Retrieval Based Time Series Forecasting

Baoyu Jing
baoyuj2@illinois.edu
University of Illinois at
Urbana-Champaign

Si Zhang
sizhang@fb.com
Meta

Yada Zhu
yzhu@us.ibm.com
IBM Research

Bin Peng
binpeng@illinois.edu
University of Illinois at
Urbana-Champaign

Kaiyu Guan
kaiyug@illinois.edu
University of Illinois at
Urbana-Champaign

Andrew Margenot
margenot@illinois.edu
University of Illinois at
Urbana-Champaign

Hanghang Tong
htong@illinois.edu
University of Illinois at
Urbana-Champaign

ABSTRACT

Time series data appears in a variety of applications such as smart transportation and environmental monitoring. One of the fundamental problems for time series analysis is time series forecasting. Despite the success of recent deep time series forecasting methods, they require sufficient observation of historical values to make accurate forecasting. In other words, the ratio of the output length (or forecasting horizon) to the sum of the input and output lengths should be low enough (e.g., 0.3). As the ratio increases (e.g., to 0.8), the uncertainty for the forecasting accuracy increases significantly. In this paper, we show both theoretically and empirically that the uncertainty could be effectively reduced by retrieving relevant time series as references. In the theoretical analysis, we first quantify the uncertainty and show its connections to the Mean Squared Error (MSE). Then we prove that models with references are easier to learn than models without references since the retrieved references could reduce the uncertainty. To empirically demonstrate the effectiveness of the retrieval based time series forecasting models, we introduce a simple yet effective two-stage method, called ReTime consisting of a relational retrieval and a content synthesis. We also show that ReTime can be easily adapted to the spatial-temporal time series and time series imputation settings. Finally, we evaluate ReTime on real-world datasets to demonstrate its effectiveness.

ACM Reference Format:
Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. 2022. Retrieval Based Time Series Forecasting. In *CIKM'22 workshop on Applied Machine Learning Methods for Time Series Forecasting (AMLTS), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Time series analysis has received paramount interest in numerous real-world applications [1–7], such as smart transportation and environmental monitoring. Accurately forecasting time series provides valuable insights for making transport policies [8] and climate change policies [9].

One of the fundamental problems for time series analysis is time series forecasting. Despite the success of recent deep learning methods for time series forecasting [1–5], a sufficient observation of the input historical time series is required. Specifically, the *ratio* of the output length (i.e., forecasting horizon) to the sum of the input and output length should be sufficiently low (e.g., 0.3). In fact, this ratio is a special case of the *missing rate* used in time series imputation, and thus for clarity and consistency with the literature, we formulate the task of time series forecasting from the perspective of time series imputation in this paper. Please refer to Section 2 for details. When the missing rate increases to a high level (e.g., 0.8), the uncertainty of the forecasting accuracy will increase significantly. In real-world applications, it is common that one wants to forecast future values based on very limited observations. In smart transportation, government administrators might be interested in forecasting the traffic conditions of a road with broken sensors [10]. In environmental monitoring, geologists always have a desire of obtaining the temperature of a specific location, where sensors cannot be easily placed [11].

To address this problem, we first formally quantify the uncertainty of the forecasting based on the entropy of the ground truth conditioned on the predicted values, and show its connections to the Mean Squared Error (MSE). Then we theoretically prove that models with reference time series are easier to train than those without references, since the references could reduce uncertainties. Motivated by the theoretical analysis, we introduce a simple yet effective two-stage time forecasting method called ReTime. Given a target time series, in the first relational retrieval stage, ReTime retrieves the references from a database based on the relations among time series. We use relational retrieval rather than content based retrieval since the input historical values of the target time series could be very unreliable when the input length is very short.

CIKM'22 AMLTS, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Thus, content-based methods could retrieve unreliable references. In comparison, the relational information is usually reliable and easy to obtain in practice [12–15], such as whether two sensors are adjacent to each other in traffic/environmental monitoring. In the second content synthesis stage, ReTime synthesizes the future values based on the content of the target and the references. Next, we show that the proposed ReTime could be easily applied to the time series imputation task and the spatial-temporal time series setting. Finally, we empirically evaluate ReTime on two real-world datasets to demonstrate its effectiveness.

The main contributions of the paper are summarized as follows:

- **Theoretical Analysis.** We theoretically quantify the uncertainty of the predicted values based on conditional entropy and show its connections to the MSE. We also theoretically demonstrate that models with references are easier to train than those without references.
- **Algorithm.** We introduce a two-stage method ReTime for time series forecasting, which is comprised of relational retrieval and content synthesis. ReTime can also be easily applied to the spatial-temporal time series and time series imputation settings.
- **Empirical Evaluation.** We evaluate ReTime on two real-world datasets and various settings (i.e., single/spatial temporal forecasting/imputation) to demonstrate its effectiveness.

2 PRELIMINARY

The ratio $L_{out}/(L_{in}+L_{out})$ of the output length L_{out} to the sum of the input length L_{in} and output length L_{out} is a special case of the missing rate used in time series imputation, and thus we formulate tasks of time series forecasting from the perspective of time series imputation. We summarize the mathematical notations in Table 1.

Definition 2.1 (Single Time Series Forecasting). *Given an incomplete target time series* $\mathbf{X} \in \mathbb{R}^{T\times v}$, *where* T *and* v *are the numbers of time steps and variates, along with its indicator mask* $\mathbf{M} \in \{0,1\}^{T\times v}$, *which indicates the absence/presence of the data points, the task aims to generate a new target time series* $\hat{\mathbf{X}} \in \mathbb{R}^{T\times v}$ *to predict the missing values* $(1-\mathbf{M}) \odot \hat{\mathbf{X}}$.

