

Spatiotemporal Transformer for Imputing Sparse Data: Application to Soil Moiture Imputation

Kehui Yao^{a,1,*}, Jingyi Huang^b, Jun Zhu^a

^a*Department of Statistics, University of Wisconsin-Madison,*

^b*Department of Soil Science, University of Wisconsin-Madison,*

Abstract

Effective management of environmental resources and agricultural sustainability heavily depends on accurate soil moisture data. However, datasets like the SMAP/Sentinel-1 soil moisture product often contain missing values across their spatiotemporal grid, which poses a significant challenge. This paper introduces a novel Spatiotemporal Transformer model (ST-Transformer) specifically designed to address the issue of missing values in sparse spatiotemporal datasets, with an application on soil moisture data. The ST-Transformer employs multiple spatiotemporal attention layers to capture the complex spatiotemporal correlations in the data and can integrate additional spatiotemporal covariates during the imputation process, thereby enhancing its accuracy. The model is trained using a self-supervised approach, enabling it to autonomously predict missing values from observed data points. Our model's efficacy is demonstrated through its application to the SMAP 1km soil moisture data over a 36 x 36 km grid in Texas. It shows superior accuracy compared to well-known imputation methods. Additionally, our simulation studies on other datasets highlight the model's broader applicability in various spatiotemporal imputation tasks.

Keywords: Spatiotemporal Imputation, Transformer Model, Soil Moisture Data

*Corresponding author

¹Email: kyao24@wisc.edu; Contact Address: Medical Sciences Center, 1300 University Ave Room 1220, Madison, USA

1. Introduction

Soil moisture is essential for enhancing weather forecasts and understanding ecosystems. The NASA Soil Moisture Active Passive (SMAP) mission, launched in January 2015, was aimed to provide high-resolution soil moisture data beneficial for regional or local studies. However, a technical failure of the SMAP radar on July 7, 2015, led to a halt in the production of high-resolution soil moisture data. In the SMAP post-radar phase, various methods were explored to resume generating high-resolution soil moisture data. Research by Das et al. [7] suggested that combining Sentinel-1A and Sentinel-1B data with SMAP radiometer data in the active-passive algorithm has the potential to retrieve soil moisture at finer spatial resolutions of 1 and 3 km. However, this integration resulted in a decreased temporal resolution of SMAP active-passive data to a range of 3 to 12 days, which poses several challenges. These include limitations in monitoring short-term soil moisture changes, potential inaccuracies in assessing long-term soil moisture trends, and difficulties in real-time decision-making due to gaps in data.

To address temporal gaps in SMAP/Sentinel-1 data, filling missing values is a common approach. Kornelsen and Coulibaly [14] evaluated methods like monthly average replacement (MAR), soil layer relative difference (SLRD), linear interpolation, Evolutionary Polynomial Regression (EPR), and Artificial Neural Networks (ANN) for this purpose. They found these methods effective for minor data gaps but less so for larger gaps. Park et al. [22] developed a Multilayer Perceptron (MLP) for long continuous gaps, but their approach is limited to single time series imputation at a specific location without considering spatial correlations. Mao et al. [18] used a two-layer machine learning-based framework for gap-filling the SMAP/Sentinel-1 3-km product. However, their method is oriented more towards the prediction of soil moisture using external covariates, which doesn't fully leverage the spatiotemporal structure within the data that requires imputation.

Beyond the specific application for soil moisture, a wide range of time series

imputation methods is available. Deep-learning-based imputation methods, in particular, have been increasingly popular for their effectiveness. These methods have shown enhanced performance compared to traditional techniques in several benchmark datasets, including those related to air quality [31] and healthcare [24]. The main strength of deep-learning approaches in time series analysis is their capability to identify and understand complex data patterns.

In the field of deep learning for time-series imputation, there are two main categories: autoregressive methods that use recurrent neural networks (RNNs), and non-autoregressive methods that utilize attention mechanisms. Cao et al. [5] introduced BRITS, an autoregressive method that uses bidirectional RNNs for imputing missing values, which was later improved by Cini et al. [6] through the integration of a graph neural network to enhance the understanding of spatial structures. RNN-based methods are known for their strong inductive bias, making them efficient even with smaller datasets. However, the training of RNNs can be slow, as they process data sequentially, and this might lead to the accumulation of errors over time, particularly in datasets with a significant number of missing values. On the other hand, attention-based methods, introduced by Vaswani et al. [28], have demonstrated remarkable performance in numerous natural language processing tasks and offer an alternative to the recurrent structure for time series imputation. For instance, Du et al. [8] introduced SAITS, a model using self-attention mechanisms for filling in missing values in multivariate time series. Tashiro et al. [25] proposed the idea of using a unique architecture of 2D attention (temporal attention and feature attention) for multivariate time series imputation. Compared to RNNs, Transformer-based models are better at capturing long-term dependencies and understanding global context, which is crucial for accurate imputation in time series with complex relationships. Also, transformer models avoid error propagation, are easier to optimize, and have a flexible attention scope, offering a more robust framework for handling sparse observations.

One limitation of the existing transformer-based imputation models is that they primarily focus on the data that requires imputation, often neglecting

the potential benefits of incorporating additional contextual information. This means these models typically concentrate on the directly observed or partially missing spatiotemporal data, but they don't fully leverage other related data that could enhance the imputation process. Another limitation is that most models are not tailored for spatiotemporal data. They usually treat spatial information as features and use a full attention layer to learn the patterns, ignoring the common sense that nearby points are likely to be more similar, which is also known as the "First Law of Geography" [27]. Additionally, those models don't scale well when the spatial field is large.

To overcome the limitations of existing models, we develop a novel transformer-based model that has two key advantages. Firstly, it allows the integration of additional data sources for imputation. In the context of soil moisture, this model utilizes external covariates like daily surface weather data to help imputation. Secondly, our model uses a shifted-window-based self-attention mechanism for spatial modeling, which enhances the model's ability to capture local spatial correlations and scalability [17].

The performance of our model was assessed through comparisons with different imputation methods, utilizing data from Texas soil moisture sites. In addition, we performed imputation tasks on various simulated datasets, such as the Healing Mnist [15] and the SMAP-Hydroblock [29]. These simulation studies offer a further understanding of the model's effectiveness in a variety of scenarios.

This paper's contributions are twofold. Firstly, it introduces a self-supervised training framework built around a new spatiotemporal transformer model designed for a broad range of spatiotemporal imputation tasks. This model is notable for its ability to include covariates in the imputation process and its scalability to large spatial areas. Secondly, to our knowledge, this study is the first to use a transformer-based model for soil moisture gap-filling. In experiments conducted within the Texas region, the model outperformed various time-series imputation methods, achieving the lowest mean absolute error (MAE) and mean relative error (MRE).

2. Method

Let S represent the discrete spatial domain and T represent the discrete temporal domain, with $s \in S$ and $t \in T$. Define $L = |T|$ as the total number of time points and $K = |S|$ as the total number of spatial locations. The observation at spatial location s and at time t is denoted as $Y(s; t)$, and the entire spatiotemporal dataset is represented as $\mathbf{Y} \in \mathbb{R}^{K \times L}$. Let $\mathbf{M} \in \mathbb{R}^{K \times L}$ denote a binary mask, where each element corresponds to the presence or absence of a value in \mathbf{Y} . Specifically, if the observation $Y(s; t)$ is missing, then $M(s; t) = 0$; otherwise, $M(s; t) = 1$. Let $\mathbf{X} \in \mathbb{R}^{K \times L \times D}$ denote the external spatiotemporal covariates, where each element $X(s; t)$ comprises a vector of D features. Furthermore, let $\tilde{\mathbf{Y}}$ denote the unknown complete data of \mathbf{Y} . The objective of spatiotemporal data imputation is to learn a model such that given inputs of \mathbf{Y} , \mathbf{M} , and \mathbf{X} , it yields an estimate $\hat{\mathbf{Y}}$ which minimizes the reconstruction error:

$$L = \frac{\sum_{s=1}^K \sum_{t=1}^L l\{\hat{Y}(s; t), \tilde{Y}(s; t)\} \cdot \{1 - M(s; t)\}}{\sum_{s=1}^K \sum_{t=1}^L \{1 - M(s; t)\}}, \quad (1)$$

where $l(\cdot, \cdot)$ denotes an element-wise loss function such as mean absolute error (MAE). Given that $\tilde{\mathbf{Y}}$ is unknown, a surrogate objective is necessary. A typical approach is to create a set of artificially masked observations. This is done by randomly selecting a subset of the observed values and masking them, resulting in a conditional spatiotemporal dataset \mathbf{Y}^o and the corresponding mask \mathbf{M}^o . The model then attempts to reconstruct \mathbf{Y} based on \mathbf{Y}^o , and the surrogate loss is evaluated on the values that have been randomly masked, expressed by:

$$L = \frac{\sum_{s=1}^K \sum_{t=1}^L l\{\hat{Y}(s; t), Y(s; t)\} \cdot \{M(s; t) - M^o(s; t)\}}{\sum_{s=1}^K \sum_{t=1}^L \{M(s; t) - M^o(s; t)\}}. \quad (2)$$

2.1. Model Overview

In this section, we develop the Spatiotemporal Transformer model (ST-Transformer). The architecture of the model is depicted in Figure 1. The ST-Transformer consists of three main components: an input encoder, a spatiotemporal transformer encoder, and an output layer.

