ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095054

NRTSI: NON-RECURRENT TIME SERIES IMPUTATION

Siyuan Shan, Yang Li, Junier B. Oliva

Department of Computer Science, University of North Carolina at Chapel Hill

ABSTRACT

Time series imputation is a fundamental task in understanding sequential data. Existing methods either rely on recurrent models that suffer heavily from error compounding or fail to exploit the hierarchical information of temporal data, both of which degrade performance severely with sparsely observed data. In this work, we reformulate time series as sets and propose a novel non-recurrent imputation model, **N**on-**R**ecurrent **T**ime **S**eries **I**mputation (NRTSI), that does not impose any recurrent structures. Taking advantage of the set formulation, we design a principled and efficient hierarchical imputation procedure. In addition, NRTSI can perform multiple-mode stochastic imputation, directly handle irregularly-sampled time series, and handle data with partially observed dimensions. Empirically, we show that NRTSI achieves state-of-the-art performance on multiple benchmarks.

***Index Terms*—** Time Series, Non-recurrent Models, Set Modeling, Transformer, Hierarchical Methods

1. INTRODUCTION

Missing values are common in real-world time series, e.g. trajectories often contain missing data due to unreliable sensors or object occlusion. Recovering those missing values is useful for the downstream analysis of time series. Modern approaches impute missing data in a data-driven fashion. For example, recurrent neural networks (RNNs) are applied in [1, 2, 3], methods built on Neural Ordinary Differential Equations (NODEs) [4] are proposed in [5, 6, 7], and a family of models called Neural Process [8] that learns a distribution over functions based on the observed data could also be leveraged. However, these existing works all have their own deficiencies. Models that are built on RNNs or NODEs usually employ a sequential imputation order, meaning that the imputed data x_t at timestep t is predicted based on the already imputed data x_{t-1} at the previous timestamp $t-1$. Since x_{t-1} inevitably contains errors, x_t is even more inaccurate and the errors will accumulate through time, resulting in poor long-horizon imputations for time series that are sparsely observed. This problem is known as *error compounding* in the fields of time series analysis [9, 10] and reinforcement learning [11]. Neural Process models [8] do not impose any recurrent structures. However, they impute all the missing data at once without exploiting the hierarchical information of temporal data.

In this work, we propose NRTSI, a **N**on-**R**ecurrent **T**ime **S**eries **I**mputation model. One of our key insights is that when imputing missing values in time series, the valuable information from the observed data is *what happened and when*. This information is most naturally represented as a set of (time, data) tuples. We propose a novel imputation model to encode observed data as a set of (time, data) tuples and impute the unobserved missing data. This is in stark contrast to previous works (e.g. NAOMI [12]) where observed data are embedded recurrently so that the temporal information (when things happened) is unnecessarily entangled with the order of points being processed. Our natural set representation not only disentangles the data processing order from the temporal information, but also enables us to design an hierarchical imputation strategy that is efficient and principled. To the best of our knowledge, we are the first to jointly leverage the set formulation of time series and hierarchical imputation. Without the set formulation, we have to use the inferior hierarchical algorithm in [12] due to the RNN sequential constraints; without the hierarchical formulation, we find directly using set formulation (e.g. [13, 8]) leads to much worse imputation performance. Despite its simplicity, we find that NRTSI effectively alleviates the problems of the existing methods in a single framework. Our contributions are as follows: (1) We reinterpret time series as a set of (time, data) tuples and propose a time series imputation approach NRTSI. (2) We propose an effective hierarchical imputation strategy that takes advantage of the non-recurrent nature of NRTSI and imputes data in a multiresolution fashion. (3) We show that NRTSI can flexibly handle irregularly-sampled data, data with partially observed time dimensions, and perform stochastic imputations for non-deterministic data. (4) We perform experiments on a wide range of datasets to demonstrate state-of-the-art performance of NRTSI. Codes: `https://github.com/lupalab/NRTSI`.

2. RELATED WORK

Time Series Imputation Deep generative models offer a flexible framework for imputation. Several variants of RNNs [2, 3, 14] are proposed to impute time series. Models based on NODEs [4], such as LatentODE [5], ODE-RNN [6] and NeuralCDE [7], are also proposed to impute irregularly-sampled data. Generative adversarial networks are leveraged in [15, 16]. However, all of these works are recurrent. NAOMI [12] performs time series imputation via a non-recurrent imputation procedure that imputes from coarse to fine-grained resolutions using a divide-and-conquer strategy. However, NAOMI relies on RNNs to process observed time points, which limits its application for irregularly-sampled time data and loses the opportunity to efficiently impute multiple time points in parallel.