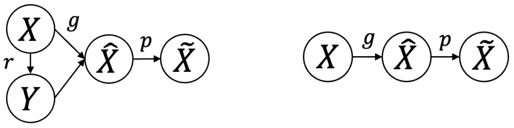
Definition 2.2 (Retrieval Based Single Time Series Forecasting). *Given an incomplete target time series* $\mathbf{X} \in \mathbb{R}^{T\times v}$, *where* T *and* v *are the numbers of time steps and variates, along with its indicator mask* $\mathbf{M} \in \{0,1\}^{T\times v}$, *the task aims to generate a new target time series* $\hat{\mathbf{X}} \in \mathbb{R}^{T\times v}$ *to predict the missing values* $(1-\mathbf{M}) \odot \hat{\mathbf{X}}$ *based on the input target time series* $\mathbf{X}$ *and the* K *reference time series* $\{\mathbf{Y}_k\}_{k=1}^{K}$ *retrieved from a database* $\{\mathbf{Y}'_n \in \mathbb{R}^{T'\times v}\}_{n=1}^{N}$, *where* $N \gg K$ *is the number of time series and* $T' \gg T$.

Definition 2.3 (Spatial-Temporal Time Series Forecasting). *Given an incomplete spatial-temporal time series* $\mathbf{X} \in \mathbb{R}^{N\times T\times v}$, *where* N, T *and* v *are the numbers of time series, time steps, and variates, along with its indicator mask* $\mathbf{M} \in \{0,1\}^{N\times T\times v}$, *where 0/1 indicates the absence/presence of the data points, and its adjacency matrix* $\mathbf{A} \in \mathbb{R}^{N\times N}$, *the task is to generate a new time series* $\hat{\mathbf{X}} \in \mathbb{R}^{N\times T\times v}$ *to predict the missing values* $(1-\mathbf{M}) \odot \hat{\mathbf{X}}$.

Note that for forecasting, missing points and zeros are concentrated at the end of the target $\mathbf{X}$ and mask $\mathbf{M}$ after a certain separation time step τ: $\mathbf{M}[\tau : T] = 0$.

Notation	Description
$\mathbf{X}$	incomplete target time series
$\tilde{\mathbf{X}}$	complete target time series
$\hat{\mathbf{X}}$	predicted target time series
$\mathbf{Y}_k$	k-th reference time series
$\mathbf{Y}'_n$	n-th time series in the database
$\mathbf{M}$	indicator mask of the target time series
$\mathbf{A}'$	adjacency matrix for the database $\{\mathbf{Y}'_n\}_{n=1}^{N}$
$\mathbf{A}$	adjacency matrix for $\{\mathbf{Y}'_n\}_{n=1}^{N}$ and $\mathbf{X}$
$\mathcal{R}$	relation between $\mathbf{X}$ and $\{\mathbf{Y}'_n\}_{n=1}^{N}$
T	length of time series
T'	length of time series in $\{\mathbf{Y}'_n\}_{n=1}^{N}$
N	number of time series in the database
K	number of reference time series
v	number of variates
d	size of hidden dimension
τ	separation time separating history and future time

Table 1: Mathematical Notations



(a) With references **(b) Without references**

Figure 1: Graphical models for methods w. or w/o references. $X, \hat{X}, \tilde{X}$ **and** Y **are random variables for incomplete, generated, complete, and retrieved time series.** g **and** r **are the generation and retrieval models.** p **is the relation between** $\hat{X}$ **and** $\tilde{X}$**.**

3 THEORETICAL MOTIVATION

An illustration of graphical models for methods with or without references is presented in Figure 1, which shows the relations among random variables. $X, \tilde{X}, \hat{X}$ and Y denote the random variables for the incomplete target, complete target, generated target, and retrieved reference time series respectively. g and r denote the generation and retrieval model respectively. p is the relation between $\hat{X}$ and $\tilde{X}$. Following the common practice for linear regression [16], which assumes a Gaussian noise between $\hat{X}$ and $\tilde{X}$, we define the relation p as:

$$p(\tilde{\mathbf{x}}|\hat{\mathbf{x}}) = \mathcal{N}(\tilde{\mathbf{x}}|\hat{\mathbf{x}}, \sigma^2\mathbf{I}) \quad (1)$$

where $\mathcal{N}$ denotes the Gaussian distribution, $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}} \in \mathbb{R}^{v}$ denote the values of $\hat{\mathbf{X}}$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{(T-\tau)\times v}$ at a future time step $t \in [\tau, T]$, σ is the standard deviation, and $\mathbf{I} \in \mathbb{R}^{v\times v}$ is the identity matrix. Note that $\mathbf{X}[:\tau] = \tilde{\mathbf{X}}[:\tau]$ for historical steps $t \in [1, \tau)$, and thus we only consider the future time steps $t \in [\tau, T]$.

We first quantify the uncertainty Δ for the accuracy of $\hat{X}$ in Definition 3.1 as the entropy of $\tilde{X}$ conditioned on $\hat{X}$. According to Equation 1 and the definition of conditional entropy, we can calculate the uncertainty Δ (Lemma 3.1). As the only parameter in Δ is the standard deviation σ in Equation 1, we can further prove that Δ is equivalent to the MSE between $\tilde{X}$ and $\hat{X}$.

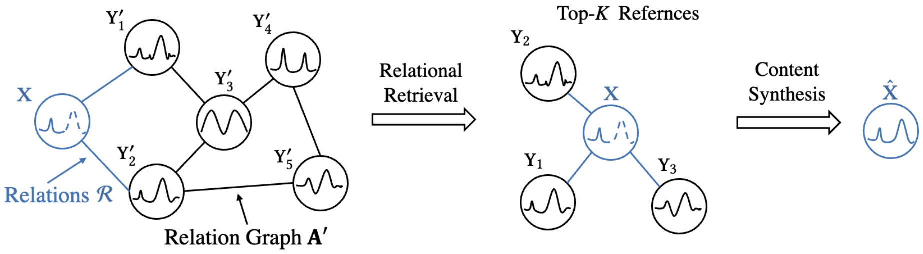


Figure 2: Overview of ReTime. Given X and $\{Y'_n\}_{n=1}^N$, ReTime first retrieves the top K references $\{Y_k\}_{k=1}^K$ based on the relations A' and $\mathcal{R}$, and then combines the content of X and $\{Y_k\}_{k=1}^K$ to generate $\hat{X}$. Solid/dashed curves are observed/unobserved values.

Definition 3.1 (Uncertainty of $\hat{X}$).

$$\Delta = H(\tilde{X}|\hat{X}) \tag{2}$$

where H denotes the entropy.