The input encoder functions by integrating covariate data (\mathbf{X}) and observed values (\mathbf{Y}) at each time point and location, transforming them into a latent representation. Following this, the spatiotemporal transformer encoder works to process the spatial and temporal dependencies present in the data. Finally, the output layer of the model is responsible for generating the imputed values.

Input Encoder

The input encoder consists of a Masked Multi-Layer Perceptron (MLP) to process the information from \mathbf{Y} , alongside a generic MLP for encoding the information in \mathbf{X} . It also consists of a positional encoding layer and a learnable spatial embedding layer to encode useful spatiotemporal positional information.

Mathematically, the process can be described as follows. Let $Y(s; t)$, $M(s; t)$, and $X(s; t)$ represent the observed data, missing data indicator, and predictive variables at location s and time t respectively. The masked MLP processes $Y(s; t)$ and $M(s; t)$ to produce an encoded representation $E_Y(s; t)$, formulated as:

$$E_Y(s; t) = \text{MLP}\{Y(s; t)\} \cdot M(s; t) + \text{mask-token} \cdot \{1 - M(s; t)\}, \quad (3)$$

where mask-token is a constant vector of dimension C . Similarly, the generic MLP processes $X(s; t)$ to yield an encoded representation $E_X(s; t)$, expressed as:

$$E_X(s; t) = \text{MLP}(X(s; t)). \quad (4)$$

The temporal position is encoded through a positional encoding layer following the approach in [28]. Specifically, the positional encoding for a position t is given by:

$$E_{\text{time}2i}(t) = \sin\left(t/10000^{\frac{2i}{C}}\right), \quad (5)$$

$$E_{\text{time}2i+1}(t) = \cos\left(t/10000^{\frac{2i}{C}}\right), \quad (6)$$

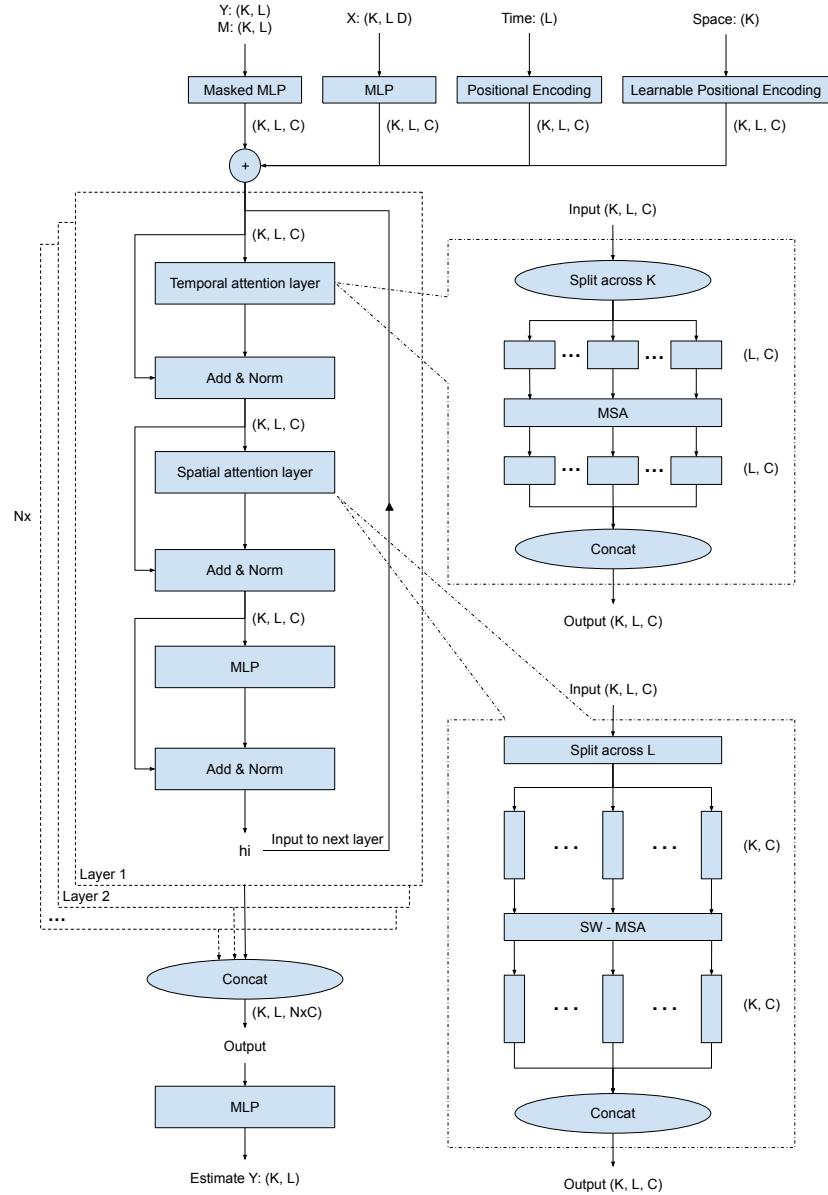


Figure 1: The architecture of the Spatiotemporal Transformer model.

where the indices $2i$ and $2i + 1$ represent the dimension indices of the positional encoding. For spatial embedding, a learnable spatial embedding layer is used to obtain a representation $E_{\text{space}}(s)$ of the spatial location s :

$$E_{\text{space}}(s) = \text{MLP}(s). \quad (7)$$

The final unified latent representation $h(s; t)$ is then obtained by summing up these four encoded representations, as shown in the following equation:

$$h(s; t) = E_Y(s; t) + E_X(s; t) + E_{\text{time}}(t) + E_{\text{space}}(s). \quad (8)$$

Therefore, $h(s; t)$ incorporates the essential information of the observed data and predictive variables, along with the spatiotemporal information.

Spatiotemporal Transformer Encoder

The Spatiotemporal Transformer Encoder aims to encode valuable information for each data point by processing the spatial, temporal, and feature dimensions in the hidden data. It consists of multiple identical spatiotemporal attention layers, each connected via residual connections and layer normalization [1]. Each layer comprises a temporal attention layer, a spatial attention layer, and an MLP, also linked through residual connections and layer normalization.

For the temporal attention layer, the input tensor is initially partitioned along the spatial axis, with each segment then processed through a multi-head self-attention layer (MSA) [28]. The processed segments are then merged to form the output of the temporal attention layer.

For the spatial attention layer, two variants are introduced. The first mirrors the temporal attention layer, differing only in the axis along which the input tensor is split and rejoined. However, for larger spatial dimensions, the conventional MSA encounters scalability issues. To mitigate this, the second variant adopts a stack of shifted-window-based multi-head self-attention (SW-MSA) [17] to replace the standard MSA to ensure efficient modeling.

MSA allows attention across varying positions within the input sequence by projecting the input data into multiple subspaces and performing self-attention

in each of these subspaces simultaneously. Mathematically, the MSA layer is formulated as follows:

$$\text{MSA}(x) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W_O, \quad (9)$$

where

$$\text{head}_i = \text{Attention}(xW_{Qi}, xW_{Ki}, xW_{Vi}), \quad (10)$$

and

$$\text{Attention}(Q, K, V) = \text{softmax}\left(QK^T/\sqrt{d_k}\right)V. \quad (11)$$

Here, W_{Qi} , W_{Ki} , W_{Vi} and W_O are learned weight matrices, h is the number of heads, $d_k = d_{\text{model}}/h$, where d_{model} is the dimensionality of the input and output vectors, and Concat denotes the concatenation operation. Each head computes self-attention independently over different projections of the input data, thereby allowing the model to learn and attend to different patterns across different subspaces. The outputs of all heads are then concatenated and linearly transformed to obtain the final output of the MSA layer.

One limitation of MSA is its quadratic complexity with respect to the input length. For large spatiotemporal datasets, the spatial dimension often spans a magnitude of 10^2 (representing a 10×10 grid) to 10^4 (representing a 100×100 grid), making MSA unscalable. Additionally, MSA computes global attention over the entire input sequence, neglecting the localized spatial correlations inherent in geospatial data.

