Deep Spatial Prediction via Heterogeneous Multi-Source Self-Supervision

MINXING ZHANG, Emory University, USA
DAZHOU YU, Emory University, USA
YUN LI, Emory University, USA
LIANG ZHAO, Emory University, USA

Spatial prediction is to predict the values of the targeted variable, such as PM2.5 values and temperature, at arbitrary locations based on the collected geospatial data. It greatly affects the key research topics in geoscience in terms of obtaining heterogeneous spatial information (e.g., soil conditions, precipitation rates, wheat yields) for geographic modeling and decision-making at local, regional, and global scales. In-situ data, collected by ground-level in-situ sensors, and remote sensing data, collected by satellite or aircraft, are two important data sources for this task. In-situ data are relatively accurate while sparse and unevenly distributed. Remote sensing data cover large spatial areas but are coarse with low spatiotemporal resolution and prone to interference. How to synergize the complementary strength of these two data types is still a grand challenge. Moreover, it is difficult to model the unknown spatial predictive mapping while handling the trade-off between spatial autocorrelation and heterogeneity. Third, representing spatial relations without substantial information loss is also a critical issue. To address these challenges, we propose a novel Heterogeneous Self-supervised Spatial Prediction (HSSP) framework that synergizes multi-source data by minimizing the inconsistency between in-situ and remote sensing observations. We propose a new deep geometric spatial interpolation model as the prediction backbone that automatically interpolates the values of the targeted variable at unknown locations based on existing observations by taking into account both distance and orientation information. Our proposed interpolator is proven to both be the general form of popular interpolation methods and preserve spatial information. The spatial prediction is enhanced by a novel error-compensation framework to capture the prediction inconsistency due to spatial heterogeneity. Extensive experiments have been conducted on real-world datasets and demonstrated our model's superiority in performance over state-of-the-art models.

CCS Concepts: • **Computing methodologies** → **Neural networks**.

Additional Key Words and Phrases: Spatial Prediction, Spatial Interpolation, Multi-Source Data Integration, In-Situ Data, Remote Sensing Data, Error-compensation, Geometric Representation, Deep Learning, Self-supervision

1 INTRODUCTION

The problem of spatial prediction is to utilize available values at locations to estimate the values of a targeted variable at these locations or other locations [53]. It is also known as predictive mapping. As one of the key research topics in geoscience, spatial prediction plays a vital role in obtaining heterogeneous spatial information (e.g., soil conditions, precipitation rates, wheat yields) for geographic modeling and decision-making at local, regional, and global scales [39]. Take ambient air pollution concentration as an example: estimating the full coverage of air pollution in high spatiotemporal resolution could be beneficial in understanding the spatiotemporal

Authors' addresses: Minxing Zhang, minxing.zhang@emory.edu, Emory University, USA; Dazhou Yu, dazhou.yu@emory.edu, Emory University, USA; Yun Li, Emory University, USA, yli230@emory.edu; Liang Zhao, Emory University, USA, liang.zhao@emory.edu.

2374-0353/2023/6-ART $15.00
https://doi.org/10.1145/3605358

patterns of air pollution distribution and discovering the precursors of extreme air pollution events for hazard preparedness and mitigation efforts [44].

Generally, spatial prediction consists of three steps. Firstly, a set of sample datasets are collected at discrete locations in the study area. Secondly, the relationships between the targeted and independent variables are discovered by analyzing the collected samples. Finally, the learned relationships are used to predict the values of the targeted variable at locations we are interested in [53]. Thus, collecting sample data and characterizing the relationships between independent and targeted variables play a vital role in the spatial prediction task.

Regarding data collection for spatial prediction, in-situ data and satellite observations are two important data sources due to the prominent development of in-situ sensors and remoting sensing instruments. In-situ sensors directly contact the medium that they measure, such as temperature, wind, and precipitation. The measured values are relatively accurate and available with a high temporal resolution, e.g., every hour or even a few minutes. However, in-situ monitoring stations are inevitably sparsely and unevenly distributed due to the high cost of deployment. Hence, only limited observations could be collected for situation awareness and spatial analysis, which would result in biased predictions. Regarding remote sensing data, remote sensors could measure the medium they sensed at a large scale. Since remote sensors measure value by satellite or aircraft at a distance from the ground, the collected data are often coarse with low spatiotemporal resolution. They are also prone to interference (e.g., clouds and bad weather) and sensitive to weak light (e.g., in the nighttime). In all, considering the respective drawbacks and complementary strengths of in-situ and remote sensing data, multi-source data fusion has recently aroused wide attention and is an active yet challenging research problem.

Spatial prediction based on multi-source data fusion is still a highly open research domain due to several major challenges: **1) Heterogeneity of in-situ and remote sensing data.** As highlighted in the region surrounded by the TSAgreen dotted line in Figure 1, in-situ data have high precision yet exist sparsely in discrete space, while remote sensing data are coarse-grained but cover continuous space regions. Only the locations with both sources will enjoy their integration, and most existing research adds values from satellite observations around locations where in-situ sensors are located into data samples for relationship learning. However, this treatment ignores valuable information in other grid cells of the satellite data, leading to substantial information loss especially considering in-situ locations are highly sparse. **2) Complex, unknown, and spatial autoregressive relation between input and targeted variables.** As highlighted in the region surrounded by the yellow dotted line in Figure 1, existing spatial prediction methods consist of i) statistical methods that learn the linear relationships among variables [1, 36], and ii) machine learning methods that aim to discover the nonlinear relationships for spatial prediction [20, 25]. Statistical methods usually require the preassumption of prescribed geostatistical models, which may not be the true sophisticated and unknown data distribution. Machine learning-based methods instead aim at fitting the relation by the data, yet fall short in sufficiently considering geographical principles, especially spatial autoregressive, spatial heterogeneity, and their trade-off. Specifically, *spatial autoregressive* [5] says that different geo-locations are correlated according to their spatial relation, while *spatial heterogeneity* [17] instead claims the pattern of an attribute at one location is different from its surrounding. It is imperative yet highly challenging to propose a technique that can automatically learn and quantify the underlying effects of spatial autoregressive and heterogeneity. **3) Information loss in spatial-relation representation.** As highlighted in the region surrounded by the orange dotted line in Figure 1, as spatial prediction requires predicting the value of the targeted variable in a new location according to those observed in the existing locations, how to represent the spatial relation between the existing locations and the new location is substantial yet currently not executed with perfection. Existing works typically utilize spatial distance as the indicator of spatial relation, which cannot fully reflect the complete spatial relation among locations. For example, suppose a new location has the same distance to four existing locations; hence, the existing ways consider it the average of the four locations' values. However, suppose three out of the four locations are very close to each other and far away from the other one, then actually the neighborhood of the three locations are relatively overestimated, and hence their weights should have been

normalized downward. However, existing works that only consider distance but not orientation cannot sense such a situation in the first place.

To address these challenges, we propose a novel Heterogeneous Self-supervised Spatial Prediction (HSSP) framework that minimizes the inconsistency between in-situ and remote sensing observations. We propose a new deep geometric spatial interpolation framework that automatically interpolates unknown locations' values of the targeted variable based on existing locations by leveraging both distance and orientation relations among spatial locations. Our proposed interpolator is proven to both be the general form of popular interpolation techniques and preserve spatial information. Moreover, we propose an error-compensation framework to capture the prediction inconsistency due to spatial heterogeneity. In short, the key contributions of this paper are summarized as follows:

(1) **Developing a new heterogeneous multi-source fusion framework for spatial prediction.** A geometric spatial interpolation model is built based on in-situ data, which is learned together with the spatial pattern collected from remote sensing via self-supervised learning. Our framework synergizes the complementary strengths of in-situ and remote sensing data and can predict with high spatial precision and coverage simultaneously.
(2) **Proposing a deep error-compensation model to jointly handle spatial relation and heterogeneity.** It leverages additional environmental ancillary attributes for different locations to characterize the error in spatial interpolation due to spatial heterogeneity.
(3) **Designing a deep geometric interpolation method to learn the underlying spatial distribution of the targeted variable's values.** Instead of using prescribed distribution or predefined kernel functions, the model we propose automatically interpolates the values of the targeted variable at unknown locations based on existing observations by learning the underlying spatial relations. Such learning is achieved by a new spatial relation representation based on distance and orientation information without substantial geometric information loss with theoretical guarantees. Our proposed interpolator is proven to be the general form of popular interpolation techniques.
(4) **Conducting comprehensive experimental analysis to validate the effectiveness of the proposed model.** Extensive experiments on two real-world datasets, the PM2.5 Concentration Dataset and the Ambient Temperature Dataset, demonstrate that our proposed framework achieves superior results in spatial prediction both qualitatively (on average, a 20% decrease in RMSE and 24% decrease in MAE on the PM2.5 Concentration Dataset and a 79% decrease in RMSE and 85% decrease in MAE on the Ambient Temperature Dataset) and quantitatively (wider coverage of spatial locations).

The rest of the paper is organized as follows. We formulate the problem of spatial prediction and introduce the in-situ data and remote sensing data in Section 2. Then, we present our Heterogeneous Self-supervised Spatial Prediction (HSSP) framework in Section 3. We further elaborate on our point-based predictor in Section 4. We evaluate the effectiveness of our model in Section 5 and report related work in Section 6. Finally, we conclude the paper in Section 7.

2 PROBLEM SETTING

For the spatial prediction task, both in-situ data (point-based) and remote sensing data (raster-based) are widely utilized as input to build up the model.

In-situ data can be treated as a finite set of points S, $\{s_1, s_2, ..., s_n\}$, and for any arbitrary point s_i in the set, we are given 1) the geographical coordinate loc_i of point $s_i \in \mathbb{R}^{2\times 1}$, which denotes the latitude and longitude, 2) the value of the targeted variable $y_i \in \mathbb{R}$ (it can be any variable in interest such as PM2.5 value and temperature), and 3) a set of k ancillary attributes Z_i, $\{z_{i1}, z_{i2}, ..., z_{ik}\}$, at point s_i (the ancillary attributes can be any other feature information such as wind, precipitation, etc.).

Remote sensing data can be treated as pixelated (or gridded) images where each pixel is associated with a specific geographical location, and the pixel's value can be any ancillary attribute. Thus, each input raster data in our setting can be defined as a multi-channel image $\mathrm{X} \in \mathbb{R}^{l \times w \times k}$, where l and w denote the length and width of the image in pixels, respectively, and $X_{i,j,k}$ is the k-th feature of the pixel at i-th row and j-th column. We use $c_{i,j} \in \mathbb{R}^2$ to denote the coordinate of the center of this pixel.

However, challenges exist for both data sources: the sparse and uneven distribution of the in-situ data renders the collected observations limited and biased, and remote sensing data are coarse with low spatiotemporal resolution and prone to interference. How to synergize the complementary strength and conquer the respective drawbacks of these two data types is still a grand challenge. Other challenges lie in the difficulty of modeling the unknown spatial predictive mapping while handling the trade-off between spatial autocorrelation and heterogeneity and perfectly representing spatial relations without substantial information loss.

