. We also use the midpoint inference (Liu et al. 2019) for the local inference during the decoding phase. This approach restricts conditioning and effectively embodies a form of sparse attention (Child et al. 2019; Tay et al. 2020; Roy et al. 2021), although the sparsity pattern is determined through a gap-filling process.

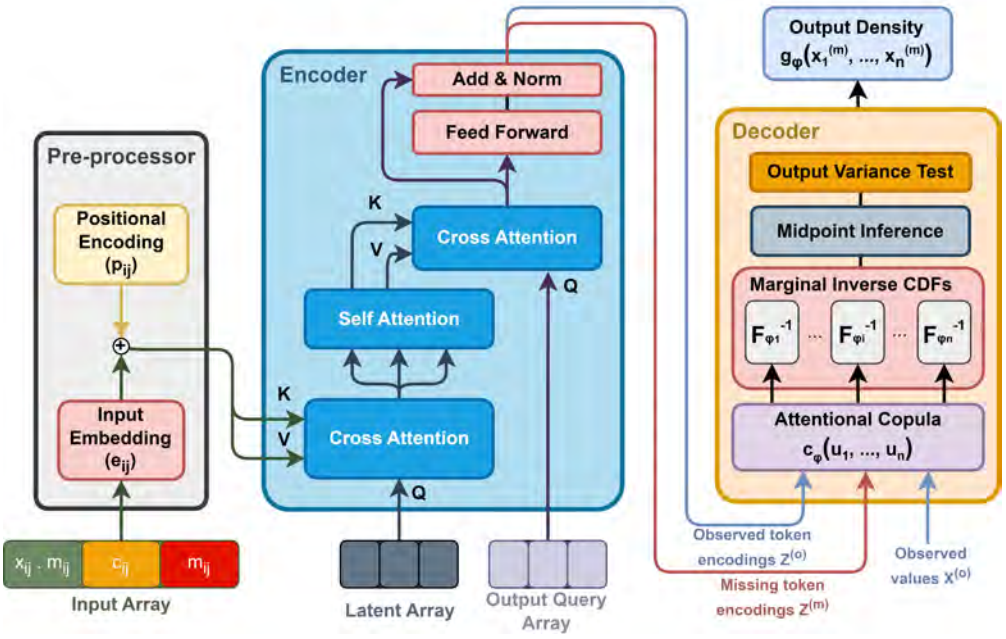


Figure 1: The overview architecture of perceiver-CDF model. The pre-processor includes input embedding, and positional encoding layers to capture temporal dependencies in the input data. The encoder uses the cross-attention mechanisms to map the embedding to a lower-dimensional latent space. The decoder constructs the joint distribution of missing data using the copula-based structure.

3 PROPOSED METHOD

We introduce a new perceiver-CDF model designed for multimodal time series prediction. The model comprises three key components: the pre-processor, the encoder, and the decoder. The overview architecture of our proposed model is illustrated in Figure 1. Drawing inspiration from transformers and other transformer-based models (Drouin et al. 2022), the pre-processor utilizes the self-attention mechanism to encode input time series variables, effectively transforming them into a sequence of generalized tokens. This transformation enables the model to process and analyze the temporal aspects of the data efficiently. Following this, the encoder, based on the perceiver model, applies cross-attention to the resulting token sequence to generate a conditional distribution of inferred variables using a parameterized copula. Particularly, it converts the complex input tokens into a compact latent space. This step is crucial for computational efficiency, as subsequent computations operate within this lower-dimensional space. Subsequently, the copula-based decoder is utilized to learn the joint distribution of missing data and observed data, facilitating future predictions. During the training process, we implement midpoint inference in the decoder for local inference instead of random imputation, contributing to further computational reduction. This mechanism also aids in establishing dependencies between nearby imputed samples. To validate and refine predictions, we introduce a variance testing mechanism for output prediction. If the prediction results exhibit instability and the variance exceeds a predefined threshold, the imputation is deemed unreliable for future predictions. It will be masked to prevent error propagation. The proposed model utilizes the advantages of both the self-attention mechanism and latent-variable-based attention mechanisms from perceivers. Notably, it enables the modeling of dependencies between covariates.

3.1 Perceiver-based Encoder

Let $\mathcal{X}$ denote the a time series of interest, with $\mathcal{X} = \{X_1, X_2, \ldots, X_i, \ldots\}$. Each element X_i is defined as a quadruple: $X_i = (v_i, c_i, t_i, m_i)$, where v_i is the value, c_i is an index identifying the variable, t_i is a time

stamp, and m_i is a mask indicating whether the data point is observed or needs to be inferred (i.e., missing data). For synchronously measured time series, we can organize it into a data matrix denoted as $X_{c,t}$. This matrix has rows corresponding to individual variables and columns corresponding to different timestamps when measurements were recorded. First, the pre-processor generates embeddings for each data point, $\vec{x}_i$, which includes the value v_i, a learned encoding for the variable c_i, an additive sinusoidal positional encoding indicating the position of t_i within the overall time series, and the mask m_i. Subsequently, these embeddings are passed through the perceiver-based encoder. Here, the encoder leverages a predefined set of learned latent vectors $\vec{u}_L$ for the cross-attention mechanism, denoted by $\mathcal{A}_C(K, Q, V)$, where K is a set of keys, Q is a query, and V is a set of values. Through the utilization of learned key and value-generating functions, $K_{\texttt{latent}}(\,)$ and $V_{\texttt{latent}}(\,)$, the encoder derives latent vectors $\vec{w}_L$, which effectively encapsulate the temporal information through cross-attention with the set of observed vectors $\vec{X}_O$ as follows:

$$\vec{w}_L = \mathcal{A}_C\left(K_{\texttt{latent}}(\vec{X}_O), \vec{u}_L, V_{\texttt{latent}}(\vec{X}_O)\right) \tag{1}$$