SW-MSA addresses the computational challenge by dividing the large spatial field into non-overlapping smaller windows, within which self-attention is computed independently within each window [17]. To enable information exchange between windows, a shifted window strategy is utilized after initial window attention. This strategy re-partitions the data based on a shifted window, creating connections between previously non-overlapping windows, and computes masked attention within each window. It's common to stack multiple SW-MSA layers with different window shift sizes to capture spatial correlations at varying scales. The high-level picture of SW-MSA is illustrated in Figure 2. In summary,

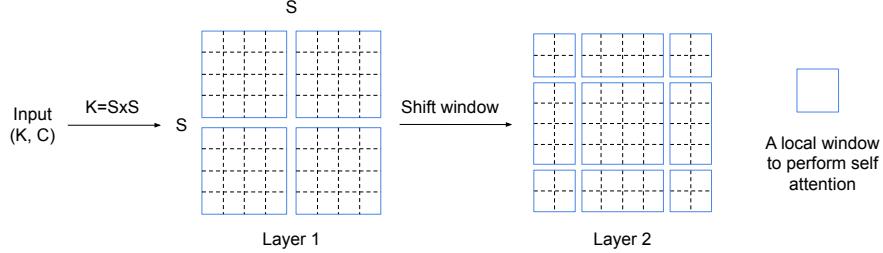


Figure 2: Illustration of the shifted-window multi-head self-attention (SW-MSA) mechanism.

shifted-window-based MSA balances local and global attention and is more scalable compared to traditional MSA. In certain configurations, the SW-MSA can be adjusted to function as the standard MSA. Unless specified otherwise, we default to using SW-MSA in the spatial attention layer in subsequent text.

Now, we demonstrate how the input tensor \mathbf{h} traverses through a spatiotemporal attention layer as follows:

$$\mathbf{h}_{\text{time}} = [\text{MSA}\{\mathbf{h}(1, \cdot)\}, \text{MSA}\{\mathbf{h}(2, \cdot)\}, \dots, \text{MSA}\{\mathbf{h}(K, \cdot)\}]' \quad (12)$$

$$\mathbf{h} = \text{LayerNorm}(\mathbf{h} + \mathbf{h}_{\text{time}}) \quad (13)$$

$$\mathbf{h}_{\text{space}} = [\text{SW-MSA}^n\{\mathbf{h}(\cdot, 1)\}, \text{SW-MSA}^n\{\mathbf{h}(\cdot, 2)\}, \dots, \text{SW-MSA}^n\{\mathbf{h}(\cdot, L)\}]' \quad (14)$$

$$\mathbf{h} = \text{LayerNorm}(\mathbf{h} + \mathbf{h}_{\text{space}}). \quad (15)$$

$$\mathbf{h} = \text{LayerNorm}\{\mathbf{h} + \text{MLP}(\mathbf{h})\} \quad (16)$$

Here, $\text{SW-MSA}^n(x)$ is a stack of SW-MSA followed by residual connections and layer normalization. The mathematical representation is:

$$\text{SW-MSA}^n(x) = \text{LayerNorm}(x + \text{SW-MSA}_n(\quad (17)$$

$$\text{LayerNorm}(x + \text{SW-MSA}_{n-1}(\quad (18)$$

$$\dots \text{LayerNorm}(x + \text{SW-MSA}_1(x)) \dots)))), \quad (19)$$

where $\text{SW-MSA}_i(x)$ is the i -th SW-MSA layer.

Suppose our encoder consists of N identical spatiotemporal attention layers. Denoting the operation of a single layer (as outlined in the preceding procedure) as $L(\cdot)$. If we represent the input to the i -th layer as \mathbf{h}_i , then the input to the $(i + 1)$ -th layer can be expressed as:

$$\mathbf{h}_{i+1} = \mathbf{h}_i + L(\mathbf{h}_i). \quad (20)$$

Even though a single layer might suffice for simpler tasks, employing multiple layers could potentially improve performance. This enhancement arises as each additional layer provides the model an opportunity to learn more abstract representations of the input data. The final output from the encoder is formed by concatenating the outputs from all spatiotemporal attention layers.

Output Layer

The Output Layer of our model is a generic MLP, which aggregates the output from the spatiotemporal transformer encoder to produce the final imputation. Firstly, hidden representations from spatiotemporal encoder \mathbf{h}_i , are concatenated along the feature dimension, formulated as:

$$\mathbf{H} = \text{Concat}(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N). \quad (21)$$

Subsequently, the concatenated representation \mathbf{H} is processed through a generic MLP to generate the final imputation, expressed as:

$$\hat{\mathbf{Y}} = \text{MLP}(\mathbf{H}). \quad (22)$$

3. Case Study: Soil Moisture Imputation

3.1. Remote-Sensed and Ground-Based Datasets

Region of Interest

The Texas Soil Observation Network (TxSON) is a well-monitored area covering 1300 km² near Fredericksburg, Texas, in the central part of the Colorado

River [4]. It has 40 soil moisture monitoring stations placed at distances of 3, 9, and 36 km within the Equal-Area Scalable Earth Grid. We selected a region of 36x36 km by setting the boundaries based on the furthest longitudinal and latitudinal coordinates of all the in situ sites within TxSON. This region was then split into a 36x36 grid, where each cell covers an area of about 1 km x 1 km. In the following sections, we will describe the datasets used for our study, all of which were downloaded for this specific region for the time period from 2016-01-01 to 2022-12-31.

SPL2SMAP_S

Our initial dataset is SMAP/Sentinel-1 L2 Radiometer/Radar 30-Second Scene 3 km EASE-Grid Soil Moisture, Version 3 (SPL2SMAP_S) [7]. This Level-2 (L2) product provides soil moisture estimates at a 1km spatial resolution, based on the cylindrical EASE-Grid 2.0 projection [3]. We will refer to this dataset as SMAP 1km in subsequent text. However, the temporal coverage of this dataset is limited due to the less frequent revisit of the Sentinel-1 satellite, which has a revisit frequency of approximately 12 days. This infrequent revisit rate results in temporal gaps in the dataset, which is exactly what we aim to impute. Due to the low temporal resolution of this product, the resulting data contains around 95% missing values.

SPL3SMP

Our second dataset is SMAP L3 Radiometer Global Daily 36 km EASE-Grid Soil Moisture, Version 8 (SPL3SMP), accessible at [https://nsidc.org/data/spl3smp/VERSIONS/8](https://nsidc.org/data/spl3smp VERSIONS/8) [21]. Within our study region, this Level-3 (L3) product offers a close to daily soil moisture estimate at a 36km spatial resolution under the cylindrical EASE-Grid 2.0 projection [3]. We will refer to this dataset as SMAP 36km in subsequent text.

Daily Surface Weather Data

Our third dataset is Daymet Version 4 [26]. This dataset offers daily weather parameters such as precipitation, solar radiation, minimum and maximum daily

temperature, and vapor pressure at 1km spatial resolution.

Soil, Topography, and Land Cover Type Data

The other datasets utilized in our study are listed as follows:

Soil Texture Data: We obtain the soil texture data from Poggio et al. [23], which provides data at a spatial resolution of 250 meters [12]. These datasets provide information on the fractions of clay and sand, soil organic carbon content, and bulk density. We aggregate this data to a coarser spatial resolution of 1km.

Elevation Data: We obtain elevation data from the U.S. Geological Survey’s (USGS) Shuttle Radar Topography Mission (SRTM) Global 1 arc-second dataset, which has a spatial resolution of about 30 meters [9]. We also aggregate this data to a 1km spatial resolution.

Land Cover Type Data: We obtain land cover type data from NASA’s Moderate Resolution Imaging Spectroradiometer (MODIS) instrument, specifically the MCD12Q1 dataset [11]. This dataset provides classifications of land cover into various categories including forest, urban, and water, among others. We specifically select the International Geosphere-Biosphere Programme (IGBP) classification band, which represents a particular classification scheme of land cover, to match the 1km spatial resolution of our study grid.

Visualization

Figure 3 displays the SMAP 1km, SMAP 36km, and daily surface weather data at a $1\text{km} \times 1\text{km}$ location over the period from January 1, 2016, to December 31, 2020.

