An Interpretable Implicit-Based Approach for Modeling Local Spatial Effects: A Case Study of Global Gross Primary Productivity Estimation

Siqi Du[1], Hongsheng Huang[2], Kaixin Shen[3], Ziqi Liu[4], Shengjun Tang[5]

[1] College of Urban and Environmental Sciences, Peking University, Beijing, P.R. China
[2] Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong, P.R. China
[3] School of Safety Science, Tsinghua University, Beijing, P.R. China
[4] College of Surveying and Geo-Informatics, Tongji University, Shanghai, P.R. China
[5] Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen, P.R. China

Keywords: AI4Science, Deep Learning, Ecology, Remote Sensing, Spatiotemporal Heterogeneity, Gross Primary Productivity.

Abstract

In Earth sciences, unobserved factors exhibit non-stationary spatial distributions, causing the relationships between features and targets to display spatial heterogeneity. In geographic machine learning tasks, conventional statistical learning methods often struggle to capture spatial heterogeneity, leading to unsatisfactory prediction accuracy and unreliable interpretability. While approaches like Geographically Weighted Regression (GWR) capture local variations, they fall short of uncovering global patterns and tracking the continuous evolution of spatial heterogeneity. Motivated by this limitation, we propose a novel perspective—that is, simultaneously modeling common features across different locations alongside spatial differences using deep neural networks. The proposed method is a dual-branch neural network with an encoder-decoder structure. In the encoding stage, the method aggregates node information in a spatiotemporal conditional graph using GCN and LSTM, encoding location-specific spatiotemporal heterogeneity as an implicit conditional vector. Additionally, a self-attention-based encoder is used to extract location-invariant common features from the data. In the decoding stage, the approach employs a conditional generation strategy that predicts response variables and interpretative weights based on data features under spatiotemporal conditions. The approach is validated by predicting vegetation gross primary productivity (GPP) using global climate and land cover data from 2001 to 2020. Trained on 50 million samples and tested on 2.8 million, the proposed model achieves an RMSE of 0.836, outperforming LightGBM (1.063) and TabNet (0.944). Visualization analyses indicate that our method can reveal the distribution differences of the dominant factors of GPP across various times and locations.

1. Introduction

In Earth science, machine learning method like XGBoost (Chen and Guestrin, 2016) is widely used to model environmental and geographical relationships, such as predicting climate change impacts on vegetation (Lu et al., 2024) and understanding tropical cyclones' effects on precipitation (Qin et al., 2024). However, a fundamental challenge in geographical modeling is the presence of spatiotemporal heterogeneity, where relationships between independent and target variables vary across both space and time. Most machine learning methods, which assume unordered samples and stationary relationships, lack the capability to model such spatiotemporal heterogeneity. This limitation calls for novel approaches specifically designed for geographical machine learning problems that can capture these varying relationships while maintaining model accuracy and interpretability.

One solution is to fit, at different spatial locations, a set of weights that vary with the coordinates by locally sampling the data, with Geographically Weighted Regression (GWR) (Fotheringham et al., 2009) serving as a representative method. However, GWR lacks temporal modeling, limiting its ability to capture the evolution of spatial heterogeneity. To address GWR's temporal limitation, Geographically and Temporally Weighted Regression (GTWR) (Fotheringham et al., 2015) was developed, using spatiotemporal metrics to model variability. Despite these advancements, current methods still fit spatial weights locally, based on neighborhood samples, leading to several challenges: 1) Local models capture spatial variability but often overlook the shared global patterns. 2) Many spatial models depend on heuristic parameter tuning, lacking automatic optimization of spatial weights. 3) Training separate models for each location is computationally inefficient for large-scale Earth science data.

In response to these challenges, we review the inherent characteristics of geographic machine learning tasks. Compared to typical machine learning problems, geographic machine learning tasks generally require the simultaneous capture of global common patterns and local spatial differences. For instance, when investigating the relationship between vegetation gross primary productivity (GPP) and various influencing factors. Although vegetation growth is governed by a general law, the dominant factors vary slightly in different locations. Based on this observation, our study proposes a novel perspective: **the geographic machine learning problem can be decomposed into two sub-tasks—learning the location-invariant objective law and capturing the conditional differences among various spatial locations.**

Based on the considerations above, we propose a novel geographic machine learning approach. This method is inspired by deep learning research that emphasizes direct pattern discovery from data and end-to-end differentiable processes for the simultaneous optimization of multiple tasks (Carion et al., 2020). The proposed approach leverages a multi-branch deep neural network to concurrently model common features shared across different locations and local spatial differences. It employs an

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-395-2025 | © Author(s) 2025. CC BY 4.0 License.

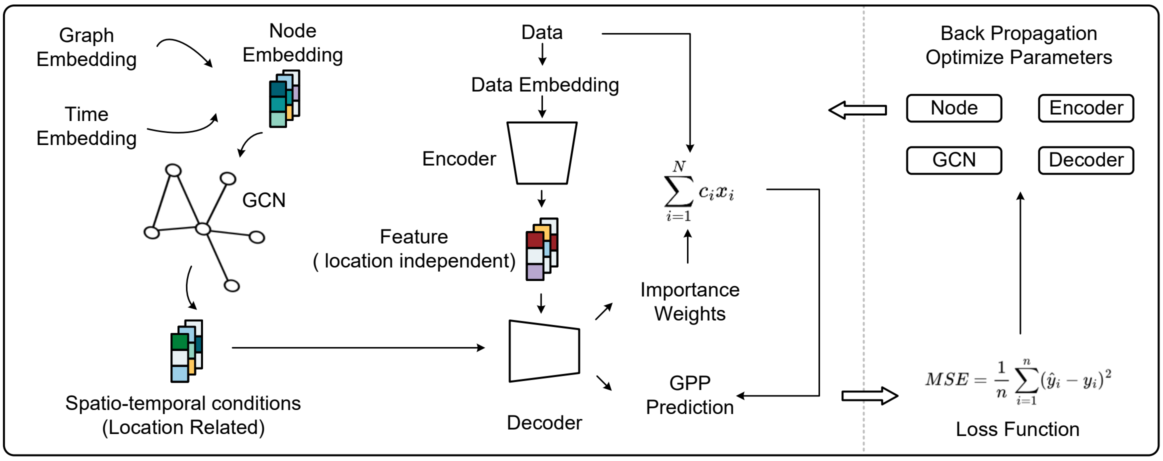


Figure 1. The overall process of the method.

