

Unveiling the Secrets: How Masking Strategies Shape Time Series Imputation

Linglong Qian^{1,2}, Zina Ibrahim^{1†*}, Wenjie Du², Yiyuan Yang^{2,3}, Richard JB Dobson^{1,4,5}

¹King’s College London, ²PyPOTS Research, ³University of Oxford

⁴University College London, ⁵Health Data Research UK

{linglong.qian, zina.ibrahim, richard.j.dobson}@kcl.ac.uk, wdu@pypots.com, yiyuan.yang@cs.ox.ac.uk

Abstract

In this study, we explore the impact of different masking strategies on time series imputation models. We evaluate the effects of pre-masking versus in-mini-batch masking, normalization timing, and the choice between augmenting and overlaying artificial missingness. Using three diverse datasets, we benchmark eleven imputation models with different missing rates. Our results demonstrate that masking strategies significantly influence imputation accuracy, revealing that more knowledge-driven and data-driven masking designs are essential for robust model evaluation. We advocate for refined experimental designs and comprehensive disclosure to better simulate real-world patterns, enhancing the practical applicability of imputation models. The source code is available at <https://github.com/LinglongQian/Mask.rethinking>.

1 Introduction

The evaluation of deep learning models designed to perform imputation tasks for time-series datasets heavily relies on the practice of masking, where certain data points are deliberately designated as missing to simulate incomplete data conditions. Masking provides a controlled way to test how an algorithm handles incomplete datasets and is therefore an essential component of performance evaluation. Currently, random masking is the predominant technique used in the literature, as noted across various studies [Wang *et al.*, 2024]. This method primarily generates Missing Completely at Random (MCAR) scenarios. However, real-world data, especially in domains like biomedicine, often exhibit more complex missingness patterns that cannot be adequately represented by MCAR. For instance, Electronic Health Records (EHRs) frequently show Missing Not at Random (MNAR) patterns due to systematic biases in data collection processes [García-Laencina *et al.*, 2010].

Additionally, the design decisions regarding masking timing, whether applied before training a deep imputation model (pre-masking) or dynamically during the training phase (in-mini-batch masking), significantly influence imputation per-

formance. Pre-masking, while straightforward, limits the model’s exposure to the full range of clinical features and their dependencies. Conversely, in-mini-batch masking introduces artificial missingness iteratively, which can lead to overfitting if the model becomes too focused on these patterns, potentially neglecting the original data structure. Simultaneously, the timing of data normalization, whether performed before or after masking, significantly impacts the consistency between experimental outcomes and the a priori hypothesis.

This paper aims to demonstrate that notable performance discrepancies depend on the timing of masking and the method employed. We evaluate various deep imputation models using PyPOTS¹, a unified interface that provides standardised masking functionalities. Our results highlight the need for more sophisticated and data-driven masking strategies to ensure robust model evaluation. We show that different masking techniques can lead to varying levels of imputation accuracy, underscoring the importance of aligning experimental designs with the real-world conditions the models are intended to address. This study emphasizes the need to refine masking techniques in deep imputation research to better align theoretical model capabilities with practical performance in addressing complex missing data patterns. Our goal is to underscore the importance of masking in model evaluation, urging researchers to pay closer attention to experimental design, especially in simulating data for algorithm assessment.

2 Related Work

Time series imputation is a well-studied field with a substantial collection of statistical and machine learning techniques. Recently, neural network-based imputation models designed for large and heterogeneous time-series datasets have outperformed traditional methods across various disciplines [Wang *et al.*, 2024]. As shown in Table 1, modern deep imputation models use a variety of neural architectures ranging from convolutional neural networks (CNNs), recurrent neural networks (RNNs), to multi-layer perceptrons (MLPs) and use neural frameworks such as Variational Autoencoders (VAEs) and diffusion models [Yang *et al.*, 2024]. There is a number of literature reviews of deep imputation models [Liu *et al.*,

^{†*}Corresponding author.

¹<https://pypots.com>

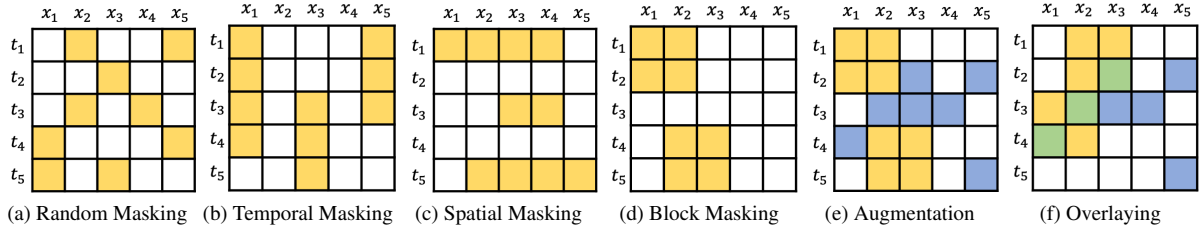


Figure 1: Masking techniques and approaches demonstrated over a time-series of five features ($x_1 \sim x_5$) and five time points ($t_1 \sim t_5$): (a) random masking, (b) temporal masking, (c) spatial masking, (d) block masking. The yellow cells indicate those labeled as missing via masking. In (e) augmentation and (f) overlaying, the blue cells indicate cells that are missing within the original data. In (e), the masked (yellow) cells have no overlap with the original missingness in the data. Green: masked data coming from both the original missingness and artificial missingness. In (f), overlaying masks cells from either the original missingness or simulates artificial missingness from non-missing data.

2023; Wang *et al.*, 2024] and all existing methods use masking to simulate missingness for algorithm evaluation and for comparison with the state of the art.

When we examined the masking techniques employed during experimental evaluation of the state of the art deep imputation models, we uncovered a significant gap between the theoretical capabilities of the imputation methods and the evaluation mechanisms of the respective algorithms. As shown in Table 1, imputation models are designed with different flavours of missingness in mind, MCAR, MNAR or missing at random (MAR). During experimental evaluation, however, all listed models use random masking (i.e., randomly selecting a subset of the dataset to mask) to generate missing datasets, predominantly producing MCAR scenarios.

The literature contains several interesting masking techniques that can capture the spatio-temporal MNAR missingness patterns of medical datasets, those include temporal masking (Figure 1 (b)), which captures missingness patterns over time, spatial masking (Figure 1 (c)), which captures cross-sectional missingness and block masking (Figure 1 (d)), which combines the two to concurrently capture different flavours of temporal and cross-sectional correlations and dependencies. Despite their direct applicability to biomedical domains, the only examples which use more sophisticated masking techniques we found the literature come from the traffic domain [Liang *et al.*, 2022; Ye *et al.*, 2021].