Set Formulation of Time Series Similar to NRTSI, SeFT [13], attentive neural process (ANP) [8] and Conditional Score-based Diffusion Models (CSDI) [17] view a temporal sequence as an unordered set. SeFT has shown that this set formulation is superior to several strong recurrent baselines for time series classification. However, only time series classification is considered in SeFT and [18], while we propose a novel model that targets at time series imputation and allows effective information exchanges between observed data and missing data. Although ANP is applicable for the imputation task, the information of what timesteps to impute (target input) is not utilized when ANP uses self-attention to compute the representations of observed data (context input/output pairs). Multi-Time Attention Networks (mTAN) [19] learn an embedding of continuous time values and use an attention mechanism for interpolation and classifica-

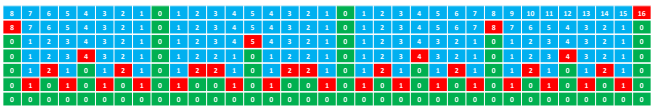


Fig. 1: Illustration of the imputation procedure. Given a time series with 2 observed values and 32 missing values, the imputation procedure starts at the first row and ends at the bottom row. Blue, green and red boxes respectively represent missing data, observed data, and data to impute next. Numbers inside each box represent the missing gap to the closest observed data and we assume the missing gap of observed data to be 0. When deciding which data to impute next, we always select the group of data with the largest missing gap.

tion. However, mTAN still contains recurrent modules (e.g. bidirenctional RNN). Besides, ANP, CSDI and mTAN do not exploit the multiresolution information of sequences, which may impact their imputation performance shown in our Experiments.

3. METHODS

Motivation To remedy the *error compounding* problem discussed in Section 1, we reinterpret time series as a set of (time, data) tuples. The set formulation allows us to conveniently develop a hierarchical scheme that reduces the number of imputation steps required compared to the sequential scheme and thus effectively alleviates the error compounding problem. It also directly enables imputing irregularly sampled time points, since the set can contain tuples for arbitrary time points. *Note that since the time information is provided in the (time, data) tuples, the sequential order of the time series is not lost, and we can easily transform the set back to a sequence.*

Formulation Throughout the paper, we denote a set as $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ with set elements $\mathbf{x}_i \in \mathcal{X}$, where $\mathcal{X}$ represents the domain of each set element. We denote a time series with N observations as a set $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^N$, where each observation $\mathbf{s}_i$ is a tuple $(t_i, \mathbf{x}_i)$, where $t_i \in \mathbb{R}^+$ denotes the observation time and $\mathbf{x}_i \in \mathbb{R}^d$ represents the observed data. Given an observed time series $\mathbf{S}$, we aim to impute the missing data based on $\mathbf{S}$. We also organize data to impute as a set $\hat{\mathbf{S}} = \{\hat{\mathbf{s}}_j\}_{j=1}^M$, where M is the number of missing time points. Each set element $\hat{\mathbf{s}}_j$ is a tuple $(\hat{t}_j, \Delta\hat{t}_j)$, where $\hat{t}_j \in \mathbb{R}^+$ is a timestep to impute and $\Delta\hat{t}_j \in \mathbb{R}^+$ denotes the missing gap (i.e. the time interval length between $\hat{t}_j$ and its closest observation time in $\mathbf{S}$). Formally, $\Delta\hat{t}_j$ is defined as $\Delta\hat{t}_j = \min_{(t_i,\mathbf{x}_i)\in\mathbf{S}} |t_i - \hat{t}_j|$. Note that both $\hat{t}_j$ and t_i can be real-valued scalars rather than fixed grid points, which enable NRTSI to handle irregularly-sampled timesteps. The missing gap $\Delta\hat{t}_j$ is essential for our hierarchical imputation procedure. As will be discussed in Sec 3, we select a subset $\mathbf{G} \subseteq \hat{\mathbf{S}}$ to impute at each hierarchy level based on the missing gap of the target time points. The imputation results are denoted as $\mathbf{H} = \{\mathbf{h}_j\}_{j=1}^{|\mathbf{G}|}$ with $\mathbf{h}_j \in \mathbb{R}^d$, where $\mathbf{H}$ is predicted using an imputation model f as $\mathbf{H} = f(\mathbf{G}; \mathbf{S})$.