encoder-decoder architecture. In the encoding stage, node information is aggregated in a spatiotemporal conditional graph using graph convolutional networks (GCN) and long short-term memory (LSTM) networks to encode location-specific spatiotemporal heterogeneity as an implicit conditional vector. Additionally, a self-attention-based encoder is utilized to extract location-invariant common features from the data. In the decoding stage, a conditional generation strategy is adopted to predict the response variables and interpretative weights based on data features under spatiotemporal conditions.

To validate our approach, we created the Climate2GPP dataset, using the ERA5 climate dataset (Muñoz-Sabater et al., 2021), the MCD12C1.061 MODIS Land Cover dataset (Friedl and Sulla-Menashe, 2022), and the PML_V2 0.1.7 GPP dataset (Zhang et al., 2019). Spanning from 2001 to 2020 with an 8-day temporal resolution, this dataset includes approximately 50 million samples for training and 2.8 million samples for testing. Our method achieved an RMSE of 0.836 on the test set, significantly outperforming GWR (RMSE 1.937), classical tabular machine learning methods like LightGBM Large (RMSE 1.063) and deep learning methods like TabNet (RMSE 0.944).

Our main contributions are summarized as follows:

- A novel multi-branch encoder-decoder architecture that jointly models global features and spatial differences.
- An integrated spatiotemporal heterogeneity model using conditional graphs, GCN, and LSTM.
- A dual-branch decoder with cross-attention for effective fusion and prediction under spatiotemporal constraints.

2. Related Works

Tabular machine learning methods have been widely applied in Earth science for tasks such as predicting environmental changes and understanding geographical phenomena. These methods, including popular algorithms like LightGBM (Ke et al., 2017) and XGBoost (Chen and Guestrin, 2016), are designed under the assumption that samples are independent and identically distributed, which limits their applicability in scenarios with spatial dependencies. While geographically weighted models, such as GWR (Fotheringham et al., 2009) and GTWR (Fotheringham et al., 2015), have been introduced to address spatial heterogeneity by adjusting coefficients locally, they face significant limitations. GWR models fail to capture temporal evolution, and while GTWR extends this capability, both models struggle with nonlinearity and exhibit high computational complexity when applied to large datasets. GWR-RF (Wang et al., 2024) combines GWR with Random Forest for nonlinearity but still suffers from local overfitting and dense weight matrices. GNNWR (Du et al., 2020) balances global patterns and spatial variability through neural network-corrected coefficients but retains the complexity of dense spatial weights. GTNNWR (Wu et al., 2021) further incorporates spatiotemporal heterogeneity but remains limited by the need to fit local variables and dense spatiotemporal weights, making these models prone to overfitting and computationally inefficient in large-scale applications.

3. Method

The proposed method is built upon a multi-branch architecture with an encoder-decoder structure that jointly addresses two key objectives: extracting location invariant common features and learning location specific spatiotemporal differential conditions. In the encoding stage, one branch utilizes a dual self-attention mechanism within a tabular data encoder to capture shared, common features from data. Concurrently, another branch focuses on learning spatiotemporal differential conditions, representing location specific variations via implicit latent vectors. These latent representations are derived through a locally shared spatiotemporal condition graph and further refined using Graph Convolutional Networks (GCN) and Long Short-Term Memory (LSTM) networks.

In the decoding stage, a multi-branch decoder integrates the location-invariant features with the spatiotemporal conditions through cross-attention interactions. This decoder is designed under two distinct objectives: predicting the target variable and estimating feature importance weights. The architecture, divided into a target variable prediction branch and a feature importance estimation branch, ensures that predictions are made under the appropriate spatiotemporal constraints while maintaining model interpretability. Finally, all parameters are jointly optimized under both the main loss and the auxiliary loss.