Following additional self-attention-based processing on the latent vectors, the perceiver-based encoder proceeds to employ cross-attention with the latent vector set $\vec{W}$, to generate tokens for each data point. This operation involves using the learned key-generating function $K_{\texttt{encode}}(\,)$, the query-generating function $Q_{\texttt{encode}}(\,)$, and the value-generating functions $V_{\texttt{encode}}(\,)$, to derive token vectors $\vec{z}_i$ as follows:

$$\vec{z}_i = \mathcal{A}_C\left(K_{\texttt{encode}}(\vec{W}), Q_{\texttt{encode}}(\vec{x}_i), V_{\texttt{encode}}(\vec{W})\right) \tag{2}$$

Aligned with the perceiver architecture, the number of latent features L is intentionally maintained at a considerably smaller scale compared to the total number of data points N. This strategic choice serves to manage computational complexity, which scales at $\mathcal{O}(NL)$. The initial cross-attention step in our model assumes a pivotal role by encoding a comprehensive global summary of the observed data from the time series into a set of concise latent vectors. These latent vectors effectively capture the essential information embedded within the entire dataset. Subsequently, our proposed perceiver-CDF model generates tokens for each individual data point by efficiently querying relevant global information from the previously obtained latent summary in the second cross-attention step. This process ensures that each token encompasses vital contextual details drawn from the overall dataset, as necessitated.

3.2 Copula-based Decoder

Next, the decoder is specifically designed to learn the joint distribution of the missing data points conditioned on the observed ones. To achieve this, the attention-based decoder is trained to mimic a non-parametric copula (Nelsen 2006). Let $x^{(m)}$ and $x^{(o)}$ represent the missing and observed data points, respectively. Let F_i be the i^{th} marginal cumulative distribution function (CDF) and f_i be the marginal probability density function (PDF). The copulas, under Sklar's theorem (Sklar 1959), allow for separate modeling of the joint distribution and the marginals, which has particular relevance to the case of sequence modeling. Similar to TACTiS (Drouin et al. 2022), we employ a normalizing flow technique known as Deep Sigmoidal Flow (Huang et al. 2018) to model the marginal CDF. The marginal PDF is obtained by differentiating the marginal CDF. The copula-based structure g_ϕ is described as follows:

$$g_\phi\left(X^{(m)}\right) = c_{\phi_c}\left(F_{\phi_1}\left(x_1^{(m)}\right), \ldots, F_{\phi_{n_m}}\left(x_{n_m}^{(m)}\right)\right) \times f_{\phi_1}\left(x_1^{(m)}\right) \times \ldots \times f_{\phi_{n_m}}\left(x_{n_m}^{(m)}\right), \tag{3}$$

where $X^{(m)} = \{x_1^{(m)}, \ldots, x_{n_m}^{(m)}\}$, c_{ϕ_c} is the density of a copula, and $c_{\phi_c}(F_{\phi_1}(x_1^{(m)}), \ldots, F_{\phi_{n_m}}(x_{n_m}^{(m)})) = c_{\phi_{c1}}(F_{\phi_1}(x_1^{(m)})) \times c_{\phi_{c2}}(F_{\phi_2}(x_2^{(m)}) | F_{\phi_1}(x_1^{(m)})) \times \ldots \times c_{\phi_{cn_m}}(F_{\phi_{n_m}}(x_{n_m}^{(m)}) | F_{\phi_1}(x_1^{(m)}), \ldots, F_{\phi_{n_m-1}}(x_{n_m-1}^{(m)}))$. During the decoding phase, our model selects a permutation, denoted as γ, from all data points, ensuring that observed data points come before those awaiting inference. The decoder then utilizes the self-attention

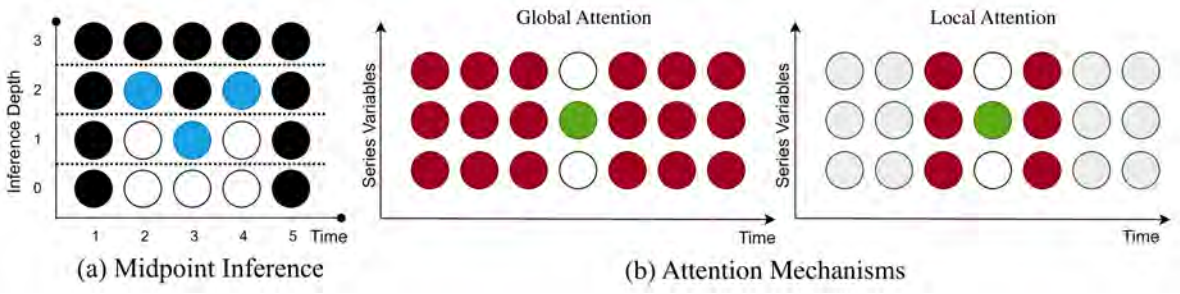


Figure 2: (a) Visualization of the midpoint inference mechanism: blue-filled points represent the points earmarked for inference at a particular depth, while black points represent those already observed or inferred at that depth and the white points are unobserved. (b) Comparison between the global attention mechanism and the local attention mechanism, which utilizes a local window containing only the nearest tokens: green-filled points indicate the currently sampled variable, while red points signify the variables to which the sampled token directs its attention during the sampling process.