Based on the above notation and challenges, the problem of spatial prediction is defined as follows:

Definition 2.1. ***Spatial Prediction.*** Given a finite set of spatial points S, $\{s_1, s_2, ..., s_i, ..., s_n\}$, with geographical coordinates $\mathbf{C} = [loc_1, loc_2, ..., loc_i, ..., loc_n]^T$ at each point, the k ancillary attributes $\mathbf{Z} = [Z_1, Z_2, ..., Z_i, ..., Z_n]^T$ at each point (where $Z_i = [z_{i1}, z_{i2}, ..., z_{ik}]^T$ contains the values of the k ancillary attributes at point s_i), and the corresponding targeted variable $\mathbf{Y} = [y_1, y_2, ..., y_i, ..., y_n]^T$, the spatial prediction problem aims to learn a model (or function) $m(\cdot)$ such that $\mathbf{Y} = m(\mathbf{C}, \mathbf{Z})$. Once the model is learned, it can be used to predict the values of the targeted variable at other spatial points based on their geographical locations and ancillary features.

3 THE HETEROGENEOUS SELF-SUPERVISED SPATIAL PREDICTION FRAMEWORK

3.1 Overall Framework

To handle both in-situ data (point-based) and remote sensing data (raster-based) as input, our model can be treated as the fusion of two modules: deep spatial prediction using in-situ data (point-based predictor) and deep spatial prediction using remote sensing data (raster-based predictor).

Deep Spatial Prediction Using In-Situ Data. Our point-based predictor, which is depicted in Figure 1(a), takes any coordinate $\in \mathbb{R}^2$ as well as the given in-situ data as input and outputs the value of the targeted variable on the targeted location. The ideas of deep geometric interpolation and error-compensation are leveraged in the point-based predictor, where we propose a multilayer perceptron-based framework to interpolate the targeted variable's value on that location using its t neighbors (points in the in-situ data that are within t-th closest to the targeted point). Besides, the same rationale is leveraged to interpolate the group of ancillary attributes on that location, and the interpolated ancillary attributes are extended to provide an error-compensation, as shown in Figure 1 "Error-compensation Framework," to make the predictions from the point-based predictor and the raster-based predictor as similar as possible. The explanation of the point-based predictor is elaborated in Section 4.

Deep Spatial Prediction Using Remote Sensing Data. Our raster-based predictor, which is depicted in Figure 1(b), takes the remote sensing data as input and extends a convolutional neural network (CNN) to handle the images, as CNN is powerful and efficient in terms of automatically detecting and extracting relevant ancillary features and proximity of pixels to each other that assist in predicting the value of the targeted variable. The idea of image-to-image translation is leveraged in our raster-based predictor. More specifically, for the raster-based predictor, we take the remote sensing data $\mathrm{X} \in \mathbb{R}^{l \times w \times k}$ as input and output a heatmap of the targeted variable's values with size $\in \mathbb{R}^{l \times w \times 1}$, where each pixel in the heatmap is associated with a specific geographical location. Each pixel can be treated as the center point of the associated small region of the entire raster image. The pixel's

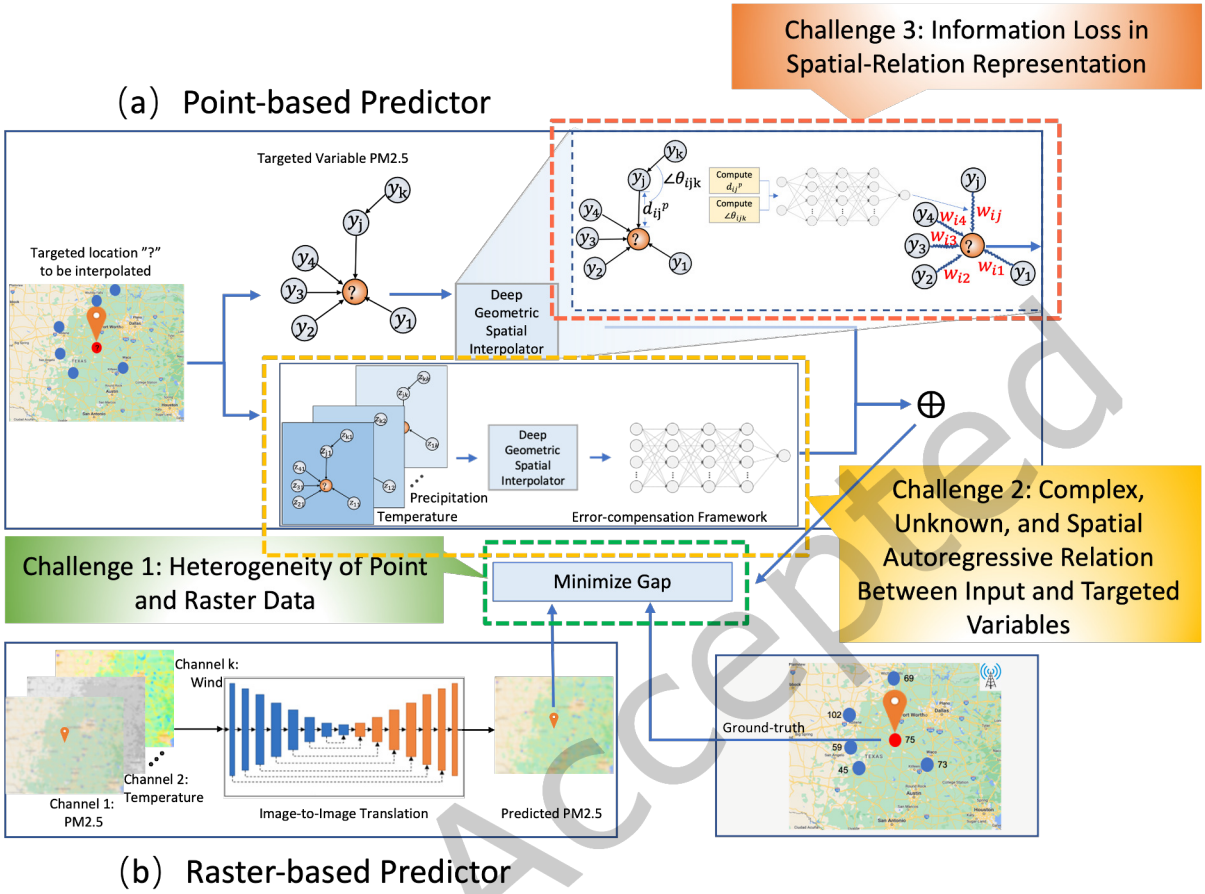


Fig. 1. The Heterogeneous Self-Supervised Spatial Prediction Framework With Highlighted Challenges and Contributions

value is the value of the targeted variable at that location, and all the points within the associated regions share the same value of the targeted variable as that of the region's center point.

3.2 Training Objective

Our training objective consists of two segments: 1) given that the in-situ data have high precision yet exist sparsely and remote sensing data are coarse with a low spatiotemporal resolution, the first training objective aims to synergize the complementary strengths of both two data sources. To fully utilize the spatial dependencies of the targeted variable, for instance, air pollution, captured by both in-situ observation and remotely sensed data, our model considers point data and raster data separately and synergizes their strengths via the loss function. In fact, these two types of data can be merged in the data processing step using spatiotemporal collocation methods. This involves assigning the value of a point data to the grid cell that contains it, allowing the creation of train and test data sets consisting of remotely sensed values, ancillary values, and in-situ values. However, it is important to note that merging raster and point data comes with a cost, as important information may be lost in the process. Specifically, the spatial dependencies of the targeted area, for instance, air pollution, captured by

Table 1. Notation Table

Symbols:	
S	A finite set of geographical points that have ground-truth values of the targeted variable
s_i	A geographical point with corresponding geographical coordinate
y_i	Ground-truth value of the targeted variable at point s_i
$\hat{y}_i$	Predicted value of the targeted variable at point s_i
Z_i	Ground-truth values of a set of k ancillary attributes at location s_i
$\hat{Z}_i$	Predicted values of a set of k ancillary attributes at location s_i
z_{ik}	Ground-truth value of the k-th ancillary attribute at location s_i
$\hat{z}_{ik}$	Predicted value of the k-th ancillary attribute at location s_i
w_{ij}	Weight (contribution) of the neighbor point s_j to the targeted point s_i
d_{ij}	Distance from the neighbor point s_j to the targeted point s_i
θ_{ijk}	Angle among the points s_i, s_j, and s_k, where s_i is the targeted point, s_j is the neighbor point of s_i, and s_k is the neighbor point of s_j
Proposed Models:	
$f(\cdot)$	Function of our point-based predictor
$g(\cdot)$	Function of our raster-based predictor
$q(\cdot)$	Function of our point-based interpolator
$a(\cdot)$	Function of our weight computation framework
$\phi(\cdot)$	Function of our proposed framework to compute distance information
$\eta(\cdot)$	Function of our proposed framework to compute orientation information
$e(\cdot)$	Function of our error estimation model
Parameters:	
λ	Balancing factor
p	Exponent factor

satellite observations cannot be fully utilized if these two types of data sets are merged together. The proposed loss function necessitates the alignment of the predictions via two data sources as strong as possible, specific to every point in space. Moreover, the outputs of two data sources are the same physics variables; hence, we require them to be the same. We cannot give such strong enforcement for hidden layers because we do not have their physical meaning. If we want to enforce the alignment of hidden layers, the expressiveness of the model will be reduced.

Thus, we aim to minimize the self-supervision loss between the predictions using the in-situ data and that using the remote sensing data, and 2) to take advantage of the in-situ data, which have ground-truth values of the targeted variable, for each point s_i in the set of the in-situ data S, we treat it as unknown and utilize other points in the set $s_j \in S - \{s_i\}$ to conduct deep spatial prediction using the point-based predictor. We try to minimize the loss of the output with respect to the given ground-truth labels. The training objective can be written as:

$$\arg \min_{g(\cdot), f(\cdot)} \sum_{r \in R(X)} \mathcal{L}(y_r, [g(X)]_r) + \sum_i \mathcal{L}(y_i, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}})) \tag{1}$$

where $R(X)$ denotes the spatial region that the raster data X covers, y_r denotes the ground-truth value of the targeted variable at pixel r, and the function of our raster-based predictor is defined as $g(\cdot)$, y_i denotes the ground-truth value of the targeted variable at point s_i, Z_i denotes a group of k ancillary attributes, and the function

of our point-based predictor is defined as $f(\cdot)$, which consists of deep geometric interpolation framework and error-compensation framework and is detailed later in Section 4.

However, y_r is not directly learnable because we do not have observation data on every pixel but only at stations, so we lack labeled data. To address this problem, we resort to the upper bound of $\sum_r \mathcal{L}(y_r, [g(X)]_r)$ according to the triangle inequality theorem in Euclidean geometry as follows:

$$\sum_{r \in R(X)} \mathcal{L}(y_r, [g(X)]_r) \leq \sum_{r \in R(X)} \mathcal{L}([g(X)]_r, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}}) + \sum_i \mathcal{L}(y_i, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}})) \quad (2)$$

By inducing Equation 2 into Equation 1, we have an upper bound for Equation 1:

$$\arg \min_{g(\cdot), f(\cdot)} \sum_{r \in R(X)} \mathcal{L}([g(X)]_r, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}}) + 2 \cdot \sum_i \mathcal{L}(y_i, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}})) \quad (3)$$

as $\mathcal{L}(y_i, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}}))$ is considered as an unknown constant to $g(\cdot)$

We can further $g(\cdot)$ into:

$$\arg \min_{g(\cdot), f(\cdot)} \lambda \cdot \sum_{r \in R(X)} \mathcal{L}([g(X)]_r, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}}) + \sum_i \mathcal{L}(y_i, f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}})) \quad (4)$$

where λ is a tunable balancing factor.