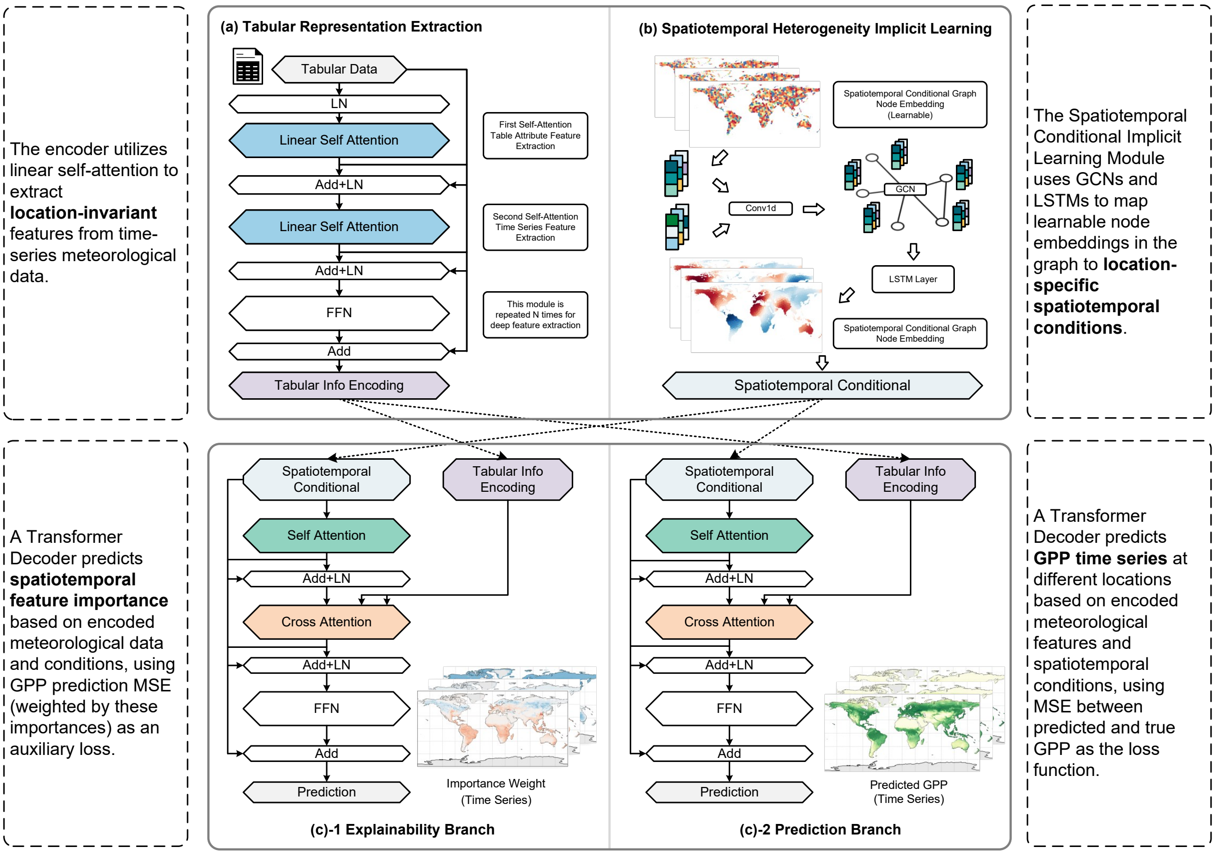

The encoder utilizes linear self-attention to extract **location-invariant** features from time-series meteorological data.
(a) Tabular Representation Extraction
Tabular Data
LN
Linear Self Attention
Add+LN
Linear Self Attention
Add+LN
FFN
Add
Tabular Info Encoding
First Self-Attention Table Attribute Feature Extraction
Second Self-Attention Time Series Feature Extraction
This module is repeated N times for deep feature extraction
(b) Spatiotemporal Heterogeneity Implicit Learning
Spatiotemporal Conditional Graph Node Embedding (Learnable)
Conv1d
GCN
LSTM Layer
Spatiotemporal Conditional Graph Node Embedding
Spatiotemporal Conditional
The Spatiotemporal Conditional Implicit Learning Module uses GCNs and LSTMs to map learnable node embeddings in the graph to **location-specific spatiotemporal conditions**.
A Transformer Decoder predicts **spatiotemporal feature importance** based on encoded meteorological data and conditions, using GPP prediction MSE (weighted by these importances) as an auxiliary loss.
Spatiotemporal Conditional
Tabular Info Encoding
Self Attention
Add+LN
Cross Attention
Add+LN
FFN
Add
Prediction
Importance Weight (Time Series)
(c)-1 Explainability Branch
Spatiotemporal Conditional
Tabular Info Encoding
Self Attention
Add+LN
Cross Attention
Add+LN
FFN
Add
Prediction
Predicted GPP (Time Series)
(c)-2 Prediction Branch
A Transformer Decoder predicts **GPP time series** at different locations based on encoded meteorological features and spatiotemporal conditions, using MSE between predicted and true GPP as the loss function.

Figure 2. The implementation details of the dual encoder and the dual decoder.

3.1 Extraction of Location-Invariant Common Features

In our task, the extraction of location-invariant common features means learning robust representations from time series meteorological data. To achieve this, we designed a module that decomposes the feature extraction task into two complementary objectives: (1) capturing inter-feature relationships among the data attributes, and (2) extracting temporal patterns inherent in the time series of each attribute.

To address these objectives, we extend a transformer encoder originally developed for tabular data (Hu et al., 2024). This enhanced encoder, as illustrated in Figure 2(a), incorporates a dual attention (DA) mechanism that simultaneously operates in both feature and temporal dimensions. To further accommodate the large-scale geoscience data and reduce the computational complexity, we implement a novel linear self-attention mechanism (Katharopoulos et al., 2020). After performing linear embedding and applying temporal position encoding (Su et al., 2024) to the input meteorological data, the stacked DA modules progressively refine the representations, ultimately yielding data features that are invariant to location-specific factors.

Dual Attention Mechanism: Given an input tensor $X \in \mathbb{R}^{L\times D}$, where L represents the sequence length (time dimension) and D is the feature dimension, the DA module sequentially computes self-attention (Katharopoulos et al., 2020) across the temporal and feature dimensions. First, temporal self-attention is applied across the time steps for each feature. Queries, keys, and values are computed as:

$$Q^{\text{temp}} = XW_Q^{\text{temp}}, \quad K^{\text{temp}} = XW_K^{\text{temp}}, \quad V^{\text{temp}} = XW_V^{\text{temp}} \tag{1}$$

The temporal attention output is then computed by applying the activation function $\phi(x) = \text{ELU}(x) + 1$ directly within the attention formula:

$$\text{Attn}^{\text{temp}} = \frac{\phi(Q^{\text{temp}})\left(\phi(K^{\text{temp}})^\top V^{\text{temp}}\right)}{\phi(Q^{\text{temp}})\left(\phi(K^{\text{temp}})^\top \mathbf{1}_L\right) + \epsilon} \tag{2}$$