Hierarchical Imputation Generative models have benefited from exploiting the hierarchical structure of data [20, 21]. Here, we propose to leverage a multi-resolution procedure for time series imputation. Specifically, we divide the missing time points into several hierarchy levels using their missing gaps (i.e. the closest distance to an observed time point). Intuitively, missing data that are far from the observed data are more difficult to impute. According to their missing gaps, we can either impute from (nearby) small gap time points to (faraway) large gap ones or vice versa. Empirically, we find starting from (faraway) large missing gaps works better (as also indicated by [12]). Given the imputed values at the current hierarchy level, the imputation at the higher hierarchy level will depend on those values. Note that the hierarchical imputation inevitably introduces some recurrent dependencies among missing time points, but since the number of hierarchy levels is typically much smaller than the number of missing time points, *the error compounding problem of NRTSI is not as severe as the sequential models.* We illustrate the imputation procedure in Fig 1 where at each hierarchy level NRTSI can impute multiple missing points in parallel thanks to the set representation of time series.

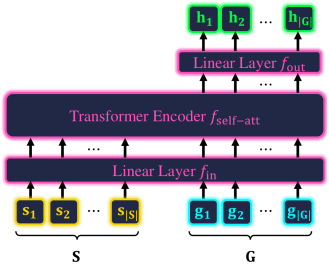


Fig. 2: Imputation model.

Imputation Model To reduce the imputation error at each level, we utilize a separate imputation model f_θ^l for each level l. The model takes in a set of known time points $\mathbf{S}$ (either observed or previously imputed at lower hierarchy levels) and imputes the values for a set of target time points $\mathbf{G}$. Theoretically, any set modeling method can be seamlessly plugged in. Representative methods include DeepSets [22], ExNODE [23], kernel methods [24, 25], and attention models [26, 27, 28, 29]. In this work, we adopt the self-attention mechanism in Transformers for its established strong capability of modeling long-range interactions.

Implementation At each hierarchy level l, a subset of missing time points $\mathbf{G}$ are first selected based on their missing gaps $\Delta\hat{t}_j$, then the imputation model f_θ^l imputes the missing values by $\mathbf{H} = f_\theta^l(\mathbf{G}; \mathbf{S})$, where $\mathbf{S} = \{(\phi(t_i), \mathbf{x}_i)\}$ and $\mathbf{G} = \{\phi(\hat{t}_j)\}$. Here $\phi : \mathbb{R} \to \mathbb{R}^\tau$ is the time encoding function proposed in [13] to provide information of time to Transformers. Note that $\Delta\hat{t}_j$ are ignored here since they are only used to define the hierarchy levels. The elements in $\mathbf{S}$ and $\mathbf{G}$ are transformed to tensors by concatenating the data $\mathbf{x} \in \mathbb{R}^d$ and the time encoding vector $\phi(t)$. Since the elements in $\mathbf{G}$ do not contain $\mathbf{x}$, we use d-dimensional zero vectors $\mathbf{0} \in \mathbb{R}^d$ as placeholders. We also add a binary scalar indicator to distinguish missing values and observed values. That is,

$$\begin{aligned} \mathbf{s}_i = (\phi(t_i), \mathbf{x}_i) \in \mathbf{S} \quad &\to \quad \mathbf{s}_i = [\phi(t_i), \mathbf{x}_i, 1] \in \mathbb{R}^{\tau+d+1}, \\ \mathbf{g}_j = \phi(\hat{t}_j) \in \mathbf{G} \quad &\to \quad \mathbf{g}_j = [\phi(\hat{t}_j), \mathbf{0}, 0] \in \mathbb{R}^{\tau+d+1}, \end{aligned} \tag{1}$$

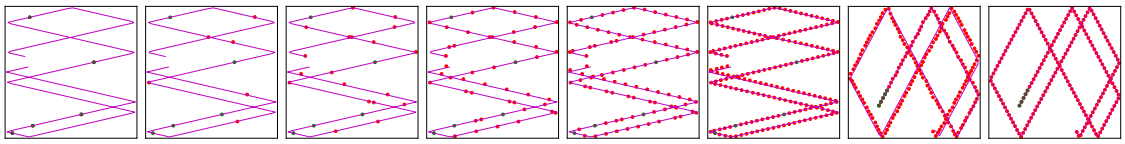
where $[\cdot]$ represents the concatenation operation. Now that the elements in $\mathbf{S}$ and $\mathbf{G}$ are all transformed to vectors with same dimensionality, we can combine them into one set and pass it through the imputation model f_θ^l, i.e. $\mathbf{H} = f_\theta^l(\mathbf{S}; \mathbf{G})$. Specifically, we implement f_θ^l by the following steps:

$$\begin{aligned} \mathbf{S}^{(1)} \cup \mathbf{G}^{(1)} &= f_{\text{in}}(\mathbf{S} \cup \mathbf{G}) \\ \mathbf{S}^{(2)} \cup \mathbf{G}^{(2)} &= f_{\text{enc}}(\mathbf{S}^{(1)} \cup \mathbf{G}^{(1)}) \\ \mathbf{H} &= f_{\text{out}}(\mathbf{G}^{(2)}). \end{aligned} \tag{2}$$