Lemma 3.1 (Uncertainty Calculation). *According to the definition of conditional entropy and Equation* (1)*, we have:*

$$\Delta = \frac{v}{2}(1 + \log 2\pi\sigma^2) \tag{3}$$

Lemma 3.2 (Equivalence between Uncertainty and MSE). *The uncertainty is equivalent to MSE.*

$$\Delta \Leftrightarrow MSE \tag{4}$$

Proof. The only parameter of Δ in Lemma 3.1 is the standard deviation σ, which can be estimated by:

$$\sigma = \sqrt{\frac{1}{Z}\sum_{z=1}^{Z}||\tilde{\mathbf{x}}_z - \hat{\mathbf{x}}_z||^2} \tag{5}$$

where Z is the total number of data pairs. The item under the square root is MSE between $\tilde{X}$ and $\hat{X}$. □

We further study the relations among the inputs X, Y, output $\hat{X}$ of the generation model g, and the complete time series $\tilde{X}$ based on their dependencies shown in Figure 1. Firstly, given $\tilde{X}$ and $\hat{X}$, we show in Lemma 3.3 that minimizing their MSE loss is equivalent to maximizing their mutual information $I(\tilde{X};\hat{X})$. Secondly, we prove that adding the retrieved reference Y to the input could reduce the uncertainty for X. Lemma 3.4 shows that methods with Y have a higher lower-bound than those without Y for the mutual information of the ground-truth $\tilde{X}$ and the predicted values $\hat{X}$: $I(\tilde{X};\hat{X}) \geq I(\tilde{X};X,Y) \geq I(\tilde{X};X)$. Finally, due to the equivalence of MSE and MI (Lemma 3.3), we can conclude that models with Y (Figure 1a) are easier to learn than models without Y (Figure 1b) under the MSE loss.

Lemma 3.3 (Equivalence between MSE and MI). *Minimizing the MSE loss of $\tilde{X}$ and $\hat{X}$ is equivalent to maximize the mutual information of $\tilde{X}$ and $\hat{X}$: $I(\tilde{X};\hat{X})$.*

$$\min \mathbb{E}_{p(\tilde{\mathbf{x}},\hat{\mathbf{x}})}||\tilde{\mathbf{x}} - \hat{\mathbf{x}}||^2 \Leftrightarrow \max I(\tilde{X},\hat{X}) \tag{6}$$

where $p(\tilde{\mathbf{x}},\hat{\mathbf{x}})$ indicates whether $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ is a true pair.

Proof. Minimizing MSE of $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ is equivalent to maximizing the log-likelihood $\log p(\tilde{\mathbf{x}}|\hat{\mathbf{x}})$ [16], where p is given in Equation (1):

$$\min \mathbb{E}_{p(\tilde{\mathbf{x}},\hat{\mathbf{x}})}||\tilde{\mathbf{x}} - \hat{\mathbf{x}}||^2 \Leftrightarrow \max \mathbb{E}_{p(\tilde{\mathbf{x}},\hat{\mathbf{x}})}[\log p(\tilde{\mathbf{x}}|\hat{\mathbf{x}})] \tag{7}$$

Therefore, we only need to prove

$$\max \mathbb{E}_{p(\tilde{\mathbf{x}},\hat{\mathbf{x}})}[\log p(\tilde{\mathbf{x}}|\hat{\mathbf{x}})] \Leftrightarrow \max I(\tilde{X};\hat{X}) \tag{8}$$

In fact,

$$I(\tilde{X};\hat{X}) = H(\tilde{X}) - H(\tilde{X}|\hat{X}) \tag{9}$$

and $H(\tilde{X})$ is a constant since the ground-truth $\tilde{X}$ is fixed in the dataset. Thus, we have

$$\max I(\tilde{X};\hat{X}) \Leftrightarrow \max -H(\tilde{X}|\hat{X}) \tag{10}$$

According to the definition of conditional entropy, we have

$$-H(\tilde{X}|\hat{X}) = \mathbb{E}_{p(\tilde{\mathbf{x}},\hat{\mathbf{x}})}[\log p(\tilde{\mathbf{x}}|\hat{\mathbf{x}})] \tag{11}$$

The proof is concluded by combining Equations (7)(10)(11). □

Lemma 3.4 (MI Monotonicity). *The following MI inequalities hold for the graphical model shown in Figure 1a.*

$$I(\tilde{X};\hat{X}) \geq I(\tilde{X};X,Y) \geq I(\tilde{X};X) \tag{12}$$

Sketch of Proof. The first inequality is derived based on the data processing inequality [17]. The second inequality holds since $I(\tilde{X};X,Y) = I(\tilde{X};X) + I(\tilde{X};Y|X)$ and $I(\tilde{X};Y|X) \geq 0$. □

4 METHODOLOGY

Figure 2 is an overview of ReTime. In the first stage, ReTime retrieves the top K references $\{\mathbf{Y}_k\}_{k=1}^K$ for the target $\mathbf{X}$ from the database $\{\mathbf{Y}'_n\}_{n=1}^N$, based on the relations $\mathcal{R}$ between the target and database and the relation graph $\mathbf{A}'$ of the database. In the second stage, ReTime combines $\mathbf{X}$ and $\{\mathbf{Y}_k\}_{k=1}^K$ to generate $\hat{\mathbf{X}}$.

4.1 Relational Retrieval

When the missing rate of the target $\mathbf{X}$ is high (e.g., 0.8), it will be hard to accurately complete $\mathbf{X}$ merely based on the observed historical content of $\mathbf{X}$. Even for recent forecasting methods [1], the uncertainty of the prediction accuracy could be very high under a high missing rate. To reduce the uncertainty, we propose to retrieve K references $\{\mathbf{Y}_k\}_{k=1}^K$ from the database $\{\mathbf{Y}'_n\}_{n=1}^N$, based on the relations $\mathcal{R}$ and $\mathbf{A}'$. We choose relational retrieval over content-based retrieval since the observed historical content could be noisy and unreliable when the missing rate is high.