mechanism $\mathcal{A}_S(K, Q, V)$ with the learned key and value functions, $K_{\texttt{decode}}(\ )$ and $V_{\texttt{decode}}(\ )$, to derive distributional parameters, $\theta_{\gamma(i)}$, for each data point awaiting inference as follows:

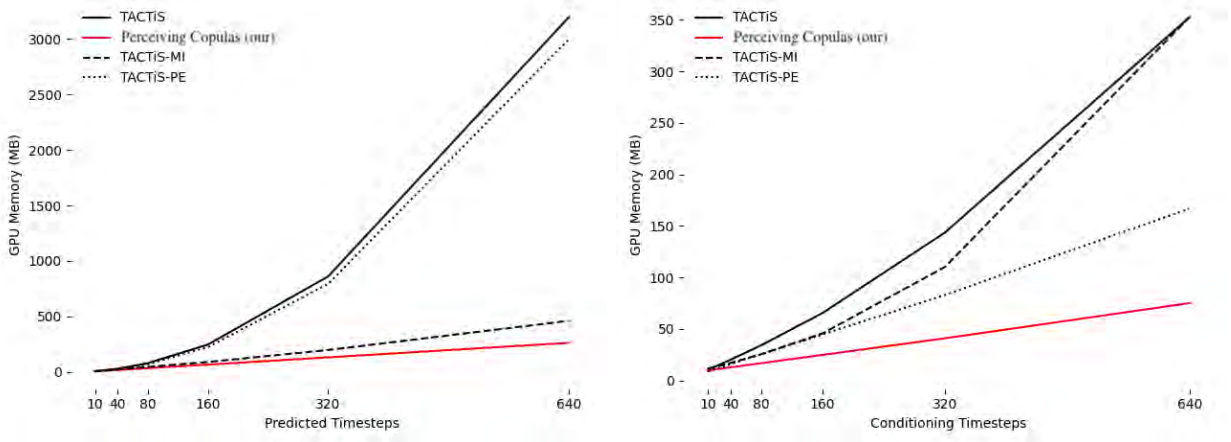
$$\theta_{\gamma(i)} = \mathcal{A}_S\Big(K_{\texttt{decode}}\left(\vec{z}_{\gamma(j)}\right), \vec{z}_{\gamma(i)}, V_{\texttt{decode}}\left(\vec{z}_{\gamma(j)}\right)\Big) \tag{4}$$

where $\gamma(j) < \gamma(i)$. Next, we use a parameterized diffeomorphism $f_{\phi,c} : (0,1) \mapsto \mathbb{R}$. When θ represents the parameters for a distribution p_θ over the interval $(0,1)$, our model proceeds by either sampling data points as $\hat{x}_i = f_{\phi,c_i}(u_i), \quad u_i \sim p_{\theta_i}$, or computing the conditional likelihood: $p_{\theta_i}(f^{-1}_{\phi,c_i}(x_i))$. Additionally, the decoder's complexity scales as $\mathcal{O}(S(S+H))$, where S represents the number of data points to be inferred and H denotes the number of observed data points.

3.3 Midpoint Inference for Local Attention

To enhance computational efficiency while maintaining the prediction performance, we propose the midpoint inference mechanism with temporally local self-attention to effectively reduce computational overhead. Instead of relying on random permutations to establish the conditioning structure, our method employs permutations that recursively infer midpoints within gaps in the observed data. When dealing with a continuous sequence of missing data points for the same variable, we determine the depth of each data point based on the number of midpoint inferences required within that sequence before considering the data point itself as a midpoint. Notably, observed data points are assigned shallower depths compared to data points that are yet to be observed. Consequently, we sample a permutation γ that positions data points with shallower assigned depths before those with deeper depths. Here, we determine midpoints by considering the number of data points between the prior observation and the next observation, as visually depicted in Figure 2. This method is well-suited for regularly or nearly-regularly sampled time series data. For each data token $\vec{z}_i$, our approach selects a set of conditioning tokens $\vec{H}_i$. These conditioning tokens comprise both past and future windows, consisting of the k closest tokens for each variable in the series that precede $\vec{z}i$ within the generated permutation γ. Figure 2(b) illustrates the proposed local-attention conditioning mechanism in comparison with the global self-attention. Here, our model employs learned key and value-generation functions, $K_{\texttt{decode}}(\ )$ and $V_{\texttt{decode}}(\ )$, to derive distributional parameters $\theta\gamma(i)$ for each data point to be inferred, following the ordering imposed by γ as follows:

$$\theta_i = \mathcal{A}_S\left(K_{\texttt{decode}}(\vec{H}_i), \vec{z}_i, V_{\texttt{decode}}(\vec{H}_i)\right)$$



(a) Varying Prediction Length

(b) Varying Conditioning Length

Figure 3: Comparison of memory consumption of perceiver-CDF model (our), TACTiS model, TACTiS model with perceiver-based encoder (TACTiS-PE), and TACTiS model with midpoint imputation (TACTiS-MI) on a synthetic dataset with (a) varying prediction length and (b) varying conditioning length.