At the first step, a linear layer $f_{\text{in}} : \mathbb{R}^{\tau+d+1} \to \mathbb{R}^{d_h}$ maps the input data to a high-dimensional space in a point by point fashion. Then, a Transformer encoder $f_{\text{enc}} : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$ is used to model the interactions between $\mathbf{S}^{(1)}$ and $\mathbf{G}^{(1)}$. The Transformer encoder is composed of multiple alternating multi-head self-attention layers

Table 1: Quantitative comparison on Billiards dataset. Statistics closer to the expert indicate better performance.

Models	Linear	KNN	GRUI	MaskGAN	ANP	mTAN	CSDI	SingleRes	NAOMI	NRTSI	Expert
Sinuosity	1.121 ± 0	1.469 ± 0	1.859 ± 0	1.095 ± 0	1.364 ± 0.012	1.099 ± 0.010	1.231 ± 0.012	1.019 ± 0	1.006 ± 0	$\mathbf{1.003 \pm 0.002}$	1.000
step change ($\times 10^{-3}$)	$\mathbf{0.961 \pm 0}$	24.59 ± 0	28.19 ± 0	15.35 ± 0	18.95 ± 2.82	14.59 ± 2.17	14.92 ± 2.05	9.290 ± 0	7.239 ± 0	5.621 ± 0.752	1.588
reflection to wall	0.247 ± 0	0.189 ± 0	0.225 ± 0	0.100 ± 0	0.134 ± 0.013	0.091 ± 0.010	0.089 ± 0.011	0.038 ± 0	0.023 ± 0	$\mathbf{0.021 \pm 0.002}$	0.018
MSE ($\times 10^{-2}$)	19.00 ± 0	5.381 ± 0	20.57 ± 0	1.830 ± 0	3.762 ± 0.659	1.102 ± 0.316	1.115 ± 0.392	0.233 ± 0	0.067 ± 0	$\mathbf{0.024 \pm 0.003}$	0.000



(a) Observed (b) Gap 16 (c) Gap 8 (d) Gap 4 (e) Gap 2 (f) Gap 1 (g) NAOMI (h) NRTSI

Fig. 3: Imputation procedure on the Billiards dataset. The red points denote imputed data while the green points denote observed data. The purple solid line is the ground-truth trajectory. The initial observed data is shown in (a), the imputed data with missing gaps 16 to 1 are shown in (b)-(f). We omit the intermediate results at missing gaps 15, 7, 6, and 3 due to the limitation of space. In (g) and (h) we show the forward prediction results of NAOMI and NRTSI.

and feedforward layers, allowing the elements in $\mathbf{S}^{(1)}$ and $\mathbf{G}^{(1)}$ to effectively exchange information, i.e. $\mathbf{G}^{(1)}$ can attend to $\mathbf{S}^{(1)}$ to gather the observed information and $\mathbf{S}^{(1)}$ can be informed about what timestamps to impute by attending to $\mathbf{G}^{(1)}$. Finally, the imputation results $\mathbf{H}$ are obtained via another linear layer $f_{\text{out}} : \mathbb{R}^{d_h} \to \mathbb{R}^d$ on $\mathbf{G}^{(2)}$. The architecture of the imputation model is illustrated in Fig 2.

Training Objective We denote our imputation model with learnable parameters θ at level l as f_θ^l, which include the two linear layers and the Transformer encoder. The optimization objective is

$$\min_\theta \mathbb{E}_{\mathbf{G}\sim p(\mathbf{G}),\mathbf{S}\sim p(\mathbf{S}),\mathbf{Y}\sim p(\mathbf{Y})}\left[\frac{1}{|\mathbf{G}|}\sum_{j=1}^{|\mathbf{G}|}\mathcal{L}(\mathbf{h}_j,\mathbf{y}_j)\right], \quad (3)$$

where $\mathbf{h}_j \in \mathbf{H} = f_\theta^l(\mathbf{G};\mathbf{S})$ is an imputed data and $\mathbf{y}_j \in \mathbf{Y}$ denotes the corresponding ground truth imputation target. For deterministic datasets, we use Mean Square Error (MSE), i.e. $\mathcal{L}(\mathbf{h}_j,\mathbf{y}_j) = ||\mathbf{h}_j - \mathbf{y}_j||_2^2$. For stochastic datasets (e.g. Football Player Trajectory in Section 4), we minimize the negative log-likelihood of a Gaussian distribution with diagonal covariance, i.e.

$$\mathcal{L}(\mathbf{h}_j,\mathbf{y}_j) = -\log\mathcal{N}(\mathbf{y}_j|\mu(\mathbf{h}_j),\text{diag}(\sigma(\mathbf{h}_j))), \quad (4)$$

where $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ are two linear mappings.