Besides, the problem is exacerbated by the lack of information in published work. Except for BRITS [Cao *et al.*, 2018] and CSDI [Tashiro *et al.*, 2021], the use of random masking is rarely mentioned in experimental design, requiring examination of the accompanying code to discern it. While random masking facilitates model evaluation, it contrasts with the complex MNAR patterns in many real-world datasets [García-Laencina *et al.*, 2010], undermining deep imputers’ capacity and leaving them under-evaluated. Standardization tools like PyPOTS [Du, 2023] offer unified masking functionalities, urging a shift to sophisticated, data-driven masking designs. Moreover, significant discrepancies exist in reporting when masking is introduced during experimental evaluation. Data can be pre-masked before model ingestion or

dynamically masked during training. Pre-masking methods limit the model’s exposure to incomplete datasets, reducing its ability to learn from the entire range of features. In contrast, in-mini-batch masking iteratively masks different subsets during training, offering a dynamic approach but risking overfitting by focusing on artificial patterns rather than original data structures. This aspect is largely overlooked in most deep imputers, except BRITS and GRUD, which mask before training, and CSDI and STAIRS, which use in-mini-batch masking. The methodology used to implement masking is also crucial. Moreover, masking can be implemented using overlaying [Du, 2023] or augmenting [Choi *et al.*, 2023]. Overlaying adds artificial missingness to the original dataset, exposing the model to a broader array of scenarios but increasing overfitting risk and evaluation complexity. Augmenting keeps artificial missingness separate, simplifying learning but potentially failing to equip the model for real-world data patterns. The specific masking implementation in deep imputers is often unclear, creating a significant gap in understanding the rigour of evaluation techniques for models addressing non-random missingness.

3 Experimental Design

In this section, we delineate the methodologies adopted to scrutinize the impact of different masking strategies, the timing of artificial missingness introduction, and the timing of data normalization on the performance of deep imputation models. The experimental design is meticulously structured to investigate these variables and their interactions comprehensively.

3.1 Artificial Missingness Strategies

To evaluate the effects of different artificial missingness strategies, we implemented two primary approaches:

- **Augmenting: In Random Masking on Existing Observations (RMEO)** (Figure 1 (e)), artificial missingness is introduced by randomly selecting a subset of the observed data points and marking them as missing. This method respects the existing missingness pattern and augments it with additional artificial missing values.

Model	Year	Architecture	Missing Mechanisms
MRNN [Yoon <i>et al.</i> , 2017]	2017	RNN	MAR
GRUD [Che <i>et al.</i> , 2018]	2018	RNN	MCAR, MNAR
BRITS [Cao <i>et al.</i> , 2018]	2018	RNN	MAR
Tiled CNN [Wang and Oates, 2015]	2015	CNN	MAR
GLIMA [Suo <i>et al.</i> , 2020]	2020	Attention	MCAR, MAR
MTSIT [Yildiz <i>et al.</i> , 2022]	2022	Attention	MCAR, MAR
SAITS [Du <i>et al.</i> , 2023]	2023	Attention	MCAR
TSL-GNN [Gordon <i>et al.</i> , 2021]	2021	GNN	MCAR
MIWAE [Mattei and Frellsen, 2019]	2019	CNN-VAE	MCAR, MAR
GP-VAE [Fortuin <i>et al.</i> , 2020]	2020	CNN-VAE	MCAR, MAR, MNAR
V-RIN [Mulyadi <i>et al.</i> , 2021]	2020	RNN-VAE	MCAR, MAR
HI-VAE [Nazabal <i>et al.</i> , 2020]	2020	MLP-VAE	MCAR
Shi-VAE [Barrejón <i>et al.</i> , 2021]	2022	RNN-VAE	MAR
supnot-MIWAE [Kim <i>et al.</i> , 2023]	2023	CNN, Attention-VAE	MNAR
CDNet [Liu <i>et al.</i> , 2022]	2022	RNN-MDN	-
VIGAN [Shang <i>et al.</i> , 2017]	2017	CNN-GAN	-
GRUI-GAN [Luo <i>et al.</i> , 2018]	2018	RNN-GAN	-
E^2 GAN [Luo <i>et al.</i> , 2019]	2019	RNN-GAN	-
SSGAN [Miao <i>et al.</i> , 2021]	2021	RNN-GAN	MCAR
Sim-GAN [Pati <i>et al.</i> , 2022]	2022	CNN-GAN	-
CSDI [Tashiro <i>et al.</i> , 2021]	2021	Attention-Diffusion	MCAR, MAR, MNAR
SSSD [Alcaraz and Strodthoff, 2022]	2023	CNN-Diffusion	MCAR, MAR, MNAR
CSBI [Chen <i>et al.</i> , 2023]	2023	CNN, Attention-Diffusion	MAR
DA-TASWDM [Xu <i>et al.</i> , 2023]	2023	Attention-Diffusion	MAR
CRU [Schirmer <i>et al.</i> , 2022]	2022	RNN-Neural ODE	MAR
CSDE [Park <i>et al.</i> , 2021]	2022	MLP-Neural ODE	-

Table 1: The overview of deep learning-based time-series imputation models.

The rationale behind this approach is to simulate scenarios where additional missing data is likely to occur in an already incomplete dataset, reflecting real-world conditions in certain applications. Augmenting allows the model to learn from the artificially introduced missingness without interference from the original missing patterns. However, it may not fully equip the model to handle the intricate missingness patterns found in real-world data.

- **Overlaying:** In **Random Masking on Overall Data (RMOD)** (Figure 1 (f)), artificial missingness is applied uniformly across the entire dataset, irrespective of the original missingness pattern. This means that some artificially introduced missing values may overlap with naturally occurring ones. The rationale for this method is to create a more comprehensive test environment by treating all data points as equally likely candidates for missingness, thereby exploring the algorithm’s capability to handle missing data more broadly. Overlaying exposes the model to a broader array of missing data scenarios, potentially leading to more robust training and effective imputation strategies. However, this method requires complex evaluation processes and increases the risk of overfitting.

3.2 Timing of Artificial Missingness Introduction

We explored two distinct timings for introducing artificial missingness to assess their effects on model performance:

- **Pre-Masking:** artificial missingness is introduced before the data is fed into the model for training. Once the dataset is masked, it remains static throughout the training process. The rationale for this method is to ensure a consistent missingness pattern throughout training, providing a stable environment for the model to learn imputation strategies. The potential benefits of this approach

include the consistency of encountering the same missingness pattern, which facilitates a controlled evaluation of the model’s imputation performance. Additionally, it simplifies the experimental setup by maintaining a single masked dataset throughout training. However, the drawbacks include limited exposure to variability in missingness patterns, potentially reducing the model’s generalizability to real-world scenarios, and the introduction of bias in the model’s learning process, leading to suboptimal performance on new, unseen missingness patterns.

- **In-Mini-Batch Masking:** introduces artificial missingness dynamically during training, with different portions of the data being masked in each mini-batch. This method mimics a more realistic scenario where missingness patterns can vary, exposing the model to a broader range of missing data conditions. The potential benefits of this method include acting as a form of data augmentation, which can enhance the model’s robustness by exposing it to various missingness patterns, and more accurately reflecting real-world conditions where missing data can occur randomly and unpredictably. However, the potential drawbacks include the risk of overfitting, as the model may become overly tuned to the specific patterns of artificial missingness introduced during training, and added complexity to the training process, which requires more sophisticated handling of dynamic missingness patterns and may lead to inconsistencies between training and evaluation phases.