4 DEEP SPATIAL PREDICTION USING IN-SITU DATA

Spatial interpolation is an open research problem with two big challenges that have not been well addressed by existing works. The first one is that the true spatial distribution of the targeted variable is typically sophisticated and unknown, which significantly challenges the existing works that use prescribed distribution or predefined kernel functions. Second, the targeted variable's value of a spatial location is also dependent on other spatial attributes and confounding, but such spatial heterogeneity is necessary yet challenging to be considered in spatial interpolation. To address both these two challenges, we propose a deep spatial prediction network where we first propose a deep geometric spatial interpolation network that can automatically learn the underlying spatial distribution for spatial interpolation. This part is elaborated in Section 4.1 and in Figure 1 "Deep Geometric Spatial Interpolator." Then the consideration of spatial heterogeneity is further leveraged to fill the gap between the interpolated value from the deep spatial interpolation network and the true value. To achieve this, we develop an error-compensation mechanism that predicts and then maps the ancillary attributes of a spatial location to its interpolation error, which also help explains spatial heterogeneity.

4.1 Deep Geometric Spatial Interpolation Framework

In this section, we propose our new deep geometric spatial interpolation framework that can take into account both distance and orientation information to fully reflect the complete spatial relations among locations. Different from popular spatial interpolation methods, which use prescribed distribution or predefined kernel functions, our model leverages the power of neural networks to automatically learn the underlying spatial distribution of the targeted variable. Hence this demonstrates our model's expressive power in automatically selecting and learning distributions among or beyond traditional prescribed-based spatial interpolation methods.

We propose a novel multilayer perceptron-based deep geometric spatial interpolation framework $q(\cdot)$ to interpolate the value of the targeted variable y_i at any arbitrary location s_i, which is reflected in Figure 1 "Deep Geometric Spatial Interpolator." Our new deep geometric interpolation method automatically learns the underlying

spatial distribution of the value of the targeted variable, which can be expressed as:

$$\hat{y}_i = q(s_i, \{s_j, y_j\}_{s_j \in S - \{s_i\}}) \tag{5}$$

where $\hat{y}_i$ denotes the interpolated value of the targeted variable at targeted point s_i

As indicated in Equation 5, we leverage the known values at the neighbor points $s_j \in S - \{s_i\}$ of the targeted point s_i to interpolate the value of the targeted variable. Therefore, our deep geometric spatial interpolator can be generalized as the computation of weights of each neighbor point and weighted sum followed by normalization, which can be expressed as:

$$\hat{y}_i = \frac{\sum_{j=1}^{t} w_{ij} y_j}{\sum_{j=1}^{t} w_{ij}} \tag{6}$$

where w_{ij} denotes the weight (contribution) of the neighbor point s_j to the targeted point s_i and y_j denotes the value of the targeted variable at the neighbor point s_j. The numerator is the weighted sum of the values of the targeted variable at these t neighbor locations. The denominator applies normalization.

As our proposed deep geometric spatial interpolator leverages the known information of the neighbor points, how to fully represent the geometric relations between the neighbors and the targeted point without substantial information loss is a critical issue. Our proposed interpolator leverages a new spatial relation representation based on distance and orientation information.

The reasons behind the consideration of both distance and orientation information are illustrated as follows. The distance information considers the ideas behind spatial autocorrelation and spatial heterogeneity, which state that 1) everything in space is connected and 2) the closer the distance, the larger the connection.

The orientation information considers that input distance only measures the relative difference between each neighbor and the interpolated point. Two neighbors sharing the same distance to the interpolated point but located at different places may have different weights. For example, suppose a new location has the same distance to four existing locations y_1, y_2, y_3, y_4, as expressed in Figure 1(a), and hence the existing ways consider equal weight for all four neighbors. However, suppose three out of the four locations y_2, y_3, y_4 are very close to each other and far away from the other one y_1, then actually, the neighborhood of the three locations are relatively overestimated, and hence their weights should have been normalized downward. Thus, the inclusion of the angle discounts the weight of less important neighbors by taking direction into consideration.

To achieve the goal of fully reflecting the complete spatial relation among locations by taking into account both distance and orientation information, our proposed deep geometric interpolator consists of a novel framework $a(\cdot)$ to compute the weight of each neighbor, which can be written as:

$$w_{ij}^{(k)} = a(\phi(s_i, s_j), \eta(s_i, s_j, s_k)) \tag{7}$$

where the output $w_{ij}^{(k)}$ is the weight of the neighbor point s_j to the interpolated point s_i, $\phi(\cdot)$ is the function to compute distance information and $\eta(\cdot)$ is the function to compute orientation information, s_j is the neighbor of the targeted point s_i where $s_j \in S - \{s_i\}$, s_k is the neighbor of the neighbor point s_j where $s_k \in S - \{s_i, s_j\}$.

More specifically, our distance computation function $\phi(\cdot)$ and orientation computation $\eta(\cdot)$ function can be respectively written as,

$$d_{ij}^{p} = \phi(s_i, s_j) = ||s_j - s_i||_2^p \tag{8}$$

which is the Euclidean norm of the vector v_{ij}, which measures the distance from neighbor point s_j to interpolated point s_i, to the power of p, where p is a hyperparameter, denoting the exponent factor.

The input angle θ_{ijk} can be expressed as:

$$\theta_{ijk} = \eta(s_i, s_j, s_k) = arccos(\langle \frac{v_{ij}}{d_{ij}}, \frac{v_{jk}}{d_{jk}} \rangle) \cdot \langle n_{ijk}, n_{xy} \rangle$$
$$n_{ijk} = \frac{v_{ij} \times v_{jk}}{||v_{ij} \times v_{jk}||_2}, n_{xy} = n_x \times n_y, v_{ij} = \overrightarrow{s_i s_j}, v_{jk} = \overrightarrow{s_j s_k} \tag{9}$$

where point s_k is the neighbor of point s_j that forms the smallest $\theta_{ijk} \in [-\pi, \pi)$, $\times$ denotes the cross product operation.

After computation of d_{ij}^p and θ_{ijk} by function $\phi(\cdot)$ and function $\eta(\cdot)$ respectively, our weight computation framework $a(\cdot)$ consists of a stack of U fully connected layers FC_u, where $u = 1, 2, ..., U$, followed by nonlinear activation function σ, where the input to the first linear layer is the concatenated output of function $\phi(\cdot)$ and function $\eta(\cdot)$, the computed distance d_{ij}^p and angle θ_{ijk}, and it can be written as:

$$w_{ij}^{(k)} = a(\phi(s_i, s_j), \eta(s_i, s_j, s_k))$$
$$= a(d_{ij}^p || \theta_{ijk}) = FC_U(\sigma(FC_{U-1}...\sigma(FC_2(\sigma(FC_1(d_{ij}^p || \theta_{ijk})))))) \tag{10}$$

Then, we introduce two important theorems that reflect the outstanding properties of our proposed interpolator.

Theorem 4.1. *Our proposed deep geometric spatial interpolation framework $q(\cdot)$ is the general form of popular spatial interpolation methods.*

Proof of Sketch: We prove that popular spatial interpolation methods are the special cases of our proposed interpolator under special parameter settings. By adjusting the parameters of our deep geometric interpolation framework, we enable our proposed model to output the same values of popular spatial interpolation methods. The details will be elaborated in Appendix A.1. ■

Theorem 4.2. *Our proposed weight computation framework $a(\cdot)$ in terms of computing the weight (contribution) of any neighbor point s_j to the targeted point s_i are rotation and translation invariant.*

Proof of Sketch: We prove that our proposed weight computation framework $a(\cdot)$ is rotation and translation invariant by proving that all the inputs to $a(\cdot)$, the computed distance $d_{ij}^p \in [0, \infty)$ and angle $\theta_{ijk} \in [-\pi, \pi)$, are rotation and translation invariance by imposing generalized rotation and translation in 2D space on them. The details will be elaborated in Appendix A.2. ■

4.2 Error-compensation Framework

The spatial interpolation of the targeted variable's values based on our deep geometric spatial interpolation method effectively learns and leverages the spatial correlation and distribution. As mentioned above, spatial correlation may not explain all the spatial patterns because different spatial locations may have different characteristics, so spatial distribution and rule may not be homogeneous. To further fill this gap, we propose an error-compensation framework that leverages additional environmental ancillary attributes for different locations to characterize the error in spatial interpolation due to spatial heterogeneity. More concretely, point-based predictor $f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}})$ is expressed as the sum of the output of our deep geometric interpolation model $q(s_i, \{s_j, y_j\}_{s_j \in S - \{s_i\}})$ and the estimated error ϵ, which is the output of the error estimation model $e(\hat{Z}_i)$, as follows:

$$f(s_i, \{s_j, y_j, Z_j\}_{s_j \in S - \{s_i\}}) = q(s_i, \{s_j, y_j\}_{s_j \in S - \{s_i\}}) + e(\hat{Z}_i) \tag{11}$$

where $\hat{Z}_i$ contains the estimated values of the group of k ancillary attributes, namely $\{\hat{z}_{i1}, \hat{z}_{i2}, ..., \hat{z}_{ik}\}$, at location s_i.

The error estimation model $e(\cdot)$ is a framework with a stack of fully connected layers; more specifically, it consists of ancillary attributes interpolation and error prediction. For ancillary attributes interpolation, as we aim to interpolate the value of the targeted variable at any location s_i, the ancillary attributes may not be available. Thus, our model is extended to interpolate the unknown group of ancillary attributes $\hat{Z}_i$ at point s_i. For each ancillary attribute, the idea of deep geometric interpolation is leveraged to compute the value of the ancillary attribute at the targeted location s_i, and therefore, the entire group of interpolated ancillary attributes $\hat{Z}_i$ can be expressed as:

$$\hat{Z}_i = q(s_i, \{s_j, Z_j\}_{s_j \in S - \{s_i\}}) \tag{12}$$

where Z_j contains the ground-truth values of the group of k ancillary attributes at the known locations.

The model architecture to interpolate each ancillary attribute is the same as that in Section 4.1. We leverage the idea of deep geometric spatial interpolation to interpolate the ancillary attributes by leveraging the weighted sum and normalization, as expressed in Equation 6. To compute the weight $w_{ij}^{(k)}$ of neighbor point s_j, the model first leverages function $\phi(\cdot)$, expressed in Equation 8, and function $\eta(\cdot)$, expressed in Equation 9, to compute the geometric information d_{ij}^p and angle θ_{ijk} based on the geographical location of the targeted location and its neighbors, which are further treated as inputs into the weight computation framework $a(\cdot)$, expressed in Equation 10, and the corresponding output is the weight $w_{ij}^{(k)}$ of each neighbor s_j to the targeted point s_i.

Thus, the strengths of acting as the general form of the popular interpolation methods, which automatically learns the underlying spatial distribution without using prescribed data distribution and kernel functions, and the full representation of geometric spatial relations are preserved.

For error prediction, we assume that the interpolated ancillary attributes may affect the distribution of the values of the targeted variable and therefore provide extra information which would benefit our model. Thus, the interpolated ancillary attributes are extended to predict the estimated error ϵ in spatial interpolation due to spatial heterogeneity, which is written as $e(\hat{Z}_i)$, where $\hat{Z}_i$ is the set of interpolated ancillary attributes $\{\hat{z}_{i1}, \hat{z}_{i2}, ..., \hat{z}_{ik}\}$ at point s_i.

