



CESI: Sparse Input Spatial Interpolation for Heterogeneous and Noisy Hybrid Wireless Sensor Networks

Chaofan Li(^[0000–0002–1859–8392], Till Riedel^[0000–0003–4547–1984], and
Michael Beigl^[0000–0001–5009–2327]

TECO Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany
{[`chaofan.li`](mailto:chaofan.li)() , [`till.riedel`](mailto:till.riedel) , [`michael.beigl`](mailto:michael.beigl)}@kit.edu

Abstract. Hybrid wireless sensor networks (HWSNs) combine sensors of varying costs to balance budget and deployment density. However, their data products often exhibit high heterogeneity and noise, presenting new challenges for spatial interpolation models. Traditional spatial interpolation models take dense input. When working on HWSN datasets, a large part of the dense input must be obtained through imputation, leading to feature distribution changes and error accumulation. To address these challenges, we propose the Context Encoder Spatial Interpolation (CESI) Model, designed to work directly with sparse, narrow-format input. CESI integrates a GraphSAGE-based backbone with a Transformer-based context embedding module, leveraging probabilistic encoding for better generalization to unseen coordinates and a self-supervised signal to balance inductive biases between the two modules. Experimental results demonstrate that CESI consistently outperforms baseline models across several publicly available real-world datasets.

Keywords: Spatial Interpolation · Graph Neural Network · Sparse Data.

1 Introduction

In-situ sensor networks are crucial in many fields, offering high temporal coverage and robustness to interference. Modern sensor networks increasingly combine sensors of varying costs to balance budget and deployment density, enabling finer-grained data collection for more detailed modeling. We hereafter refer to them as hybrid wireless sensor networks (HWSNs) [13, 20].

Low-cost sensors, while economical, often compromise accuracy and reliability, leading to highly heterogeneous and noisy HWSN datasets. This poses significant challenges for spatial interpolation models, which are traditionally designed based on homogeneous, high-quality sensor data [2, 7, 12, 15, 18]. These methods typically require dense, wide-format input (Figure 1 left), which is increasingly incompatible with modern IoT protocols like OGC SensorThings API [16] that favor narrow, sparse data formats (Figure 1 right) to cope with the

Location X	Location Y	Humidity	Wind Speed	Temperature
25	25			
75	30	20%	2	
80	69	17%		10
121	105	14%	4	
...

Location X	Location Y	Property	Value
75	30	Humidity	20%
75	30	Wind Speed	2
80	69	Humidity	17%
80	69	Temperature	10
121	105	Humidity	14%
121	105	Wind Speed	4
...

Fig. 1: An example of the wide format (left) and narrow format (right) of the same input data entry of spatial interpolation models.

heterogeneity of HWSN. Consequently, there is a pressing need for spatial interpolation models tailored to sparse input in HWSNs, which will introduce the following potential benefits:

First, dense input models require extensive imputation to handle missing values in HWSN datasets. While traditional spatial interpolation methods also discuss data imputation, the causes of missing values in HWSNs differ significantly, resulting in much higher imputation workloads. In traditional sensor networks, missing values are mainly caused by occasional sensor failures. With high-quality sensors, such issues are infrequent. Thus, recent spatial interpolation studies still consider simple techniques like linear interpolation [8] or removing incomplete rows/columns [23] acceptable. In contrast, HWSNs face far more frequent failures from low-cost sensors, compounded by heterogeneity in sensor types, where sensors at different locations may only measure subsets of the observed properties. For instance, applying PE-GNN [12] to the SmartAQnet dataset [13] required imputing over 50% of the inputs, with all rows containing missing cells. In such cases, removing incomplete data is infeasible, while excessive imputation alters feature distributions and accumulates errors, degrading model performance. By using sparse input, we can prevent the data imputation step and the above-mentioned disadvantages (see Figure 1 as an example).

Second, dense input models typically encode all properties at the same location (a row in Figure 1 left) and focus on location-level correlations. In contrast, sparse input models encode each observation (a row in Figure 1 right), directly capturing observation-level correlations. This allows sparse models to learn fine-grained relationships more efficiently.

Despite these benefits, sparse input introduces challenges. The high dimensional nature of the sparse input makes it harder for models to learn generalizable representations, and direct exposure to noisy observations makes sparse input models more sensitive to the high noise in HWSN datasets. In contrast, for dense input models, the imputed values that occupy a considerable part of input are obtained by referring to multiple observations. This helps neutralize the noise from individual sensors, making the dense input models more robust to noise.

Based on the above insights, we propose the Context Encoder Spatial Interpolation (CESI) Model with the following contributions:

- CESI is among the first spatial interpolation models tailored for narrow-format sparse input, effectively addressing the heterogeneity in HWSN datasets and achieving significant performance gains.
- We designed a self-supervised context embedding module to handle the sparse input series. This module uses variational inference to learn the probabilistic encoding of the input observations and uses a self-supervised loss signal to achieve an adaptive balance of inductive bias with other modules. Thus, the model’s robustness against noise and universality across different tasks is significantly improved.
- We tested our model on three publicly available real-world HWSN datasets from different fields and with different characteristics. Compared to the baselines, whose performance is shaky across different datasets, our model consistently outperforms baselines on all three datasets.