Feature self-attention is then computed along the feature dimension using the same process. Residual connections are employed at each step to ensure gradient flow and model stability:

$$X^{\text{feat}} = X^{\text{temp}} + \text{Attn}^{\text{feat}} \tag{3}$$

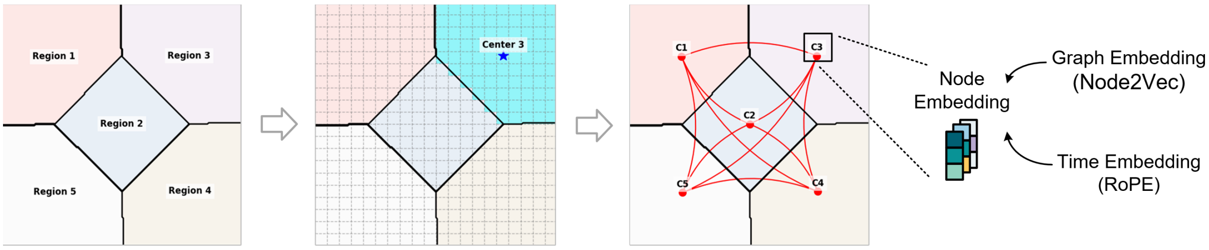
Feedforward Network: Finally, a position-wise feedforward network is applied to each element:

$$Y = X^{\text{feat}} + \text{FFN}(X^{\text{feat}}) \tag{4}$$

By employing both temporal and feature self-attention, followed by a feedforward network, this model captures rich representations from tabular Earth science data.

3.2 Learning Spatiotemporal Differential Conditions

The Learning Spatiotemporal Differential Conditions subsection is dedicated to directly learning implicit condition vectors that capture the distinct spatiotemporal variations present across locations. To achieve this, each location is assigned a hidden vector representing its unique spatiotemporal condition. These vectors, together with their spatial relationships, form the Spatiotemporal Conditional Graph. By leveraging a combination of Graph Convolutional Networks (GCN) and Long Short-Term



1. K-means clustering for region partitioning.

2. Each grid cell within a region shares a Spatiotemporal Conditions Vector.

3.Connect adjacent regions.

4.Apply embedding on the Spatiotemporal Conditions Vector

Figure 3. Spatiotemporal Conditional Graph Construction with Local Weight Sharing.

Memory (LSTM) networks, our approach effectively aggregates the spatial dependencies and temporal dynamics among these vectors, resulting in a unified spatiotemporal condition vector that is subsequently input into the decoder for further prediction tasks.

Spatiotemporal Conditional Graph Construction (STCG): The STCG is defined as $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each node $v_{i,t} \in V$ represents a spatiotemporal point (λ_i, ϕ_i, t), where λ_i and ϕ_i are the longitude and latitude of node i, and t is the time. Each node has an embedding $\boldsymbol{v}_{i,t}$ that captures the spatiotemporal condition at that point. The prediction for a spatiotemporal location is influenced by the embedding $\boldsymbol{v}_{i,t}$ of the corresponding node in the STCG. The construction of the STCG involves the following steps. The construction of STCG is shown in Figure **??**

Graph Node Generation: To determine the geographical coordinates (λ_i, ϕ_i) of each node $v_{i,t} \in V$, we apply K-means clustering to the global land grid. This step reduces computational complexity by grouping spatial regions into clusters, where each cluster shares a common spatiotemporal condition. Specifically, the cluster centers $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ are determined by minimizing the sum of squared distances between all spatial points and their nearest cluster centers:

$$\boldsymbol{C} = \arg\min_{\boldsymbol{C}} \sum_{p=1}^{n} \min_{j} \|(\lambda_p, \phi_p) - \boldsymbol{c}_j\|^2 \tag{5}$$

Here, (λ_p, ϕ_p) represents any spatial point p in the global grid, and n is the total number of such points. Each node $v_{i,t} \in V$ is then assigned the spatial coordinates of the corresponding cluster center: $(\lambda_i, \phi_i) = \boldsymbol{c}_i$.

Cyclic Graph Construction: To ensure connectivity between the eastern and western hemispheres, we map the geographical coordinates of the cluster centers to spherical coordinates and construct the adjacency matrix $\boldsymbol{A}$ using the K-nearest neighbors (KNN) method on the sphere. Specifically, for each cluster center $v_{i,t}$, we first project its geographical coordinates (λ_i, ϕ_i) (longitude and latitude) onto a 3D unit sphere using the following transformation:

$$x_i = \cos(\phi_i)\cos(\lambda_i), \quad y_i = \cos(\phi_i)\sin(\lambda_i), \quad z_i = \sin(\phi_i) \tag{6}$$

where (x_i, y_i, z_i) represents the 3D spherical coordinates of node $v_{i,t}$. Using the spherical coordinates $\boldsymbol{p}_i = (x_i, y_i, z_i)$, we compute the adjacency matrix $\boldsymbol{A}$ of the graph by defining the k-nearest neighbors for each node $v_{i,t}$. The adjacency matrix $\boldsymbol{A}$ is constructed as follows:

$$\boldsymbol{A}_{i,j} = \begin{cases} 1, & j \in \arg\text{top-k}\, \|\boldsymbol{p}_i - \boldsymbol{p}_j\| \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Here, $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are the 3D spherical coordinates of nodes $v_{i,t}$ and $v_{j,t}$, respectively. This approach ensures that the graph is cyclic, connecting locations on opposite sides of the globe, which is particularly important for capturing the circular nature of the Earth.

Node Embedding Calculation: We use Node2vec (Grover and Leskovec, 2016) to compute the initial embeddings for the nodes. For each time dimension t, we add a time embedding using the Rotational Position Embedding (RoPE) (Su et al., 2024) method: $\boldsymbol{v}_{i,t} = \text{Node2vec}(i) + \text{RoPE}(t)$ where $\boldsymbol{v}_{i,t}$ is the embedding of node $v_{i,t}$ at time t.