5 EVALUATION

In this section, the performance of the proposed model HSSP is evaluated using two real-world datasets. First, the experimental setup is introduced. The effectiveness of HSSP is then evaluated against eleven existing methods. Then, several ablation studies are conducted to validate the effectiveness of different components of our HSSP framework. Finally, the impact of important parameters on our HSSP framework and its scalability is explored.

5.1 Experimental Setup

Data. We evaluate our proposed methods using two real-world datasets in the United States: the PM2.5 Concentration Dataset (AQ-PM2.5) and the Ambient Temperature Dataset.

(1) PM2.5 Concentration Dataset: This dataset is derived from fusing in-situ purple air sensor data, the Moderate Resolution Imaging Spectroradiometer (MODIS) TERRA and AQUA satellite observations [34], and MERRA-2 reanalysis data [15] across the continental united states. It contains MAIAC [38] Aerosol optical depth (AOD) value, meteorological variables such as humidity, surface pressure, wind speed, and the corresponding ambient PM2.5 value in the location. Interpolation methods could help create high spatiotemporal coverage of ambient PM2.5 products for better air pollution mitigation strategies and urban planning.

Table 2. Datasets Summary

Dataset	AQ-PM2.5		Temperature	
Data Source Type	Point	Raster	Point	Raster
Targeted Variable	PM2.5	PM2.5	Temperature	Temperature
Location Representation	Coordinate (latitude, longitude)	Pixel position in raster	Coordinate (latitude, longitude)	Pixel position in raster
Number of Ancillary Attributes	9	9	9	9
Dataset Size	155 points	$2485 * 5781$ pixels	90 points	$70 * 70$ pixels
Average Train Size	124 points	$1.149 * 10^7$ pixels	71 points	3920 pixels
Average Test Size	31 points	$1.437 * 10^6$ pixels	19 points	490 pixels

(2) Ambient Temperature Dataset: Ambient temperature datasets are collected in the Los Angeles region to analyze the spatiotemporal pattern of extreme heat events in one of the largest cities in the world [34]. The sparsely distributed in-situ air temperature data are collected from Weather Underground. Land surface temperature (LST) values are derived from MODIS satellite observations. Meteorological variables such as wind and humidity are also collected from the MERRA-2 dataset.

We report the average performance of all days in December 2019. The summary of both in-situ data and remote sensing data is shown in Table 2. The size of the two datasets is relatively small as they were obtained from two sources: in-situ sensors and satellite observations. In-situ sensors are known to be expensive to install and maintain, resulting in sparse and uneven distribution, particularly in rural regions. As a result, access to in-situ sensor observations is restricted, and in our study, we have gathered data from all sensors situated in California State. To mitigate the problems caused by limited dataset size, we adopt an 80/20 split for train and test sets in terms of data splitting. 10-fold cross-validation is leveraged for hyperparameter tuning.

Comparison Methods. We compare our proposed framework to three types of state-of-the-art approaches for spatial prediction. Firstly, our point-based predictor can be treated as the fusion of a deep geometric spatial interpolation framework and an error-compensation framework. Our proposed deep geometric spatial interpolation framework is proven to be the general form of popular spatial interpolation methods and, therefore, theoretically outperforms them by automatically selecting and learning underlying spatial distributions instead of using prescribed distributions or predefined kernel functions. For verification, we compare our proposed framework to a group of popular spatial interpolation methods, including Inverse Distance Weighting (IDW) [27], Kriging [13], Radial Basis Function Interpolation (RBF) [22], and Nearest Neighbor Interpolation (Nearest Neighbor) [13]. Moreover, we add two recently published interpolator methods reported as advanced interpolators. They are Value Propagation-based Spatial Interpolation (VPint) [2] and Random Forest Spatial Interpolation (RFSI) [45].

Then, both our raster-based predictor, which leverages additional environmental ancillary attributes to estimate the values of the targeted variable, and error-compensation framework, which takes into account the inconsistency in spatial interpolation from multi-source data due to spatial heterogeneity, extend the power of ancillary attributes to spatial prediction. For verification, we compare our proposed method to three ML-based spatial prediction methods, including Geographically Weighted Regression (GWR), Random Forest (RF), and Gradient Boosting (GB) [7, 20, 25], which use ancillary attributes to conduct spatial prediction. However, as we aim to estimate the values of the targeted variable at any location s_i, the ancillary attributes may not be available at that location. The unavailability of the ancillary attributes at any arbitrary location invalidates these ML-based spatial prediction methods that necessitate the knowledge of them. For a fair comparison, we first use the popular spatial interpolation method Inverse Distance Weighting to interpolate the ancillary attributes on all the test locations. Next, we test the performance of these three ML-based spatial prediction methods on the test set.

Finally, we compare our proposed method to two neural networks-based state-of-the-art spatial prediction methods: Artificial Neural Network (ANN) [51] and Geographically and Temporally Weighted Neural Network (GTWNN) [14]. Similar to the property of ML-based approaches, these two methods also require additional environmental ancillary attributes on the test locations. Thus, we use Inverse Distance Weighting to interpolate the ancillary and then use these models to conduct spatial prediction for a fair comparison.

In short, the following eleven methods are included in the performance comparison. For each model, we leverage 10-fold cross-validation for hyperparameter tuning. The detailed settings are reported after the introduction of each baseline method.

- Inverse Distance Weighting (IDW) [27]: the values of the targeted variable to unknown points are calculated as a weighted average of the values available at the known points. The weight of each known point is derived by the inverse of the corresponding distance. The number of neighbors is set to 20 for the PM2.5 Concentration Dataset and 15 for the Ambient Temperature Dataset; for both datasets, power is set to 1.0, reg is set to 0.0, and eps is set to 0.2.
- Kriging [13]: given a set of points S, with $\mathbf{s}$ is the vector containing the value of the target variable y of all the points $\in$ S, it constructs a covariance matrix $\underline{C}$. Then, a covariance vector $\mathbf{c}$ for the interpolated point s_i relative to the given set of points S is defined. It finally computes the weight vector $\mathbf{w}$, which contains the weights of all the points in the in-situ data with respect to the interpolated point s_i, by solving a linear least square problem. Kriging typically starts with a prior distribution over functions, and this prior takes the form of a Gaussian process. For the PM2.5 Concentration Dataset, variogram model is set to linear, weight is set to False, and nlags is set to 6; for the Ambient Temperature Dataset, variogram model is set to hole-effect, weight is set to True, and nlags is set to 2.
- Radial Basis Function Interpolation (RBF) [22]: a univariate radial function that maps distance to weight is utilized. The weights for each neighbor are computed by solving a linear system of equations. The multiquadric kernel is typically used by RBF interpolation. For the PM2.5 Concentration Dataset, smoothing is set to 0.5, and the number of neighbors is set to 8; for the Ambient Temperature Dataset, smoothing is set to 3, and the number of neighbors is set to 16. Kernel is set to linear for both two datasets.
- Nearest Neighbor Interpolation (Nearest Neighbor) [13]: the value of the targeted variable at an unknown point is determined by the value of the known point that has the shortest distance to the targeted point. Rescale is set to True for both two datasets.
- Value Propagation-based Spatial Interpolation (VPint) [2]: VPint operates locally but applies recursion to implicitly account for global spatial relationships in the entire system via Markov reward processes. N is set to 5 for the PM2.5 Concentration Dataset and 6 for the Ambient Temperature Dataset.
- Random Forest Spatial Interpolation (RFSI) [45]: RFSI takes spatial autocorrelation between observations into consideration and incorporates covariates representing neighbors' information in the random forest model. For the PM2.5 Concentration Dataset, the number of neighbors is set to 30, criterion is set to squared_error, max features is set to log2, bootstrap is set to True, oob_score is set to False, n_estimators is set to 500, max_depth is set to None, min_samples_split is set to 2, min_samples_leaf is set to 1, min_weight_fraction_leaf is set to 0.0, max_leaf_nodes is set to None, and min_impurity_decrease is set to 0.0. For the Ambient Temperature Dataset, the number of neighbors is set to 30, criterion is set to absolute_error, max_features is set to sqrt, bootstrap is set to True, oob_score is set to True, n_estimators is set to 400, max_depth is set to None, min_samples_split is set to 2, min_samples_leaf is set to 1, min_weight_fraction_leaf is set to 0.0, max_leaf_nodes is set to None, and min_impurity_decrease is set to 0.0.
- Geographically Weighted Regression (GWR) [7]: GWR extends the ordinary least squares regression and adds a level of modeling sophistication by enabling the relationships between the independent and

dependent variables to vary by locality. GWR constructs a separate ordinary least squares equation for every location in the dataset, which incorporates the dependent and explanatory variables of locations falling within the bandwidth of each targeted location. For both datasets, bandwidth is obtained using Sel_BW with criterion set to AIC for the PM2.5 Concentration Dataset and CV for the Ambient Temperature Dataset. For the PM2.5 Concentration Dataset, fixed is set to False, kernel is set to exponential, and constant is set to True; for the Ambient Temperature Dataset, fixed is set to False, kernel is set to exponential, constant is set to True.

- Random Forest (RF) [20]: it is an ensemble learning method that operates by constructing a multitude of decision trees at training time. The mean or average prediction of the individual trees is returned. For the PM2.5 Concentration Dataset, criterion is set to squared_error, max_features is set to sqrt, bootstrap is set to True, oob_score is set to False, n_estimators is set to 100, max_depth is set to None, min_samples_split is set to 2, min_samples_leaf is set to 1, min_weight_fraction_leaf is set to 0.0, max_leaf_nodes is set to None, and min_impurity_decrease is set to 0.0. For the Ambient Temperature Dataset, criterion is set to friedman_mse, max_features is set to None, bootstrap is set to True, oob_score is set to True, n_estimators is set to 100, max_depth is set to None, min_samples_split is set to 2, min_samples_leaf is set to 1, min_weight_fraction_leaf is set to 0.0, max_leaf_nodes is set to None, and min_impurity_decrease is set to 0.0.
- Gradient Boosting (GB) [25]: it gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. For the PM2.5 Concentration Dataset, loss is set to absolute_error, learning_rate is set to 0.1, n_estimators is set to 400, and criterion is set to friedman_mse; for the Ambient Temperature Dataset, loss is set to squared_error, learning_rate is set to 0.1, n_estimators is set to 100, and criterion is set to friedman_mse.
- Artificial Neural Network (ANN) [51]: the spatial variables (i.e., latitude and longitude) could be regarded as two general variables as other variables like temperature, wind speed, and pressure to make a prediction of the values of the targeted variable. For both datasets, the Adam optimizer and ReLU activation function are leveraged. The number of hidden layers is set to 4.
- Geographically and Temporally Weighted Neural Network (GTWNN) [14]: it integrates artificial neural networks into geographically and temporally weighted regression to capture the spatial and temporary non-stationarity in identifying the relationship between predictors and the response variables. GTWNN consists of two fully connected neural networks. It starts with a weight estimation neural network learning the spatial-temporal weight of each independent variable from coordinates. Then, the learned weights are multiplied by the corresponding independent variables and serve as input to the second neural network which performs the nonlinear transformation on the input variables to obtain the output as the response value. For both datasets, the number of spatial features is set to 2, and the number of non-spatial features is set to 9. The Adam optimizer is leveraged.

Implementation Details. We leverage MLP frameworks with 4 hidden layers. In terms of optimization, we use the Adam optimizer with a learning rate of 0.001. The mean squared error (MSE) serves as the training loss function. The number of hidden units is set to 512, and the number of epochs is set to 100 for all datasets. Our model demonstrates flexibility by being able to accommodate different targeted variables via the activation function. Given that the range of the targeted variable is non-negative, such as PM2.5, precipitation, and traffic flow, we leverage ReLU as the nonlinear activation function; given that the targeted variable can be negative, such as temperature and wind speed, Leaky ReLU is leveraged.