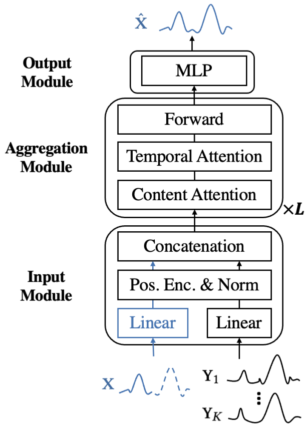


Figure 3: Content Synthesis

Given $\mathbf{A}' \in \mathbb{R}^{N\times N}$ and $\mathcal{R}$, we first construct a new adjacency matrix $\mathbf{A} \in \mathbb{R}^{(N+1)\times(N+1)}$ by appending $\mathcal{R}$ to the last row and column of $\mathbf{A}'$. Then we use the Random Walk with Restart (RWR) [18] to obtain the relational proximity scores between $\mathbf{X}$ and $\{\mathbf{Y}'_n \in \mathbb{R}^{T'\times v}\}_{n=1}^{N}$, whose closed-form solution is given by:

$$\mathbf{p} = (1-c)(\mathbf{I} - c\tilde{\mathbf{A}})^{-1}\mathbf{e} \tag{13}$$

where $\tilde{\mathbf{A}}$ is the normalized adjacency matrix, $\mathbf{I}$ is the identity matrix and $c \in (0,1)$ is the tunable damping factor; $\mathbf{e} \in \{0,1\}^{N+1}$ is the indicator vector of $\mathbf{X}$, where $\mathbf{e}[N+1] = 1$ and $\mathbf{e}[n] = 0$ for $\forall n \in [1,\cdots,N]$; $\mathbf{p} \in \mathbb{R}^{N+1}$ is the relational proximity vector where $\mathbf{p}[n]$ describes the proximity between $\mathbf{X}$ and $\mathbf{Y}'_n$. Given $\mathbf{p}$, we retrieve the top K time series $\{\mathbf{Y}_k\}_{k=1}^{K}$ as the references.

Instead of using the entire $\mathbf{Y}_k \in \mathbb{R}^{T'\times v}$ as the reference, we only use a T-length ($T \ll T'$) snippet of it, since using the entire sequence could introduce much irrelevant noisy information. Let T_S and T'_S be the start time of the target $\mathbf{X}$ and the reference snippet respectively. Many time series have clear periodicity patterns, such as the weekly pattern of traffic monitoring data and the yearly pattern of the air temperature. In this paper, we exploit the periodicity patterns and set the time difference between the target and $\Delta T = T_S - T'_S$ as the length of one period.

4.2 Content Synthesis

Despite the relational closeness of $\mathbf{X}$ and $\{\mathbf{Y}_k\}_{k=1}^{K}$, they usually have content discrepancies. For example, peaks and valleys of $\mathbf{X}$ and $\mathbf{Y}_k$ might be different, and $\mathbf{Y}_k$ might be noisy. Besides, there are also temporal dependencies among different time steps. Therefore, it is necessary to build a model to combine their content.

Figure 3 presents an illustration of the content synthesis model, which is comprised of the input, aggregation, and output modules. The input module maps $\mathbf{X} \in \mathbb{R}^{T\times v}$ and $\{\mathbf{Y}_k \in \mathbb{R}^{T\times v}\}_{k=1}^{K}$ into embeddings $\mathbf{H} \in \mathbb{R}^{(K+1)\times T\times d}$, where d is the size of hidden dimension. Then the aggregation module aggregates the content of $\mathbf{H}$ across the $K+1$ time series and T time steps into the aggregated embeddings $\mathbf{H}$. Finally, the output module generates the completed time series $\hat{\mathbf{X}}$ based on the aggregated embeddings $\mathbf{H}$.

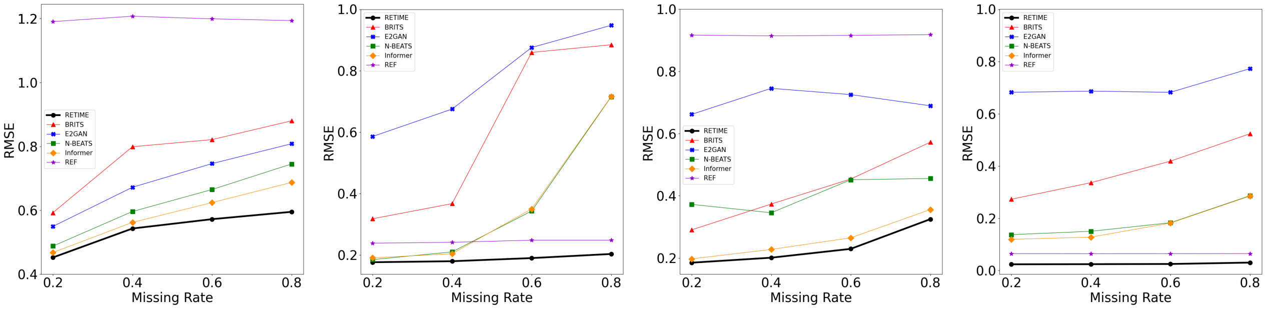
A - Input Module. The input module maps the time series into the embedding space. Given $\mathbf{X}$ and $\{\mathbf{Y}_k\}_{k=1}^{K}$, we first apply separate linear layers to them. Then we apply the position encoding [19] and layer norm [20] to obtain the embeddings $\mathbf{H} \in \mathbb{R}^{T\times d}$ and $\{\mathbf{H}'_k \in \mathbb{R}^{T\times d}\}_{k=1}^{K}$. Finally, they are concatenated into $\hat{\mathbf{H}} \in \mathbb{R}^{(K+1)\times T\times d}$, where $\hat{\mathbf{H}}[1:K] = [\mathbf{H}'_1,\ldots,\mathbf{H}'_K]$ and $\hat{\mathbf{H}}[K+1] = \mathbf{H}$.
B - Aggregation Module. The aggregation module jointly considers the content discrepancies between $\mathbf{H}$ and $\{\mathbf{H}'_k\}_{k=1}^{K}$ at each time step and the temporal dependencies across time steps. Based on the multi-head self-attention [19], we build content and temporal attention models to handle the content discrepancies and temporal dependencies. The aggregation module sequentially applies the content and temporal attention models to $\hat{\mathbf{H}} \in \mathbb{R}^{(K+1)\times T\times d}$, which is comprised of L blocks, and the structure of the block is shown in Figure 3. Within the block, firstly, the content attention computes content attention scores over the first dimension ($K+1$ time series) of $\hat{\mathbf{H}}$ for each step $t \in [1,\cdots,T]$, and produces the content embeddings $\mathbf{Z} \in \mathbb{R}^{(K+1)\times T\times d}$. Secondly, the temporal attention computes temporal attention scores over the second dimension (T steps) of $\mathbf{Z}$ for each time series $k \in [1,\cdots,K+1]$, and encodes temporal information into $\mathbf{Z}$. Finally, a forward layer [19] is applied to $\mathbf{Z}$ to obtain the aggregated embeddings $\hat{\mathbf{H}} \in \mathbb{R}^{(K+1)\times T\times d}$.
C - Output Module. The output module maps the aggregated embeddings $\hat{\mathbf{H}}$ from the embedding space $\mathbb{R}^{d}$ back into the original space $\mathbb{R}^{V}$ of the time series. Specifically, the output module takes input as $\hat{\mathbf{H}}[K+1]$, which is the embeddings of the target $\mathbf{X}$. Then a Multi-Layer Perceptron (MLP) is applied over $\mathbf{H}[K+1]$ to generate the predicted target time series $\hat{\mathbf{X}}$.