3.3 Normalization Procedures

The sequence of data normalization relative to masking is another critical factor that we examined:

- **Normalization Before Masking (NBM):** data normalization is performed on the complete dataset before any artificial missingness is introduced. This approach assumes access to the full data distribution, which may lead to more stable normalization parameters. However, it does not reflect the practical situation where normalization is based on incomplete data.
- **Normalization After Masking (NAM):** data normalization is conducted after the artificial missingness is introduced. This method mirrors real-world scenarios more closely, where normalization must be performed on incomplete data. While it better simulates practical conditions, it may lead to inconsistencies between training and test data distributions, affecting performance evaluation.

3.4 Masking Rates

To comprehensively understand the impact of masking rates, we experimented with different proportions of artificially introduced missing values: Low Masking Rate (5%), Medium Masking Rate (10%), and High Masking Rate (20%). Introducing missingness to 5% of the data points simulates scenarios with minimal additional missing data. This setup allows us to observe model performance under conditions with low levels of missingness, providing insight into how well the model can handle slight data loss. Introducing missingness

to 10% of the data points creates a balanced condition that is neither too sparse nor too dense in terms of missingness. This intermediate level helps in understanding the model’s performance under moderate data loss, which is often encountered in real-world applications. Introducing missingness to 20% of the data points challenges the model’s imputation capabilities under substantial missingness. This high masking rate allows us to evaluate how well the model performs when faced with extensive data loss, providing a stringent test of its robustness and effectiveness.

4 Experimental Setup

In this section, we describe the datasets used, the models evaluated, the implementation details, and the metrics employed to evaluate the performance of various imputation strategies.

4.1 Datasets

We used three diverse datasets to benchmark the performance of our imputation models, ensuring a comprehensive evaluation across different domains. The first dataset, the **Beijing Multi-Site Air-Quality Data**, provides hourly data on air pollutants from 12 locations, which we sourced from the UCI repository to align with open benchmarking practices [Zhang *et al.*, 2017]. This dataset captures environmental data critical for assessing air quality trends and health impacts. The second dataset, **PhysioNet 2012**, is a public medical dataset containing records of 12,000 48-hour ICU stays [Silva *et al.*, 2012]. This dataset includes a variety of physiological signals and clinical variables, making it essential for testing imputation methods in the healthcare domain. The third dataset, **ETTm1**, consists of two years of data from two separated counties in China and is a crucial indicator in the electric power long-term deployment [Zhou *et al.*, 2021]. This dataset captures vital statistics for power consumption and environmental variables. Table 3 summarizes the characteristics of these datasets, selected to demonstrate performance across diverse characteristics, application scenarios, and domains.

The characteristics of these datasets are summarized in Table 3. These datasets are selected to demonstrate performance on datasets with diverse characteristics, application scenarios, and domains.

4.2 Models Evaluated

We evaluated eleven imputation methods, including three naive approaches and eight state-of-the-art deep learning models. The naive methods, **Mean**, **Median**, and **Last Observation Carried Forward (LOCF)**, serve as baselines. The eight representative deep-learning models selected for experimental studies span various neural architectures and methodologies. These include **M-RNN** [Yoon *et al.*, 2017], a recurrent neural network-based model; **GP-VAE** [Fortuin *et al.*, 2020], which combines Gaussian processes with variational autoencoders; **BRITS** [Cao *et al.*, 2018], a bidirectional recurrent model specifically designed for missing value imputation; **USGAN** [Miao *et al.*, 2021], which uses generative adversarial networks; **CSDI** [Tashiro *et al.*, 2021], a conditional score-based diffusion model for imputation; **TimesNet** [Wu *et al.*, 2022], a neural network tailored for time

series; **Transformer** [Du *et al.*, 2023], utilizing the transformer architecture for time series imputation and **SAITS** [Du *et al.*, 2023]. Experiments were performed with PyPOTS [Du, 2023], a unified interface providing standardized masking functionalities.

4.3 Implementation Details

The experiments were implemented using PyPOTS², ensuring standardized masking functionalities and reproducibility. All experiments were conducted on a machine equipped with an NVIDIA A100 GPU, 80GB of RAM. Each model was trained with its recommended hyperparameters, tuned through 5-fold cross-validation. All code, including the data preprocessing scripts, model configurations, and training scripts, are publicly available in the GitHub repository. This ensures transparency and allows for reproducibility of our experiments by the research community.

4.4 Evaluation Metrics

The performance of the imputation models was evaluated using two metrics to provide a comprehensive assessment. Mean Absolute Error (MAE) was computed to determine the average of the absolute differences between imputed values and actual values, with lower MAE signifying more accurate imputation. Mean Squared Error (MSE) was used to measure the average of the squared differences between imputed values and actual values, with lower MSE indicating better imputation performance. Additionally, we evaluated the model size and training time to assess the computational efficiency of each method. Model size was measured in terms of the number of parameters, and training time was recorded to provide insight into the computational resources required for training each model. These metrics collectively ensured a robust evaluation of imputation accuracy, computational efficiency, and effectiveness across different datasets and models.

5 Analysis of Results

The results of the imputation experiments using different methods on the ETTm1, Air Quality, and PhysioNet 2012 datasets are presented in Tables 4, 2, 5, and 6. These tables report the Mean Absolute Error (MAE), Mean Squared Error (MSE), and training time for each imputation method across various masking strategies and conditions. The overall conclusion aligns with expectations: different datasets and different models respond differently to various setting changes, and there is no unified trend. Here, we delve deeper into the findings and implications from multiple perspectives: datasets, algorithms, and settings.

5.1 Analysis by Datasets

For the **ETTm1** dataset, performance fluctuations were observed across various imputation models and settings. **TimesNet** consistently achieved the lowest MAE and MSE, demonstrating robustness across different masking strategies. This dataset, characterized by its regular temporal patterns and long-term dependencies, benefits from models that can effectively capture such patterns. **BRITS** and **Transformer** also

²<https://github.com/WenjieDu/PyPOTS>

Table 2: Performances with different imputation methods on Physionet 2012 dataset.