Evaluation Metrics. We report the average statistics of three metrics to evaluate the performance of our proposed model. 1) Root-mean-square error (RMSE): the objective of the spatial prediction problem can be treated as minimizing the differences between values predicted by the model and the true values observed. The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. 2) Mean absolute error (MAE): similar to RMSE, it is a measure of errors between paired observations expressing the same phenomenon. MAE is computed as the sum of absolute errors divided by the sample size. 3) Coefficient of determination (R^2): it denotes the proportion of the variation in the true values that are predictable from the predictions, which measures how differences in the predictions can be explained by the difference in true values, assessing the strength of the linear relationship between the predictions and the true values.

5.2 Effectiveness Comparison with the State-of-the-arts

We present the performance comparison of all methods in Table 3. Based on the table, our proposed model performs consistently the best over all the metrics on both datasets.

Effectiveness on Spatial Prediction. On the PM2.5 Concentration Dataset, it is apparent from the left segment of Table 3 that HSSP excels other approaches regarding MAE, RMSE, and R^2. For instance, HSSP outperforms other methods with, on average, a 20% decrease in RMSE and a 24% decrease in MAE. HSSP achieves the best result by excelling the second-best method in terms of an 8% decrease in MAE, a 6% decrease in RMSE, and a 27% increase in R^2. Specifically, HSSP achieves RMSE as low as 3.3, while the average RMSE of other methods is over 4.1. HSSP achieves MAE as low as 2.3, while the average MAE of other methods is over 3.0. The R^2 of most of the baselines are around 0.2, even with some of them lower than 0, and HSSP consistently achieves R^2 over 0.4.

On the Ambient Temperature Dataset, as can be seen from the right segment of Table 2, HSSP still achieves the best performance among all methods in each evaluation metric. Specifically, HSSP outperforms other methods with, on average, a 79% decrease in RMSE and an 85% decrease in MAE. HSSP performs 25% better than the second-best model in MAE, 9% in RMSE, and 3% in R^2. For MAE, HSSP achieves a value as low as 0.36, which is 0.12 lower than the second-best method and 2.04 lower than the average of all methods. For RMSE, HSSP achieves a value as low as 0.6, which is 0.05 lower than the second-best method and 2.25 lower than the average of all methods.

Stability. On both datasets, the average RMSE, MAE, and R^2 of all days in December 2019, followed by standard deviation, are reported. The standard deviation is an effective metric to reflect the stableness of the method. Being able to consistently produce predictions without big fluctuation is an important criterion for spatial prediction approaches. It is apparent that HSSP excels over other approaches in terms of stability.

On the PM2.5 Concentration Dataset, the standard deviations of HSSP on three metrics are in the lowest category. In particular, for MAE, HSSP performs at least 20% better than the second-best stable method in terms of standard deviation and 70% than the average of all the baselines. Regarding RMSE, HSSP performs at least 2% better than the second-best stable method in terms of standard deviation and 20% than the average of all the baselines.

On the Ambient Temperature Dataset, HSSP has the lowest standard deviation on RMSE, MAE, and R^2. Regarding MAE, HSSP excels over the second-best stable method 8% in standard deviation and outperforms the average of other methods in terms of a 70% decrease in standard deviation. Regarding R^2, HSSP achieves a standard deviation as low as 0.03, while the second-best stable method is higher than 0.06.

Table 3. Experimental results on two real-world datasets. The best performance is in boldface.

	AQ-PM2.5			Temperature		
Interpolation Method	MAE	RMSE	R^2	MAE	RMSE	R^2
IDW	2.543±0.765	3.493±1.057	0.327±0.319	0.476±0.109	0.647±0.151	0.779±0.058
Kriging	3.800±1.174	4.894±1.376	-0.239±0.283	1.180±0.268	1.439±0.348	-0.055±0.064
RBF	2.469±0.698	3.489±1.025	0.286±0.468	0.529±0.134	0.686±0.162	0.749±0.073
Nearest Neighbor	2.810±0.656	4.097±1.042	-0.086±0.963	0.762±0.093	1.090±0.129	0.322±0.272
VPint	4.945±1.773	6.308±2.096	-1.237±1.170	16.260±1.095	18.491±1.116	-226.924±154.933
RFSI	2.492±0.727	3.468±1.056	0.335±0.316	0.811±0.114	1.232±0.166	0.046±0.602
IDW+GWR	2.918±0.987	3.875±1.266	0.208±0.201	0.852±0.136	1.042±0.168	0.467±0.057
IDW+RF	2.855±0.880	3.895±1.151	0.206±0.212	0.767±0.203	0.974±0.234	0.449±0.331
IDW+GB	2.946±0.892	4.008±1.162	0.161±0.222	0.791±0.209	1.008±0.233	0.416±0.323
IDW+ANN	2.586±0.986	3.620±1.402	0.051±0.957	2.813±0.325	3.265±0.351	-6.407±5.424
IDW+GTWNN	2.789±0.900	3.851±1.238	-0.099±0.988	1.118±0.450	1.468±0.549	-1.051±3.828
HSSP	**2.276±0.525**	**3.274±1.004**	**0.424±0.270**	**0.357±0.086**	**0.592±0.111**	**0.800±0.031**

Flexibility. Although our point-based predictor leverages additional environmental ancillary attributes to predict the values of the targeted variable, it can estimate the values of the targeted variable at arbitrary locations, at which the ancillary attributes may not be available, since we propose a novel deep geometric spatial interpolation framework to first automatically learn the underlying spatial distribution of all the ancillary environmental attributes and then interpolate them. Consequently, our point-based predictor demonstrates more flexibility than all the ML-based spatial prediction methods and the two neural networks-based state-of-the-art methods, as these methods necessitate the knowledge of ancillary attributes at the targeted locations. Thus, these baselines cannot perform individually but require the assistance of spatial interpolation methods first to interpolate the required ancillary attributes.

Moreover, HSSP demonstrates more flexibility by being able to take both point data and raster data as inputs, compared with popular traditional spatial interpolation methods that cannot handle raster data as input data sources.

5.3 Ablation Study

Effectiveness of the Proposed Deep Geometric Spatial Interpolator. We compare the performance of our proposed interpolator with the interpolation methods, which are IDW, Kriging, RBF, and Nearest Neighbor, as well as the state-of-the-art VPint and RFSI on our datasets. The results are summarized as follows and are also illustrated in Table 4. Our proposed deep geometric spatial interpolator consistently outperforms the baselines on both datasets.

As shown in the upper segment of Table 4, on the PM2.5 Concentration Dataset, our proposed interpolator outperforms the traditional interpolation models by at least 7% and on average 28% in MAE. Our proposed interpolator also has the best performance in terms of RMSE and R^2 (outperforms the second-best method by 5% and 19%, respectively). Similarly, on the Ambient Temperature Dataset, our proposed interpolator outperforms the second-best method by 22% and 89% on average regarding MAE. In terms of RMSE, our proposed interpolator outperforms the second-best by 6% and 85% on average. In terms of the comparison with the more recent interpolators (VPint and RFSI), as shown in the middle segment of Table 4, our interpolator consistently performs the best in all the metrics. More specifically, on the PM2.5 Concentration Dataset, our proposed interpolator outperforms RFSI by 8% in MAE, 5% in RMSE, and 0.06 in terms of an increase in R^2. Our proposed interpolator outperforms VPint by 3 in terms of a decrease in MAE and RMSE. Moreover, on the Ambient Temperature Dataset,

Table 4. Ablation study of our proposed deep geometric spatial interpolator on two real-world datasets. The best performance is in boldface and the second best is underlined.

	AQ-PM2.5			Temperature		
Interpolation Method	MAE	RMSE	R^2	MAE	RMSE	R^2
IDW	2.543±0.765	3.493±1.057	0.327±0.319	0.476±0.109	0.647±0.151	0.779±0.058
Kriging	3.800±1.174	4.894±1.376	-0.239±0.283	1.180±0.268	1.439±0.348	-0.055±0.064
RBF	2.469±0.698	3.489±1.025	0.286±0.468	0.529±0.134	0.686±0.162	0.749±0.073
Nearest Neighbor	2.810±0.656	4.097±1.042	-0.086±0.963	0.762±0.093	1.090±0.129	0.322±0.272
VPint	4.945±1.773	6.308±2.096	-1.237±1.170	16.260±1.095	18.491±1.116	-226.924±154.933
RFSI	2.492±0.727	3.468±1.056	0.335±0.316	0.811±0.114	1.232±0.166	0.046±0.602
Our Interpolator	<u>2.293±0.411</u>	<u>3.305±1.021</u>	<u>0.397±0.279</u>	<u>0.372±0.091</u>	<u>0.609±0.124</u>	<u>0.786±0.044</u>
HSSP	**2.276±0.525**	**3.274±1.004**	**0.424±0.270**	**0.357±0.086**	**0.592±0.111**	**0.800±0.031**

our proposed interpolator outperforms RFSI by 54% in MAE, 51% in RMSE, and 0.7 in terms of an increase in R^2. Our proposed interpolator outperforms VPint by 16 in terms of a decrease in MAE and RMSE. VPint generates the worst result as it can only take squared gridded data as input; therefore, huge information loss exists during the transformation of the data points to the gridded version, which results in the worst performance. It also reflects our model's flexibility that can take both point data and raster data as input without a strict requirement of data format.

Additionally, as indicated in the lower segment of Table 4, our HSSP consistently outperforms our proposed interpolator on two datasets in terms of all the metrics, which reflects the effectiveness of our error-compensation framework.

Effectiveness of Leveraging Both the Distance Information and the Orientation Information in the Proposed Deep Geometric Spatial Interpolator. As introduced in section 4.1, we leverage the distance and orientation information in our proposed deep geometric spatial interpolator. To verify the effectiveness of introducing both pieces of information, we conduct another ablation study on our proposed deep geometric spatial interpolator by removing one piece of information at a time, where the first one removes the orientation information (named "Our Interpolator-orientation"), and the second one removes the distance information (named "Our Interpolator-distance").

As indicated in Table 5, we can see that our proposed interpolator consistently outperforms the two baselines with the incomplete input information. On the PM2.5 Concentration Dataset, our proposed interpolator outperforms the baselines by, on average, 4% in MAE, 3% in RMSE, and 3% in R^2. Moreover, on the Ambient Temperature Dataset, our proposed interpolator outperforms the baselines by achieving MAE as low as 0.37; the average MAE of the two baselines is 0.45. Regarding RMSE, our proposed interpolator reaches 0.61, while the average of the two baselines is 0.63.

Effectiveness of the Error-compensation Framework. We compare our HSSP against two new comparison methods, where the first one combines our proposed interpolator with ANN (named "Our Interpolator+ANN"), and the second one combines our proposed interpolator with GTWNN (named "Our Interpolator+GTWNN").

As shown in Table 6, our HSSP consistently outperforms "Our Interpolator+ANN" and "Our Interpolator+GTWNN" on both two datasets in terms of all the metrics. More specifically, on the PM2.5 Concentration Dataset, our HSSP outperforms the second-best by 6% in MAE and 0.04 in an increase in R^2. On average, our HSSP outperforms all the models by 13% in MAE and 11% RMSE. On the Ambient Temperature Dataset, our

Table 5. Ablation study of leveraging both the distance information and the orientation information in our proposed deep geometric spatial interpolator. The best performance is in boldface.