2 Related Work

2.1 Challenges in HWSN Datasets

HWSN datasets are heterogeneous and noisy in several ways, typically including but not limited to the following:

- **Heterogeneity from various sensor models:** HWSNs are usually a mix of multiple sensor models. They may not observe all properties at the same location, which is often one of the underlying assumptions of studies based on traditional sensor networks.
- **Heterogeneity from dynamic sensor network topology:** Unlike the long-lived traditional measuring stations, the lifespan of low-cost sensors is unstable. Sometimes, deploying new ultra-low-cost sensors is even more affordable than retrieving and repairing old ones. These factors keep the spatial structure of HWSNs changing. Figure 2a illustrates how the total device amount of one of such sensor networks changes over time, while Figure 2b shows when sensors in this network successfully returned data and when did not. It is easy to figure out that the topology of HWSNs is dynamic. Furthermore, in addition to stationary sensors, some sensor networks also partially [13, 4] or fully [14] employ moveable sensors, which further enhances the heterogeneity of the spatial structure of the sensor network.
- **Heterogeneity due to low-power wireless communication protocols:** Deployment of traditional sensors often faces administrative difficulties, such as applying for land, power supply, and network access from local administrations. As a result, many HWSNs turn to using low-power wireless communication technologies such as LoRaWAN, Zigbee, BLE, etc. These communication protocols allow sensors to operate on batteries alone for a considerable period and send their observations to the data center wirelessly. The cost of this is usually a restricted uplink bandwidth and transmission time window. Even if the network delivers complete data at the end, the real-time heterogeneity induced by these protocols must be considered when considering the actual deployment of the model for real-time usage.

- **Uncertainty in sensor readings:** The accuracy of a sensor, not only in terms of its accuracy on its observed properties but also its position recorded by the mounted GPS module, is generally related to its price level. Some studies have also pointed out that how low-cost sensors are assembled and the environment they operate in can also harm their measurements. In severe cases, it can even return only qualitative results [3]. Moreover, maintaining HWSNs is a complex, long-term task requiring much manual logging during the installation, repair, and transfer of sensors. Considering that the operation period of HWSNs is often measured in years, human errors are almost unavoidable, and a significant portion of them are challenging to identify and fix in quality checks.

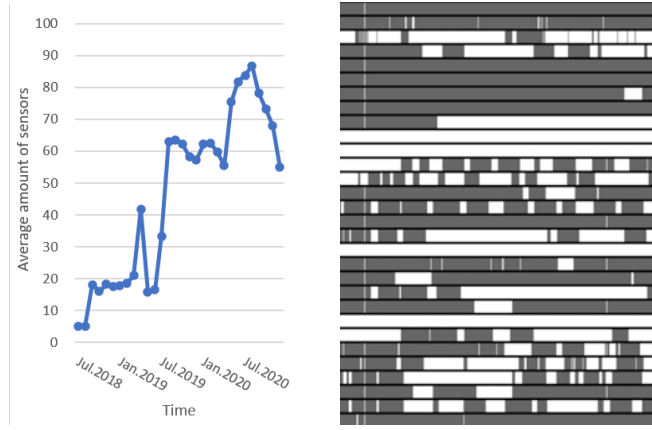


Fig. 2: (a). The curve of the monthly average active sensors in the SmartAQnet dataset [13] (Jul. 2018 to Dec. 2020). (b). Daily activity status of low-cost sensors in the SmartAQnet dataset (2021.01.01 to 2022.01.01). Each row represents a sensor, with white indicating no readings collected in the day and black indicating the opposite.

2.2 Spatial Interpolation

Spatial interpolation aims to predict values of a target property at any location (mostly locations without historical observations) according to known observations. It is an essential spatial data analysis task widely used in atmosphere, geology, and urban studies.

Early machine learning approaches, such as K-Nearest Neighbors and Random Forest, are simple statistical models and struggle to capture complex and dynamic correlations [10]. Gaussian Processes (GP, also Kriging) [5, 17] offer greater flexibility with custom kernel functions and provide reliable probabilistic estimates.

Deep learning methods have gained popularity in spatial interpolation, with two prominent families: Graph Neural Networks (GNNs) and Transformers. GNN-based models treat locations with known observations as graph nodes, capturing patterns of information transfer through message-passing mechanisms. Numerous GNN models [9, 11, 22] have been migrated to the field with promising results. Researchers have also improved these GNN models regarding the specific needs of the spatial interpolation task [1, 12, 18].

Transformer-based models interpret the input as a sequence of tokens. They adaptively extract the correlation between input tokens with the multi-head self-attention mechanism. However, we also note that existing transformer-based spatial interpolation models still use dense input that takes all known observations of the same location as a token, benefiting from its homogeneity to learn stable representations. For example, Fan et al. [7] put known observations on grid maps and processed them with Vision Transformer, Yu et al. [23] removes all sensing stations with more than 25% missing data, and Feng et al. [8] interpolates the missing data with linear interpolation. However, we believe that Transformer could also treat sparse observations as variable-length token sequences and, therefore, be highly compatible with the heterogeneity of HWSN datasets. This paper will explore whether we can extract stable, generalizable representations for spatial interpolation tasks from the sparse HWSN data.

3 Methodology

3.1 Preliminaries

Notations We regard a HWSN dataset $D = \{F_j \mid j = 1, 2, \dots, n\}$ as a collection of Frames F_j . Each Frame $F_j = \{O_i \mid i = 1, 2, \dots, m\}$ contains all the Observations O_i recorded at a same time, which an example is illustrated as the table in Figure 1 (right). Each Observation $O_i = (P, C, V)$ is a triplet of a one-hot encoded Property P , a two- or three-dimensional Coordinate C , and a Value V , which an example is illustrated as a row in Figure 1 (right).