3.4 Output Variance Testing

By incorporating midpoint inference and local attention mechanisms, the decoder adeptly captures dependencies among neighboring imputed samples. However, it is crucial to acknowledge that this enhancement introduces a susceptibility to errors, potentially hindering the training process. To address this concern and prevent error propagation, we propose an output variance testing mechanism for each imputed data point. In this mechanism, for every imputation, we conduct a series of forecasts by sampling from the derived joint distribution of the missing data. Subsequently, we calculate the output variance of the generated outcomes and compare it with a threshold set to align with the input data variance. If the output variance surpasses four times the threshold, we identify the predicted sample for exclusion in future imputations. In other words, this predicted data point is masked to insulate it from influencing subsequent imputation processes. With a fixed window size, the decoder's complexity can be characterized as $\mathcal{O}(nN)$, where n represents the number of time series variables, and N is the total number of data points in the time series.

4 EXPERIMENTAL STUDY

We present comprehensive experiments to showcase the computational efficiency of our proposed perceiver-CDF model. First, we conduct memory consumption scaling experiments using synthetic random walk data to demonstrate the memory efficiency of our proposed model. Next, we evaluate the predictive capabilities of our model, comparing it against the state-of-the-art approaches, such as deep autoregressive AR (Kalliovirta et al. 2015), GPVar (Salinas et al. 2019), SSAE-LSTM (Zhu et al. 2021), and TACTiS (Drouin et al. 2022). Our evaluation spans across three unimodal time series datasets from the Monash Time Series Forecasting Repository (Godahewa et al. 2021), including `electricity`, `traffic`, and `fred-md`, for short-term and long-term prediction tasks. Moreover, we evaluate the multi-modality capabilities of our perceiver-based model in three multimodal time series datasets from the UCI Machine Learning Repository (Dua and Graff 2017), including `room occupation` (Candanedo 2016), `interstate traffic` (Hogue 2019), and `air quality` (Chen 2019) datasets. The model parameters employed for these experiments were adopted from the configuration used by TACTiS (Drouin, Marcotte, and Chapados 2022). We adopt these parameters as the foundation for establishing the perceiver-CDF model. Below, in Table 1, we provide a comprehensive listing of the model parameters utilized for our perceiver-CDF and TACTiS models.

Table 1: The architectures and parameters of perceiver-CDF and TACTiS models.

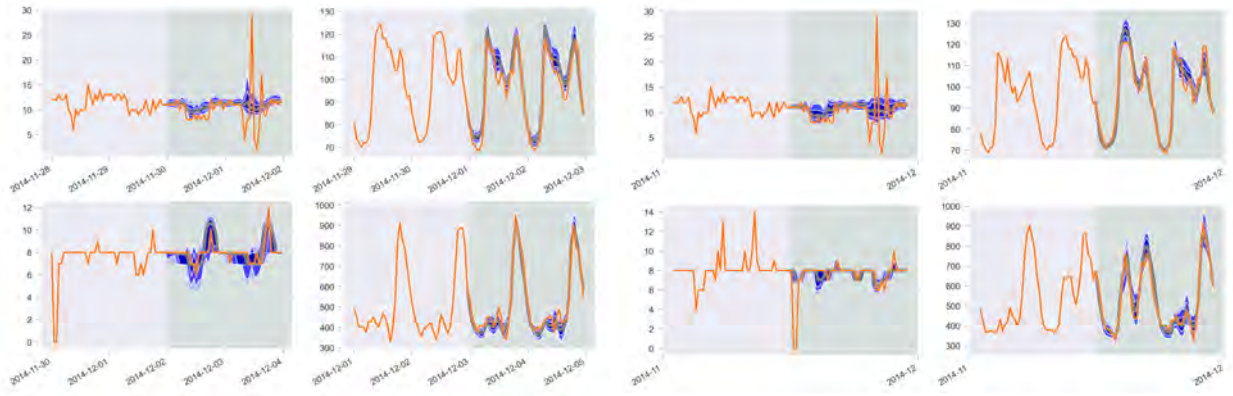
(A) PERCEIVER-CDF MODEL	
INPUT ENCODING	
SERIES EMBEDDING DIM.	5
INPUT ENCODER LAYERS	3
POSITIONAL ENCODING	
DROPOUT	0.01
PERCEIVER ENCODER	
NUM. LATENTS	256
LATENT DIM.	48
ATTENTION LAYERS	2
SELF-ATTENTION HEADS	3
CROSS-ATTENTION HEADS	3
DROPOUT	0.01
PERCEIVER DECODER	
CROSS-ATTENTION HEADS	3
COPULA DECODER	
MIN. U	0.05
MAX. U	0.95
ATTENTIONAL COPULA	
ATTENTION LAYERS	1
ATTENTION HEADS	3
ATTENTION DIM.	16
FEEDFORWARD DIM.	48
FEEDFORWARD LAYERS	1
RESOLUTION	20
MARGINAL FLOW	
FEEDFORWARD LAYERS	1
FEEDFORWARD DIM.	48
FLOW LAYERS	3
FLOW DIM.	16

(B) TACTIS MODEL	
INPUT ENCODING	
SERIES EMBEDDING DIM.	5
INPUT ENCODER LAYERS	3
POSITIONAL ENCODING	
DROPOUT	0.01
TEMPORAL ENCODER	
ATTENTION LAYERS	2
ATTENTION HEADS	2
ATTENTION DIM.	24
ATTENTION FEEDFORWARD DIM.	24
DROPOUT	0.01
COPULA DECODER	
MIN. U	0.05
MAX. U	0.95
ATTENTIONAL COPULA	
ATTENTION LAYERS	1
ATTENTION HEADS	3
ATTENTION DIM.	16
FEEDFORWARD DIM.	48
FEEDFORWARD LAYERS	1
RESOLUTION	20
MARGINAL FLOW	
FEEDFORWARD LAYERS	1
FEEDFORWARD DIM.	48
FLOW LAYERS	3
FLOW DIM.	16

The experimental results show the efficacy of our proposed model over other approaches in prediction performance and memory utilization.