	Size	Augmentation Mini-Batch Mask NBM			Augmentation Pre-Mask NBM			Augmentation Pre-Mask NAM		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.211±0.003	0.268±0.004	0.0282	0.267±0.002	0.287±0.001	0.0127	0.267±0.007	0.290±0.001	0.0144
Transformer	13M	0.222±0.001	0.274±0.003	0.0242	0.283±0.002	0.301±0.005	0.0077	0.283±0.002	0.303±0.003	0.0069
TimesNet	44.3M	0.289±0.014	0.330±0.019	0.0080	0.288±0.002	0.278±0.007	0.0053	0.290±0.002	0.279±0.005	0.0051
CSDI	0.3M	0.239±0.012	0.759±0.517	1.2005	0.237±0.006	0.302±0.057	0.9102	0.241±0.017	0.430±0.151	2.3030
GPVAE	2.5M	0.425±0.011	0.511±0.017	0.0249	0.399±0.002	0.402±0.004	0.0338	0.396±0.001	0.401±0.004	0.0398
USGAN	0.9M	0.298±0.003	0.327±0.005	0.4154	0.294±0.003	0.261±0.004	0.3880	0.293±0.002	0.261±0.003	0.4125
BRITS	1.3M	0.263±0.003	0.342±0.001	0.1512	0.257±0.001	0.256±0.001	0.1384	0.257±0.001	0.258±0.001	0.1519
MRNN	0.07M	0.685±0.002	0.935±0.001	0.0165	0.688±0.001	0.899±0.001	0.0163	0.690±0.001	0.901±0.001	0.0161
LOCF	/	0.411±5.551	0.613±0.0	/	0.404±0.0	0.506±0.0	/	0.404±0.0	0.507±0.0	/
Median	/	0.690±0.0	1.049±0.0	/	0.690±0.0	1.019±0.0	/	0.691±0.0	1.022±0.0	/
Mean	/	0.707±0.0	1.022±0.0	/	0.706±0.0	0.976±0.0	/	0.706±0.0	0.979±1.110	/

	Size	Overlay Mini-Batch Mask NBM			Overlay Pre-Mask NBM			Overlay Pre-Mask NAM		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.206±0.002	0.227±0.005	0.0274	0.274±0.006	0.326±0.005	0.0134	0.271±0.006	0.325±0.004	0.0164
Transformer	13M	0.219±0.004	0.222±0.007	0.0234	0.289±0.002	0.336±0.002	0.0077	0.290±0.002	0.336±0.003	0.0079
TimesNet	44.3M	0.273±0.011	0.242±0.018	0.0138	0.293±0.003	0.290±0.011	0.0054	0.291±0.003	0.288±0.007	0.0051
CSDI	0.3M	0.226±0.010	0.279±0.051	2.4022	0.253±0.005	0.461±0.074	0.8606	0.239±0.006	0.344±0.109	0.9562
GPVAE	2.5M	0.427±0.006	0.453±0.008	0.0149	0.412±0.007	0.484±0.013	0.0323	0.420±0.009	0.489±0.008	0.0235
USGAN	0.9M	0.295±0.002	0.261±0.007	0.3517	0.297±0.002	0.284±0.005	0.4158	0.299±0.003	0.287±0.004	0.4182
BRITS	1.3M	0.254±0.001	0.265±0.001	0.1334	0.262±0.000	0.288±0.003	0.1428	0.263±0.001	0.294±0.003	0.1461
MRNN	0.07M	0.682±0.000	0.905±0.001	0.0168	0.685±0.001	0.926±0.002	0.0162	0.684±0.001	0.923±0.002	0.0167
LOCF	/	0.411±0.0	0.532±0.0	/	0.408±0.0	0.540±0.0	/	0.409±0.0	0.540±0.0	/
Median	/	0.687±0.0	1.019±0.0	/	0.686±0.0	1.030±0.0	/	0.686±0.0	1.030±0.0	/
Mean	/	0.705±0.0	0.990±1.110	/	0.702±0.0	1.001±0.0	/	0.702±0.0	1.000±0.0	/

Table 3: Characteristics of the three datasets.

Dataset	Size	Time Window	Number of Features	Missing Rate
Air Quality	1458	24	132	1.60%
Physionet 2012	4000	48	37	80%
ETTM1	722	96	7	0%

performed well, likely due to their ability to model sequential dependencies. **CSDI** showed high variability, particularly in dynamic settings, indicating sensitivity to the introduction of artificial missingness during training. These fluctuations can be attributed to the complexity of the time series data and the inherent difficulty in accurately imputing long gaps. Models like **GPVAE** and **MRNN** struggled, highlighting their limitations in handling the intricate temporal dynamics of the ETTm1 dataset.

In the **Air Quality** dataset, **CSDI** emerged as the top performer, achieving the lowest MAE and MSE in most settings. The dataset’s spatial-temporal characteristics, with measurements from multiple locations over time, may favor **CSDI**’s approach to modeling distributions. **SAITS** and **BRITS** also demonstrated strong performance, whereas **Transformer** showed subpar results compared to **BRITS**. This variability can be linked to the dataset’s sensitivity to the masking strategy employed. The differences in performance under various settings suggest that models like **CSDI** and **SAITS** can better adapt to the spatial-temporal dependencies present in the air quality data. The higher error rates observed in models like **GPVAE** and **MRNN** reflect their inability to effectively handle these complex dependencies, which are crucial for accurate imputation in this dataset.

For the **PhysioNet 2012** dataset, **SAITS** achieved the best

overall performance, particularly excelling under the **Overlay Mini-Batch Mask NBM** setting. This dataset, with its high-dimensional medical data, challenges models to accurately capture the intricate relationships between various physiological signals. The strong performance of **SAITS** and **CSDI** can be attributed to their sophisticated modeling techniques that effectively handle such complexity. **Transformer** also showed strong results under **Mini-Batch Mask NBM**, indicating its capability in dealing with dynamic missing data patterns. The moderate performance of **USGAN**, **BRITS**, and **GPVAE** highlights their relative effectiveness, while **MRNN** consistently underperformed, emphasizing its limitations in handling high-dimensional and complex medical data.

5.2 Analysis by Algorithms

Among the evaluated algorithms, **SAITS** showed its strength particularly in the **PhysioNet 2012** dataset and under specific settings like **Mini-Batch Mask NBM**, where it achieved the best results. However, in other datasets and settings, its performance was not as strong, often being outperformed by **BRITS**. This suggests that while **SAITS** has potential in specific high-dimensional and complex data scenarios, it may not be the best all-around performer across diverse datasets. Additionally, **SAITS** has a relatively large model size (43.6M parameters) and moderate training time, which should be considered when evaluating its practical deployment.

BRITS consistently exhibited strong performance across all datasets and settings, particularly excelling in the **ETTM1** and **Air Quality** datasets. Its robustness to various masking strategies and missing rates highlights its sophisticated approach to modeling sequential dependencies, making it a

Table 4: Performances with different imputation methods on ETTm1 and Air dataset.