We refer to the Property that needs to be interpolated as the Target Property, abbreviated as P_{tgt} . For our model, we only consider one Target Property at a time. Since the spatial distribution of the Target Property is usually not only affected by spatial correlation but also correlated with some other Properties, HWSN datasets also observe these correlated Properties. They are called Support Properties, abbreviated as P_{sup} . Thus, a Frame F can be further divided into two parts: Target Sequence F_{tgt} includes all the Observations of P_{tgt} and Support Sequence F_{sup} includes all the Observations of P_{sup} .

Spatial Interpolation Task Given an input Frame F' (F' may not in D), the spatial interpolation task is to predict the value V' of the P_{tgt} at any arbitrary target location C' . The basis for interpolation comes from the spatial correlation with the known values in F'_{tgt} and the effect of F'_{sup} on this correlation, which can be learned from Frames provided in D .

3.2 Framework

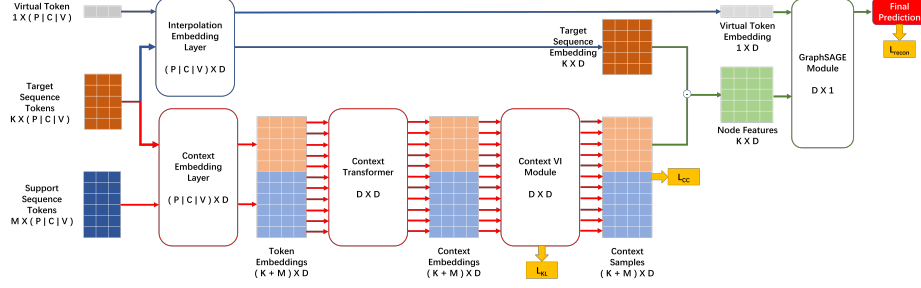


Fig. 3: An overview of the CESI model. The upper half is the GI Module, which mainly models the spatial correlations. The bottom half shows the TCE Module, which models the influence of the Support Sequence on the spatial correlations.

The main challenge of sparse input models on HWSN datasets is the contradiction between the requirement of adaptively discovering complex correlations and defective datasets, mainly manifested in low spatial coverage limited by the amount of the sensor and high noise due to the introduction of low-cost sensors. Such problems are usually solved in other fields by obtaining more data sources or using data augmentation approaches. However, in spatial interpolation tasks, such methods are generally limited. We can no longer return to the past to collect data from more locations, and we also lack prior knowledge of those Target Properties affected by complex systems for artificially creating more data. It's worth noting that some heuristics widely used in other fields, such as translation and transposition, are also risky in fields like meteorology, where spatial correlations are significantly affected by longitude, latitude, and azimuth.

These challenges necessitate a robust model design. Models with weak inductive biases, like Transformers [21], excel at capturing complex correlations but heavily depend on data quality and quantity, making their results unstable on HWSN datasets. Conversely, models with strong inductive biases, such as KCN [1] or even Inverse Distance Weighting Interpolation, while based on simple assumptions, perform surprisingly strongly on specific datasets. Nevertheless, they also risk their inductive biases being mismatched with the dataset. To address this, we propose a hybrid strategy: a strong inductive bias module serves as the backbone, complemented by a weak inductive bias module as an auxiliary component. A self-supervised signal dynamically balances the two modules, enabling better adaptation to different tasks.

Transformer-based Context Embedding (TCE) Module We design a Transformer-based module as our auxiliary component, whose structure is illustrated as the bottom part of Figure 3. It learns the observation-level influence of

the Support Properties on the spatial correlation of the Target Property. This influence is eventually encoded as Context Samples, which are subsequently used to correct the inputs of the GraphSAGE Module.

The TCE Module starts with input centering, that is, replacing the absolute coordinates in each observation of the input Frame F with its relative coordinates to the target location C' :

$$F_{cen} = \{(P_i, C_i - C', V_i) \mid i = 1, 2, \dots, m\} \quad (1)$$

With input centering, we hide the information of specific coordinates in the input Frame, forcing the TCE Module to concentrate on more generalizable spatial correlations. F_{cen} is then embedded by a multi-layer perceptron (MLP) and further processed by the Context Transformer. The Context Transformer is without positional embedding, making it order-independent for the input sequence. With the multi-head self-attention mechanism, each output token of the Context Transformer is obtained after referring to the information of all tokens in F_{cen} . In our design, the underlying intuition here is: for each input token, assuming that all other tokens are noise-free, how much should we adjust its embedding?