Edge Weight Calculation: The weight of each edge $w_{i,j}$ in the graph is computed using a log-Gaussian kernel, which incorporates two sequential normalization steps to effectively capture the similarity between cluster centers in a 3D space. The weight of each edge $w_{i,j}$ can be calculated by:

$$w_{i,j} = \begin{cases} \exp\left(-\frac{1}{2\sigma^2}\left(1 - \exp\left(-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|}{\mu}\right)\right)^2\right) \\ 0 \end{cases} \tag{8}$$

The above equation defines $w_{i,j}$ such that when

$$j \in \arg\text{top-k}\, \|\mathbf{p}_i - \mathbf{p}_j\|, \tag{9}$$

the weight is given by the expression in the first case; otherwise, it is 0.

Spatiotemporal Conditional Encoding: Although our earlier constructed spatiotemporal conditional graph enables each node to directly learn its spatiotemporal condition vector via backpropagation, this method alone is not sufficient to capture the intricate interactions between regions. To address this limitation, we incorporate a spatiotemporal aggregation operation that leverages 1D temporal convolution, graph convolution, and LSTM. For each node $v_{i,t}$ in the STCG, we update its embedding $\boldsymbol{v}_{i,t}$ through these aggregation processes.

Temporal Aggregation: We first apply a 1D convolution operation along the time dimension to capture local temporal dependencies. This can be formulated as:

$$V^{temp} = V * W^{time} \tag{10}$$

where V is the matrix of node embeddings $\boldsymbol{v}_{i,t}$, W^{time} is the learnable temporal convolution kernel, and $*$ denotes the convolution operation.

Spatial Aggregation: Following the temporal aggregation, we employ a graph convolution operation to aggregate spatial information:

$$V^{spatial} = \sigma(D^{-\frac{1}{2}}\boldsymbol{A}WD^{-\frac{1}{2}}V^{temp}H) \tag{11}$$

where $\boldsymbol{A}$ is the adjacency matrix, W is the edge weight matrix $w_{i,j}$, D is the degree matrix d_i, H is the learnable weight matrix, and σ is a non-linear activation function. The final embedding for each node $v_{i,t}$, incorporating both temporal and spatial information, can be expressed as:

$$\boldsymbol{v}_{i,t}^{spatial} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{d_i d_j}} w_{i,j} H \left(\sum_{\tau=-k}^{k} w_{\tau}^{time} \cdot \boldsymbol{v}_{j,t+\tau}\right)\right) \tag{12}$$

where $\mathcal{N}(i)$ is the set of neighboring nodes of node $v_{i,t}$, d_i and d_j are the degrees of nodes i and j, $w_{i,j}$ is the edge weight between nodes $v_{i,t}$ and $v_{j,t}$, and w_{τ}^{time} are the elements of the temporal convolution kernel W^{time}. By applying these operations sequentially, we obtain a rich representation $\boldsymbol{v}_{i,t}^{spatial}$ for each spatiotemporal point $v_{i,t}$, which encapsulates both local and global spatiotemporal dependencies.

LSTM Aggregation: After performing spatial aggregation, these spatial features are further processed by an LSTM network to capture long-term temporal dependencies.

$$V^{final} = \text{LSTM}(V^{spatial}) \tag{13}$$

3.3 Decoder Design for Constrained Prediction

In the prediction stage, a conditional prediction approach is adopted to estimate the target variables corresponding to meteorological data under spatiotemporal conditions. The prediction can be formulated under the maximum a posteriori (MAP) framework as follows:

$$\hat{y} = \arg\max_{y}\, p\Big(y \,\Big|\, c(x,y,t),\, X\Big) \tag{14}$$

where $c(x, y, t)$ denotes a function of spatial coordinates (x, y) and time (t) representing the spatiotemporal condition and X stands for the input data. To achieve a higher level of interpretability similar to geographically weighted regression (GWR), our design employs two independent branches: one for predicting the target variable and another for estimating the interpretable weights associated with the intarget variables.

Decoder Structure: For the decoder, we utilize the standard Transformer decoder (Vaswani, 2017), which naturally serves as a conditional predictor. Its core component is the cross-attention mechanism. In our framework, the query Q and key K are embedded from the data features obtained in the **Extraction of Location-Invariant Common Features**, while the value V is given by V^{final}, which aggregates the spatiotemporal conditional graph information. The cross-attention mechanism is formulated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{15}$$

where d_k is the dimension of the key vectors.

target variable Prediction Branch: In this branch, the Transformer decoder takes as input the spatiotemporal conditions and the encoded data features to directly predict the target variable. The optimization of this branch is performed using the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{dep}} = \frac{1}{N}\sum_{i=1}^{N}\Big(y_i - \hat{y}_i\Big)^2, \tag{16}$$

where y_i and $\hat{y}_i$ denote the ground-truth and predicted values, respectively.

Feature Importance Estimation Branch: For the estimation of interpretable weights, we employ a separate Transformer decoder branch to predict an importance weight for each intarget variable. The predicted interpretable weights, denoted as w_j, are then used to perform a weighted linear combination of the input intarget variables:

$$\hat{y}_{\text{interp}} = \sum_{j=1}^{M} w_j\, x_j, \tag{17}$$

where x_j represents the jth intarget variable and M is the total number of intarget variables. An auxiliary MSE loss is computed between $\hat{y}_{\text{interp}}$ and the true target variable y to further refine the interpretability of this branch:

$$\mathcal{L}_{\text{interp}} = \frac{1}{N}\sum_{i=1}^{N}\Big(y_i - \hat{y}_{\text{interp},i}\Big)^2. \tag{18}$$

The overall training objective is to minimize a combination of both loss functions $\mathcal{L}_{\text{interp}} + \mathcal{L}_{\text{dep}}$, thereby ensuring accurate prediction while obtaining interpretable feature importance scores.