ETTm1													
	Size	Augmentation Mini-Batch Mask			Augmentation Pre-Mask			Overlay Mini-Batch Mask			Overlay Pre-Mask		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.164±0.019	0.054±0.014	0.007	0.169±0.018	0.058±0.012	0.0056	0.165±0.013	0.052±0.008	0.0057	0.164±0.015	0.054±0.012	0.0075
Transformer	13M	0.154±0.018	0.043±0.008	0.0036	0.129±0.004	0.036±0.002	0.0049	0.132±0.009	0.034±0.004	0.0049	0.136±0.003	0.039±0.002	0.0042
TimesNet	44.3M	0.119±0.008	0.030±0.004	0.0034	0.110±0.002	0.026±0.001	0.0016	0.110±0.005	0.026±0.003	0.0039	0.109±0.001	0.026±0.001	0.0019
CSDI	0.3M	0.507±0.436	5.487±7.678	0.0390	0.136±0.006	0.048±0.003	0.0420	0.203±0.049	0.509±0.569	0.0378	0.145±0.022	0.062±0.039	0.0386
GPVAE	2.5M	0.279±0.011	0.165±0.010	0.0079	0.275±0.012	0.159±0.013	0.0066	0.293±0.012	0.177±0.009	0.0069	0.279±0.009	0.163±0.009	0.0057
USGAN	0.9M	0.140±0.003	0.052±0.004	0.2961	0.150±0.005	0.057±0.002	0.2139	0.142±0.003	0.049±0.002	0.2074	0.148±0.007	0.057±0.006	0.2592
BRITS	1.3M	0.134±0.001	0.054±0.001	0.0615	0.140±0.012	0.058±0.008	0.0567	0.127±0.003	0.048±0.002	0.0582	0.130±0.006	0.050±0.005	0.0637
MRNN	0.07M	0.730±0.089	1.307±0.216	0.0063	0.616±0.032	1.038±0.039	0.0096	0.640±0.049	1.072±0.088	0.0079	0.590±0.021	0.993±0.030	0.0113
LOCF	/	0.138±0.0	0.077±0.0	/	0.138±0.0	0.077±0.0	/	0.135±0.0	0.072±0.0	/	0.135±0.0	0.072±0.0	/
Median	/	0.655±0.0	0.824±0.0	/	0.655±0.0	0.824±0.0	/	0.657±0.0	0.825±0.0	/	0.657±0.0	0.825±0.0	/
Mean	/	0.661±0.0	0.807±0.0	/	0.661±0.0	0.807±0.0	/	0.663±0.0	0.809±0.0	/	0.663±0.0	0.809±0.0	/

Air													
	Size	Augmentation Mini-Batch Mask			Augmentation Pre-Mask			Overlay Mini-Batch Mask			Overlay Pre-Mask		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.147±0.003	0.119±0.004	0.0384	0.152±0.002	0.155±0.003	0.0343	0.145±0.002	0.106±0.002	0.0391	0.155±0.001	0.176±0.003	0.0290
Transformer	13M	0.159±0.003	0.130±0.003	0.0208	0.171±0.001	0.179±0.001	0.0099	0.160±0.005	0.123±0.007	0.0167	0.172±0.001	0.204±0.003	0.0094
TimesNet	44.3M	0.163±0.004	0.165±0.003	0.0066	0.166±0.006	0.218±0.009	0.0073	0.156±0.002	0.151±0.004	0.0089	0.167±0.002	0.248±0.005	0.0069
CSDI	0.3M	0.108±0.004	0.182±0.049	0.4620	0.106±0.005	0.223±0.077	0.4521	0.103±0.004	0.124±0.034	0.4886	0.111±0.005	0.240±0.018	0.4591
GPVAE	2.5M	0.282±0.008	0.247±0.013	0.0062	0.276±0.019	0.267±0.024	0.0053	0.296±0.008	0.256±0.011	0.0036	0.277±0.011	0.311±0.023	0.0051
USGAN	0.9M	0.174±0.003	0.125±0.004	0.1403	0.149±0.001	0.125±0.007	0.1335	0.172±0.003	0.111±0.004	0.1051	0.153±0.001	0.150±0.002	0.1155
BRITS	1.3M	0.143±0.000	0.115±0.001	0.2274	0.142±0.000	0.131±0.002	0.1878	0.142±0.000	0.105±0.001	0.1934	0.144±0.000	0.163±0.002	0.1883
MRNN	0.07M	0.524±0.001	0.624±0.003	0.0299	0.518±0.001	0.658±0.001	0.0321	0.523±0.001	0.618±0.002	0.0261	0.522±0.001	0.693±0.001	0.0349
LOCF	/	0.206±2.776	0.244±2.776	/	0.205±0.0	0.345±0.0	/	0.206±2.776	0.267±0.0	/	0.209±2.776	0.376±0.0	/
Median	/	0.668±0.0	1.018±0.0	/	0.663±0.0	1.058±0.0	/	0.660±0.0	1.001±0.0	/	0.666±0.0	1.090±0.0	/
Mean	/	0.697±0.0	0.954±0.0	/	0.695±0.0	0.997±1.110	/	0.692±0.0	0.943±0.0	/	0.697±0.0	1.031±0.0	/

reliable and versatile choice for time series imputation tasks. **BRITS** also benefits from a smaller model size (1.3M parameters) and reasonable training times, enhancing its practical utility. **TimesNet** also showed robust performance, especially in the **ETTm1** dataset, maintaining low MAE and MSE across different settings. Its effectiveness in capturing long-term dependencies makes it particularly well-suited for datasets with regular temporal patterns. Despite its large model size (44.3M parameters), **TimesNet** demonstrated relatively quick training times, particularly under Augmentation Pre-Mask settings, which makes it an efficient option for time-sensitive applications.

CSDI exhibited high variability but excelled in the **Air Quality** dataset, maintaining robustness with increased missing rates. Its sophisticated modeling of underlying data distributions makes it effective, though it requires longer training times and has a moderate model size (0.3M parameters). **USGAN** performed reasonably well, showing benefits from pre-masking strategies and demonstrating moderate robustness across datasets. However, its longer training times may limit its practicality in time-sensitive applications.

GPVAE and **MRNN** showed higher error rates and struggled with the complexity of the datasets, highlighting their limitations in effectively handling intricate temporal dynamics and high-dimensional data. **GPVAE** has a moderate model size (2.5M parameters) but fails to deliver competitive performance. **MRNN**, despite its small model size (0.07M parameters), consistently underperformed, indicating that its simplicity is inadequate for the evaluated datasets. Naive methods (**LOCF**, **Median**, **Mean**) consistently showed the highest error rates across all datasets and settings. Their simplicity fails to capture complex data patterns, reaffirming the necessity for more advanced imputation techniques. The consistently poor performance of naive methods underscores their limitations and the need for more sophisticated

approaches to handle missing data in time series.

5.3 Analysis by Settings

The choice between **augmentation** and **overlay** masking strategies had a significant impact on model performance. Augmentation masking, both in **pre-mask** and **mini-batch** settings, generally showed better performance in handling static missing patterns. For instance, models like **Transformer** and **TimesNet** exhibited strong results under **augmentation** masking, benefiting from the consistent exposure to missingness during training, which helped in learning more robust representations. In contrast, **overlay** masking, particularly in dynamic mini-batch settings, introduced higher variability in model performance. This variability is likely due to the intermittent introduction of missingness, which can confuse models and lead to less stable learning. **CSDI**, which performed exceptionally well in some settings, was particularly affected by overlay masking, suggesting that its sophisticated modeling of underlying distributions might be disrupted by inconsistent missing patterns introduced during training.