The output of the Context Transformer is a deterministic encoding that maps each token to a specific point in the latent space. As the reconstruction error decreases, the model risks overfitting noise in the dataset, leading to degraded performance. We use Variational Inference (VI) to learn a smooth, probabilistic latent space to address this. In probabilistic encoding, the data with noise is treated as a sample of the learned distribution. We construct a continuous and smooth latent space by repeatedly sampling from the learned distribution and ensuring these samples yield consistent outputs. This approach significantly enhances the model's generalization ability while providing meaningful uncertainty estimates for the final output. Specifically, we assume the posterior distribution in the latent space $q(z|x)$ follows a Gaussian distribution. The deterministic encoding x is passed through two MLPs to predict the mean μ and variance σ^2 of $q(z|x)$, respectively. Using the reparameterization trick, we sample a random Context Sample from $q(z|x)$:

$$z = \mu + \sigma\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (2)$$

To align the learned posterior $q(z|x)$ with the standard normal prior $p(z) \sim \mathcal{N}(0, \mathbf{I})$, we minimize their KL divergence:

$$L_{KL} = D_{KL}(q(z|x) \parallel p(z)) \quad (3)$$

GraphSAGE-based Interpolation (GI) Module We select GraphSAGE as our backbone module, whose structure is illustrated as the upper part of Figure 3. GraphSAGE assumes that the message-passing process follows the graph's topology, exchanging information within local neighborhoods through shared aggregation and update functions. This represents a relatively strong inductive bias. First, we construct a Virtual Token representing the target location in the format of an observation, in which the Value is filled as zero:

$$O_v = (P_{tgt}, C', 0) \quad (4)$$

The Virtual Token, along with the tokens in F_{tgt} , is then encoded by an MLP, which is the Interpolation Embedding Layer in Figure 3, resulting in the Virtual Token Embedding and the Target Sequence Embeddings. Next, we use the Context Samples from the TCE Module to correct their corresponding Target Sequence Embeddings, resulting in Node Features. The Virtual Token Embedding and the Node Features are then together treated as the node feature matrix of the input graph. The adjacency matrix of the input graph is constructed using the k-nearest neighbors heuristic. Then, GraphSAGE is applied to process this graph. Finally, the GraphSAGE output corresponding to the Virtual Token is fed into an MLP Head to produce the interpolation result V' . We use the mean absolute error between V' and the label L as part of the supervisory signal for model training, named reconstruction loss:

$$L_{recon} = MAE(V', L) \quad (5)$$

Context Correction Loss In addition to L_{recon} and L_{KL} , we introduce another self-supervision loss signal, named Context Correction Loss L_{CC} , to automatically balance the inductive bias of the two modules. It is the average of the L1 Norm of all Context Samplings:

$$L_{CC} = \frac{1}{n} \sum_{i=1}^n \|CS_i\|_1 \quad (6)$$

Introducing L_{CC} can bring the following benefits that stabilize the model’s performance. First, since the input of the GraphSAGE Module is a linear combination of Target Sequence Embeddings and Context Samples, by limiting the Context Samples to the global minimum, the L_{CC} can make sure that the GraphSAGE Module dominates the training when backpropagating the L_{recon} . This ensures that the GraphSAGE module keeps being the central component of the pipeline, restricting the Transformer’s strong trend of overfitting as a module with weak inductive bias. Second, since the Context Samples are sampled from a Gaussian distribution q learned by the Context VI Module, minimizing L_{CC} can constrain the standard deviation of q , preventing the model from identifying the major part of the input as noise and converging to suboptimal results. Third, the L_{CC} encourages the TCE Module to correct the inputs with the minimum possible corrections. This can be thought of as an Occam’s razor-based heuristic. When a simple and a complex correction achieves similar results on a poorly sampled dataset, we will prefer the simpler one, thus reducing the overfitting.

The final loss Signal of the model pipeline is a linear combination of L_{recon} , L_{KL} , and L_{CC} :

$$L = L_{recon} + L_{KL} + L_{CC} \quad (7)$$

Table 1: Comparison of Datasets Included in this Study

Name	Sensor Type	Noise Level	P_{sup} Channels	Average F Length	Spatial Coverage Rate ¹	Missing rate ²
SAQN	All fixed-location	High	8	200.31	12.30%	52.54%
ABO	All movable	Low	2	460.52	4.68%	0.38%
Marine	mixed	Pass quality inspection	5	339.62	97.96%	21.15%

1. How many grids have been observed at least once in the entire dataset
2. How many input cells are missing when expressed as dense input

4 Experiments

4.1 Experimental Setup

Datasets We evaluate CESI on three publicly available real-world datasets: the SmartAQnet dataset (SAQN) [13], the NOAA Aircraft Based Observation dataset (ABO) [19], and the Copernicus In-situ Marine Observation dataset (Marine) [4]. Table 1 provides detailed dataset information.

The SAQN dataset is a typical fixed-location HWSN dataset that monitors urban air quality and meteorological conditions. It is characterized by a high missing rate and considerable noise due to deploying numerous low-cost sensors. Furthermore, its spatial coverage is limited as it relies exclusively on fixed-location sensors. In contrast, the ABO and Marine datasets reflect the trend of incorporating movable sensors in HWSN datasets for higher spatial coverage, which leads to more complex sensor topologies. The ABO dataset, which monitors meteorological parameters using sensors mounted on commercial aircraft, is distinguished by its low noise and extremely low missing rate. However, despite its larger number of observations in each Frame, the ABO dataset still exhibits limited spatial coverage as it is the only three-dimensional dataset included in our analysis. The Marine dataset, on the other hand, measures hydrological and meteorological parameters. Its use of a wide array of movable sensors results in high spatial coverage. This dataset also resembles a traditional dataset, given its relatively low missing data rate and the implementation of strict quality inspection processes. Experiments using the Marine dataset also provide an opportunity to evaluate the effectiveness of our model on more conventional datasets.

Baselines We involve GraphSAGE [9] and Transformer [21] into baselines, as they are the base components of our model. From the GNN-based spatial interpolation models, we involve GAT [22], KSAGE [1], PE-SAGE [12], LSPE [6], and SPONGE [18]. From the attention-based spatial interpolation models, we involve SSIN [15], and SMACNP [2]. Experiment codes are provided in the additional materials.