Irregularly-sampled Time Series For irregularly-sampled time series, missing time points tend to have unique missing gaps. Therefore, imputing from large missing gaps to small missing gaps will reduce to an autoregressive model, which could incur a high computation demand. Therefore, we choose to impute time points with similar missing gaps together.

Stochastic Time Series For stochastic datasets, our model can impute multiple potential time series conditioned on the observed points (see Fig 4). When sampling from the distribution (4), the samples may be incongruous if we sample for all the elements in $\mathbf{G}$ simultaneously. This is because the intermediate imputations should affect the conditional distribution of later imputed steps. To solve this problem, we impute data with large missing gaps one by one. Based on the observation that missing data with small missing gaps are almost deterministic, they can be imputed simultaneously in parallel to avoid the high complexity of sampling sequentially.

Partially Observed Time Series In practice, a timestep may be only partially observed, i.e. only a subset of features is missing at that time. Our hierarchical imputation procedure can be easily extended to this scenario. We modify the hierarchical algorithm to impute the timesteps with the most missing dimensions first rather than the timesteps with the largest missing gap. We also modify the data representation to $\mathbf{s}_i = [\phi(t_i), \mathbf{x}_i, \mathbf{m}_i] \in \mathbb{R}^{d+\tau+d}$ where $\mathbf{m}_i \in \{0,1\}^d$ is a binary mask indicating which dimensions are observed.

Table 2: Traffic data MSE loss ($\times 10^{-4}$).

Linear	GRUI	KNN	MaskGAN	ANP	CSDI	mTAN	SingleRes	NAOMI	NRTSI
15.59 ± 0	15.24 ± 0	4.58 ± 0	6.02 ± 0	6.93 ± 1.91	5.89 ± 0.52	5.23 ± 0.44	4.51 ± 0	3.54 ± 0	$\mathbf{3.22 \pm 0.12}$

4. EXPERIMENTS

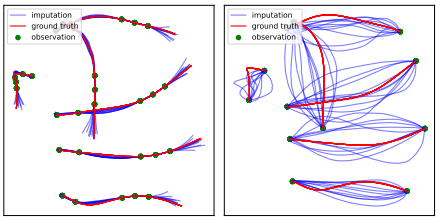
In our experiments, extensive hyperparameter searching is performed for all the baselines. For fair comparisons, we use the same training/validation/testing splits for all the methods. *During training, we follow previous works [12, 14, 5] to randomly mask out a subset of observed data and use the masked data as the ground truth imputation target to train models.* We use the same method to randomly mask out data for all the methods. Experiments conducted in this paper are repeated 5 times to compute the standard deviations.

Billiards Ball Trajectory Billiards dataset [30] contains regularly-sampled trajectories of Billiards balls in a rectangular world. Each trajectory is rolled out for 200 timesteps. We report MSE, *Sinuosity*, *step change* and *reflection to wall*, as reported in [12] to assess the realism of the imputed trajectories. We follow the setting in [12] and compare to all baselines mentioned there. We also include ANP [8], mTAN [19] and CSDI [17] as baselines. Results are reported in Table 1, where `Expert` denotes the ground truth trajectories. Following [12], we randomly select 180 to 195 timesteps as missing for each trajectory. From Table 1, we can see NRTSI reduces the L_2 loss by 64% compared to NAOMI and compares favorably to other baselines. In Fig 3, we visualize the hierarchical imputation procedure. The final imputed trajectory not only aligns well with the ground truth but also maintains a constant speed and straight lines between collisions. In Fig 3 (g), (h), we respectively show the forward prediction (predict the last 195 missing values based on the first 5 observed values) results of NAOMI and NRTSI. The trajectories predicted by NRTSI is more accurate and realistic compared to NAOMI, indicating the advantage of using a non-recurrent imputation model.