4.3 Adaptation to Other Settings

A - Spatial-Temporal Time Series. Spatial-temporal time series is ubiquitous and have attracted a lot of attention [21–24]. Different from the retrieval-based single time series forecasting, where the target is a single time series *out of* the database $\mathbf{X} \notin \{\mathbf{Y}_n\}_{n=1}^{N}$, the target of the spatial-temporal time series is the entire dataset $\mathbf{X} = \{\mathbf{Y}_n\}_{n=1}^{N}$. ReTime can be naturally adapted to spatial-temporal time series by treating each single time series of spatial-temporal time series as the target and the rest time series as the database. The relational retrieval can be interpreted as the diffusion graph kernel [18], which identifies the most important neighbors, and the content synthesis aggregates the information of the neighbors.
B - Time Series Imputation. It is obvious that ReTime can be naturally applied to the time series imputation task for regularly-sampled time series.

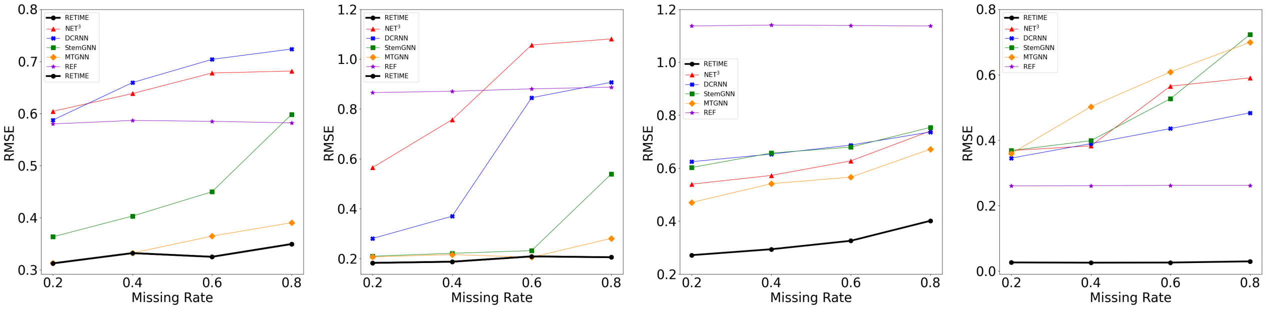
4.4 Training

During training, given a complete target time series $\tilde{\mathbf{X}}$, we generate a binary mask $\mathbf{M}$ to obtain the incomplete target time series $\mathbf{X} = \mathbf{M}\odot\tilde{\mathbf{X}}$, where $\odot$ is the Hadamard product. For forecasting, the values after the pre-defined time step τ are set as zeros. For imputation, we randomly generate the values for the masks according to the pre-defined missing rate. Then we feed $\mathbf{X}$ with its references $\{\mathbf{Y}_k\}_{k=1}^{K}$, which are retrieved by the relational retrieval model, to the content synthesis model to obtain $\hat{\mathbf{X}}$. We use the standard Mean Squared Error (MSE) between $\hat{\mathbf{X}}$ and $\tilde{\mathbf{X}}$ to train the model.



(a) Traffic Forecasting **(b) Temperature Forecasting** **(c) Traffic Imputation** **(d) Temperature Imputation**

Figure 4: RMSE scores on single time series forecasting and imputation. The lower the better.



(a) Traffic Forecasting **(b) Temperature Forecasting** **(c) Traffic Imputation** **(d) Temperature Imputation**

Figure 5: RMSE scores on spatial-temporal time series forecasting and imputation. The lower the better.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets. We evaluate ReTime on two real-world datasets, where the shapes of the datasets are formulated as $N \times T \times v$:

Traffic dataset is collected from Caltrans Performance Measurement System (PeMS)[1]. It contains hourly average speed and occupancy collected from 2,000 sensor stations in District 7 of California during June 2018. The size of the dataset is $2000 \times 720 \times 2$. The relation between two stations is whether they are adjacent.

Temperature is a subset of version 3 of the 20th Century Reanalysis[2] data [25]. It contains the monthly average temperature from 2001 to 2015, which covers a 30×30 area of North America, from 30° N to 60° N, 80° W to 110° W. The shape is $900 \times 180 \times 1$. The relation between two locations is whether they are adjacent.