Data Preprocessing The SAQN dataset uses SmartAQnet data from January 1, 2017, to December 31, 2021. The time interval of the Frame is 1 hour. The observed area is a rectangular area within 14 kilometers north and east from

10.7992° E and 48.421° N. The target OP is PM10. Support OPs include PM2.5, temperature, relative humidity, air pressure, longitudinal wind speed, latitudinal wind speed, precipitation, and solar radiation.

For the ABO dataset, we selected all observations in this dataset from July 1, 2001, to April 1, 2004, located in the range of 74° W to 77° W, and 39° N to 42° N. The time interval of the Frame is 1 hour. The target variable is air temperature, and the support variables include wind speed and wind direction.

For the Marine dataset, we selected all observations in this dataset from January 1, 1900, to December 31, 2010, located in the range of 36.0° W to 11.0° W, and 31.0° N to 56.0° N. The time interval of the Frame is 4 hours. The target variable is water temperature, and the support variables include air temperature, air pressure, dew point, wind speed, and wind direction.

The following preprocessing steps are common to all the datasets:

- **Step 1:** Exclude outliers. In this step, we use the threshold method to exclude outliers that do not comply with physical laws. The preprocessing code provides further detail.
- **Step 2:** Split the Frames. We split the observations in the dataset into different Frames according to the time intervals mentioned above.
- **Step 3:** Spatial aggregation. We further partition the horizontal space into a 250×250 grid for each Frame. Then, we aggregate the readings with the same coordinates by averaging.
- **Step 4:** Filter the Frames. We only retain Frames that provide at least 5 nodes for training and one node for evaluation. For datasets that still have more than 20,000 Frames after filtering, we retain the 20,000 Frames with the latest timestamps.

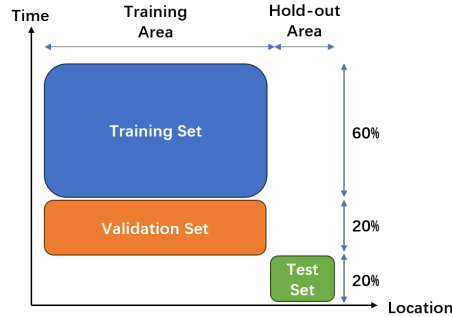


Fig. 4: Our strategy for dividing the dataset. With this strategy, we ensure that the models are tested only at times and locations that have never been seen during training and validation.

Evaluation Strategy Figure 4 illustrates our strategy for dividing the dataset.

In the temporal dimension, we divide all the Frames into three parts according to their temporal order: 60%, 20%, and 20% each, which are used in the model’s training, validation, and testing, respectively.

In the spatial dimension, we divided the study area into four equal parts by dividing the length and width of the horizontal area equally. We adopt the leave-one-area-out cross-validation method and, in turn, use four areas as hold-out areas. The training and validation of the model are performed only with the label within the three non-hold-out regions, while the model is tested only with the label in the hold-out region. With the above evaluation strategy, we ensure that the models are tested only at times and locations that have never been seen during training and validation.

Further, test locations might be densely surrounded by other sensors, making it hard to evaluate whether the model performs well in the target locations that are very remote. Therefore, when testing the model, for each target location, in addition to testing with the complete Frame, we also use two other Frames that remove all nodes within 20 or 50 pixels by Manhattan distance of the target location, simulating the situation that target locations of different levels of remoteness.

As in other literature in the field, we choose the mean absolute error (MAE) and coefficient of determination (R^2) between the model output and the label as the metrics to evaluate the model performance. We first calculate the performance of four-fold leave-one-area-out cross-validations under each random seed and then calculate the mean and standard deviation of the results between different random seeds.

Environments We conduct our experiments on an HPC cluster. Models that demand lower computational resources, including GCN, GraphSAGE, KCN, and PE-GNN, are trained on CPU nodes equipped with 20 Intel Xeon Gold 6230 CPUs and 192 GB of memory. Other models are trained on GPU nodes equipped with 20 Intel Xeon Gold 6230 CPUs, 192 GB of CPU memory, 2 NVIDIA Tesla V100 GPUs, and 64 GB of GPU memory.

The system used for all nodes is Red Hat Enterprise Linux (RHEL) 8.4. The training environment is based on Python 3.10.12, Pytorch 2.1.0 + CUDA 12.1, Pytorch-geometric 2.4.0, DGL 2.2.1, Numpy 1.26.1, Pandas 2.1.1, Scikit-learn 1.3.1, and Scipy 1.11.3.

4.2 Overall Performance

After random searches on hyperparameters, each model was evaluated with four-fold leave-one-area-out cross-validations and five random seeds (1, 2, 3, 4, and 5). Table 2 shows the overall performance.

Q1: Which model demonstrates the best overall performance?