	AQ-PM2.5			Temperature		
Method	MAE	RMSE	R^2	MAE	RMSE	R^2
Our Interpolator-orientation	2.436±0.432	3.463 ±1.037	0.341±0.354	0.466±0.107	0.640±0.133	0.780±0.051
Our Interpolator-distance	2.349±0.551	3.332±1.061	0.385±0.399	0.441±0.103	0.623±0.132	0.782±0.047
Our Interpolator	2.293±0.411	3.305±1.021	0.397±0.279	0.372±0.091	0.609±0.124	0.786±0.044

Table 6. Ablation study of our error-compensation framework on two real-world datasets. The best performance is in boldface. In each segment of the table, the better performance is underlined.

	AQ-PM2.5			Temperature		
Method	MAE	RMSE	R^2	MAE	RMSE	R^2
IDW+ANN	2.586±0.986	3.620±1.402	0.051±0.957	2.813±0.325	3.265±0.351	-6.407±5.424
Our Interpolator+ANN	2.424±0.863	3.425±1.226	0.382±0.963	1.256±0.813	2.255±1.102	-0.218±2.449
IDW+GTWNN	2.789±0.900	3.851±1.238	-0.099±0.988	1.118±0.450	1.468±0.549	-1.051±3.828
Our Interpolator+GTWNN	2.617±0.974	3.741±1.221	0.040±0.833	0.873±0.470	1.125±0.529	0.453±2.339
HSSP	**2.276±0.525**	**3.274±1.004**	**0.424±0.270**	**0.357±0.086**	**0.592±0.111**	**0.800±0.031**

HSSP outperforms the second-best by 59% in MAE, 47% in RMSE, and 0.35 in terms of an increase in R^2. On average, our HSSP outperforms all the models by 76% in MAE and 71% in RMSE.

More specifically, we can see the trend that the combination of IDW and ANN/GTWNN consistently performs the worst, the combination of our proposed interpolator and ANN/GTWNN is consistently ranked in the middle, and our HSSP consistently performs the best, which reflects not only the potency of our error-compensation framework that provide additional information for spatial prediction but also the effectiveness of our proposed interpolator than the typical traditional interpolation method.

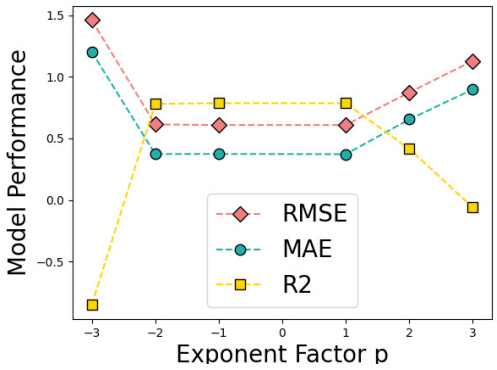
5.4 Impact of Important Parameters

We conduct experiments to explore the impact of important parameters on our model's performance. We leverage the metrics, RMSE, MAE, and R^2, as the measurement of model performance and report the statistics in Figure 2(a) and Figure 2(b).

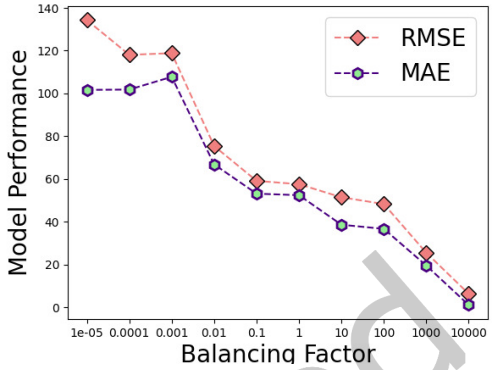
Exponent Factor p. As shown in equation 8, the exponent factor p affects the value of the computed distance information. To explore the impact of different exponent factors on the model performance, we take a wide range of p from -3 to 3 and explore the model performance, reflected by RMSE, MAE, and R^2.

As indicated in Figure 2(a), the model performs consistently better when p ranges from -2 to 1, with the performance of all three metrics achieving the best. As the value of p increases to a number greater than 1 or decreases to less than 2, the model's accuracy starts to decrease.

Balancing Factor λ. The performance of our raster-based predictor is the worst when the balancing factor $\lambda = 0$, which makes sense as the factor λ controls the loss of the raster-based predictor as expressed in the first term in Equation 4. Given $\lambda = 0$, our point-based predictor still enjoys the availability of ground-truth labels, as indicated in the second term in Equation 4, but our raster-based predictor receives no training, resulting in the worst performance. However, as λ increases from 0 to 1, our raster-based predictor's accuracy increases, enabling our model to take remote sensing data as input and enjoy its strength in covering large spatial areas. To

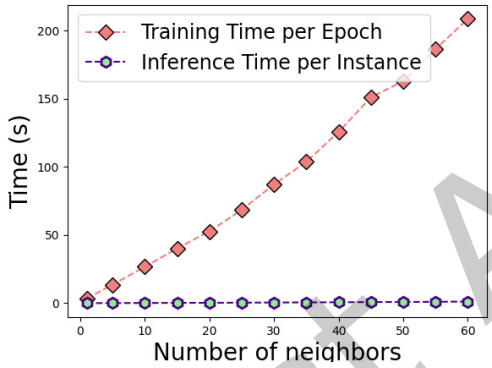


(a) Point-based predictor's performance w.r.t the exponent factor p

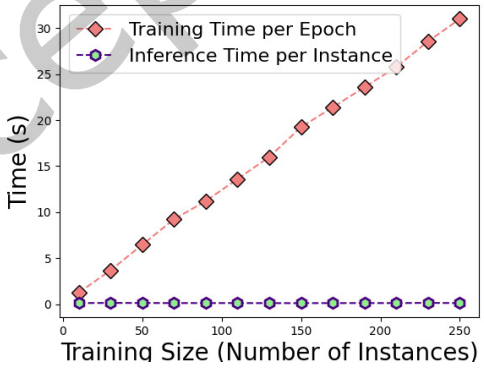


(b) Raster-based predictor's performance w.r.t the balancing factor λ

Fig. 2. Model performance w.r.t important parameters p and λ



(a) HSSP's scalability versus number of neighbors



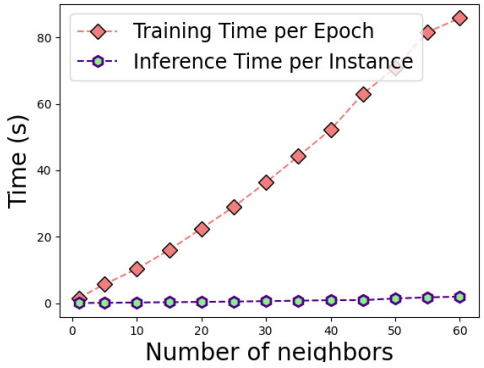
(b) HSSP's scalability versus training size

Fig. 3. HSSP's predictor's scalability testing against the number of neighbors and training size

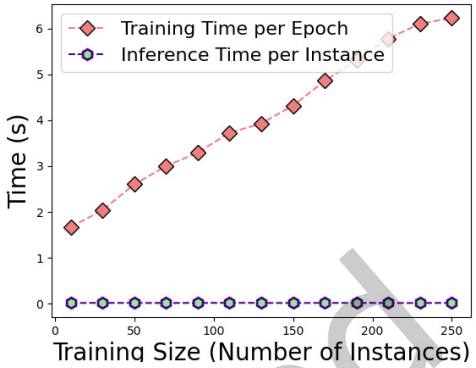
demonstrate that the increase in the balancing factor λ allows the improvement in accuracy of the raster-based predictor via quantitative analysis, we set the balancing factor λ with a wide range of values and record the performance of our raster-based predictor in terms of MAE and RMSE. As supported by Figure 2(b), when we increase the balancing factor from 10^{-5} to 10^{4}, the RMSE of our raster-based predictor drops from 134 to 6, and the corresponding MAE drops from 101 to 1. The downward trend reflected by Figure 2(b) verifies our claim.

5.5 Model Scalability

As the idea of deep geometric spatial interpolation is leveraged in our point-based predictor, where we propose a multilayer perceptron-based framework to interpolate the value of the targeted variable on the targeted location



(a) Point-based predictor's scalability versus number of neighbors



(b) Point-based predictor's scalability versus training size

Fig. 4. Point-based predictor's scalability testing against the number of neighbors and training size

using its t neighbors (points in the in-situ data that are within t-th closest to the targeted point). Thus, the impact of the number of neighbors is crucial. We conduct a scalability test that measures our proposed model's ability to scale up or down, reflected by its training and testing time, as a reaction to the variation of the number of neighbors selected. Besides, the impact of the variation of the train set's size on our proposed model's training and testing time is investigated.

HSSP's Scalability. For our proposed model HSSP, Figure 3(a), indicating the variation of the training time (seconds per epoch) with respect to the change in the number of neighbors selected, reflects their linear relationship. The invariance property (a horizontal line) of inference time per instance as a reaction to the increase in neighbors is also reflected in the same figure.

Similar patterns are reflected in our proposed framework's reaction to an increase in the number of train set's size, where a linear relationship between the training time per epoch with respect to the increase in train set's size and the invariance of the testing time are both reflected in Figure 3(b).

Point-based Predictor's Scalability. As an indispensable component of HSSP, the point-based predictor inherits the entire HSSP's patterns in terms of the model Scalability. More specifically, as reflected in Figure 4(a) and Figure 4(b), the point-based predictor reacts to an increase in the number of neighbors and the increase in the train set's size by demonstrating a linear relationship in terms of training time and invariance in terms of inference time.

6 RELATED WORK

6.1 Spatial Prediction and Interpolation

Many approaches have been proposed to construct the relationships between the targeted variable and independent variables in a spatial prediction task, which could be categorized into two main types, including statistical methods that learn the linear relationships among variables and machine learning methods aiming to discover the nonlinear relationships for spatial prediction. In terms of statistical methods, several interpolation and regression methods have been developed to predict the values of the targeted variable in the same location or other locations

based on available values in some locations according to the first law [48] and the second law of geography [17]. Traditional interpolation methods contain Kriging, Inversed Distance Weighting (IDW), Radial Basis Function Interpolation (RBF), Nearest Neighbor Interpolator, and so on [1, 36]. The interpolant of all these methods can be formatted as a weighted sum of values on known points [10, 13]. Specifically, IDW computes a weighted average of the values at the known locations for unknown locations [9, 27, 35]. Kriging also applied a weighted average of known values to calculate values at unknown locations[13]. The Nearest Neighbor Interpolation method assigns the value of the nearest neighbor to the targeted point [13, 49]. The Radial Basis Function Interpolation (RBF) approach computes interpolation value by utilizing a weighted sum of radial basis functions [22]. In terms of regression methods, geographically weight regression performs a local form of linear regression to model spatially varying relationships [7]. Spatial autoregressive regression [28] models spatial dependency by adding spatial lags of targeted variables. Although these traditional methods focus on modeling spatial dependency and heterogeneity with mathematical expressions, they ignore the nonlinear nature of spatial relationships.