4. Experiments

4.1 Experimental Setup

Dataset: To validate our approach, we created the Climate2GPP dataset. We used data from Google Earth Engine spanning January 1, 2001, to December 17, 2020. The data sources include:

- **ERA5-Land Daily Aggregated** (ECMWF): Global historical meteorological data aggregated every 8 days (Muñoz-Sabater et al., 2021).
- **MCD12C1.061 MODIS Land Cover Type** (NASA): Yearly global land cover changes at 0.05 degree resolution (Friedl and Sulla-Menashe, 2022).
- **PML_V2 0.1.7**: Global gross primary productivity (GPP) data aggregated every 8 days (Zhang et al., 2019).

From ERA5, we selected 26 climate parameters, with solar radiation, evaporation, and precipitation summed over 8-day periods, while the rest were averaged. GPP was similarly summed over 8-day intervals. Data from 2001 to 2019 was used for training (52M samples), and data from 2020 served as the test set (2.8M samples).

Training Setting: Our method is implemented in PyTorch 2.1.2 with CUDA 11.8. All features, except GPP, are normalized.

The AdamW optimizer is used with a batch size of 256, an initial learning rate of 0.001, decayed to 0.0001 after 10 epochs, for a total of 20 epochs.

Comparison machine learning method is using AutoGluon 1.1.1 (Erickson et al., 2020) with default hyperparameters. These models, along with deep learning comparisons (TabNet, ResNet, ExcelFormer, FFTransformer implemented in PyTorchFrame 0.2.3 (Hu et al., 2024)), are trained on RTX 4090 GPU, 64-core Intel Xeon Platinum 8352V and 120GB of RAM. All deep learning models use the same optimizer settings as our method.

4.2 Result Comparison

Table 1. Comparison of Different Methods

Method	RMSE	R^2
Tabular Machine Learning		
LightGBM Large	1.063	0.886
KNeighborsDist	1.093	0.879
KNeighborsUnif	1.096	0.879
LightGBM	1.108	0.876
XGBoost	1.124	0.872
LightGBMXT	1.126	0.872
NeuralNetFastAI	1.142	0.868
CatBoost	1.152	0.866
RandomForestMSE	1.182	0.859
Tabular Deep Learning		
TabNet	0.944	0.901
ExcelFormer	1.001	0.878
ResNet	1.014	0.878
FTTransformer	1.158	0.850
Our Method		
Ours	**0.836**	**0.932**

Experiment Setting: To validate the suitability of our proposed method for machine learning tasks in the Earth sciences, we conducted a comparative evaluation against a range of widely adopted machine learning baselines, including Random Forest (Breiman, 2001), XGBoost (Chen and Guestrin, 2016), CatBoost (Prokhorenkova et al., 2018), the LightGBM family (Ke et al., 2017), and KNN. Additionally, we compared our method with state-of-the-art tabular deep learning approaches, including TabNet (Arik and Pfister, 2019), ExcelFormer (Chen et al., 2024), ResNet (Gorishniy et al., 2021), and FTTransformer (Gorishniy et al., 2021). All models were trained using the complete dataset of 50 million samples to assess their scalability and performance on large-scale data. The prediction accuracy of each method was evaluated on the Climate2GPP test set for estimating the total gross primary productivity (GPP) for the year 2020, as detailed in the table (1).

Comparison: As shown in Table 1, the best-performing machine learning and deep learning methods on this task were LightGBM Large and TabNet, achieving RMSEs of 1.063 and 0.944, respectively. However, our proposed method outperformed both, achieving a lower RMSE of 0.836 and an R^2 of 0.932. These results underscore the superior capability of our approach in handling large-scale Earth science data.

4.3 Comparison of Spatial and Spatiotemporal Heterogeneity Methods

Experiment Setting: Furthermore, we aim to compare our method with other approaches that are capable of modeling spatial or spatiotemporal heterogeneity. The computational complexity of the GWR series methods is proportional to the number of spatial locations in the dataset and require exactly one sample point per spatial location. Given these limitations, all experiments were conducted on a few-shot dataset. Specifically, we uniformly sampled 6,000 grids from the land grid as training data, with each grid containing data from all available time points. Since GWR series methods require a one-to-one correspondence between samples and locations, we used the average data from every 8 days over 19 years as the training samples. For GWR (Fotheringham et al., 2009) and GNNWR (Du et al., 2020), which cannot model spatiotemporal heterogeneity, we fit separate weekly temporal models. For all models, we selected results from Weeks 1, 10, 20, 30, and 40, which are representative of different seasons, for comparison. The results are shown in Table 2.

Table 2. Comparison of Spatial and Spatiotemporal Heterogeneity Methods (RMSE / R^2)

	Spatial		Spatiotemporal	
	GWR	GNNWR	GTWR	Ours
Week-01	1.990/0.43	0.871/0.89	1.761/0.56	**0.779/0.91**
Week-10	2.097/0.42	1.066/0.85	1.859/0.55	**0.813/0.91**
Week-20	2.184/0.59	1.330/0.86	2.475/0.48	**1.073/0.91**
Week-30	2.060/0.51	1.178/0.84	2.070/0.50	**0.931/0.89**
Week-40	1.958/0.43	0.835/0.90	1.689/0.58	**0.700/0.92**

Linear Model vs non-Linear Model: Based on the results shown in Table 2, non-linear models such as GNNWR and our method consistently achieve better performance than the linear approaches (GWR and GTWR). For instance, in Week-01, the RMSE values for GWR and GTWR are 1.990 and 1.761, respectively, whereas GNNWR and our method obtain lower RMSE values of 0.871 and 0.779, with corresponding R² values of 0.89 and 0.91. Similar advantages are evident in Week-40, where GWR and GTWR report RMSE values of 1.958 and 1.689, compared to 0.835 and 0.700 for GNNWR and our approach, along with higher R² values (0.90 and 0.92). These numerical findings clearly illustrate that non-linear methods are more adept at capturing complex spatial and spatiotemporal variations, leading to more accurate model fitting.