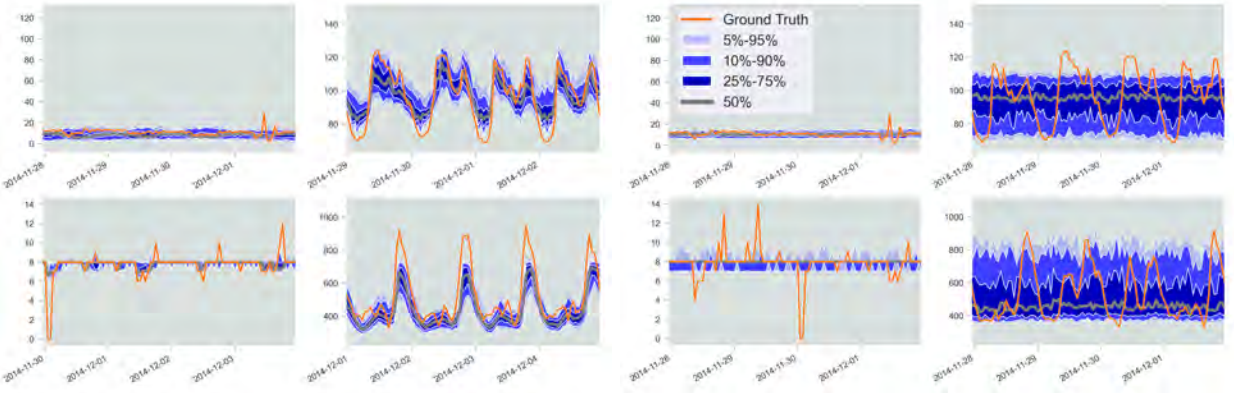
4.1 Memory Consumption Scaling

In this experiment, we evaluate the computational costs associated with our proposed perceiver-CDF model and the state-of-the-art TACTiS model with respect to the quantity of observed and inferred data. Here, we use the synthetic Random Walk data with a synchronously-measured time series consisting of 10 variables, 10 observed time-steps, and 10 to-be-inferred time-steps. Additionally, we vary the number of observed and inferred time-steps to assess their impact. Our analysis extends to comparing our model with TACTiS-PE, which leverages the perceiver-based encoder architecture for the TACTiS model. Additionally, we consider the TACTiS model with a midpoint inference mechanism, called TACTiS-MI. This model deduces data points using midpoint imputation for temporally local attention. A comprehensive comparison of memory usage among these models when applied to a single input series is illustrated in Figure 3. Firstly, it shows the quadratic relationship between the computational cost of TACTiS and the quantity of



(a) Perceiver-CDF (b) TACTiS

Figure 4: The predicted samples by the perceiver-CDF and TACTiS for one-month forecasts, corresponding to 48 time-steps, conditioned on two-day historical data in `electricity` dataset.



(a) Perceiver-CDF (b) TACTiS

Figure 5: The predicted samples by the perceiver-CDF and TACTiS for one-month forecasts, corresponding to 672 time-steps, conditioned on one-month historical data in `electricity` dataset.

input data. Secondly, it underscores the remarkable efficiency of the proposed model in terms of memory utilization. Additionally, it showcases the improvements achieved by utilizing perceiver-based encoder and midpoint inference mechanism for TACTiS model. TACTiS-PE, which utilizes a global inference mechanism, operates quadratically when dealing with inferred variables, thereby maintaining its quadratic scaling with respect to the number of predicted time-steps. Conversely, TACTiS-MI employs TACTiS' encoder, preserving its quadratic scaling with respect to the number of observed time steps. Overall, these results underscore the success of the perceiver-CDF model and the proposed midpoint inference mechanism in efficiently mitigating the inherent quadratic scaling issue within TACTiS.

4.2 Forecasting on Unimodal Datasets

We evaluate our proposed model's computational cost and inference performance across three real-world unimodal datasets. To begin, we employ the `fred-md` time series dataset, consisting of 20 input variables, each comprising 24 observed samples, to predict 24 time-steps into the future. Table 2 presents a comparative analysis of performance metrics for perceiver-CDF, TACTiS, GPVar, and AR models. We evaluate these

Table 2: Comparison of memory usage and prediction performance between Perceiver-CDF and other approaches in unimodal time series datasets, such as fred-md, traffic, and electricity.