A growing number of related research adopt machine learning models to learn the nonlinear relationship for spatial prediction [20]. Random forests and gradient boosting shares are commonly applied for predicting unknown variables given available variable values at the same location [20, 25]. With the development of deep learning technologies, the Artificial Neural Network (ANN) model and its variants have been widely utilized due to their learning capability to capture the real world's complex relationships. For a general ANN model, the spatial variables (i.e., latitude and longitude) could be regarded as two general variables as other variables like temperature, wind speed, and pressure to make a prediction [51]. Geographically and Temporally Weighted Neural Network (GTWNN) integrated artificial neural networks into geographically and temporally weighted regression to capture the spatial and temporary non-stationarity in identifying the relationship between predictors and the response variables [14]. GTWNN consists of two fully connected neural networks. It starts with a weight estimation neural network learning the spatial-temporal weight of each independent variable from coordinates. Then, the learned weights are multiplied by the corresponding independent variables and serve as input to the second neural network which performs the nonlinear transformation on the input variables to obtain the output as the response value.

6.2 Self-supervised Learning

Self-supervised learning learns representations of unlabeled data that can be used for downstream tasks by leveraging the data's inherent co-occurrence relationships as the self-supervision. As a promising alternative to supervised learning, self-supervised learning has drawn massive attention for its ability to take advantage of massive amounts of unlabeled data. The self-supervision can be categorized into three main types [37], including 1) Generative: train an encoder to encode input into an explicit vector and a decoder to reconstruct the input from the explicit vector, 2) Contrastive: train an encoder to encode input into an explicit vector to measure similarity, and 3) Generative-Contrastive: train an encoder-decoder to generate fake samples and a discriminator to distinguish them from real samples.

In terms of generative self-supervised learning, many important methods have been developed, including auto-regressive (AR) models, flow-based models, auto-encoding (AE) models, and hybrid generative models. Auto-regressive (AR) models [41, 43, 50] can be viewed as a directed graph model where the joint distribution can be factorized as a product conditional and the probability of each variable is dependent on the previous variables. The flow-based models [11, 30] stacks a series of transforming functions to estimate the complex high-dimensional densities from data. The auto-encoding models reconstruct inputs from inputs and further contain basic AE model [3], context prediction model [4], denoising AE model [46], and variational AE model [31]. The hybrid generative models can either combine AR and AE models [29] or combine AE and flow-based models [24].

Contrastive self-supervised learning aims to "learn to compare" through a Noise Contrastive Estimation objective [19] and can be divided into two types [37]: context-instance contrast and instance-instance contrast. The context-instance contrast methods [12, 16, 23] models the belonging relationship between the local feature of a sample and its global context representation, where the mutual information is maximized. The instance-instance contrast methods [8, 18, 21, 47] ignore mutual information and directly learn the relationships between different samples' instance-level local representations.

Generative-contrastive self-supervised learning is also called adversarial representation learning, which leverages the discriminative loss function as the objective to reconstruct the original data distribution. There are two types of models under this category: 1) Generate with complete input, which is represented by Generative Adversarial Networks (GAN) [42] and its variants [6, 26, 40]. 2) Recover with partial input, which asks models to recover the remaining parts given partial input [32, 33, 52].

7 CONCLUSION

Spatial prediction is an essential yet challenging task. The fusion of in-situ data and remote sensing observations to synergize the complementary strength is still a highly open research domain due to the heterogeneity of multi-source data. Besides, the complex, unknown, and spatial autoregressive relation of the targeted variable and the information loss in spatial-relation representation invalidate the existing spatial prediction approaches. This paper proposes a novel yet generic framework, namely HSSP, that synergizes multi-source data while effectively conquering the problems of spatial heterogeneity. Specifically, HSSP leverages a novel error-compensation framework to capture the prediction inconsistency to enhance the effectiveness of spatial prediction. This paper proposes a new deep geometric spatial interpolation model as the prediction backbone that automatically learns the underlying spatial distribution and interpolates the values of the targeted variable at unknown locations based on existing observations. Popular interpolation methods are proven to be the special cases of our proposed interpolator under special parameter settings. Moreover, spatial relation is fully represented without substantial information loss by HSSP, which takes into account both distance and orientation information. Extensive experiments and case studies conducted on two real-world datasets demonstrate the consistent and superior performance of HSSP in spatial prediction. HSSP excels in existing works by generating more accurate, stable, and generalizable performance. Specifically, HSSP outperforms other approaches with, on average, a 20% decrease in RMSE and 24% decrease in MAE on the PM2.5 Concentration Dataset and a 79% decrease in RMSE and 85% decrease in MAE on the Ambient Temperature Dataset.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, Computing Research Association/NSF Sub 2021CIF-Emory-05, the Department of Homeland Security (DHS) Grant No. 17STCIN00001, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).

REFERENCES

[1] Halit Apaydin, F Kemal Sonmez, and Y Ersoy Yildirim. 2004. Spatial interpolation techniques for climate data in the GAP region in Turkey. *Climate Research* 28, 1 (2004), 31–40.

[2] Laurens Arp, Mitra Baratchi, and Holger Hoos. 2022. VPint: value propagation-based spatial interpolation. *Data Mining and Knowledge Discovery* (2022), 1–32.

[3] Dana H Ballard. 1987. Modular learning in neural networks.. In *Aaai*, Vol. 647. 279–284.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics* 5 (2017), 135–146.

[5] Kenneth Ewart Boulding. 2018. *Conflict and defense: A general theory.* Pickle Partners Publishing.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).
[7] Chris Brunsdon, Stewart Fotheringham, and Martin Charlton. 1998. Geographically weighted regression. *Journal of the Royal Statistical Society: Series D (The Statistician)* 47, 3 (1998), 431–443.
[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
[9] Tao Chen, Liliang Ren, Fei Yuan, Xiaoli Yang, Shanhu Jiang, Tiantian Tang, Yi Liu, Chongxu Zhao, and Liming Zhang. 2017. Comparison of spatial interpolation schemes for rainfall data and application in hydrological modeling. *Water* 9, 5 (2017), 342.
[10] Masoomeh Delbari, Peyman Afrasiab, and Samane Jahani. 2013. Spatial interpolation of monthly and annual rainfall in northeast of Iran. *Meteorology and Atmospheric Physics* 122, 1 (2013), 103–113.
[11] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).
[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*. 1422–1430.
[13] Paul Daniel Dumitru, Marin Plopeanu, and Dragos Badea. 2013. Comparative study regarding the methods of interpolation. *Recent advances in geodesy and Geomatics engineering* 1 (2013), 45–52.
[14] Luwei Feng, Yumiao Wang, Zhou Zhang, and Qingyun Du. 2021. Geographically and temporally weighted neural network for winter wheat yield prediction. *Remote Sensing of Environment* 262 (2021), 112514.
[15] Ronald Gelaro, Will McCarty, Max J Suárez, Ricardo Todling, Andrea Molod, Lawrence Takacs, Cynthia A Randles, Anton Darmenov, Michael G Bosilovich, Rolf Reichle, et al. 2017. The modern-era retrospective analysis for research and applications, version 2 (MERRA-2). *Journal of climate* 30, 14 (2017), 5419–5454.
[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).
[17] Michael F Goodchild. 2004. The validity and usefulness of laws in geographic information science and geography. *Annals of the Association of American Geographers* 94, 2 (2004), 300–303.
[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems* 33 (2020), 21271–21284.
[19] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 297–304.
[20] KS Harishkumar, KM Yogesh, Ibrahim Gad, et al. 2020. Forecasting air pollution particulate matter (PM2. 5) using machine learning regression models. *Procedia Computer Science* 171 (2020), 2057–2066.
[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.
[22] Alfa RH Heryudono and Tobin A Driscoll. 2010. Radial basis function interpolation on irregular domain through conformal transplantation. *Journal of Scientific Computing* 44, 3 (2010), 286–300.
[23] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).
[24] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. 2019. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*. PMLR, 2722–2730.
[25] Xuefei Hu, Jessica H Belle, Xia Meng, Avani Wildani, Lance A Waller, Matthew J Strickland, and Yang Liu. 2017. Estimating PM2. 5 concentrations in the conterminous United States using the random forest approach. *Environmental science & technology* 51, 12 (2017), 6936–6944.
[26] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–14.
[27] Kevin Johnston, Jay M Ver Hoef, Konstantin Krivoruchko, and Neil Lucas. 2001. *Using ArcGIS geostatistical analyst*. Vol. 380. Esri Redlands.
[28] Harry H Kelejian and Ingmar R Prucha. 1998. A generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances. *The Journal of Real Estate Finance and Economics* 17, 1 (1998), 99–121.
[29] Ahmad Khajenezhad, Hatef Madani, and Hamid Beigy. 2020. Masked autoencoder for distribution estimation on small structured data sets. *IEEE Transactions on Neural Networks and Learning Systems* 32, 11 (2020), 4997–5007.
[30] Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems* 31 (2018).
[31] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[32] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning representations for automatic colorization. In *European conference on computer vision*. Springer, 577–593.
[33] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.
[34] RC Levy, S Mattoo, LA Munchak, LA Remer, AM Sayer, F Patadia, and NC Hsu. 2013. The Collection 6 MODIS aerosol products over land and ocean. *Atmospheric Measurement Techniques* 6, 11 (2013), 2989–3034.
[35] Jin Li and Andrew D Heap. 2008. A review of spatial interpolation methods for environmental scientists. (2008).
[36] ZH Lin, XG Mo, HX Li, and HB Li. 2002. Comparison of three spatial interpolation methods for climate variables in China. *Acta Geographica Sinica* 57, 1 (2002), 47–56.
[37] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[38] Alexei Lyapustin, John Martonchik, Yujie Wang, Istvan Laszlo, and Sergey Korkin. 2011. Multiangle implementation of atmospheric correction (MAIAC): 1. Radiative transfer basis and look-up tables. *Journal of Geophysical Research: Atmospheres* 116, D3 (2011).
[39] Bradley O Parks, Louis T Steyaert, and Michael F Goodchild. 1993. *Environmental modeling with GIS*. Oxford university press.
[40] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.
[41] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. 2019. MolecularRNN: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372* (2019).
[42] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
[43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[44] Xiang Ren, Zhongyuan Mi, and Panos G Georgopoulos. 2020. Comparison of Machine Learning and Land Use Regression for fine scale spatiotemporal estimation of ambient air pollution: Modeling ozone concentrations across the contiguous United States. *Environment international* 142 (2020), 105827.
[45] Aleksandar Sekulić, Milan Kilibarda, Gerard Heuvelink, Mladen Nikolić, and Branislav Bajat. 2020. Random forest spatial interpolation. *Remote Sensing* 12, 10 (2020), 1687.
[46] Wilson L Taylor. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly* 30, 4 (1953), 415–433.
[47] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *European conference on computer vision*. Springer, 776–794.
[48] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.
[49] Yue Xing, Qifan Song, and Guang Cheng. 2019. Benefit of interpolation in nearest neighbor algorithms. *arXiv preprint arXiv:1909.11720* (2019).
[50] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32 (2019).
[51] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*. 1–4.
[52] Richard Zhang, Phillip Isola, and Alexei A Efros. 2017. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1058–1067.
[53] A-Xing Zhu, Guonian Lu, Jing Liu, Cheng-Zhi Qin, and Chenghu Zhou. 2018. Spatial prediction based on Third Law of Geography. *Annals of GIS* 24, 4 (2018), 225–240.