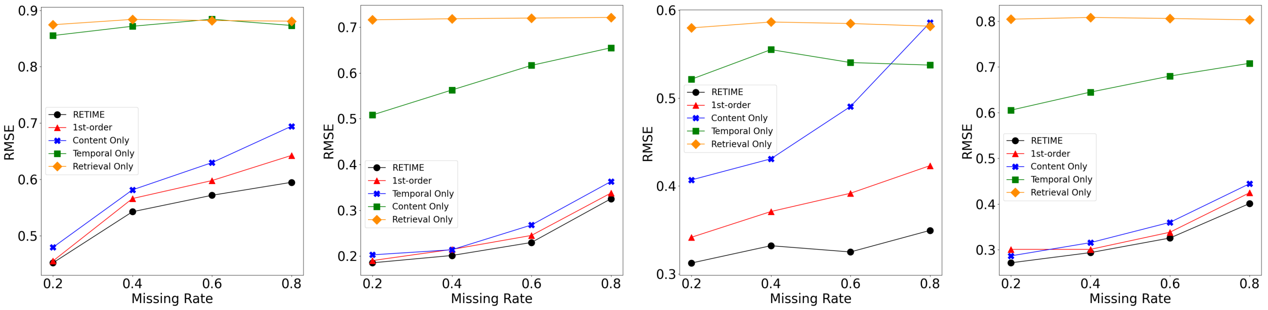
For the single time series setting, given N time series, we randomly select 10%/10% time series for validation and testing. As a result, the train/validation/test splits are 1600/200/200 and 720/90/90 for the traffic and temperature datasets. For the spatial-temporal setting, we select 10%/10% *snippets* from the entire datasets for validation/test. After splitting data, we segment each time series into 24/12-length snippets for traffic/temperature datasets respectively, which cover one day/one year. Time series are normalized according to the mean and standard deviation of the training sets.

Evaluation Tasks. We evaluate ReTime for both single time series and spatial-temporal time series, and on both forecasting and imputation tasks. For each setting, we fix the snippet length of the target and change the missing rate from 0.2 to 0.8.

Comparison Methods. We compare ReTime with the following baselines. The methods for single time series setting include block-style methods: N-BEATS [26] and Informer [1], and RNN based methods: BRITS [27], E2GAN [28]. The methods for spatial-temporal time series setting include block-style methods: StemGNN [22] and MTGNN [23], and RNN based methods: DCRNN [29], NET^3 [24]. Additionally, we also use the reference (REF) with the highest relational score to the target as a baseline.

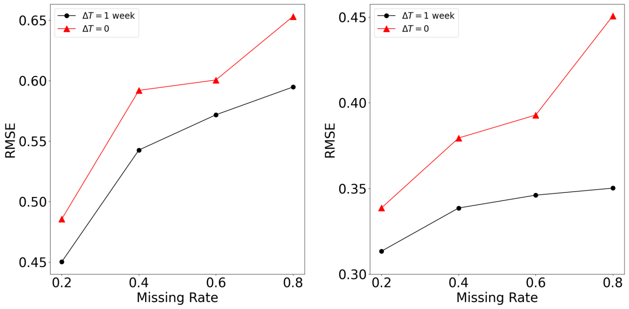
Implementation Details. The hidden dimensions for the traffic/temperature datasets are 256/64. The numbers of blocks and attention heads are 8/4 for the traffic/temperature datasets. The learning rates are tuned within [0.001, 0.0001]. Early stopping is applied on the validation set to prevent over-fitting. K is tuned within [1, 5, 10, 20]. For the relational retrieval, we set the damping factor of RWR as $c = 0.9$. Given a target time series snippet, which starts from the time T_S, we select reference snippets starting from T'_S, and denote their difference as $\Delta T = T_S - T'_S$. For forecasting, ΔT is one week/one year for the traffic/temperature datasets, which is generally the length of one cycle. For imputation, we set $\Delta T = 0$, which has the best performance in experiments. When applying forecasting models to imputation tasks, we take the entire $\mathbf{X}$ as input and force the models to predict the entire $\tilde{\mathbf{X}}$.

[1] https://dot.ca.gov/programs/traffic-operations/mpr/pems-source
[2] https://psl.noaa.gov/data/gridded/data.20thC_ReanV3.html



(a) Single Forecasting **(b) Single Imputation** **(c) Spatial-Temporal Forecasting** **(d) Spatial-Temporal Imputation**

Figure 6: Ablation study on the traffic dataset.



(a) Single Forecasting **(b) Spatial-Temporal Forecasting**

Figure 7: Effect of using prior snippets for forecasting.

5.2 Main Results

We compare ReTime with various baselines for single/spatial-temporal time series forecasting/imputation tasks for different missing rates.

Single Time Series. The results for single time series are presented in Figure 4, where the first and second rows show the Root Mean Squared Error (RMSE) for forecasting and imputation respectively. For both forecasting and imputation, compared with RNN based methods, i.e., BRITS and E2GAN, recent block-style methods, i.e., N-BEATS and Informer, have lower RMSE scores. ReTime has much lower RMSE scores than N-BEATS and Informer, demonstrating the effectiveness of the proposed strategy.

An interesting observation for the temperature dataset is that the simple retrieval baseline REF performs significantly better than other state-of-the-art baselines, corroborating the power of the reference time series. To be more specific, firstly, as indicated by the performance of the state-of-the-art methods, there might exist some complex temporal patterns for the temperature data, which could not be easily captured by models merely based on the observed content of the targets. Secondly, the superior performance of REF over the state-of-the-art methods demonstrates that the retrieved time series can indeed significantly reduce uncertainty.

Spatial-Temporal Time Series. The results for spatial-temporal time series are presented in Figure 5. The general observation is similar to the single time series that ReTime consistently performs better than the state-of-the-art methods.

5.3 Ablation Study

We study the impact of each component of ReTime on the traffic dataset, which is the largest dataset.

Impact of the Attentions. As shown in Figure 6, compared with the full model ReTime , if we only use either the content or temporal attention, the performance will drop. Besides, spatial attention alone performs worse than temporal attention alone, showing the importance of modeling temporal dependencies.

Beyond the First Order Neighbors. In Figure 6, the “1st-order” is ReTime using the 1st-order neighbors of the targets as references. ReTime is better than “1st-order”, indicating that it is important to choose appropriate neighbors as references. RWR used in the relational retrieval stage could capture global proximity scores.