A1: CESI achieves the best average MAE and R^2 on all datasets. On the ABO dataset, Transformer and SMACNP rank second and third, respectively, while Transformer and SSIN occupy these positions on the Marine dataset. On

Table 2: Overall Result of all models. Bold indicates the best performer, underline indicates the second place

Model	ABO		SAQN		Marine	
Metrics	MAE	R^2	MAE	R^2	MAE	R^2
GraphSAGE	10.293 \pm 0.044	0.451 \pm 0.004	5.863 \pm 0.048	<u>0.317 \pm 0.009</u>	1.993 \pm 0.038	0.631 \pm 0.012
Transformer	<u>1.811 \pm 0.639</u>	<u>0.972 \pm 0.025</u>	6.041 \pm 0.437	0.184 \pm 0.104	<u>0.971 \pm 0.144</u>	<u>0.903 \pm 0.027</u>
KSAGE	14.268 \pm 0.021	0.012 \pm 0.002	<u>5.535 \pm 0.048</u>	0.301 \pm 0.010	3.128 \pm 0.018	0.198 \pm 0.007
PE-SAGE	3.302 \pm 0.258	0.927 \pm 0.008	6.115 \pm 0.243	0.217 \pm 0.047	1.315 \pm 0.042	0.835 \pm 0.011
LSPE	13.844 \pm 0.424	-0.390 \pm 0.409	6.205 \pm 0.115	0.171 \pm 0.030	1.660 \pm 0.084	0.721 \pm 0.026
SPONGE	3.918 \pm 0.296	0.913 \pm 0.013	6.388 \pm 0.138	0.249 \pm 0.019	1.593 \pm 0.071	0.768 \pm 0.023
SSIN	18.800 \pm 0.469	-0.420 \pm 0.062	6.197 \pm 0.084	0.167 \pm 0.034	1.035 \pm 0.052	0.893 \pm 0.014
SMACNP	3.241 \pm 0.281	0.884 \pm 0.025	6.237 \pm 0.337	0.201 \pm 0.062	1.741 \pm 0.044	0.287 \pm 0.127
CESI	1.426 \pm 0.040	0.987 \pm 0.001	5.362 \pm 0.110	0.334 \pm 0.008	0.944 \pm 0.036	0.910 \pm 0.009

the SAQN dataset, however, KSAGE and GraphSAGE take second and third place, as the above models experience significant degradation. In conclusion, CESI consistently outperforms all baselines across all three datasets, highlighting its adaptability.

Q2: Is sparse input a beneficial choice for spatial interpolation?

A2: Sparse input is beneficial but presents challenges. On ABO and Marine datasets, even the Vanilla Transformer surpasses dense input baselines on average performance. However, sparse input models are more sensitive to noise and bias in lower-quality datasets, such as the Transformer failure on the SAQN dataset, and its performance is volatile on all the datasets. CESI effectively addresses this challenge, with MAE standard deviations 93.7%, 74.8%, and 75.0% lower than Transformer on ABO, SAQN, and Marine datasets, respectively. CESI’s stability is competitive even against dense input models.

Q3: Why do many models degrade performance on the SAQN dataset?

A3: First, the SAQN dataset only contains fixed-location sensors, coupled with a low spatial coverage rate, resulting in a high location-related bias in the dataset. Models that employ learnable location-based encodings (e.g., PE-SAGE, LSPE, SPONGE, SSIN) are particularly susceptible to these biases, leading to significant performance degradation. Second, the SAQN dataset has the highest heterogeneity and noise level. Models lacking stable inductive bias (Transformer and SMACNP) tend to overfit the noise, resulting in volatile performances. Our model, on the contrary, successfully overcomes these challenges.

Q4: Why is the performance on the ABO dataset so polarized?

A4: Models like GraphSAGE, KSAGE, and SSIN use Euclidean distance-based heuristics for encoding spatial relationships, and unlike PE-SAGE and CESI, they do not incorporate additional location-based embeddings. The hidden inductive bias of such heuristics is the spatial isotropy of the Euclidean distance. However, on the ABO dataset, the Target Property (air temperature) has an evident stratification along the altitude dimension. This reminds us again that we should be cautious when introducing inductive bias into model design. When the inductive bias of the model is consistent with the actual situation of the dataset, we can learn a good model with less and worse data. However, when the

Table 3: Result of Ablation Study. Bold indicates the best performer, underline indicates the second place

Model	ABO		SAQN		Marine	
Metrics	MAE	R^2	MAE	R^2	MAE	R^2
CESI	1.426 ± 0.040	0.987 ± 0.001	5.362 ± 0.110	0.334 ± 0.008	0.944 ± 0.036	0.910 ± 0.009
CESI w/o L_{KL}	2.020 ± 0.187	0.972 ± 0.005	6.018 ± 0.156	0.170 ± 0.030	<u>0.900 ± 0.025</u>	<u>0.915 ± 0.008</u>
CESI w/o L_{CC}	<u>1.489 ± 0.045</u>	<u>0.986 ± 0.001</u>	<u>6.044 ± 0.500</u>	<u>0.214 ± 0.108</u>	0.980 ± 0.019	0.896 ± 0.008
CESI Null	2.285 ± 0.064	0.968 ± 0.002	8.548 ± 1.170	-0.672 ± 0.654	0.865 ± 0.031	0.922 ± 0.005

model’s inductive bias conflicts with the dataset’s actual situation, the model’s performance will be negatively affected.

4.3 Ablation Study

We use the following ablation models to study the effectiveness of each module: CESI w/o L_{KL} model removes the probabilistic encoding and its associated L_{KL} , CESI w/o L_{CC} model removes the Context Correction Loss L_{CC} , and CESI Null model simultaneously removes the both. All experiment settings are the same as above. Table 3 shows the results of the ablation study.