		FRED-MD – 24 TIMESTEPS PREDICTION				
APPROACH	**PARAMS**	**MEMORY**	**BATCHES/S**	**NLL**	**RMSE-CM**	**CRPS**
AR(24)	6K	3.7 MB	37.4	–	7.0±0.5	1.10±0.05
GPVAR	78K	1.39 GB	11.7	42.3±0.6	6.8±0.5	0.86±0.06
TACTIS	91K	1.51 GB	12.3	42.3±0.4	6.1±0.4	0.74±0.05
PERCEIVERCDF	122K	1.66 GB	16.6	34.2±0.3	**6.0±0.4**	**0.71±0.06**
		TRAFFIC – 48 TIMESTEPS PREDICTION				
APPROACH	**PARAMS**	**MEMORY**	**BATCHES/S**	**NLL**	**RMSE-CM**	**CRPS**
AR(48)	20K	11.5 MB	10.31	–	0.053±0.005	0.431±0.004
GPVAR	78K	4.67 GB	5.81	204.6±0.8	0.044±0.003	0.215±0.008
TACTIS	91K	5.52 GB	5.84	198.7±0.6	0.035±0.002	0.181±0.009
PERCEIVERCDF	122K	2.75 GB	5.95	188.7±0.6	**0.028±0.002**	**0.162±0.006**
		ELECTRICITY – 48 TIMESTEPS PREDICTION				
APPROACH	**PARAMS**	**MEMORY**	**BATCHES/S**	**NLL**	**RMSE-CM**	**CRPS**
AR(48)	20K	11.6 MB	10.34	–	90±0.1	0.149±0.001
GPVAR	78K	4.78 GB	5.76	185.6±0.5	62±0.1	0.060±0.001
TACTIS	91K	5.42 GB	5.81	182.3±0.6	49±0.1	0.060±0.001
PERCEIVERCDF	122K	2.73 GB	5.93	177.8±0.8	**42±0.1**	**0.056±0.001**
		ELECTRICITY – 672 TIMESTEPS PREDICTION				
APPROACH	**PARAMS**	**MEMORY**	**BATCHES/S**	**NLL**	**RMSE-CM**	**CRPS**
AR(672)	270K	47.7 MB	1.74	–	159±0.8	0.290±0.02
GPVAR	78K	4.81 GB	3.48	350±0.4	147±0.5	0.198±0.005
TACTIS	91K	4.81 GB	3.65	280±0.2	141±0.3	0.186±0.006
PERCEIVERCDF	122K	372 MB	18.3	185±0.9	**98±0.1**	**0.133±0.001**

models based on negative log-likelihoods (NLL), root-mean-squared-errors of conditional expectations (RMSE-CM), and continuous ranked probability scores (CRPS). Our proposed model outperforms GPVar and AR while achieving competitive results with TACTiS in both RMSE-CM and CRPS metrics.

Next, we utilize `traffic` time series data with 20 input variables, each with 48 observed samples to predict 48 time-steps ahead. The performance comparisons between our model and other methods are demonstrated in Table 2. Our proposed model demonstrates a significant performance advantage over TACTiS, GPVar, and AR, excelling in both RMSE-CM and CRPS metrics. Notably, we achieve 20% improvement over TACTiS in terms of RMSE-CM. The number of parameters and memory usage also highlight the efficiency of the proposed model, which utilizes less than 50% of the memory compared to TACTiS and GPVar.

Lastly, we evaluate our model in the context of short-term and long-term prediction tasks using the `electricity` dataset. In the short-term prediction experiment, we utilize 20 variables, each spanning 48 observed time-steps, to forecast 48 time-steps into the future. Visual representations of the predictions from perceiver-CDF and TACTiS are shown in Figure 4. As shown in Table 2, our proposed model significantly outperforms other approaches, boasting a 14% improvement in RMSE-CM compared to TACTiS, all while utilizing just 50% of available memory. For the long-term prediction task, we work with 10 variables, each encompassing 672 observed time-steps, to predict the subsequent 672 time-steps. This experiment provides valuable insights into the capabilities of these models on a large-scale dataset. Visual representations of the predictions from perceiver-CDF and TACTiS are shown in Figure 5. In this scenario, our model demonstrates a significant performance advantage over TACTiS, excelling in both RMSE-CM and CRPS

Table 3: Comparison of memory usage and prediction performance between Perceiver-CDF and other approaches in multimodal time series datasets, such as room occupation, interstate traffic, air quality.

	ROOM OCCUPATION – 6 FEATURES ATTRIBUTIONS				
APPROACH	**PARAMS**	**MEMORY**	**RMSE-CM**	**USAGE**	**CO_2 LEVEL**
SSAE-LSTM	76K	5.22 GB	0.056±0.002	97.1%	96.5%
TACTiS	91K	6.38 GB	0.031±0.001	98.1%	97.7%
PERCEIVERCDF	122K	3.09 GB	**0.018**±0.001	**98.9%**	**98.4%**
	INTERSTATE TRAFFIC – 8 FEATURES ATTRIBUTIONS				
APPROACH	**PARAMS**	**MEMORY**	**RMSE-CM**	**RAIN**	**TRAFFIC LEVEL**
SSAE-LSTM	76K	5.68 GB	0.083±0.004	95.3%	94.6%
TACTiS	91K	7.13 GB	0.065±0.003	96.7%	96.1%
PERCEIVERCDF	122K	3.22 GB	**0.027**±0.003	**98.2%**	**97.8%**
	AIR QUALITY – 12 FEATURES ATTRIBUTIONS				
APPROACH	**PARAMS**	**MEMORY**	**RMSE-CM**	**RAIN**	**PM2.5 LEVEL**
SSAE-LSTM	76K	6.17 GB	0.106±0.006	93.7%	93.4%
TACTiS	91K	8.83 GB	0.074±0.005	95.8%	94.9%
PERCEIVERCDF	122K	3.41 GB	**0.022**±0.004	**98.5%**	**98.1%**

while utilizing only 10% of available memory. Additionally, our perceiver-CDF model manages to capture the seasonal patterns in the data, albeit not as accurately as in the short-term task. Conversely, TACTiS and other methods face inherent challenges when dealing with extended time series. In particular, TACTiS struggles to model the underlying seasonal structures within the data, resulting in less reliable performance when tasked with long-term predictions.