The choice between **pre-mask** and **mini-batch mask** strategies had a significant impact on model performance. **Pre-masking** involves introducing missing data before the training process, allowing the model to learn with a consistent set of missing data throughout the training. This method generally led to stable performance for models like **BRITS** and **TimesNet**, which were able to effectively adapt to the fixed missing patterns. On the other hand, **mini-batch masking**, which introduces missing data dynamically during training, showed varied results. **SAITS**, for instance, performed exceptionally well with mini-batch masking, particularly under the NBM setting, as this approach likely provided a form of data augmentation that improved the model's robustness. However, other models like **CSDI** showed sensitivity to this dynamic approach, indicating that the constant change in

missing data patterns during training could disrupt the learning process for certain models.

The impact of normalization timing whether applied before or after masking on model performance was found to be minimal for many models, as observed in the results from the **PhysioNet 2012** dataset. However, the sophisticated model **CSDI** exhibited variable performance depending on the normalization strategy employed. Specifically, while pre-normalization, which scales the entire dataset uniformly before masking, often provides stability and consistency, **CSDI** sometimes performed better with post-normalization. This latter approach, which normalizes data after masking and thus reflects real-world scenarios where models encounter missing values in their original scale, can enhance the robustness of models capable of adapting to diverse data distributions. For instance, **CSDI** achieved superior results in certain settings with pre-normalization but excelled in other configurations with post-normalization, indicating its sensitivity to the timing of normalization and its ability to leverage the data characteristics presented by each strategy. This analysis suggests that while the timing of normalization may have negligible effects on simpler models, it can significantly influence the performance of more complex models like **CSDI**, emphasizing the importance of aligning preprocessing strategies with model capabilities and dataset intricacies.

The impact of different missing rates on model performance was profound. Increased missing rates (20%) generally degraded the performance of most models, highlighting their sensitivity to the extent of missing data. However, some models demonstrated robustness under higher missing rates. For example, **TimesNet** and **CSDI** managed to maintain relatively lower MAE and MSE despite the increased missingness, indicating their strong capacity to handle substantial data gaps. In contrast, models like **GPVAE** and **MRNN** suffered significant performance drops, indicating their inability to effectively impute data when a large portion of it is missing. Naive methods were the most affected by higher missing rates, consistently showing the highest error rates.

6 Discussion and Conclusion

In this study, we explored the impact of various preprocessing and masking strategies on the performance of deep learning models for time series imputation. Our findings indicate that while many models show minimal performance differences between pre-normalization and post-normalization, sophisticated models like **CSDI** exhibit significant variability depending on these strategies. Pre-normalization generally provides stability and consistency, whereas post-normalization better simulates real-world scenarios by normalizing data after masking, enhancing robustness. Additionally, the choice between pre-mask and mini-batch mask strategies affects model outcomes, with pre-masking offering stable performance through consistent missing patterns and mini-batch masking providing data augmentation benefits but introducing variability. Overlaying masking typically resulted in better performance for certain models by capturing realistic missingness. Overall, this study underscores the necessity for careful selection of preprocessing and masking techniques

tailored to the dataset and model characteristics, advocating for detailed and transparent experimental designs to ensure robust and reliable imputation outcomes in practice.

References

- [Alcaraz and Strodthoff, 2022] Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2022.
- [Barrejón *et al.*, 2021] Daniel Barrejón, Pablo M Olmos, and Antonio Artés-Rodríguez. Medical data wrangling with sequential variational autoencoders. *IEEE Journal of Biomedical and Health Informatics*, 26(6):2737–2745, 2021.
- [Cao *et al.*, 2018] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.
- [Che *et al.*, 2018] 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.
- [Chen *et al.*, 2023] Yu Chen, Wei Deng, Shikai Fang, Fengpei Li, Nicole Tianjiao Yang, Yikai Zhang, Kashif Rasul, Shandian Zhe, Anderson Schneider, and Yuriy Nevmyvaka. Provably convergent schrödinger bridge with applications to probabilistic time series imputation. In *International Conference on Machine Learning*, pages 4485–4513. PMLR, 2023.
- [Choi *et al.*, 2023] Tae-Min Choi, Ji-Su Kang, and Jong-Hwan Kim. Rdis: Random drop imputation with self-training for incomplete time series data. *IEEE Access*, 2023.
- [Du *et al.*, 2023] Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023.
- [Du, 2023] Wenjie Du. Pypots: A python toolbox for data mining on partially-observed time series. *arXiv preprint arXiv:2305.18811*, 2023.
- [Fortuin *et al.*, 2020] 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.
- [García-Laencina *et al.*, 2010] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.
- [Gordon *et al.*, 2021] David Gordon, Panayiotis Petousis, Henry Zheng, Davina Zamanzadeh, and Alex AT Bui. Tsig-nn: Extending graph neural networks to handle missing data in temporal settings. *Frontiers in big Data*, 4:693869, 2021.
- [Kim *et al.*, 2023] SeungHyun Kim, Hyunsu Kim, Eunggu Yun, Hwangrae Lee, Jaehun Lee, and Juho Lee. Probabilistic imputation for time-series classification with missing data. In *International Conference on Machine Learning*, pages 16654–16667. PMLR, 2023.
- [Liang *et al.*, 2022] Yuebing Liang, Zhan Zhao, and Lijun Sun. Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns. *Transportation Research Part C: Emerging Technologies*, 143:103826, 2022.
- [Liu *et al.*, 2022] Yuxi Liu, Shaowen Qin, Zhenhao Zhang, and Wei Shao. Compound density networks for risk prediction using electronic health records. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1078–1085. IEEE, 2022.
- [Liu *et al.*, 2023] Mingxuan Liu, Siqi Li, Han Yuan, Marcus Eng Hock Ong, Yilin Ning, Feng Xie, Seyed Ehsan Safari, Yuqing Shang, Victor Volovici, Bibhas Chakraborty, and Nan Liu. Handling missing values in healthcare data: A systematic review of deep learning-based imputation techniques. *Artificial Intelligence in Medicine*, 142:102587, 2023.
- [Luo *et al.*, 2018] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.
- [Luo *et al.*, 2019] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the 28th international joint conference on artificial intelligence*, pages 3094–3100. AAAI Press Palo Alto, CA, USA, 2019.
- [Mattei and Frellsen, 2019] Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*, pages 4413–4423. PMLR, 2019.
- [Miao *et al.*, 2021] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8983–8991, 2021.
- [Mulyadi *et al.*, 2021] Ahmad Wisnu Mulyadi, Eunji Jun, and Heung-Il Suk. Uncertainty-aware variational-recurrent imputation network for clinical time series. *IEEE Transactions on Cybernetics*, 52(9):9684–9694, 2021.
- [Nazabal *et al.*, 2020] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.
- [Park *et al.*, 2021] Sung Woo Park, Kyungjae Lee, and Jun-seok Kwon. Neural markov controlled sde: Stochastic optimization for continuous-time data. In *International Conference on Learning Representations*, 2021.
- [Pati *et al.*, 2022] Soumen Kumar Pati, Manan Kumar Gupta, Rinita Shai, Ayan Banerjee, and Arijit Ghosh. Missing value estimation of microarray data using simgan. *Knowledge and Information Systems*, 64(10):2661–2687, 2022.
- [Schirmer *et al.*, 2022] Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular

- time series with continuous recurrent units. In *International Conference on Machine Learning*, pages 19388–19405. PMLR, 2022.
- [Shang *et al.*, 2017] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. Vigan: Missing view imputation with generative adversarial networks. In *2017 IEEE International conference on big data (Big Data)*, pages 766–775. IEEE, 2017.
- [Silva *et al.*, 2012] Ikaro Silva, George Moody, Roger Mark, and Leo Anthony Celi. Predicting mortality of icu patients: The physionet/computing in cardiology challenge 2012. *Predicting Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge*, p. v1, 2012.
- [Suo *et al.*, 2020] Qiuling Suo, Weida Zhong, Guangxu Xun, Jianhui Sun, Changyou Chen, and Aidong Zhang. Glima: Global and local time series imputation with multi-directional attention learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 798–807. IEEE, 2020.
- [Tashiro *et al.*, 2021] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.
- [Wang and Oates, 2015] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3939–3945, 2015.
- [Wang *et al.*, 2024] Jun Wang, Wenjie Du, Wei Cao, Keli Zhang, Wenjia Wang, Yuxuan Liang, and Qingsong Wen. Deep learning for multivariate time series imputation: A survey. *arXiv preprint arXiv:2402.04059*, 2024.
- [Wu *et al.*, 2022] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.
- [Xu *et al.*, 2023] Jingwen Xu, Fei Lyu, and Pong C Yuen. Density-aware temporal attentive step-wise diffusion model for medical time series imputation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2836–2845, 2023.
- [Yang *et al.*, 2024] Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, et al. A survey on diffusion models for time series and spatio-temporal data. *arXiv preprint arXiv:2404.18886*, 2024.
- [Ye *et al.*, 2021] Yongchao Ye, Shiyao Zhang, and James J. Q. Yu. Spatial-temporal traffic data imputation via graph attention convolutional network. In Igor Farkas, Paolo Masulli, Sebastian Otte, and Stefan Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 241–252, 2021.
- [Yıldız *et al.*, 2022] A Yarkin Yıldız, Emirhan Koç, and Aykut Koç. Multivariate time series imputation with transformers. *IEEE Signal Processing Letters*, 29:2517–2521, 2022.
- [Yoon *et al.*, 2017] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Multi-directional recurrent neural networks: A novel method for estimating missing data. In *Time series workshop in international conference on machine learning*, 2017.
- [Zhang *et al.*, 2017] 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*, 473(2205):20170457, 2017.
- [Zhou *et al.*, 2021] 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 *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

A Performance on different missing rate

Table 5 and table 6 show the performance of 11 methods on ETTm1 and air datasets under different missing rates.

Table 5: Performances with different imputation methods on ETTm1 dataset with 5% and 20% missing rate.

	Size	Augmentation Mini-Batch Mask (5%)			Augmentation Mini-Batch Mask (20%)			Augmentation Pre-Mask (5%)			Augmentation Pre-Mask (20%)		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.150±0.011	0.043±0.006	0.0078	0.163±0.008	0.052±0.005	0.0069	0.170±0.020	0.059±0.013	0.0068	0.156±0.002	0.050±0.004	0.0079
Transformer	13M	0.123±0.004	0.029±0.002	0.0040	0.141±0.009	0.039±0.003	0.0052	0.136±0.003	0.038±0.001	0.0040	0.145±0.006	0.041±0.003	0.0045
TimesNet	44.3M	0.118±0.008	0.029±0.003	0.0028	0.122±0.005	0.031±0.002	0.0035	0.106±0.002	0.024±0.001	0.0017	0.115±0.000	0.029±0.000	0.0031
CSDI	0.3M	0.255±0.123	1.222±1.262	0.0382	0.166±0.032	0.128±0.098	0.0512	0.158±0.010	0.061±0.005	0.0312	0.132±0.015	0.044±0.009	0.0428
GPVAE	2.5M	0.282±0.014	0.167±0.011	0.0073	0.298±0.009	0.190±0.011	0.0078	0.274±0.009	0.156±0.007	0.0082	0.302±0.023	0.195±0.021	0.0068
USGAN	0.9M	0.140±0.005	0.049±0.004	0.2884	0.164±0.005	0.069±0.002	0.2364	0.139±0.008	0.050±0.004	0.2883	0.159±0.007	0.065±0.005	0.1611
BRITS	1.3M	0.129±0.009	0.049±0.006	0.0718	0.143±0.005	0.062±0.003	0.0702	0.129±0.006	0.051±0.005	0.0693	0.144±0.007	0.062±0.005	0.0603
MRNN	0.07M	0.709±0.036	1.196±0.094	0.0074	0.706±0.092	1.249±0.217	0.0076	0.627±0.028	1.045±0.065	0.0107	0.585±0.010	0.982±0.015	0.0103
LOCF	/	0.135±0.000	0.071±0.000	/	0.143±0.000	0.083±0.000	/	0.135±0.000	0.071±0.000	/	0.143±0.000	0.083±0.000	/
Median	/	0.652±0.000	0.822±0.000	/	0.653±0.000	0.821±1.110	/	0.652±0.000	0.822±0.000	/	0.653±0.000	0.821±1.110	/
Mean	/	0.657±0.000	0.804±0.000	/	0.658±0.000	0.801±0.000	/	0.657±0.000	0.804±0.000	/	0.658±0.000	0.801±0.000	/
	Size	Overlay Mini-Batch Mask (5%)			Overlay Mini-Batch Mask (20%)			Overlay Pre-Mask (5%)			Overlay Pre-Mask (20%)		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.141±0.009	0.039±0.005	0.0069	0.154±0.011	0.048±0.006	0.0087	0.163±0.019	0.051±0.010	0.0069	0.166±0.016	0.054±0.011	0.0068
Transformer	13M	0.122±0.006	0.029±0.002	0.0044	0.145±0.012	0.041±0.006	0.0087	0.138±0.010	0.038±0.004	0.0048	0.137±0.007	0.039±0.004	0.0046
TimesNet	44.3M	0.117±0.005	0.028±0.002	0.0016	0.122±0.006	0.032±0.003	0.0037	0.102±0.001	0.023±0.001	0.0016	0.115±0.001	0.029±0.000	0.0027
CSDI	0.3M	0.149±0.021	0.145±0.124	0.0436	0.258±0.092	1.123±1.271	0.0470	0.154±0.029	0.057±0.023	0.0403	0.133±0.011	0.042±0.007	0.0421
GPVAE	2.5M	0.270±0.005	0.152±0.004	0.0078	0.295±0.007	0.185±0.008	0.0081	0.262±0.007	0.145±0.010	0.0091	0.289±0.004	0.178±0.007	0.0077
USGAN	0.9M	0.141±0.009	0.051±0.005	0.2467	0.162±0.010	0.066±0.006	0.2538	0.142±0.005	0.051±0.004	0.3209	0.159±0.006	0.063±0.004	0.2540
BRITS	1.3M	0.125±0.006	0.046±0.004	0.0631	0.139±0.003	0.056±0.003	0.0836	0.126±0.005	0.047±0.003	0.0687	0.141±0.003	0.057±0.002	0.0837
MRNN	0.07M	0.702±0.035	1.209±0.109	0.0073	0.720±0.086	1.271±0.200	0.0070	0.658±0.024	1.090±0.071	0.0118	0.653±0.116	1.121±0.265	0.0092
LOCF	/	0.133±0.000	0.072±0.000	/	0.142±0.000	0.079±0.000	/	0.133±0.000	0.072±0.000	/	0.142±0.000	0.079±0.000	/
Median	/	0.667±0.000	0.862±0.000	/	0.653±0.000	0.815±1.110	/	0.667±0.000	0.862±0.000	/	0.653±0.000	0.815±1.110	/
Mean	/	0.673±0.000	0.844±1.110	/	0.659±0.000	0.799±0.000	/	0.673±0.000	0.844±1.110	/	0.659±0.000	0.799±0.000	/