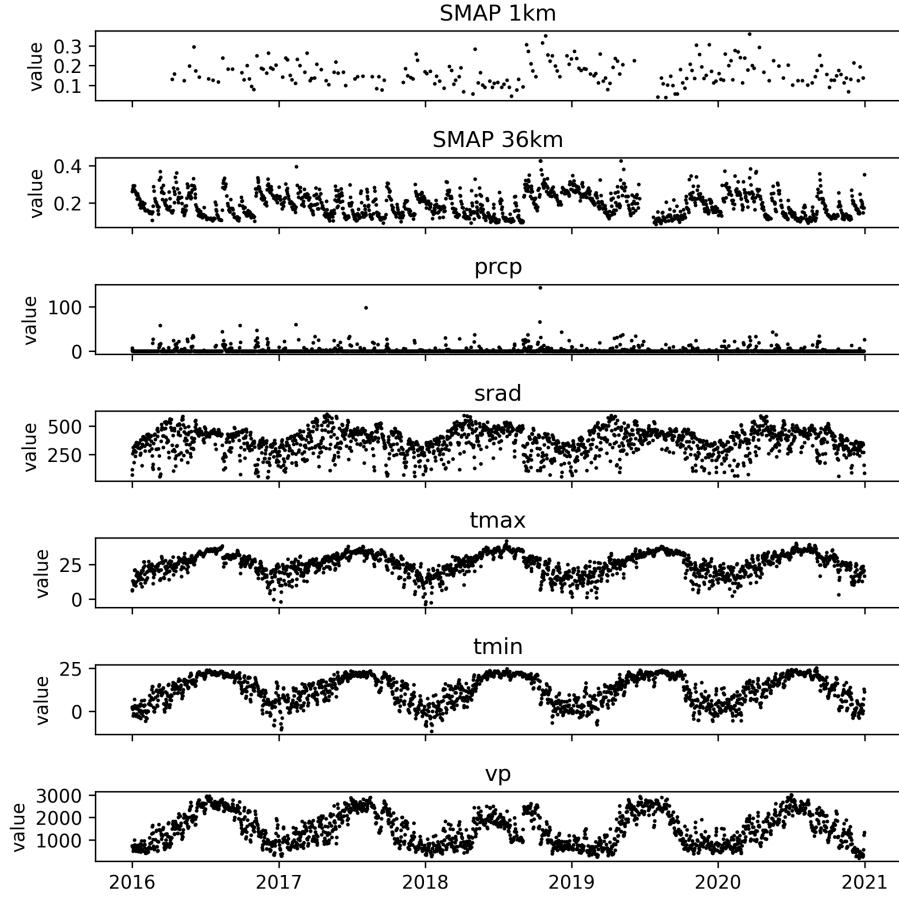


Figure 3: SMAP 1km, SMAP 36km, and other daily surface weather data at a $1\text{km} \times 1\text{km}$ location over the period from January 1, 2016, to December 31, 2020. The weather data include prcp (precipitation), srad (solar radiation), tmax (maximum temperature), tmin (minimum temperature), and vp (vapor pressure).

3.2. Partitioning Data for Training, Validation, and Testing

Let S represent our spatial domain, which is a 36×36 km grid, and T our temporal domain, spanning from 2016/01/01 to 2022/12/31. This configuration yields 1296 spatial locations and 2557 time points. We initially divide all the data temporally into two segments. The segment from 2016/01/01 to 2020/12/31 is allocated for training and validation, while the segment from 2021/01/01 to 2022/12/31 is reserved for out-of-sample testing.

In both segments, artificial missingness is injected in SMAP 1km data to facilitate model validation. We use two distinct approaches for setting missing data: one being “missing not at random (MNAR)”, and the other “missing completely at random (MCAR)”. In the former approach, we randomly select a proportion p of time points, marking the data at all locations for these time points as missing. These points are used for validation. This approach to validation data selection aims to closely mirror the actual missingness patterns observed in real-world data, ensuring a meaningful reflection of model performance on unseen data. On the other hand, the later approach randomly selects a proportion p of points and marks them as missing. This approach seeks to establish more general missing patterns in a spatiotemporal dataset. For both approaches, p is set to 0.2. Figure 4 describes how these two approaches work.

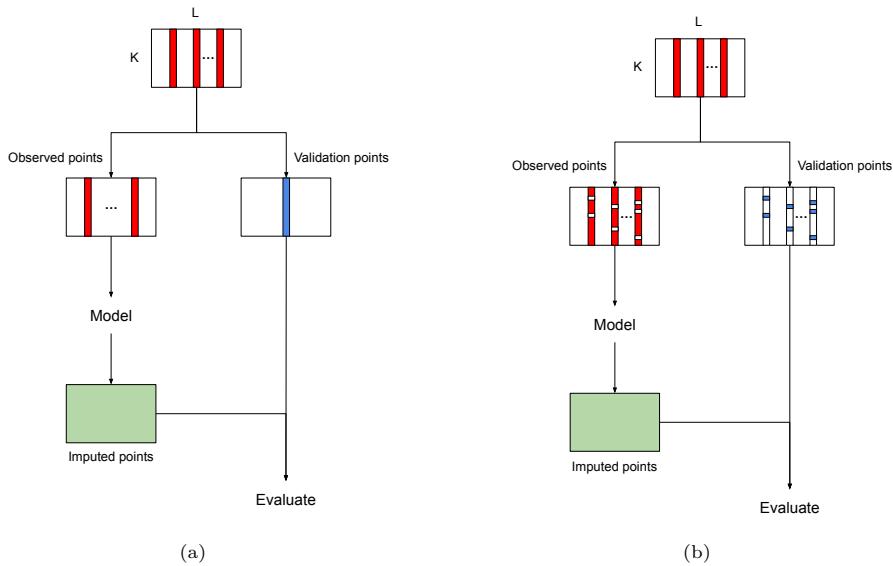


Figure 4: The image on the left describes the “missing not at random” approach. The image on the right describes the “missing completely at random” approach. In both figures, white represents missing data, red represents observed data, blue represents validation data, and green represents imputed data.

3.3. Imputation via Spatiotemporal Transformer Network

Initial Setup

Let $\mathbf{Y}_{\text{train}}$ denote the training dataset of SMAP 1km, with its corresponding missing mask represented by $\mathbf{M}_{\text{train}}$. The predictive variables, including SMAP 36km, Daily Surface Weather Data, Soil, Topography, and Land cover type data, are denoted by $\mathbf{X}_{\text{train}}$. In this setup, the total number of spatial locations is 1296, the total number of time points is 1827, and the total number of predictive variables is 15.

It's important to note that $\mathbf{X}_{\text{train}}$ may also contain missing values. To tackle this, we create a binary tensor $\mathbf{Z}_{\text{train}}$ with the same shape of $\mathbf{X}_{\text{train}}$, where $\mathbf{Z}_{\text{train}}$ pinpoints the missing positions within $\mathbf{X}_{\text{train}}$. Then we concatenate $\mathbf{X}_{\text{train}}$ and $\mathbf{Z}_{\text{train}}$ along the feature axis, yielding an updated version of $\mathbf{X}_{\text{train}}$. As a result, the number of predictive variables becomes 30.

Next, we divide the training data into multiple training samples. According to Orth and Seneviratne [20], soil moisture memory can last up to 40 days. Therefore, we partition $\mathbf{Y}_{\text{train}}$ along the temporal dimension using sliding windows with a fixed length of 72 and a stride of 12. The chosen window length ensures that the model has access to a sufficient range of temporal data to capture the relevant short-term patterns in soil moisture without unnecessarily extending the memory window to periods where the relevance of data has decayed. This operation results in 152 temporal subsets, as illustrated in Figure 5.

In determining the appropriate split size for the spatial grid, several factors were taken into consideration. Firstly, a grid size that is too small might result in the loss of crucial spatial correlations between points. Conversely, an overly large grid size, while potentially capturing spatial correlations, tends to include distant points that may not be spatially correlated, and significantly increases the computational cost. Therefore, a balanced grid size is needed. To determine a reasonable grid size, we use a variogram, a tool commonly used to quantify spatial correlations. Here, a linear model was fitted at each location using

the training data, from which the residuals were computed. Next, a sample variogram was derived from these residuals to assess the spatial correlations, see Figure 6. By analyzing the variogram, we identified that the variance stabilizes when the distance reaches 12 km, indicating a level where the spatial correlation between the furthest points within a grid becomes minimal. Following this, we further segment each of these temporal subsets along the spatial axis by subdividing the 36x36 km grid into 9 equally sized 12x12 km grids, illustrated in Figure 5.

This entire process yields a total of 1368 spatiotemporal subsets. The same procedure is replicated for $\mathbf{X}_{\text{train}}$ and $\mathbf{M}_{\text{train}}$. Consequently, we have 1368 training samples, where each sample consists of $\mathbf{Y}_i \in \mathbb{R}^{144 \times 72}$, $\mathbf{X}_i \in \mathbb{R}^{144 \times 72 \times 30}$, and $\mathbf{M}_i \in \mathbb{R}^{144 \times 72}$. The purpose of segmentation is twofold. First, it functions as a data augmentation technique, providing more samples for model training. Second, it enhances the computational efficiency of the transformer model by mitigating the complexity associated with managing long sequences in the attention mechanism.