4.3 Forecasting on Multimodal Datasets

We first evaluate the predictive capabilities of the perceiver-CDF model on the `room occupation` dataset (Candanedo 2016). This dataset is multimodal, consisting of 6 feature attributes related to room conditions, such as temperature, humidity, and CO_2 levels. The evaluation of predictive performance is based on the average RMSE-CM across all six attributes. Furthermore, we undertake two classification tasks: the first task involves predicting room occupancy, while the second task focuses on detecting high CO_2 levels (i.e., levels exceeding 700 ppm). Here, we conduct a comparative analysis with TACTiS (Drouin et al. 2022) and SSAE-LSTM (Zhu et al. 2021). Both of these methods employ a strategy of concatenating all feature attributes at each time-step for prediction. The performance results, as presented in Table 3, consist of measures such as average RMSE-CM, room occupation detection accuracy, and high CO_2 detection accuracy. The memory usage is also provided to highlight the efficiency of our model when achieving 40% reduction in RMSE-CM compared to TACTiS while utilizing only half of the computational resources.

Next, we extend our experimentation to the `interstate traffic` dataset (Hogue 2019). This dataset comprises 8 feature attributes related to weather conditions (e.g., temperature, snow), holiday status, and traffic volume. To assess predictive performance, we utilize RMSE-CM calculated across all eight attributes. Additionally, we investigate two classification tasks: firstly, identifying instances of rainy weather conditions, and secondly, detecting periods of high traffic volume (i.e., volumes exceeding 2000 cars). Table 3 illustrates that the proposed perceiver-CDF model significantly outperforms other approaches while maintaining linear memory usage. Notably, our approach achieves a 58% improvement in RMSE-CM compared to TACTiS and consistently excels in prediction tasks related to detecting rain and high traffic.

Lastly, we evaluate the performance of our approach on the `air quality` dataset (Chen 2019), which encompasses 12 variables, each with 12 feature attributes, including 6 pollution-related features (e.g., PM2.5, PM10) and 6 weather-related features (e.g., temperature, rain). To assess the quality of our

predictions, we employ the average RMSE-CM calculated across all attributes. Moreover, we tackle two classification tasks: firstly, identifying instances of rainy weather conditions, and secondly, detecting periods with elevated PM2.5 levels, specifically those exceeding 80 $\mu g/m^3$. Table 3 showcases the performance comparison between perceiver-CDF and other approaches, with our model achieving a remarkable 70% improvement in RMSE-CM compared to TACTiS while utilizing only 40% of the memory resources.

5 CONCLUSION

We present a new method for modeling multimodal time series, leveraging cross-attention and copula-attention mechanisms. Our model adeptly encodes the global patterns within partially observed multimodal time series into latent representations, effectively streamlining computational complexity. It also incorporates temporally local attention via midpoint inference, focusing token attention on those with the utmost temporal relevance to their conditioning for precise conditional modeling. Our experiments demonstrate that our proposed model exhibits heightened efficiency as prediction length and the number of feature attributes increases. Perceiver-based encoding proves highly effective in addressing the challenges posed by complex multimodal datasets. In future work, we aim to extend the applications of this approach by enhancing the structure of the copula-based model.

ACKNOWLEDGMENTS

We extend our gratitude to Duke University and the Rhodes Family for their generous partial support of this work. Vahid Tarokh was partially supported by the Office of Naval Research (ONR) under grant number N00014-21-1-2590.

REFERENCES

Ashok, A., É. Marcotte, V. Zantedeschi, N. Chapados and A. Drouin. 2024. "TACTiS-2: Better, Faster, Simpler Attentional Copulas for Multivariate Time Series". In *The Twelfth International Conference on Learning Representations*.

Benidis, K., S. S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, C. Turkmen, , , , *et al*. 2022. "Deep learning for time series forecasting: Tutorial and literature survey". *ACM Computing Surveys* 55(6):1–36.

Box, G. E., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Candanedo, L. 2016. "Occupancy Detection ". DOI: https://doi.org/10.24432/C5X01N.

Chen, S. 2019. "Beijing Multi-Site Air-Quality Data". DOI: https://doi.org/10.24432/C5RK5G.

Chen, Y., Y. Kang, Y. Chen, and Z. Wang. 2020. "Probabilistic forecasting with temporal convolutional neural network". *Neurocomputing* 399:491–501.

Child, R., S. Gray, A. Radford, and I. Sutskever. 2019. "Generating Long Sequences with Sparse Transformers". *arXiv preprint arXiv:1904.10509*.

Drouin, A., É. Marcotte, and N. Chapados. 2022. "TACTiS: Transformer-Attentional Copulas for Time Series". In *International Conference on Machine Learning*, 5447–5493. PMLR.

Dua, D. and C. Graff. 2017. "UCI Machine Learning Repository".

Godahewa, R., C. Bergmeir, G. I. Webb, R. J. Hyndman and P. Montero-Manso. 2021. "Monash time series forecasting archive". *arXiv preprint arXiv:2105.06643*.

Größer, J. and O. Okhrin. 2022. "Copulae: An overview and recent developments". *Wiley Interdisciplinary Reviews: Computational Statistics* 14(3):e1557.

Hogue, J. 2019. "Metro Interstate Traffic Volume". DOI: https://doi.org/10.24432/C5X60B.