Table 3. Comparison of Spatial and Spatiotemporal Heterogeneity Methods

Metric	GNNWR	Ours
Train RMSE	0.478	0.627
Train R^2	0.931	0.942
Test RMSE	0.835	0.700
Test R^2	0.896	0.922
Gen. Gap	0.357	0.073

Local learning vs Global learning: Our model's innovation lies in leveraging the entire spatial sample to learn the spatiotemporal conditions at each location, as opposed to the traditional local fitting approach used in GWR, GTWR, and GNNWR. This global learning strategy significantly reduces the tendency to overfit in local regions. As evidenced in Table 3, although our model exhibits a slightly higher training RMSE (0.627 versus 0.478) and a marginally improved training R² (0.942 versus 0.931) compared to GNNWR, it achieves superior performance on the test set. Specifically, our method records a lower test RMSE (0.700 compared to 0.835) and a higher test R² (0.922 versus 0.896), leading to a much smaller generalization gap (0.073 versus 0.357). This outcome confirms that employing a global learning approach effectively mitigates overfitting and enhances predictive robustness.

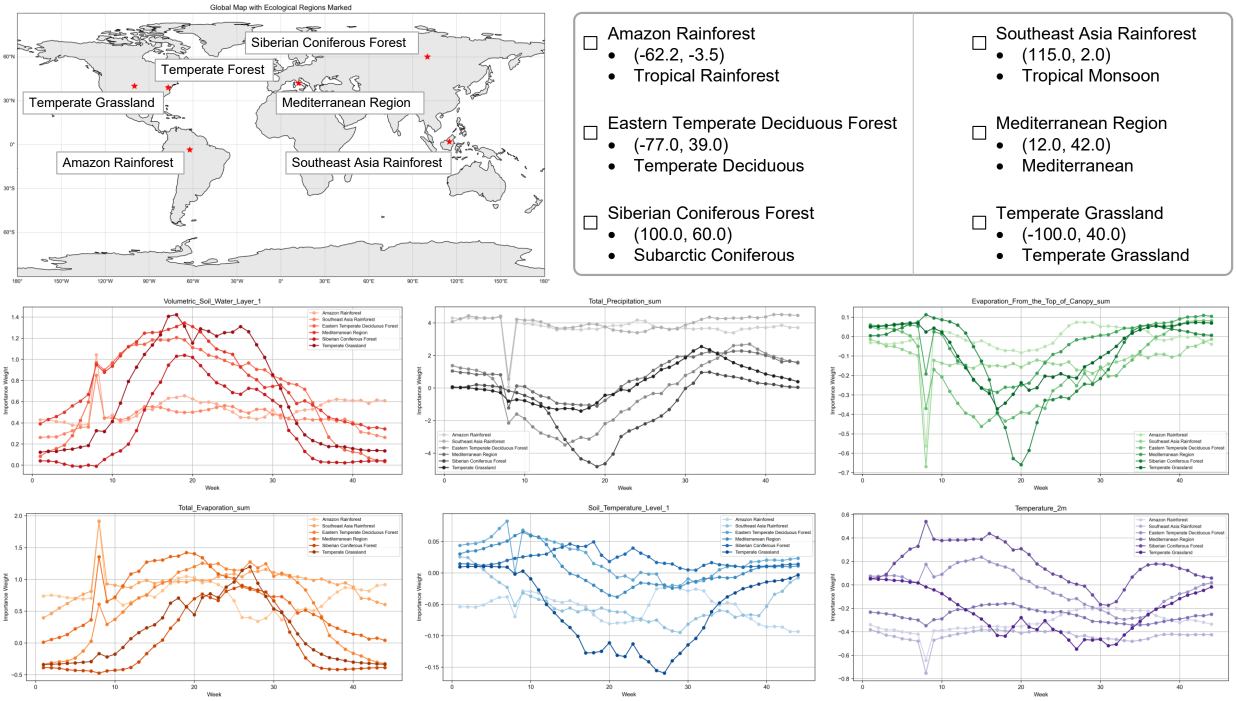


Figure 4. Temporal Variation of Feature Importance in Six Typical Regions.

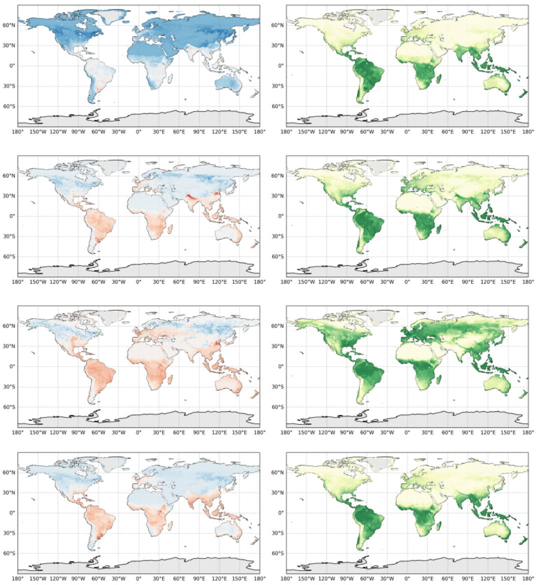


Figure 5. Spatial Distribution of temperature_2m Importance Weights and GPP Prediction Results at 01, 10, 20, and 30.

4.4 Visualization

Our visualization analysis aims to demonstrate the potential of our method in spatial analysis through three sets of results. The Figure 4 presents the temporal variation curves of explanatory variables at different spatial locations, covering six distinct climate-vegetation zones: ”Amazon Rainforest”, ”Southeast Asia Rainforest”, ”Eastern Temperate Deciduous Forest”, ”Mediterranean Region”, ”Siberian Coniferous Forest”, and ”Temperate Grassland”. The Figure 5 illustrates the spatial distribution of the explanatory weights for the temperature_2m variable at Weeks 01, 10, 20, and 30, along with the corresponding model-predicted GPP. These visualizations are designed to highlight two main points: first, that our approach can capture the variation in dominant factors influencing GPP across different regions; and second, that the spatial patterns in the temperature_2m weights reveal both beneficiary and affected areas under the context of global warming. It is important to note that this visualization analysis represents an initial exploration, and further work is needed to fully develop the interpretability of the model.