Model Training

For each training sample, we introduce artificial missing values to \mathbf{Y}_i using the same two approaches as used in obtaining validation data. In the case where validation data are “missing not at random”, we randomly select $p_i \in \{0.2, 0.5, 0.8\}$, and mark p_i proportion of time points in \mathbf{Y}_i as missing. In the “missing completely at random” scenario, we randomly mark p_i proportion of entries in \mathbf{Y}_i as missing. This procedure results in different positions and proportions of missingness across training samples, enabling the model to learn from a diverse set of scenarios and prevent overfitting.

Model Configuration

In our model, MLP is a feed-forward network comprising two linear transformations with a ReLU activation function in between, as expressed below:

$$\text{MLP}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \quad (23)$$

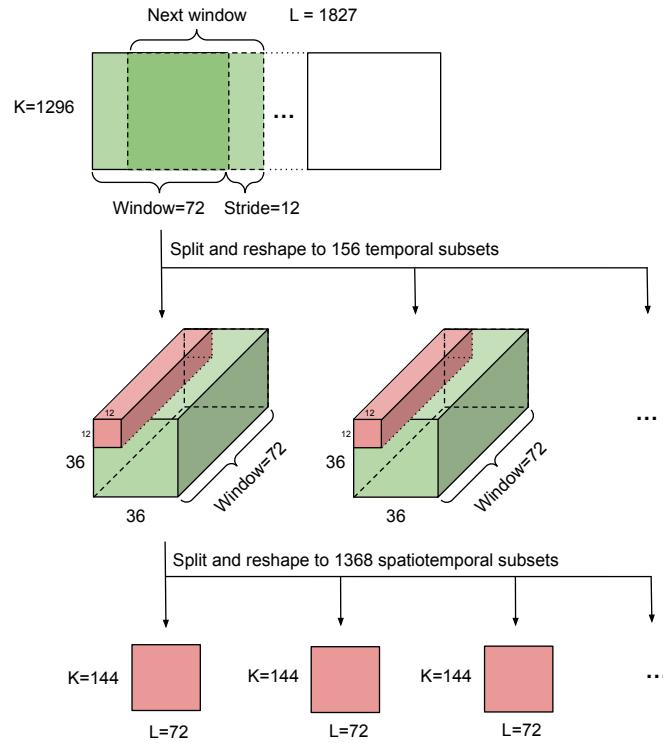


Figure 5: The process of segmenting training data into multiple training samples.

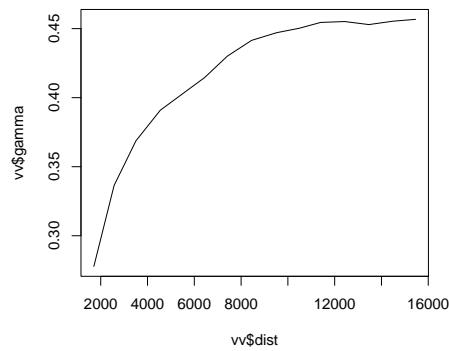


Figure 6: The semi-variogram derived from the analysis of spatial residuals. The variance stabilizes at a lag distance of approximately 12 kilometers.

Within the input encoder, both the hidden and output dimensions of the MLP are set to 64, alongside a spatial and temporal encoding dimension of 64.

For the spatiotemporal transformer encoder, we use four spatiotemporal attention layers. Within each layer, the temporal attention layer uses a single-layer MSA with one attention head, while the spatial layer is explored in two variants: one with a single-layer MSA, and the other with two layers of SW-MSA. In the latter, the first layer uses a window size of 4×4 with no shift (shift-size of 0), and the second layer uses a window size of 4×4 with a shift-size of 2. The hidden dimension of the MLP within the encoder is set to 64.

In the output layer, we set the hidden dimension to 64 and the output dimension to 1 for the MLP.

As for hyperparameters, we set the batch size as 16 and, the number of training epochs to 200. For optimization, We use Adam with a cosine scheduled learning rate, tapering from 0.001 down to 0.0001 over the training period [13].

Model Evaluation

After model training, the model’s performance is assessed through the imputation of the validation points. During evaluation, the training dataset is again divided into smaller subsets, akin to the training phase. However, unlike the training phase, no additional masks are applied to \mathbf{Y}_i during evaluation. After obtaining imputations for all training samples, we reverse the splitting process to reconstruct $\mathbf{Y}_{\text{train}}$. In cases where multiple imputations are made at a specific point in the original data, we take the average. Finally, we evaluate the imputation performance using the mean absolute error (MAE) and mean relative error (MRE). Suppose y_i is the ground truth of i th validation point, \hat{y}_i is the corresponding imputed value, and N is the number of validation points in total. Then MAE and MRE are defined as:

$$\text{MAE} = \sum_{i=1}^N |\hat{y}_i - y_i|/N \quad (24)$$

$$\text{MRE} = \sum_{i=1}^N |\hat{y}_i - y_i| / \sum_{i=1}^N |y_i| \quad (25)$$

Next, we apply the same procedure to assess the model’s performance on the test data. Since the test data is untouched during training, it serves as a reliable measure of the model’s generalization ability.

3.4. Comparison with Baseline Methods

Our model is benchmarked against various simple and advanced imputation methods. These methods include:

Mean: For each location, missing values are replaced with the monthly mean.

Linear Interpolation: Missing values are estimated by drawing a straight line between two known values and finding the value at the missing point along this line.

Matrix Factorization: Matrix Factorization is used to impute missing values by decomposing the original data matrix into two lower-dimensional matrices, capturing underlying patterns and structures. The missing values are then estimated by reconstructing the data matrix from these factorized matrices.

Mice: Multiple imputation by chained equations (MICE) works by performing multiple imputations through a series of regression models, each time estimating missing values, thus creating multiple imputed datasets [30]. The final imputation could be an average of these datasets.

ImputeTS: ImputeTS is an R package for missing value imputation, which utilizes the state space model and Kalman smoothing [19].

Random Forest: Random Forest utilizes an ensemble of decision trees to predict missing values, considering each missing entry as a target variable while treating external covariates as features [2].

GRIN: Graph Recurrent Imputation Network (GRIN) uses a Graph-based Bidirectional Recurrent Neural Network to capture spatial and temporal dependencies in data, leveraging these relationships to impute missing values [6].

CSDI: Conditional Score-based Diffusion models for Imputation (CSDI) introduce a conditional score-based diffusion model for probabilistic imputation [25].

Among the discussed methods, Mean, Linear Interpolation, Matrix Factorization, and ImputeTS do not utilize any predictive variables. In contrast, with minor modifications, the GRIN and CSDI models can incorporate covariates to assist in imputation. It’s also worth noting that methods such as Mean, Matrix Factorization, MICE, and Random Forest solely rely on predictive variables to impute missing values, disregarding the inherent time series structural information. However, other methods do consider the time series information in the imputation process.

Additionally, methods such as MF and MICE do not offer a general model form for imputation, which requires retraining when applied to unseen datasets, thereby limiting their generalizability.

3.5. Results

Missing Not at Random

Initially, we focus on data imputation where validation points are “missing not at random”. For robustness, each imputation algorithm is independently executed five times. During each run, the selection of validation points is randomized, which may lead to variability in chosen points across runs.

The imputation results are presented in Table 1. Our ST-Transformer models outperform the baseline models in both validation and testing datasets.

Specifically, the model using SW-MSA for spatial modeling is slightly better than the one using global attention. Furthermore, models that incorporate covariate information tend to perform better than those that do not, indicating the informative nature of external covariates in imputing SMAP 1km data. Additionally, the deep learning-based methods outperform traditional methods and show greater robustness (as evidenced by smaller standard errors in five replicates). This enhanced performance is likely due to their ability to model complex, non-linear relationships in the data, which traditional imputation methods might fail to capture.

In terms of training efficiency, the ST-Transformer model demonstrates a comparative advantage over certain other deep learning methods. For instance, when trained on an NVIDIA T4 GPU, the GRIN model requires approximately 8 hours to complete its training process. In contrast, the ST-Transformer model completes its training in about 2 hours. Additionally, while the CSDI model matches the ST-Transformer in training duration, it necessitates an extra 4 hours for generating imputation results.

Figure 7 displays the imputation performance of the ST-Transformer (SW-MSA) at a 1km \times 1km location in the test dataset. Figure 8 presents the imputation results across all locations within the study region over a span of 14 days from July 21, 2021, to August 3, 2021.