Effect of the Content Synthesis Model. The “Retrieval Only” in Figure 6 uses the average of the K retrieved references as the prediction. Compared with the retrieval-only model, ReTime performs better, demonstrating the effectiveness of the content synthesis.

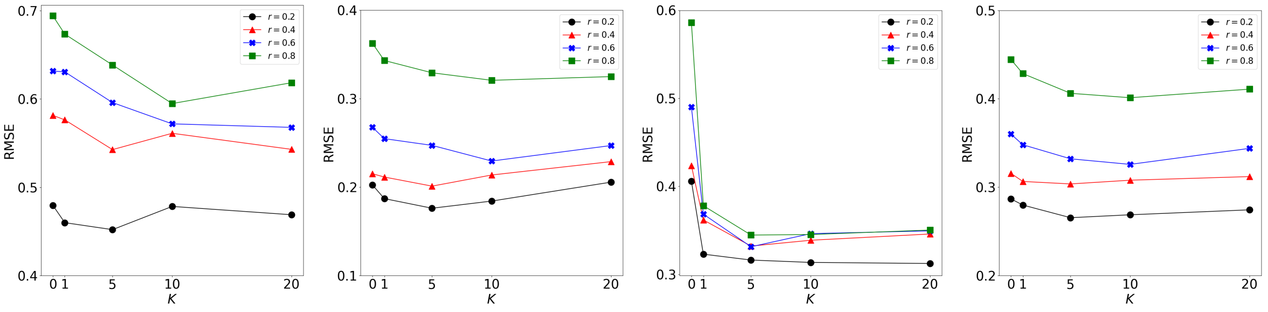
Effect of Using Prior Snippets for Forecasting. When the time series has clear periodic patterns, it is natural to resort to the historical snippets in the database for help. In Figure 7, “$\Delta T = 0$” means that the target and reference snippets have the same start time, and thus the values of both the targets and references are zeros after the separation time τ. “$\Delta T = 1$ week” means that the start time of the reference snippets is one week before the targets. Figure 7 shows that using prior snippets can significantly improve the model's performance.

5.4 Impact of K

We study the performance of ReTime and its time/memory usage w.r.t. the number of references $K \in \{0, 1, 5, 10, 20\}$. The experiments are conducted on the traffic dataset.

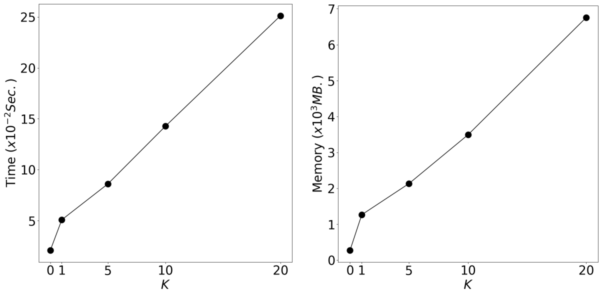
Performance of ReTime. The performance of ReTime w.r.t. the number of references K is presented in Figure 8, where r denotes the missing rate. ReTime achieves the best performance when $K \in \{5, 10\}$ Note that for spatial-temporal forecasting, the top 1 reference snippet of a target is its own historical snippet.

Time and Memory Usage. We fix the batch size as 100 and record the average training time ($\times 10^{-2}$ seconds) of each iteration and the GPU memory usage ($\times 10^{3}$ MegaBytes) for different K. The results in Figure 9 show that the training time and memory usage grow linearly w.r.t. K.



(a) Single Forecasting **(b) Single Imputation** **(c) Spatial-Temporal Forecasting** **(d) Spatial-Temporal Imputation**

Figure 8: The performance of ReTime w.r.t. different K.



(a) Time Usage **(b) Memory Usage**

Figure 9: Time and Memory Usage.

6 RELATED WORK

6.1 Single Time Series Methods

Many deep learning methods have been proposed to generate time series, including Recurrent Neural Network (RNN) based methods [27, 30] and block-style methods [1, 26]. For example, Cao et al. [27] introduce BRITS which leverages the recurrent dynamics for both correlated and uncorrelated multivariate time series. Fortuin et al. [31] combines the Gaussian process to capture the temporal dynamics and reconstruct missing values by VAE [32]. Luo et al. [33] introduce GRUI and propose a two-stage GAN [34] model, the generator and discriminator of which are based on GRUI. E2GAN [28] further simplifies the generator by combining GRUI and denoising autoencoder such that the GAN-based imputation can be trained in an end-to-end manner. Gamboa et al. [2] explore different neural networks for time series analysis. Oreshkin et al. [26] introduce N-BEATS for explainable time series forecasting. Li et al. [3] propose an enhanced version of Transformer [19] for forecasting. Zhou et al. [1] propose Informer for long time series forecasting. Wu et al. [4] introduce AutoFormer, which reduces the complexity of Transformer. However, when the missing rate grows to a high level, the performance of these methods drops significantly. ReTime address this issue by retrieving relevant reference time series as an augmentation.

6.2 Spatial-Temporal Time Series Methods

Time series often co-evolve with each other. Networks/graphs are commonly used data structures to model the relations among objects [18, 35–37], which have also been used to model relations in spatial-temporal time series. Traditional methods leverage probabilistic graphical models to introduce graph regularizations [38, 39]. Recently, Li et al. [29] introduce DCRNN combining the diffusion graph kernel with RNN. Yu et al. [40] and Zhao et al. [41] respectively propose STGCN and TGCN for modeling spatial-temporal time series. Jing et al. [24] introduce NET^3 which captures both explicit and implicit relations among time series. Cao et al. [22] introduce StemGNN which combines graph and discrete Fourier transform to jointly model spatial and temporal relations. Wu et al. [23] introduce MTGNN which learns the relation graph of time series. However, these methods do not allow models to refer to the historical snippets when forecasting future values, which perform worse than ReTime.