5. Conclusion

In this paper, we have identified key challenges in geographical machine learning, notably the difficulty of capturing spatiotemporal heterogeneity. While traditional statistical learning methods fail to model these variations, existing approaches such as the GWR family can capture spatiotemporal heterogeneity but rely on locally sampled data, which often leads to overfitting in spatial regions.

To address these issues, we have proposed a novel deep learning–based framework that decomposes a geographical regression task into learning location-independent data features and capturing spatiotemporal variations. Unlike conventional methods, our approach optimizes over the entire data space, significantly alleviating overfitting in local areas.

We validated our method on a large-scale dataset built from ERA5 and PML_V2, comprising 50 million training samples and 2.8 million test samples. The experimental results demonstrate the effectiveness of our design: our method achieved a test RMSE of 0.836, outperforming GWR (RMSE 1.937), classical tabular machine learning methods like LightGBM Large (RMSE 1.063), and other deep learning approaches such as TabNet (RMSE 0.944). These findings confirm that our proposed framework not only captures spatiotemporal heterogeneity effectively, but also delivers superior predictive performance in geographical machine learning tasks.

6. Discussion

While our proposed method shows promising performance, several aspects remain open for future improvement. First, regarding interpretability, our visualizations indicate that our approach holds potential for explaining complex spatiotemporal relationships. However, to accurately characterize causal effects, we plan to integrate additional causal constraints and develop methods that infer dominant factors directly from the explanatory weights. Second, the current modeling of spatial interactions relies on a simple linear mechanism using graph convolutions, where interaction patterns are predetermined and fixed. In future work, we intend to explore more sophisticated and dynamic spatial interaction models, including refined embedding techniques for spatiotemporal condition graphs and improved strategies for handling multimodal information during prediction. Lastly, our model has so far been validated only on the Climate2GPP dataset; additionally, we plan to test our method on simulated datasets and other real-world tasks to further verify its generalizability and robustness.

References

Arik, S. Ö., Pfister, T., 2019. TabNet: Attentive Interpretable Tabular Learning. *CoRR*, abs/1908.07442. http://arxiv.org/abs/1908.07442.

Breiman, L., 2001. Random forests. *Machine learning*, 45, 5–32.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. *European conference on computer vision*, Springer, 213–229.

Chen, J., Yan, J., Chen, Q., Chen, D. Z., Wu, J., Sun, J., 2024. Excelformer: A neural network surpassing gbdts on tabular data.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Du, Z., Wang, Z., Wu, S., Zhang, F., Liu, R., 2020. Geographically neural network weighted regression for the accurate estimation of spatial non-stationarity. *International Journal of Geographical Information Science*, 34(7), 1353–1377.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A., 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.

Fotheringham, A. S., Brunsdon, C., Charlton, M., 2009. Geographically weighted regression. *The Sage handbook of spatial analysis*, 1, 243–254.

Fotheringham, A. S., Crespo, R., Yao, J., 2015. Geographical and temporal weighted regression (GTWR). *Geographical Analysis*, 47(4), 431–452.

Friedl, M., Sulla-Menashe, D., 2022. Modis/terra+aqua land cover type yearly l3 global 0.05deg cmg v061.

Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A., 2021. Revisiting Deep Learning Models for Tabular Data. *CoRR*, abs/2106.11959. https://arxiv.org/abs/2106.11959.

Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Hu, W., Yuan, Y., Zhang, Z., Nitta, A., Cao, K., Kocijan, V., Leskovec, J., Fey, M., 2024. PyTorch Frame: A Modular Framework for Multi-Modal Tabular Learning. *arXiv preprint arXiv:2404.00776*.

Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F., 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. H. D. III, A. Singh (eds), *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 119, PMLR, 5156–5165.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Lu, Q., Liu, H., Wei, L., Zhong, Y., Zhou, Z., 2024. Global prediction of gross primary productivity under future climate change. *Science of The Total Environment*, 912, 169239.

Muñoz-Sabater, J., Dutra, E., Agustí-Panareda, A., Albergel, C., Arduini, G., Balsamo, G., Boussetta, S., Choulga, M., Harrigan, S., Hersbach, H. et al., 2021. ERA5-Land: A state-of-the-art global reanalysis dataset for land applications. *Earth system science data*, 13(9), 4349–4383.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A., 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Qin, L., Zhu, L., Liu, B., Li, Z., Tian, Y., Mitchell, G., Shen, S., Xu, W., Chen, J., 2024. Global expansion of tropical cyclone precipitation footprint. *Nature Communications*, 15(1), 4824.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y., 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 127063.

Vaswani, A., 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, S., Gao, K., Zhang, L., Yu, B., Easa, S. M., 2024. Geographically weighted machine learning for modeling spatial heterogeneity in traffic crash frequency and determinants in US. *Accident Analysis & Prevention*, 199, 107528.

Wu, S., Wang, Z., Du, Z., Huang, B., Zhang, F., Liu, R., 2021. Geographically and temporally neural network weighted regression for modeling spatiotemporal non-stationary relationships. *International Journal of Geographical Information Science*, 35(3), 582–608.

Zhang, Y., Kong, D., Gan, R., Chiew, F. H., McVicar, T. R., Zhang, Q., Yang, Y., 2019. Coupled estimation of 500 m and 8-day resolution global evapotranspiration and gross primary production in 2002–2017. *Remote sensing of environment*, 222, 165–182.