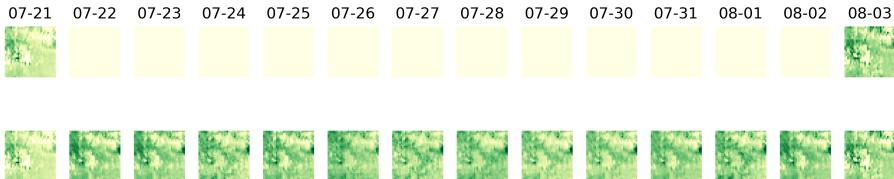


Figure 8: Imputation of SMAP 1km data spanning from July 21, 2021, to August 3, 2021. The top row displays the observed SMAP 1km data, where white spaces represent missing data. The bottom row illustrates the imputed soil moisture data.

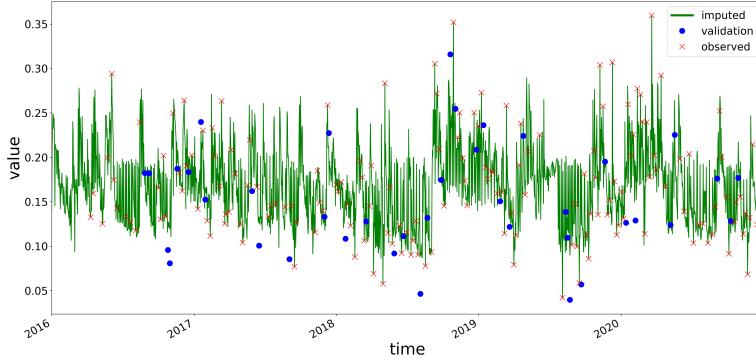


Figure 7: Imputation of SMAP 1km from 2021-01-01 to 2022-12-31 at a 1km × 1km location. The land cover is grassland. Here, the green line is the imputed soil moisture data, red \times represents the observed data, and blue \circ represents the validation data.

Missing Completely at Random

The “missing not at random” scenario is less common in spatiotemporal data. Typically, missing values in spatiotemporal data are distributed randomly across space and time. To ensure the generalizability of our method to these situations, we also evaluate all imputation methods on the “missing completely at random” scenario. Similar to the earlier scenario, every imputation algorithm is run five times independently. The outcomes are presented in Table 2. Our ST-Transformer models consistently outperform all baseline models. Methods like Mean, Linear interpolation, ImputeTS, Mice, and Random Forest, which do not model spatial correlation, exhibit relatively stable performance. Interestingly, Matrix Factorization outperforms the covariate-based methods and even matches the performance of some deep learning methods. This suggests the presence of non-missing data at certain spatial locations significantly aids the imputation process. Additionally, our model with the SW-MSA layer significantly outperforms the model using the standard MSA layer. This indicates that the SW-MSA layer is more effective in capturing spatial correlations compared to the global attention approach.

3.6. Ablation Studies

To highlight the effectiveness of spatiotemporal attention, we conduct an ablation study comparing four model variations: the complete ST-Transformer, a variant without spatial or temporal attention, one with only temporal attention, and another with only spatial attention. We evaluate these models on the validation data under both “missing not at random” and “missing completely at random” scenarios, with the results presented in Table 3. In both scenarios, incorporating temporal and spatial attention significantly enhances imputation accuracy. Interestingly, while spatial attention doesn’t significantly improve results in the “missing not at random” scenario, it brings substantial improvement in the “missing at random” scenario. One reason is that in the former scenario, observations are missing for all locations, leaving spatial information unexploited. Conversely, in the latter scenario, the model can directly leverage observed data from nearby locations for imputation.

Additionally, we conducted another ablation study to show the impact of covariates on imputation. We categorize the covariates into two groups: time-varying features, which include SMAP 36km and daily surface weather data, and static features like soil characteristics, topography, and land cover type data.

We evaluate four model variations. The first model incorporates both time-varying and static covariates, the second model includes no covariates, the third model only uses time-varying features, and the fourth model solely uses static features. We assess these model variations under both missing data scenarios.

Table 4 presents the imputation performance. In the “missing not at random” scenario, we observe that including time-varying covariates significantly enhances the model performance, while incorporating static covariates also provides a slight improvement. Conversely, in the “missing completely at random” scenario, the inclusion of covariates doesn’t contribute much to the performance. This is probably because the model imputes missing values by exploiting spatial relations, making the additional covariate information less influential.

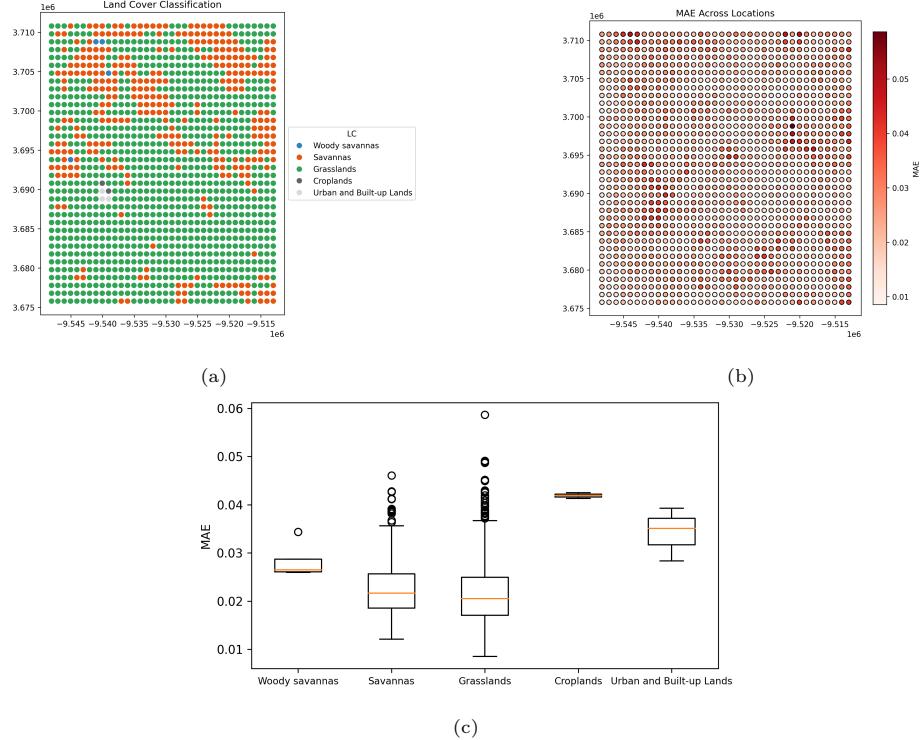


Figure 9: Figure (a) illustrates the spatial distribution of the land cover types. Figure (b) shows the spatial variation of the MAE. Figure (c) provides box plots of the MAE across different land cover types.

3.7. Spatial variability of the Imputation

In our study, we evaluated the imputation accuracy of different land covers by measuring the Mean Absolute Error (MAE) for various locations. Under the “missing not at random” scenario, we plot the MAE across space in Figure 9. We observed distinct variations in imputation accuracy among different land covers. Specifically, grasslands showed the best performance, followed by savannas, woody savannas, urban areas, and croplands. The poorer imputation in cropland and urban areas could be attributed to a scarcity of training data.

4. Conclusion

In this paper, we introduced the Spatiotemporal Transformer (ST-Transformer) model, designed to address the challenge of imputing missing values in sparse spatiotemporal datasets, with an application to the sparse soil moisture dataset. Compared with traditional soil moisture methods, our model has two key advantages. Firstly, it incorporates both time-varying and static covariates to guide imputation in cases of extreme data sparsity, offering a more robust approach to addressing missing soil moisture data. Secondly, it employs a spatiotemporal transformer to effectively model spatial and temporal correlations from the available data. Additionally, the utilization of SW-MSA layers allows the model to efficiently handle large spatial fields.