Huang, C.-W., D. Krueger, A. Lacoste, and A. Courville. 2018. "Neural autoregressive flows". In *International Conference on Machine Learning*, 2078–2087. PMLR.

Jaegle, A., S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, , , , *et al*. 2021. "Perceiver IO: A General Architecture for Structured Inputs & Outputs". *arXiv preprint arXiv:2107.14795*.

Kalliovirta, L., M. Meitz, and P. Saikkonen. 2015. "A Gaussian mixture autoregressive model for univariate time series". *Journal of Time Series Analysis* 36(2):247–266.

Krupskii, P. and H. Joe. 2020. "Flexible copula models with dynamic dependence and application to financial data". *Econometrics and Statistics* 16:148–167.

Lim, B., S. Ö. Arık, N. Loeff, and T. Pfister. 2021. "Temporal fusion transformers for interpretable multi-horizon time series forecasting". *International Journal of Forecasting* 37(4):1748–1764.

Lim, B. and S. Zohren. 2021. "Time-series forecasting with deep learning: a survey". *Philosophical Transactions of the Royal Society A* 379(2194):20200209.

Liu, Y., R. Yu, S. Zheng, E. Zhan and Y. Yue. 2019. "NAOMI: Non-Autoregressive Multiresolution Sequence Imputation". *Advances in Neural Information Processing Systems* 32.

Lopez-Paz, D., J. Hernández-lobato, and B. Schölkopf. 2012. "Semi-supervised domain adaptation with non-parametric copulas". *Advances in neural information processing systems* 25.

Mayer, A. and D. Wied. 2023. "Estimation and inference in factor copula models with exogenous covariates". *Journal of Econometrics*.

Müller, S., N. Hollmann, S. P. Arango, J. Grabocka and F. Hutter. 2021. "Transformers can do bayesian inference". *arXiv preprint arXiv:2112.10510*.

Nelsen, R. B. 2006. *An introduction to copulas*. Springer.

Roy, A., M. Saffar, A. Vaswani, and D. Grangier. 2021. "Efficient Content-Based Sparse Attention with Routing Transformers". *Transactions of the Association for Computational Linguistics* 9:53–68.

Salinas, D., M. Bohlke-Schneider, L. Callot, R. Medico and J. Gasthaus. 2019. "High-Dimensional Multivariate Forecasting with Low-Rank Gaussian Copula Processes". *Advances in Neural Information Processing Systems* 32.

Shih, S.-Y., F.-K. Sun, and H.-y. Lee. 2019. "Temporal pattern attention for multivariate time series forecasting". *Machine Learning* 108:1421–1441.

Shukla, S. N. and B. M. Marlin. 2021. "Multi-time attention networks for irregularly sampled time series". *arXiv preprint arXiv:2101.10318*.

Sklar, M. 1959. "Fonctions de répartition à n dimensions et leurs marges". 8(3):229–231.

Tashiro, Y., J. Song, Y. Song, and S. Ermon. 2021. "Csdi: Conditional score-based diffusion models for probabilistic time series imputation". *Advances in Neural Information Processing Systems* 34:24804–24816.

Tay, Y., D. Bahri, L. Yang, D. Metzler and D.-C. Juan. 2020. "Sparse Sinkhorn Attention". In *International Conference on Machine Learning*, 9438–9447. PMLR.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al*. 2017. "Attention Is All You Need". *Advances in Neural Information Processing Systems* 30.

Zhu, Q., S. Zhang, Y. Zhang, C. Yu, M. Dang and L. Zhang. 2021. "Multimodal Time Series Data Fusion Based on SSAE and LSTM". In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–7. IEEE.

AUTHOR BIOGRAPHIES

CAT P. LE received a B.S. degree in electrical and computer engineering from Rutgers University, an M.S. degree in electrical engineering from the California Institute of Technology (Caltech), and a Ph.D. degree in electrical and computer engineering from Duke University. His research interests include image processing, computer vision, and machine learning, with a focus on transfer learning, and continual learning. His email address is cat.le@duke.edu.

CHRIS CANNELLA received a B.S. degree in astrophysics from the Caltech, and an M.S. and a Ph.D. degree in electrical and computer engineering from Duke University. He has extensive research experience in machine learning, signal processing, data compression, and error correction. His email address is christopher.cannella@duke.edu.

ALI HASAN received a B.S. degree from the University of North Carolina, Chapel Hill (UNC), and a Ph.D. degree in biomedical engineering from Duke University. His research interests include image processing, machine learning, and stochastic differential equations. His email address is ali.hasan@duke.edu.

YUTING NG received a B.S. degree in electrical engineering and an M.S. degree in aerospace engineering from the University of Illinois at Urbana-Champaign (UIUC). She graduated with a Ph.D. degree in electrical and computer engineering from Duke University. Her research interests include machine learning, radar signal processing, GNSS signal tracking, navigation, and time synchronization. Her e-mail address is yuting.ng@duke.edu.

VAHID TAROKH joined as an Associate Professor at the Massachusetts Institute of Technology (MIT) in 2000. In 2002, he joined Harvard University as a Hammond Vinton Hayes Senior Fellow of Electrical Engineering and a Perkins Professor of Applied Mathematics. In 2018, he joined as a Rhodes Family Professor of electrical and computer engineering, computer science, and mathematics, and a Bass Connections Endowed Professor with Duke University. He was a Gordon Moore Distinguished Research Fellow at Caltech. His email address is vahid.tarokh@duke.edu.