Traffic Time Series The PEMS-SF traffic [31] is a multivariate dataset with 963 dimensions at each time point, which represents the freeway occupancy rate from 963 sensors. The occupancy rate is regularly-sampled every 10 minutes throughout the day, resulting in the length of each time series being 144. Time series in this dataset is *non-stationary* as statistical properties (e.g. mean of the occupancy rate) are not constant over time. Similar to the Billiards experiment above, we train and evaluate using MSE loss. We also compare to the same set of baselines as the Billiards experiment. Following [12], we generate masked sequences with 122 to 140 missing values at random and repeat the testing set 100 times. The MSE losses are

Table 3: MuJoCo dataset MSE loss (10^{-3}) comparison.

Method	10%	20%	30%	50%
RNN GRU-D	19.68 ± 0	14.21 ± 0	11.34 ± 0	7.48 ± 0
ODE-RNN	16.47 ± 0	12.09 ± 0	9.86 ± 0	6.65 ± 0
NeuralCDE	13.52 ± 0.71	10.71 ± 0.57	8.35 ± 0.49	6.09 ± 0.41
Latent-ODE	**3.60 ± 0**	2.95 ± 0	3.00 ± 0	2.85 ± 0
ANP	7.65 ± 0.47	4.37 ± 0.38	3.21 ± 0.36	2.97 ± 0.33
CSDI	6.64 ± 0.35	3.79 ± 0.37	2.96 ± 0.31	2.62 ± 0.32
mTAN	5.90 ± 0.45	3.17 ± 0.36	2.51 ± 0.32	2.35 ± 0.28
NAOMI	4.42 ± 0.41	2.32 ± 0.35	1.46 ± 0.13	0.93 ± 0.11
NRTSI	4.06 ± 0.38	**1.22 ± 0.11**	**0.63 ± 0.09**	**0.26 ± 0.02**



(a) 5 timesteps observed (b) 2 timesteps observed

Fig. 4: Imputed trajectories of football players.

reported in Table 2.

MuJoCo Physics Simulation MuJoCo is a physical simulation dataset created by [5] using the "Hopper" model from the Deepmind Control Suite [32]. Initial positions and velocities of the hopper are randomly sampled such that the hopper rotates in the air and falls on the ground. The dataset is 14-dimensional. MSE loss is used to train and evaluate NRTSI. Baseline models include Latent-ODE [5], ODE-RNN [5], GRU-D [2], NeuralCDE [7], ANP [8], mTAN [19], CSDI [17] and NAOMI [12]. We report the MSEs with different observation rates in Table 3. NRTSI compares favorably to all baselines with 20%, 30% and 50% observed data. When only 10% data are observed, NRTSI is comparable to Latent-ODE and NAOMI.

Football Player Trajectory This dataset is from the NFL Big Data Bowl 2021 [33], which contains the 2D trajectories of 6 offensive players and is therefore 12-dimensional. During training and testing, we treat all players in a trajectory equally and randomly permute their orders. Every time series contain 50 regularly-sampled time points. This dataset is stochastic since there could be many possible trajectories based on the sparsely observed data. Therefore, we follow [34] to use *min*MSE to evaluate the precision and the ratio between *avg*MSE and *min*MSE to evaluate the diversity of multiple imputed trajectories. Similar to [12], we also use *average trajectory length* and *step change* to assess the quality of imputation. For this dataset, we minimize the negative log-likelihood as in (4). For each trajectory, we randomly select 40 to 49 timesteps as missing. According to the discussion in Sec 3, data with missing gaps larger than 4 are imputed one by one, while data with smaller missing gaps are imputed in parallel. We compare to baselines such as Latent-ODE, NAOMI, CSDI and ANP that can impute stochastically. As shown in Table 4, NRTSI compares favorably to the baselines.

Irregularly-sampled Time Series We evaluate NRTSI on an irregularly-sampled Billiards dataset. The only difference between this dataset and the regularly-sampled Billiards dataset is that this dataset is irregularly-sampled. We compare NRTSI with two representative ODE-based approaches that can deal with irregularly-sampled data, i.e. Latent-ODE and NeuralCDE. We also modify NAOMI to handle irregularly-sampled data, which we call NAOMI-Δ_t. The time gap information between observations is provided to the RNN update function of NAOMI-Δ_t. We also compare to ANP, mTAN and CSDI which can handle irregularly-sampled data. According to Table 5, NRTSI outperforms the baselines by a large margin despite extensive hyperparameter search for these baselines. To investigate the poor performance of Latent-ODE, NeuralCDE, and ANP, we visualize their imputed trajectories with different numbers of observed data and find that when the observation is dense (150 points observed), they all perform well. However, they have difficulty predicting the correct trajectories when the observation becomes sparse (e.g. with only 5 points observed). The excellent performance of NRTSI and NAOMI-Δ_t indicates the benefits of the multiresolution imputation procedure. Furthermore, the superiority of NRTSI over NAOMI-Δ_t demonstrates the advantage of the proposed set modeling approach.