Furthermore, we established a self-supervised training framework and demonstrated its efficacy in imputing SMAP 1km data. The model’s ability to generalize to other spatiotemporal imputation tasks was also investigated. In A’s simulation study (shown in the Appendix), the model was applied to the Healing Mnist dataset, where it surpassed the performance of the current leading benchmark. In both simulation studies and real data analysis, our ST-Transformer model showed enhanced accuracy and computational efficiency compared to existing imputation methods.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.
- [2] Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

- [3] Brodzik, M. J., Billingsley, B., Haran, T., Raup, B., and Savoie, M. H. (2012). Ease-grid 2.0: Incremental but significant improvements for earth-gridded data sets. *ISPRS International Journal of Geo-Information*, 1(1):32–45.
- [4] Caldwell, T. G., Bongiovanni, T., Cosh, M. H., Jackson, T. J., Colliander, A., Abolt, C. J., Casteel, R., Larson, T., Scanlon, B. R., and Young, M. H. (2019). The texas soil observation network: A comprehensive soil moisture dataset for remote sensing and land surface model validation. *Vadose Zone Journal*, 18(1):1–20.
- [5] Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. (2018). Brits: Bidirectional recurrent imputation for time series. *Advances in Neural Information Processing Systems*, 31.
- [6] Cini, A., Marisca, I., and Alippi, C. (2021). Filling the g_ap_s: Multivariate time series imputation by graph neural networks. *arXiv preprint arXiv:2108.00298*.
- [7] Das, N. N., Entekhabi, D., Dunbar, R. S., Chaubell, M. J., Colliander, A., Yueh, S., Jagdhuber, T., Chen, F., Crow, W., O'Neill, P. E., et al. (2019). The smap and copernicus sentinel 1a/b microwave active-passive high resolution surface soil moisture product. *Remote Sensing of Environment*, 233:111380.
- [8] Du, W., Côté, D., and Liu, Y. (2023). Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619.
- [9] Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., et al. (2007). The shuttle radar topography mission. *Reviews of Geophysics*, 45(2).
- [10] Fortuin, V., Baranchuk, D., Rätsch, G., and Mandt, S. (2020). Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661. PMLR.

- [11] Friedl, M. and Sulla-Menashe, D. (2022). Modis/terra+ aqua land cover type yearly l3 global 500m sin grid v061. *NASA EOSDIS Land Processes DAAC: Sioux Falls, SD, USA*.
- [12] Hengl, T. (2018). Clay content in%(kg/kg) at 6 standard depths (0, 10, 30, 60, 100 and 200 cm) at 250 m resolution (v0. 2)[data set]. *Zenodo. Available online: <https://zenodo.org/record/2525663#.Y-7VR2lByUk>* (accessed on 10 October 2022).
- [13] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [14] Kornelsen, K. and Coulibaly, P. (2014). Comparison of interpolation, statistical, and data-driven methods for imputation of missing values in a distributed soil moisture dataset. *Journal of Hydrologic Engineering*, 19(1):26–43.
- [15] Krishnan, R. G., Shalit, U., and Sontag, D. (2015). Deep kalman filters. *arXiv preprint arXiv:1511.05121*.
- [16] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [17] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.
- [18] Mao, H., Kathuria, D., Duffield, N., and Mohanty, B. P. (2019). Gap filling of high-resolution soil moisture for smap/sentinel-1: a two-layer machine learning-based framework. *Water Resources Research*, 55(8):6986–7009.
- [19] Moritz, S. and Bartz-Beielstein, T. (2017). imputets: time series missing value imputation in r. *R J.*, 9(1):207.

- [20] Orth, R. and Seneviratne, S. I. (2012). Analysis of soil moisture memory from observations in europe. *Journal of Geophysical Research: Atmospheres*, 117(D15).
- [21] O'Neill, P., Chan, S., Njoku, E., Jackson, T., Bindlish, R., Chaubell, J., and Colliander, A. (2021). Smap enhanced l3 radiometer global and polar grid daily 9 km ease-grid soil moisture version 5. *NASA National Snow and Ice Data Center DAAC, Boulder, CO, USA*.
- [22] Park, J., Müller, J., Arora, B., Faybushenko, B., Pastorello, G., Varadharajan, C., Sahu, R., and Agarwal, D. (2023). Long-term missing value imputation for time series data using deep neural networks. *Neural Computing and Applications*, 35(12):9071–9091.
- [23] Poggio, L., De Sousa, L. M., Batjes, N. H., Heuvelink, G., Kempen, B., Ribeiro, E., and Rossiter, D. (2021). Soilgrids 2.0: producing soil information for the globe with quantified spatial uncertainty. *Soil*, 7(1):217–240.
- [24] Silva, I., Moody, G., Scott, D. J., Celi, L. A., and Mark, R. G. (2012). Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE.
- [25] Tashiro, Y., Song, J., Song, Y., and Ermon, S. (2021). CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816.
- [26] Thornton, M., Shrestha, R., Wei, Y., Thornton, P., Kao, S., and Wilson, B. (1840). Daymet: daily surface weather data on a 1-km grid for north america, version 4. ornl daac, oak ridge, tennessee, usa.
- [27] Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(sup1):234–240.

- [28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [29] Vergopolan, N., Chaney, N. W., Pan, M., Sheffield, J., Beck, H. E., Ferguson, C. R., Torres-Rojas, L., Sadri, S., and Wood, E. F. (2021). Smap-hydroblocks, a 30-m satellite-based soil moisture dataset for the conterminous us. *Scientific Data*, 8(1):264.
- [30] White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: issues and guidance for practice. *Statistics in Medicine*, 30(4):377–399.
- [31] Yi, X., Zheng, Y., Zhang, J., and Li, T. (2016). St-mvl: filling missing values in geo-sensory time series data. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Dataset	Method	MAE	MRE
Validation	Mean	0.0473 +/- 0.001	31.12% +/- 2.10%
	Linear Interpolation	0.0484 +/- 0.000	32.55% +/- 0.32%
	Matrix Factorization	0.0491 +/- 0.003	43.25% +/- 6.54%
	ImputeTS	0.0413 +/- 0.002	27.63% +/- 4.83%
	Mice	0.0312 +/- 0.001	27.32% +/- 2.36%
	Random Forest	0.0316 +/- 0.001	27.25% +/- 2.68%
	GRIN	0.0287 +/- 0.000	16.76% +/- 0.42%
	CSDI	0.0283 +/- 0.000	16.85% +/- 0.50%
	ST-Transformer (MSA)	0.0240 +/- 0.000	15.14% +/- 0.59%
	ST-Transformer (SW-MSA)	0.0223 +/- 0.000	14.66% +/- 0.63%
Testing	Mean	0.0482 +/- 0.000	32.24% +/- 0.91%
	Linear Interpolation	0.0485 +/- 0.000	32.59% +/- 0.30%
	Matrix Factorization	0.0408 +/- 0.005	34.22% +/- 4.64%
	ImputeTS	0.0415 +/- 0.003	25.11% +/- 3.30%
	Mice	0.0325 +/- 0.001	28.10% +/- 2.27%
	Random Forest	0.0344 +/- 0.001	30.81% +/- 1.86%
	GRIN	0.0289 +/- 0.000	17.10% +/- 0.49%
	CSDI	0.0284 +/- 0.000	17.51% +/- 0.65%
	ST-Transformer (MSA)	0.0247 +/- 0.001	16.62% +/- 0.87%
	ST-Transformer (SW-MSA)	0.0231 +/- 0.001	15.51% +/- 0.91%

Table 1: Average performance of imputation methods over five independent runs. The data follows a “missing not at random” pattern. The table compares various imputation algorithms.

Dataset	Method	MAE	MRE
Validation	Mean	0.0483 +/- 0.000	32.13% +/- 0.85%
	Linear Interpolation	0.0466 +/- 0.000	30.98% +/- 0.094%
	Matrix Factorization	0.0240 +/- 0.000	19.33% +/- 0.12%
	ImputeTS	0.0410 +/- 0.002	27.12% +/- 0.92%
	Mice	0.0226 +/- 0.001	18.61% +/- 0.17%
	Random Forest	0.0319 +/- 0.001	26.55% +/- 0.21%
	GRIN	0.0215 +/- 0.000	14.78% +/- 0.43%
	CSDI	0.0211 +/- 0.000	14.01% +/- 0.48%
	ST-Transformer (MSA)	0.0192 +/- 0.000	12.76% +/- 0.74%
	ST-Transformer (SW-MSA)	0.0144 +/- 0.000	9.58% +/- 0.34%
Testing	Mean	0.0484 +/- 0.000	32.55% +/- 0.00%
	Linear Interpolation	0.0467 +/- 0.000	31.45% +/- 0.00%
	Matrix Factorization	0.0240 +/- 0.000	19.27% +/- 0.13%
	ImputeTS	0.0415 +/- 0.003	25.11% +/- 3.25%
	Mice	0.0230 +/- 0.000	19.07% +/- 0.17%
	Random Forest	0.0364 +/- 0.000	32.91% +/- 0.30%
	GRIN	0.0220 +/- 0.000	15.31% +/- 0.22%
	CSDI	0.0214 +/- 0.000	14.96% +/- 0.51%
	ST-Transformer (MSA)	0.0195 +/- 0.001	13.16% +/- 0.40%
	ST-Transformer (SW-MSA)	0.0146 +/- 0.001	9.80% +/- 0.18%

Table 2: Average performance of imputation methods over five independent runs. The data follows a “missing completely at random” pattern. The table compares various imputation algorithms.