A PROOF OF THEOREMS

THEOREM A.1. *Our proposed deep geometric spatial interpolation framework $q(\cdot)$ is the general form of popular spatial interpolation methods.*

PROOF. Based on Equation 5, Equation 6, and the illustration in Section 4.1, we can prove that the same techniques, weighted sum and normalization, are leveraged for both the popular interpolation methods and our proposed interpolator. Thus, we only need to prove that our proposed weight computation framework, expressed in Equation 10, of the neighbors to the targeted point can be generalizable to compute the weights of popular spatial interpolation methods.

We first express our weights computation scheme as:

$$H = \sigma(\vec{x}W^{(1)} + \vec{b}^{(1)}), O = HW^{(2)} + \vec{b}^{(2)}$$
$$W^{(1)} = \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1h}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2h}^{(1)} \end{pmatrix}, \vec{b}^{(1)} = \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)} \end{pmatrix} \tag{13}$$
$$W^{(2)} = \begin{pmatrix} w_1^{(2)} & w_2^{(2)} & \dots & w_h^{(2)} \end{pmatrix}^T, \vec{b}^{(2)} = b^{(2)}, \sigma = ReLU(\cdot)$$

where $\vec{x}$ denotes the input vector $\langle d_{ij}^p, \theta_{ijk} \rangle \in \mathbb{R}^{1\times 2}$, $W^{(1)}, \vec{b}^{(1)}, W^{(2)}, \vec{b}^{(2)}$ are our model's parameters, σ is the activation function, H denotes the hidden-layer variable, O denotes the output, and h denotes the hidden size.

Although we leverage more complicated MLP framework with more hidden layers, we only need to show the simplest case in Equation 13 is the general form of the weight computation schemes of popular interpolation methods as the number of layers is one of the hyperparameters that we can control for.

(1) Nearest Neighbor Interpolation: all the weights are assigned to the nearest point and does not assign any weight to other neighbor points

$$w_{ij} = \begin{cases} 1 & \text{if point } s_j \text{ is the nearest neighbor of } s_i \\ 0 & \text{if point } s_j \text{ is not the nearest neighbor of } s_i \end{cases}$$

To prove that Nearest Neighbor Interpolation is one of the special cases of our proposed interpolator under particular parameter settings, we first set $p = -1$, as p is another hyperparameter of our model that we can control for. Then, we set our framework's parameters to be,

$$W^{(1)} = \begin{pmatrix} d_{min} & d_{min} & \dots & d_{min} \\ 0 & 0 & \dots & 0 \end{pmatrix}, \vec{b}^{(1)} = \begin{pmatrix} -d_{min2}^{-1}d_{min} & -d_{min2}^{-1}d_{min} & \dots & -d_{min2}^{-1}d_{min} \end{pmatrix}$$
$$W^{(2)} = \begin{pmatrix} d_{min2}(h(d_{min2} - d_{min}))^{-1} & d_{min2}(h(d_{min2} - d_{min}))^{-1} & \dots & d_{min2}(h(d_{min2} - d_{min}))^{-1} \end{pmatrix}^T, \vec{b}^{(2)} = 0 \tag{14}$$

where d_{min} and d_{min2} are the shortest distance and the second shortest distance between the neighbor point in the in-situ data to the interpolated point s_i respectively. Then, we can compute the weight of neighbor point s_j to the targeted point s_i,

$$H = \sigma(\vec{x}W^{(1)} + \vec{b}^{(1)})$$
$$= \sigma(\begin{pmatrix} d_{ij}^{-1} & \theta_{ijk} \end{pmatrix} \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1h}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2h}^{(1)} \end{pmatrix} + \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)} \end{pmatrix})$$
$$= \sigma(\begin{pmatrix} d_{ij}^{-1}w_{11}^{(1)} + \theta_{ijk}w_{21}^{(1)} & d_{ij}^{-1}w_{12}^{(1)} + \theta_{ijk}w_{22}^{(1)} & \dots & d_{ij}^{-1}w_{1h}^{(1)} + \theta_{ijk}w_{2h}^{(1)} \end{pmatrix} + \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)} \end{pmatrix})$$
$$= \sigma(\begin{pmatrix} d_{ij}^{-1}w_{11}^{(1)} & d_{ij}^{-1}w_{12}^{(1)} & \dots & d_{ij}^{-1}w_{1h}^{(1)} \end{pmatrix} + \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)} \end{pmatrix})$$
$$= \sigma\begin{pmatrix} d_{ij}^{-1}d_{min} & d_{ij}^{-1}d_{min} & \dots & d_{ij}^{-1}d_{min} \end{pmatrix} + \begin{pmatrix} -d_{min2}^{-1}d_{min} & -d_{min2}^{-1}d_{min} & \dots & -d_{min2}^{-1}d_{min} \end{pmatrix}$$
$$= \sigma(\begin{pmatrix} d_{ij}^{-1}d_{min} - d_{min2}^{-1}d_{min} & d_{ij}^{-1}d_{min} - d_{min2}^{-1}d_{min} & \dots & d_{ij}^{-1}d_{min} - d_{min2}^{-1}d_{min} \end{pmatrix}) \tag{15}$$

We can see that if d_{ij} is not equal to d_{min}, then every element in H will be zero after the ReLU activation function; d_{ij} must be equal to d_{min} so that the element in H is greater than 0. Thus, given d_{ij} not equal to d_{min}, $H = \vec{0}$ and therefore $O = 0$; given $d_{ij} = d_{min}$, we have:

$$H = \begin{pmatrix}(d_{min2} - d_{min})d_{min2}^{-1} & (d_{min2} - d_{min})d_{min2}^{-1} & \dots & (d_{min2} - d_{min})d_{min2}^{-1}\end{pmatrix}$$
$$O = HW^{(2)} + \vec{b}^{(2)}$$
$$= \begin{pmatrix}(d_{min2} - d_{min})d_{min2}^{-1} & (d_{min2} - d_{min})d_{min2}^{-1} & \dots & (d_{min2} - d_{min})d_{min2}^{-1}\end{pmatrix} \begin{pmatrix} d_{min2}(h(d_{min2} - d_{min}))^{-1} \\ d_{min2}(h(d_{min2} - d_{min}))^{-1} \\ \dots \\ d_{min2}(h(d_{min2} - d_{min}))^{-1} \end{pmatrix} + 0$$
$$= 1 \tag{16}$$

Thus, we can conclude that when d_{ij} is equal to d_{min}, the output will be 1 and otherwise, the output will be 0. In short, we can prove that Nearest Neighbor Interpolation is one of the special cases of our proposed inteprolator under special parameter settings.

(2) Inverse Distance Weighting (IDW): the weight w_{ij} is the inverse distance d_{ij} between the neighbor point s_j and interpolated point s_i, and p is a hyperparameter

$$w_{ij} = d_{ij}^{-p} \tag{17}$$

To prove that IDW is also a special case of our proposed interpolator under special parameter settings, we set our framework's parameters to be:

$$W^{(1)} = \begin{pmatrix}1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0\end{pmatrix}, \vec{b}^{(1)} = \begin{pmatrix}0 & 0 & \dots & 0\end{pmatrix}$$
$$W^{(2)} = \begin{pmatrix}h^{-1} & h^{-1} & \dots & h^{-1}\end{pmatrix}^T, \vec{b}^{(2)} = 0 \tag{18}$$

Thus, we have,

$$H = \vec{x}W^{(1)} + \vec{b}^{(1)}$$
$$= \begin{pmatrix} d_{ij}^{-p} & \theta_{ijk}\end{pmatrix} \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1h}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2h}^{(1)}\end{pmatrix} + \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)}\end{pmatrix}$$
$$= \begin{pmatrix} d_{ij}^{-p}w_{11}^{(1)} + \theta_{ijk}w_{21}^{(1)} & d_{ij}^{-p}w_{12}^{(1)} + \theta_{ijk}w_{22}^{(1)} & \dots & d_{ij}^{-p}w_{1h}^{(1)} + \theta_{ijk}w_{2h}^{(1)}\end{pmatrix} + \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)}\end{pmatrix}$$
$$= \begin{pmatrix} d_{ij}^{-p}w_{11}^{(1)} & d_{ij}^{-p}w_{12}^{(1)} & \dots & d_{ij}^{-p}w_{1h}^{(1)}\end{pmatrix} + \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \dots & b_h^{(1)}\end{pmatrix}$$
$$= \begin{pmatrix} d_{ij}^{-p} & d_{ij}^{-p} & \dots & d_{ij}^{-p}\end{pmatrix}$$
$$O = HW^{(2)} + b^{(2)}$$
$$= \begin{pmatrix} d_{ij}^{-p} & d_{ij}^{-p} & \dots & d_{ij}^{-p}\end{pmatrix} \begin{pmatrix} h^{-1} & h^{-1} & \dots & h^{-1}\end{pmatrix}^T + 0$$
$$= \frac{1}{h} d_{ij}^{-p} * h$$
$$= d_{ij}^{-p} \tag{19}$$

We can conclude that the output will be d_{ij}^{-p} and therefore prove that IDW is one of the special cases of our proposed interpolator under special parameter settings.

In short, we can prove that popular interpolation methods, Nearest Neighbor Interpolation and IDW, are special cases of our proposed interpolators under special parameter settings.

To conclude, these popular traditional interpolation methods assume a particular distribution of the targeted variable and therefore use a predefined kernel function to compute the weight of each neighbor location. The comparison demonstrates our model's expressive power in automatically selecting and learning distributions among or beyond traditional prescribed-based spatial interpolation methods. Thus, our proposed deep spatial interpolation theoretically outperforms popular traditional spatial interpolation methods in terms of automatically selecting and learning underlying spatial distributions, instead of using prescribed distributions or predefined kernel functions.

■

Theorem A.2. *Our proposed weight computation framework $a(\cdot)$ in terms of computing the weight (contribution) of any neighbor point s_j to the targeted point s_i are rotation and translation invariant.*

Proof. We prove that our proposed weight computation framework $a(\cdot)$ is rotation and translation invariant by proving that all the inputs to $a(\cdot)$, the computed distance $d_{ij}^p \in [0, \infty)$, expressed in Equation 8 and angle $\theta_{ijk} \in [-\pi, \pi)$, expressed in Equation 9, are rotation and translation invariance. For any translation transformations T in 2D space, since only relative coordinates are used in terms of distance and angle, instead of specific coordinates, the input distance and angle of our model $a(\cdot)$ are invariant.

Now we show d_{ij}^p and θ_{ijk} are invariant for any rotation R in 2D space. We set p to 1 for proving purpose, as p is the hyperparameter of our framework that we can control for. Given the identity equations,

$$\begin{aligned} \langle x, y \rangle &= \langle Rx, Ry \rangle \\ R(x \times y) &= (Rx) \times (Ry) \end{aligned} \tag{20}$$

we can prove that,

$$\begin{aligned} d_{ij} = ||v_{ij}||_2 = \langle v_{ij}, v_{ij} \rangle &= \langle Rv_{ij}, Rv_{ij} \rangle \\ arccos(\langle \frac{v_{ij}}{d_{ij}}, \frac{v_{jk}}{d_{jk}} \rangle) &= arccos(\langle \frac{Rv_{ij}}{d_{ij}}, \frac{Rv_{jk}}{d_{jk}} \rangle) \\ \langle n_{ijk}, n_{xy} \rangle &= \langle Rn_{ijk}, Rn_{xy} \rangle \end{aligned} \tag{21}$$

Given Equation 9 and that both two factors of the angle θ_{ijk} are rotation invariant, we prove that the input angle is invariant under rotation.

In short, we prove that the input distance and angle are invariant under all translation and rotation. ■