On the ABO dataset, both modules contribute to performance improvement. The main contribution comes from probabilistic encoding, while L_{CC} further refines the performance. On the SAQN dataset, the contribution on average MAE from both modules is roughly the same, and L_{CC} provides more stability improvement than probabilistic encoding.

However, our modules had a slight adverse effect on the Marine dataset. The probabilistic encoding and L_{CC} are designed to address bias and noise in datasets. However, these occurred less in the Marine dataset. First, the dataset has undergone strict quality checks, making it generally noise-free. Second, it boasts exceptionally high spatial coverage (up to 97.96%), minimizing location-related bias. This led to misattributions of our modules, where the probabilistic encoding mistakenly interpreted some genuine correlations as noise, resulting in the significant performance drop of CESI w/o L_{CC} . From the performance of CESI and CESI w/o L_{KL} , we observed that L_{CC} effectively served its intended purpose of constraining such misattributions yet did not fully mitigate the performance decline. Nevertheless, as the overall results demonstrated, this did not prevent the model from achieving state-of-the-art performance. This highlights that our model’s competitive edge relies not solely on exploiting flawed datasets but also on learning fine-grained observation-level correlations.

We conducted additional experiments on robustness to missing rates and noise using the Marine dataset to validate our explanation.

4.4 Experiments on Robustness

In the robustness experiment, we randomly mask 20%, 40%, 60%, and 80% of the observations from each Frame in the Marine Dataset to increase its missing rate, and we randomly add multiple Gaussian noise with different standard deviations

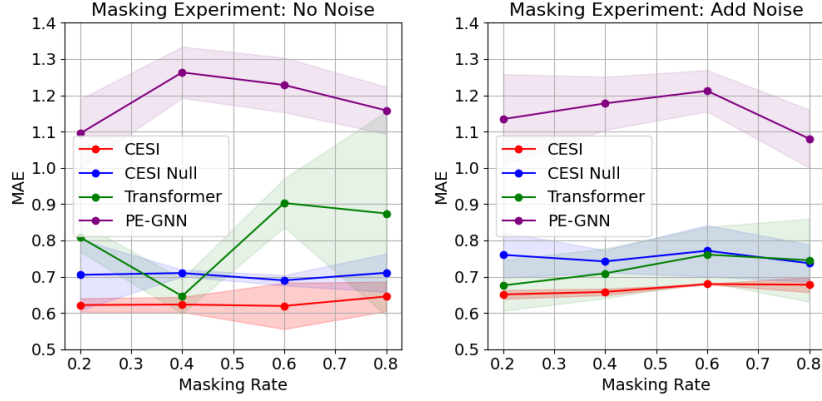


Fig. 5: Result of Robustness Experiment, the shaded area marks the standard deviation

Table 4: Result of Robustness Experiment. Bold indicates the best performer, underline indicates the second place

Masking Rate	20%				40%				60%				80%			
Metrics	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
Without additional noise																
CESI	0.622 ± 0.019	0.874 ± 0.005	0.623 ± 0.021	0.878 ± 0.006	0.619 ± 0.064	0.878 ± 0.026	0.645 ± 0.041	0.872 ± 0.017								
CESI Null	0.705 ± 0.097	0.850 ± 0.040	0.710 ± 0.009	0.843 ± 0.005	0.690 ± 0.014	0.857 ± 0.008	0.711 ± 0.053	0.848 ± 0.019								
Transformer	0.809 ± 0.040	0.797 ± 0.013	0.646 ± 0.048	0.864 ± 0.025	0.903 ± 0.067	0.722 ± 0.042	0.874 ± 0.281	0.690 ± 0.234								
PE-GNN	1.095 ± 0.094	0.631 ± 0.066	1.263 ± 0.071	0.508 ± 0.054	1.228 ± 0.075	0.525 ± 0.067	1.158 ± 0.065	0.589 ± 0.045								
With additional noise																
CESI	0.651 ± 0.012	0.869 ± 0.003	0.658 ± 0.009	0.867 ± 0.001	0.679 ± 0.001	0.862 ± 0.001	0.678 ± 0.021	0.856 ± 0.004								
CESI Null	0.760 ± 0.063	0.834 ± 0.025	0.742 ± 0.031	0.840 ± 0.012	0.771 ± 0.071	0.829 ± 0.028	0.737 ± 0.051	0.841 ± 0.018								
Transformer	0.676 ± 0.070	0.859 ± 0.028	0.709 ± 0.069	0.845 ± 0.032	0.761 ± 0.076	0.823 ± 0.033	0.745 ± 0.114	0.826 ± 0.051								
PE-GNN	1.134 ± 0.124	0.603 ± 0.089	1.178 ± 0.073	0.594 ± 0.049	1.212 ± 0.057	0.559 ± 0.057	1.080 ± 0.080	0.651 ± 0.047								

to varying proportions of data. Then, we train CESI, CESI Null, Transformer, and PE-GNN models on these datasets. We train with random seeds 1, 2, and 3 for each model, respectively. The results are summarized in Table 4 and Figure 5.

Obviously, (1). the CESI model performs best in all experiments. (2). Although we added noise with different standard deviations to different proportions of data, the noise didn't significantly affect the performance of the CESI model. Since the Gaussian noise added is consistent with the preset of probabilistic encoding, after getting rid of the misattribution, the stability of the model is even improved. (3). Dense input models represented by PE-GNN are hardly affected by the missing rate and noise because the model and data augmentation provide very stable inductive biases. However, as a price, it sacrifices the ability to discover fine-grained correlations, so the overall performance is the worst.