Scenario	Method	MAE	MRE
MNAR	ST-Transformer	0.0223 +/- 0.000	14.66% +/- 0.63%
	ST-Transformer (no space, no time)	0.0301 +/- 0.000	19.89% +/- 0.73%
	ST-Transformer (no space)	0.0257 +/- 0.000	17.17% +/- 0.62%
	ST-Transformer (no time)	0.0262 +/- 0.000	17.89% +/- 0.15%
MCAR	ST-Transformer	0.0144 +/- 0.000	9.58% +/- 0.34%
	ST-Transformer (no space, no time)	0.0300 +/- 0.000	19.80% +/- 0.11%
	ST-Transformer (no space)	0.0213 +/- 0.000	14.17% +/- 0.42%
	ST-Transformer (no time)	0.0171 +/- 0.000	11.50% +/- 0.11%

Table 3: Average imputation performance over five independent runs at two missing scenarios. The models compared are ST-Transformers with different structures.

Scenario	Method	MAE	MRE
MNAR	ST-Transformer	0.0223 +/- 0.000	14.66% +/- 0.63%
	ST-Transformer (no covariates)	0.0404 +/- 0.001	24.75% +/- 1.51%
	ST-Transformer (time-varying)	0.0254 +/- 0.001	16.17% +/- 0.66%
	ST-Transformer (static)	0.0385 +/- 0.001	22.39% +/- 0.50%
MCAR	ST-Transformer	0.0144 +/- 0.000	9.58% +/- 0.34%
	ST-Transformer (no covariates)	0.0155 +/- 0.001	10.31% +/- 0.38%
	ST-Transformer (time-varying)	0.0147 +/- 0.001	9.79% +/- 0.11%
	ST-Transformer (static)	0.0150 +/- 0.001	9.93% +/- 0.15%

Table 4: Average imputation performance over five independent runs at two missing scenarios. The models compared are ST-Transformers with different covariates.

Appendix A. Simulation

To assess the effectiveness of our ST-Transformer model under different scenarios, we also conducted two distinct simulation studies. The first focused on the imputation tasks on the Healing MNIST dataset, and the second focused on imputation using another Soil Moisture dataset.

Appendix A.1. Healing MNIST

The Healing MNIST dataset [15] is a variant of the popular MNIST dataset [16]. The original MNIST dataset comprises a large collection of handwritten digits (0 through 9), each represented as a 28x28 pixel grayscale image. The Healing MNIST transforms them into time series or sequential data by applying various changes to the digits like rotation or scaling, creating a “video” of each digit undergoing transitions. For testing our imputation model, we introduced missing values into each frame and then utilized the remaining observed data to predict these missing elements. Following the approach of Fortuin et al. [10], we generated missing values in two ways: missing completely at random (MCAR), and missing not at random (MNAR). In the latter scenario, the likelihood of a white pixel becoming missing was 2 times higher than that of a black pixel. In both scenarios, approximately 50% of each frame’s pixels were masked as missing. The dataset comprised 50,000 time series of digits for training purposes and 10,000 for testing.

We evaluated our model’s performance against the GP-VAE model [10], which is the state-of-the-art for this type of task, along with basic methods like mean and interpolation. Attempts were made to implement other deep learning methods such as GRIN and CSDI. However, these were found to be computationally intensive, even during initial test runs. Therefore, we decided not to include them in our analysis. Our results, detailed in Table A.5, show that the ST-Transformer model significantly outperformed the GP-VAE across both missing data scenarios, particularly in the MNAR scenario where the improvement was substantial. Figure A.10 displays the imputation results in this

scenario, illustrating the model’s effectiveness in accurately imputing the missing digits.

Scenario	Method	MSE
MCAR	ST-Transformer	0.025 +/- 0,000
	GP-VAE	0.036 +/- 0.000
	Mean	0.113 +/- 0.001
	Interpolation	0.074 +/- 0.001
MNAR	ST-Transformer	0.054 +/- 0.000
	GP-VAE	0.114 +/- 0.002
	Mean	0.210 +/- 0.004
	Interpolation	0.135 +/- 0.002

Table A.5: Average imputation performance on the Healing MNIST dataset over 5 independent runs.

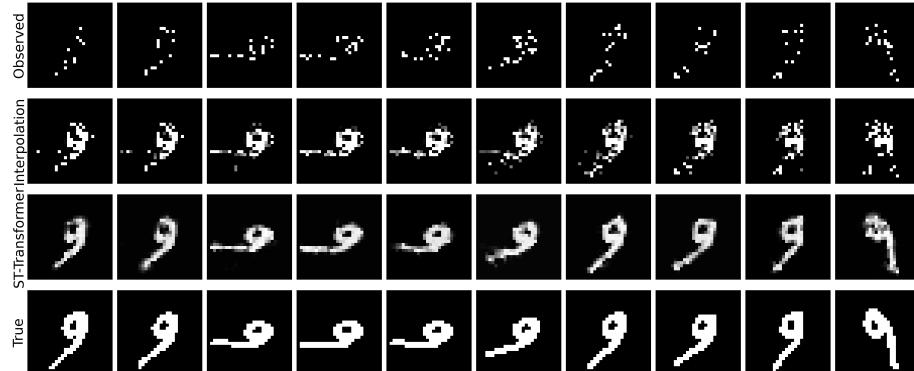


Figure A.10: Reconstructions from Healing MNIST in the missing not at random scenario. The first row displays a time series of digits with missing values. The second row shows the imputation result using linear interpolation. The third row shows the imputation result using ST-Transformer. The fourth row shows the true sequences of the digits.

Method	MAE	MRE
Mean	0.0456 +/- 0.002	28.34% +/- 0.41%
Linear Interpolation	0.0415 +/- 0.001	25.16% +/- 0.080%
Matrix Factorization	0.0483 +/- 0.000	35.29% +/- 1.36%
ImputeTS	0.0410 +/- 0.002	27.12% +/- 0.92%
Mice	0.0286 +/- 0.001	20.09% +/- 0.60%
Random Forest	0.0294 +/- 0.001	20.35% +/- 0.397%
GRIN	0.0259 +/- 0.000	16.11% +/- 0.43%
CSDI	0.0264 +/- 0.000	16.50% +/- 0.48%
ST-Transformer (MSA)	0.0240 +/- 0.000	15.29% +/- 0.43%
ST-Transformer (SW-MSA)	0.0210 +/- 0.000	13.17% +/- 0.24%

Table A.6: Average imputation performance over 5 independent runs. The data used is from the SMAP HydroBlocks dataset. The table compares various imputation algorithms.

Appendix A.2. SMAP-HydroBlocks

SMAP-HydroBlocks (SMAP-HB), introduced in Vergopolan et al. [29], is a high-resolution, satellite-based surface soil moisture dataset covering the conterminous United States from 2015 to 2019 at a 30-m resolution. The dataset includes a post-processed, aggregated version at 1 km and 6-hour resolution, available at <https://zenodo.org/records/5206725>. To ensure consistency with our real data analysis, we extract data from a 36×36 km grid within the study region and incorporate covariates to help in the imputation process. We replicate the missing patterns found in the original SMAP 1km dataset and use only the observed data for imputation.

The imputation result, presented in Table A.6, shows that our ST-Transformer model outperforms other methods. Additionally, Figure A.11 provides a visual representation of the imputation results, showing that the data imputed by our model aligns closely with the actual data.

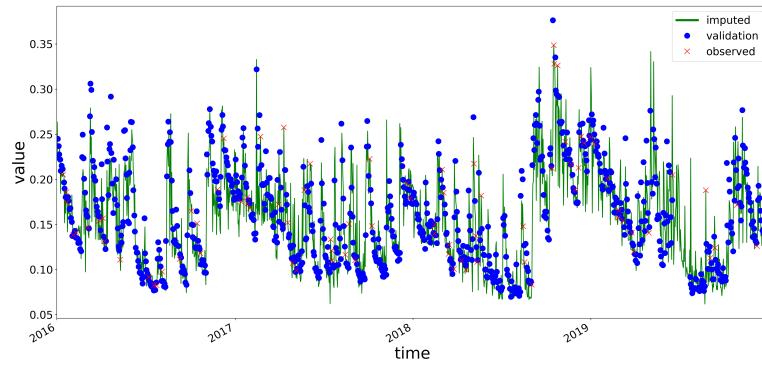


Figure A.11: Imputation of SMAP HydroBlocks 1km from 2016-01-01 to 2019-12-31 at a 1km \times 1km location using ST-Transformer. The land cover is grassland. Here, the green line is the imputed soil moisture data, red \times represents the observed data, and blue \circ represents the validation data.