Table 4: Quantitative comparison on Football Player Trajectory. A larger *avg*MSE / *min*MSE indicate better diversity. Other statistics closer to the expert indicate better performance.

Models	Latent-ODE	NAOMI	CSDI	ANP	NRTSI	Expert
step change ($\times 10^{-3}$)	1.473±0.154	3.227±0.216	1.543±0.297	1.754±0.211	**2.401±0.087**	2.482
avg length	0.136±0.009	0.236±0.008	0.201±0.009	0.145±0.007	**0.175±0.004**	0.173
***min*MSE** ($\times 10^{-3}$)	19.53±1.44	4.079±0.487	8.142±1.005	6.652±0.881	**1.908±0.101**	0.000
***avg*MSE / *min*MSE**	1.16±0.09	1.12±0.07	1.53±0.07	1.19±0.10	**2.13± 0.08**	—

Table 5: Irregularly-sampled Billiards data L_2 loss ($\times 10^{-2}$).

Latent-ODE	NeuralCDE	ANP	mTAN	CSDI	NAOMI-Δ_t	NRTSI
19.48±1.64	34.01±1.99	29.31 ±1.53	3.542±0.447	3.823±0.521	1.121±0.265	**0.042±0.008**

Table 6: The MSE comparison under different missing rates.

Dataset	Method	missing rate 20%	40%	60%	80%
Air	Latent-ODE	.2954±.0109	.3291±.0118	.3569±.0124	.3762±.0127
	NeuralCDE	.3129±.0271	.3524±.0285	.4074±.0290	.4865±.0319
	BRITS	.2076 ±.0000	.2088 ±.0000	.2660 ±.0000	.3421 ±.0000
	RDIS	.1807±.0000	.1977±.0000	.2528±.0000	.3178±.0000
	CSDI	.1236±.0032	.1411±.0041	.1648±.0044	.2155±.0057
	mTAN	.1192±.0034	.1261±.0033	.1403±.0046	.1885±.0049
	NRTSI	**.1155±.0035**	**.1250±.0038**	**.1378±.0039**	**.1790±.0041**
Gas	Latent-ODE	.1282±.0039	.1299±.0041	.1387±.0044	.1979±.0049
	NeuralCDE	.0773±.0024	.1044±.0028	.1538±.0045	.3011±.0097
	BRITS	.0226 ±.0000	.0279 ±.0000	.0406 ±.0000	.1595 ±.0000
	RDIS	.0226 ±.0000	.0251 ±.0000	.0321 ±.0000	.0837 ±.0000
	CSDI	.0297±.0009	.0273±.0009	.0352±.0011	.0591±.0017
	mTAN	.0215±.0007	.0259±.0008	.0497±.0012	.0886±.0016
	NRTSI	**.0195±.0007**	**.0229±.0007**	**.0311±.0010**	**.0445±.0012**

Partially Observed Time Series The air quality dataset [35] and the gas sensor dataset [36] are used to evaluate the partially observed scenario. Data in these datasets are 11 and 19-dimensional respectively. For both datasets, we follow RDIS [14] to select 48 consecutive timesteps to construct one regularly-sampled time series. We compare NRTSI to RDIS, BRITS, Latent-ODE, NeuralCDE, CSDI and mTAN. In Table 6, we report the MSEs by randomly masking out some dimensions for all timesteps with different missing rates. NRTSI outperforms the baselines on all of the missing rates.

5. CONCLUSION AND DISCUSSION

In this work, we introduce a novel time-series imputation approach named NRTSI. NRTSI represents time series as a set and leverages a Transformer-based architecture to impute the missing values. We also propose a hierarchical imputation procedure where missing data are imputed in the order of their missing gaps. NRTSI is broadly applicable to numerous applications, such as irregularly-sampled time series, partially observed time series, and stochastic time series. Extensive experiments demonstrate that NRTSI achieves state-of-the-art performance on commonly used imputation benchmarks. Throughout the experiments, we conduct an extensive hyperparameter searching for the baselines to make sure they perform as well as they can be. We find that the best configurations of these baselines are not improved by increasing their model capacities. Thus, the superiority of NRTSI is due to the novel architecture rather than naively using more parameters.

Acknowledgements This research was partly funded by NSF grant IIS2133595 and by NIH 1R01AA02687901A1.

6. REFERENCES

[1] Yonghong Luo, Xiangrui Cai, Ying ZHANG, Jun Xu, and Yuan xiaojie, "Multivariate time series imputation with generative adversarial networks," in *NeurIPS*. 2018.