6.3 Retrieval Based Generation

The main idea underlying retrieval-based methods is to retrieve references from databases to guide generation. Cao et al. [42] propose Re^{3}Sum to generate document summaries based on the retrieved templates. Song et al. [43] propose to generate dialogues based on the retrieved references. Lewis et al. [44] introduce Retrieval-Augmented Generation (RAG) for knowledge-intensive natural language processing tasks e.g., summarization [45]. Tseng et al. [46] propose RetrievalGAN to generate images by retrieving relevant images. Ordonez et al. [47] generate image descriptions based on the retrieved captions. To the best of our knowledge, we present the first retrieval-based deep generation model for time series data.

7 CONCLUSION

In this paper, we theoretically quantify the uncertainty of the predicted values and prove that retrieved references could help to reduce the uncertainty for prediction results. To empirically demonstrate the effectiveness of retrieval-based forecasting, we build a simple yet effective method called ReTime, which is comprised of a relational retrieval stage and a content synthesis stage. The experimental results on real-world datasets demonstrate the effectiveness of ReTime.

REFERENCES

[1] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

[2] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[3] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *NeurIPS*, 2019.

[4] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[5] Dawei Zhou, Lecheng Zheng, Yada Zhu, Jianbo Li, and Jingrui He. Domain adaptive multi-modality neural attention network for financial forecasting. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2230–2240. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380288. URL https://doi.org/10.1145/3366423.3380288.

[6] Yada Zhu, Hanghang Tong, Baoyu Jing, JinJun Xiong, Nitin Gaur, and Anna Wanda Topol. Network of tensor time series, September 8 2022. US Patent App. 17/184,880.

[7] Ruihan Yang, Dingsu Wang, Ziyu Wang, Tianyao Chen, Junyan Jiang, and Gus Xia. Deep music analogy via latent representation disentanglement. *arXiv preprint arXiv:1906.03626*, 2019.

[8] Xin Li, Liyu Wu, and Xianfeng Yang. Exploring the impact of social economic variables on traffic safety performance in hong kong: A time series analysis. *Safety science*, 2018.

[9] Md Abdur Rashid Sarker, Khorshed Alam, and Jeff Gow. Exploring the relationship between climate change and rice yield in bangladesh: An analysis of time series data. *Agricultural Systems*, 112:11–16, 2012.

[10] Xiaokun Wang and Kara M Kockelman. Forecasting network data: Spatial interpolation of traffic counts from texas data. *Transportation research record*, 2105(1): 100–108, 2009.

[11] Ian A Nalder and Ross W Wein. Spatial interpolation of climatic normals: test of a new method in the canadian boreal forest. *Agricultural and forest meteorology*, 92(4):211–225, 1998.

[12] Philippe Bonnet, Johannes Gehrke, and Praveen Seshadri. Towards sensor database systems. In *International Conference on mobile Data management*, 2001.

[13] Tuomas Pelkonen, Scott Franklin, Justin Teller, Paul Cavallaro, Qi Huang, Justin Meza, and Kaushik Veeraraghavan. Gorilla: A fast, scalable, in-memory time series database. *Proceedings of the VLDB Endowment*, 8(12):1816–1827, 2015.

[14] Sean Rhea, Eric Wang, Edmund Wong, Ethan Atkins, and Nat Storer. Littletable: A time-series database and its uses. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 125–138, 2017.

[15] Dongjin Song, Ning Xia, Wei Cheng, Haifeng Chen, and Dacheng Tao. Deep r-th root of rank supervised joint binary embedding for multivariate time series retrieval. In *KDD*, pages 2229–2238, 2018.

[16] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[17] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[18] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622. IEEE, 2006.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[20] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[21] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *NeurIPS*, 2016.

[22] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In *NeurIPS*, 2020.

[23] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD*, 2020.

[24] Baoyu Jing, Hanghang Tong, and Yada Zhu. Network of tensor time series. In *The WebConf 2021*, pages 2425–2437, 2021.

[25] Laura C Slivinski, Gilbert P Compo, Jeffrey S Whitaker, Prashant D Sardeshmukh, Benjamin S Giese, Chesley McColl, Rob Allan, Xungang Yin, Russell Vose, Holly Titchner, et al. Towards a more reliable historical reanalysis: Improvements for version 3 of the twentieth century reanalysis system. *Quarterly Journal of the Royal Meteorological Society*, 145(724):2876–2908, 2019.

[26] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *ICLR*, 2020.

[27] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *NeurIPS*, 2018.

[28] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *IJCAI*, pages 3094–3100, 2019.

[29] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *ICLR*, 2018.

[30] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[31] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661. PMLR, 2020.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[33] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, and Xiaojie Yuan. Multivariate time series imputation with generative adversarial networks. In *NeurIPS*, 2018.

[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[35] Baoyu Jing, Chanyoung Park, and Hanghang Tong. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*, pages 2414–2424, 2021.

[36] Bolian Li, Baoyu Jing, and Hanghang Tong. Graph communal contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 1203–1213, 2022.

[37] Baoyu Jing, Yuejia Xiang, Xi Chen, Yu Chen, and Hanghang Tong. Graph-mvp: Multi-view prototypical contrastive learning for multiplex graphs. *arXiv preprint arXiv:2109.03560*, 2021.

[38] Yongjie Cai, Hanghang Tong, Wei Fan, and Ping Ji. Fast mining of a network of coevolving time series. In *SDM*. SIAM, 2015.

[39] Yongjie Cai, Hanghang Tong, Wei Fan, Ping Ji, and Qing He. Facets: Fast comprehensive mining of coevolving high-order time series. In *KDD*, 2015.

[40] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *IJCAI*, 2018.

[41] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9), 2019.

[42] Ziqiang Cao, Wenjie Li, Furu Wei, Sujian Li, et al. Retrieve, rerank and rewrite: Soft template based neural summarization. In *ACL*, 2018.

[43] Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *IJCAI*, 2018.

[44] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, 2020.

[45] Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. Multiplex graph neural network for extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, 2021.

[46] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. Retrievegan: Image synthesis via differentiable patch retrieval. In *ECCV*, 2020.

[47] Vicente Ordonez, Xufeng Han, Polina Kuznetsova, Girish Kulkarni, Margaret Mitchell, Kota Yamaguchi, Karl Stratos, Amit Goyal, Jesse Dodge, Alyssa Mensch, et al. Large scale retrieval and generation of image descriptions. In *IJCV*, 2016.