Table 6: Performances with different imputation methods on Air dataset with 5% and 20% missing rate.

	Size	Augmentation Mini-Batch Mask (5%)			Augmentation Mini-Batch Mask (20%)			Augmentation Pre-Mask (5%)			Augmentation Pre-Mask (20%)		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.146±0.003	0.094±0.003	0.0403	0.151±0.001	0.179±0.001	0.0458	0.148±0.001	0.100±0.001	0.0352	0.160±0.002	0.144±0.003	0.0314
Transformer	13M	0.161±0.003	0.109±0.005	0.0191	0.167±0.005	0.203±0.009	0.0181	0.166±0.003	0.119±0.003	0.0088	0.182±0.002	0.170±0.003	0.0096
TimesNet	44.3M	0.167±0.003	0.144±0.006	0.0052	0.163±0.001	0.234±0.003	0.0094	0.169±0.007	0.153±0.011	0.0070	0.173±0.002	0.218±0.004	0.0031
CSDI	0.3M	0.102±0.003	0.091±0.016	0.4681	0.108±0.004	0.184±0.019	0.5944	0.105±0.008	0.169±0.084	0.6129	0.106±0.003	0.146±0.015	0.4911
GPVAE	2.5M	0.281±0.010	0.221±0.012	0.0060	0.310±0.009	0.335±0.018	0.0061	0.282±0.006	0.226±0.006	0.0062	0.287±0.015	0.266±0.017	0.0061
USGAN	0.9M	0.165±0.013	0.102±0.010	0.1288	0.208±0.001	0.195±0.003	0.1169	0.140±0.001	0.086±0.003	0.1856	0.164±0.001	0.130±0.003	0.1469
BRITS	1.3M	0.138±0.001	0.091±0.001	0.2324	0.155±0.001	0.165±0.001	0.2315	0.137±0.000	0.095±0.000	0.2341	0.151±0.000	0.132±0.001	0.2344
MRNN	0.07M	0.523±0.001	0.602±0.002	0.0319	0.530±0.002	0.706±0.003	0.0292	0.516±0.001	0.592±0.001	0.0383	0.521±0.001	0.634±0.001	0.0366
LOCF	/	0.201±0.000	0.198±0.000	/	0.221±0.000	0.334±0.000	/	0.198±0.000	0.215±2.776	/	0.219±0.000	0.347±0.000	/
Median	/	0.670±0.000	1.004±0.000	/	0.667±0.000	1.094±0.000	/	0.665±0.000	0.998±1.110	/	0.663±0.000	1.028±0.000	/
Mean	/	0.697±0.000	0.935±0.000	/	0.697±0.000	1.030±0.000	/	0.695±0.000	0.935±0.000	/	0.694±0.000	0.968±0.000	/
	Size	Overlay Mini-Batch Mask (5%)			Overlay Mini-Batch Mask (20%)			Overlay Pre-Mask (5%)			Overlay Pre-Mask (20%)		
		MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)	MAE ↓	MSE ↓	Time (h)
SAITS	43.6M	0.142±0.003	0.098±0.003	0.0454	0.151±0.003	0.112±0.003	0.0418	0.151±0.000	0.217±0.002	0.0370	0.162±0.002	0.157±0.001	0.0309
Transformer	13M	0.160±0.004	0.120±0.006	0.0173	0.168±0.008	0.131±0.009	0.0172	0.169±0.002	0.253±0.004	0.0095	0.184±0.001	0.187±0.002	0.0091
TimesNet	44.3M	0.154±0.003	0.122±0.003	0.0109	0.163±0.004	0.156±0.004	0.0098	0.171±0.002	0.323±0.016	0.0084	0.174±0.003	0.258±0.010	0.0049
CSDI	0.3M	0.105±0.010	0.191±0.138	0.4735	0.111±0.002	0.157±0.060	0.4919	0.123±0.024	0.540±0.383	0.5646	0.111±0.004	0.199±0.039	0.5123
GPVAE	2.5M	0.275±0.007	0.221±0.011	0.0059	0.307±0.005	0.263±0.006	0.0051	0.282±0.015	0.353±0.021	0.0066	0.285±0.017	0.285±0.025	0.0064
USGAN	0.9M	0.155±0.003	0.100±0.002	0.1398	0.207±0.002	0.134±0.002	0.1053	0.144±0.003	0.178±0.004	0.1574	0.167±0.001	0.142±0.002	0.1421
BRITS	1.3M	0.137±0.001	0.095±0.001	0.2338	0.153±0.000	0.111±0.001	0.2414	0.139±0.001	0.185±0.004	0.2418	0.153±0.000	0.150±0.000	0.2327
MRNN	0.07M	0.520±0.002	0.605±0.002	0.0312	0.527±0.001	0.621±0.002	0.0314	0.528±0.001	0.766±0.001	0.0380	0.522±0.001	0.658±0.001	0.0374
LOCF	/	0.198±0.000	0.211±0.000	/	0.220±2.776	0.280±0.0	/	0.204±0.000	0.403±0.000	/	0.222±2.776	0.337±0.0	/
Median	/	0.660±0.000	0.990±1.110	/	0.662±0.000	1.002±0.000	/	0.670±0.000	1.169±0.000	/	0.664±0.000	1.052±0.000	/
Mean	/	0.692±0.000	0.931±0.000	/	0.694±0.000	0.944±0.000	/	0.702±0.000	1.109±0.000	/	0.695±0.000	0.993±0.000	/