[2] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, 2018.

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

[4] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud, "Neural ordinary differential equations," in *NeurIPS*, 2018.

[5] Yulia Rubanova, Ricky TQ Chen, and David Duvenaud, "Latent odes for irregularly-sampled time series," *NeurIPS*, 2019.

[6] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau, "Gru-ode-bayes: Continuous modeling of sporadically-observed time series," in *NeurIPS*, 2019.

[7] Patrick Kidger, James Morrill, James Foster, and Terry Lyons, "Neural controlled differential equations for irregular time series," *NeurIPS*, 2020.

[8] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh, "Attentive neural processes," in *ICLR*, 2018.

[9] Arun Venkatraman, Martial Hebert, and J Bagnell, "Improving multi-step prediction of learned time series models," in *AAAI*, 2015.

[10] Yan Xu, Siyuan Shan, Ziming Qiu, Zhipeng Jia, Zhengyang Shen, Yipei Wang, Mengfei Shi, and Eric I-Chao Chang, "End-to-end subtitle detection and recognition for videos in east asian languages via cnn ensemble," *Signal Processing: Image Communication*, vol. 60, pp. 131–143, 2018.

[11] Kavosh Asadi, Dipendra Misra, and Michael Littman, "Lipschitz continuity in model-based reinforcement learning," in *ICML*, 2018.

[12] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue, "Naomi: Non-autoregressive multiresolution sequence imputation," in *NeurIPS*, 2019.

[13] Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt, "Set functions for time series," in *ICML*, 2020.

[14] Tae-Min Choi, Ji-Su Kang, and Jong-Hwan Kim, "Rdis: Random drop imputation with self-training for incomplete time series data," *AAAI*, 2021.

[15] Jinsung Yoon, James Jordon, and Mihaela Schaar, "Gain: Missing data imputation using generative adversarial nets," in *ICML*, 2018, pp. 5689–5698.

[16] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al., "Multivariate time series imputation with generative adversarial networks," in *NeurIPS*, 2018.

[17] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," *NeurIPS*, vol. 34, 2021.

[18] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan, "Self-attention with functional time representation learning," in *NeurIPS*, 2019.

[19] Satya Narayan Shukla and Benjamin Marlin, "Multi-time attention networks for irregularly sampled time series," in *ICLR*, 2021.

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *ICLR*, 2018.

[21] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer, "Generating wikipedia by summarizing long sequences," in *ICLR*, 2018.

[22] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola, "Deep sets," in *NIPS*, 2017.

[23] Yang Li, Haidong Yi, Christopher Bender, Siyuan Shan, and Junier B Oliva, "Exchangeable neural ode for set modeling," *NeurIPS*, 2020.

[24] Siyuan Shan, Vishal Athreya Baskaran, Haidong Yi, Jolene Ranek, Natalie Stanley, and Junier B Oliva, "Transparent single-cell set classification with kernel mean embeddings," in *ACM-BCB*, 2022, pp. 1–10.

[25] Vishal Athreya Baskaran, Jolene Ranek, Siyuan Shan, Natalie Stanley, and Junier B Oliva, "Distribution-based sketching of single-cell samples," in *ACM-BCB*, 2022, pp. 1–10.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[27] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *ICML*, 2019.

[28] Siyuan Shan, Yang Li, and Junier B Oliva, "Meta-neighborhoods," in *NeurIPS*, 2020.

[29] Siyuan Shan, Lamtharn Hantrakul, Jitong Chen, Matt Avent, and David Trevelyan, "Differentiable wavetable synthesis," in *ICASSP 2022*, 2022, pp. 4598–4602.

[30] Tim Salimans and Durk P Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *NIPS*, 2016, pp. 901–909.

[31] Dheeru Dua and Casey Graff, "UCI machine learning repository," 2017.

[32] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al., "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.

[33] Kaggle, "Nfl big data bowl 2021," `https://www.kaggle.com/c/nfl-big-data-bowl-2021`.

[34] Seong Hyeon Park, Gyubok Lee, Manoj Bhat, Jimin Seo, Minseok Kang, Jonathan Francis, Ashwin R Jadhav, Paul Pu Liang, and Louis-Philippe Morency, "Diverse and admissible trajectory forecasting through multimodal context understanding," *ECCV*, 2020.

[35] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen, "Cautionary tales on air-quality improvement in beijing," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2017.

[36] Javier Burgués, Juan Manuel Jiménez-Soto, and Santiago Marco, "Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models," *Analytica chimica acta*, vol. 1013, pp. 13–25, 2018.