The above concludes that our design works as expected and can maintain the model's performance and stability under different noise and missing rates.

5 Conclusion

We propose the CESI Model for HWSN datasets. Our model directly takes the narrow format sparse input and learns their correlations. Since HWSN datasets usually exhibit small-scale, low spatial sampling rates and considerable noise, we use probabilistic encoding and a self-supervision signal named Context Correction Loss to extract encodings conducive to better generalizing to coordinates not present in the training set. As a result, we effectively improve the model’s performance and stability. Experiments across several publicly available real-world HWSN datasets with different characteristics show the CESI Model holds significant potential for broader applications, such as enhancing data-driven decision-making in environmental monitoring, urban planning, and other domains reliant on sparse spatial data.

Acknowledgments

We thank the Helmholtz European Partnership for Technological Advancement (HEPTA) for supporting this study. The state of Baden-Württemberg also supported this work through bwHPC.

References

1. Appleby, G., Liu, L., Liu, L.P.: Kriging convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3187–3194 (2020)
2. Bao, L.L., Zhang, J.S., Zhang, C.X.: Spatial multi-attention conditional neural processes. *Neural Networks* **173**, 106201 (2024)
3. Budde, M., Schwarz, A.D., Müller, T., Laquai, B., Streibl, N., Schindler, G., Köpke, M., Riedel, T., Dittler, A., Beigl, M., et al.: Potential and limitations of the low-cost sds011 particle sensor for monitoring urban air quality. *ProScience* **5**(6), 12 (2018)
4. CDS: Global marine surface meteorological variables from 1851 to 2010 from comprehensive in-situ observations. Copernicus Climate Change Service (C3S) Climate Data Store (CDS) (2021), <https://cds.climate.copernicus.eu/cdsapp#!/dataset/10.24381/cds.27f643d7>, dOI: 10.24381/cds.27f643d7
5. Cui, T., Pagendam, D., Gilfedder, M.: Gaussian process machine learning and kriging for groundwater salinity interpolation. *Environmental Modelling & Software* **144**, 105170 (2021)
6. Dwivedi, V.P., Luu, A.T., Laurent, T., Bengio, Y., Bresson, X.: Graph neural networks with learnable structural and positional representations. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=wTTjnvGphYj>
7. Fan, H., Cheng, S., de Nazelle, A.J., Arcucci, R.: An efficient vit-based spatial interpolation learner for field reconstruction. In: International Conference on Computational Science. pp. 430–437. Springer (2023)
8. Feng, Y., Kim, J.S., Yu, J.W., Ri, K.C., Yun, S.J., Han, I.N., Qi, Z., Wang, X.: Spatiotemporal informer: A new approach based on spatiotemporal embedding and attention for air quality forecasting. *Environmental Pollution* **336**, 122402 (2023)

9. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. *Advances in neural information processing systems* **30** (2017)
10. Hu, J., Liang, Y., Fan, Z., Chen, H., Zheng, Y., Zimmermann, R.: Graph neural processes for spatio-temporal extrapolation. *arXiv preprint arXiv:2305.18719* (2023)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016)
12. Klemmer, K., Safir, N.S., Neill, D.B.: Positional encoder graph neural networks for geographic data. In: *International Conference on Artificial Intelligence and Statistics*. pp. 1379–1389. PMLR (2023)
13. Li, C., Budde, M., Tremper, P., Schäfer, K., Riesterer, J., Redelstein, J., Petersen, E., Khedr, M., Liu, X., Köpke, M., et al.: Smartaqnet 2020: a new open urban air quality dataset from heterogeneous pm sensors. *Proscience* **8** (2022)
14. Li, J.J., Faltings, B., Saukh, O., Hasenfratz, D., Beutel, J.: Sensing the air we breathe—the opensense zurich dataset. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 26, pp. 323–325 (2012)
15. Li, J., Shen, Y., Chen, L., Ng, C.W.W.: Ssin: Self-supervised learning for rainfall spatial interpolation. *Proceedings of the ACM on Management of Data* **1**(2), 1–21 (2023)
16. Liang, S., Khalafbeigi, T., van Der Schaaf, H., Miles, B., Schleidt, K., Grellet, S., Beaufls, M., Alzona, M.: Ogc sensorthings api part 1: Sensing version 1.1. In: *Open geospatial consortium* (2021)
17. Lucas, M.P., Longman, R.J., Giambelluca, T.W., Frazier, A.G., Mclean, J., Cleveland, S.B., Huang, Y.F., Lee, J.: Optimizing automated kriging to improve spatial interpolation of monthly rainfall over complex terrain. *Journal of Hydrometeorology* **23**(4), 561–572 (2022)
18. Njifon, M.A., Schuhmacher, D.: Graph convolutional networks for spatial interpolation of correlated data. *Spatial Statistics* **60**, 100822 (2024)
19. NOAA: Aircraft based observation (abo) dataset (2023), https://madis.ncep.noaa.gov/madis_acars.shtml
20. Petrova-Antonova, D., Jelyazkov, J., Pavlova, I.: Air quality monitoring platform with multiple data source support. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* pp. 1–17 (2021)
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017)
23. Yu, M., Masrur, A., Blaszcak-Boxe, C.: Predicting hourly pm_{2.5} concentrations in wildfire-prone areas using a spatiotemporal transformer model. *Science of The Total Environment* **860**, 160446 (2023)