

From Deterministic to Generative Deep Learning for Urban Air Quality Reconstruction from Sparse Observations

Abhishek Ajit Sabnis^{1,2}, Mihai Mitrea², Lya Lugon¹,
Karine Sartelet¹, Marc Bocquet¹, Xiaoyuan Cheng³,
Shupeng Zhu⁴, Sib0 Cheng^{1*}

^{1*}CEREA, ENPC, Institut Polytechnique de Paris, Île-de-France,
France.

²Department of Computing, Imperial College London, London, UK.

³Dynamic Systems Lab, University College London, London, UK.

⁴Department of Atmospheric Sciences, School of Earth Sciences,
Zhejiang University, Hangzhou, China.

*Corresponding author(s). E-mail(s): sibo.cheng@enpc.fr;

Contributing authors: abhishek-ajit.sabnis@enpc.fr;

mihai.mitrea24@imperial.ac.uk; lya.lugon@enpc.fr;

karine.sartelet@enpc.fr; marc.bocquet@enpc.fr;

xiaoyuan.cheng.21@ucl.ac.uk; shupengz@zju.edu.cn;

Abstract

Full-field reconstruction of air pollution is essential for evaluating pollution exposure and supporting public health decision-making. However, the complex interactions among pollutants, hard-to-predict weather patterns, and limited monitoring-station coverage make this a complex task. We apply deep learning techniques to provide fast and accurate reconstructions from sparse observations of four key pollutants: NO₂, O₃, PM_{2.5} and PM₁₀. Models are trained on full-field simulation data and evaluated on real-world observations collected from **9** to **28** monitoring stations in the city of Paris. We introduce a diffusion-based generative framework for multi pollutant reconstruction and benchmark its performance against deterministic deep learning models. Despite noisy observations and strong spatial variability, the models achieve high structural similarity on simulated validation data and produce realistic spatial patterns on real-world observations, as

indicated by power-spectrum analysis. We introduce data augmentation methods that enable transfer to real-world observations without retraining, allowing the models to generalise beyond the training period. These findings highlight the potential of ML models for reliable real-world deployment in air pollution reconstruction tasks.

Keywords: Air pollution, Deep learning, Generative AI, Sparse observations

1 Introduction

Increased human activity makes it difficult to predict variations in air quality. Exposure to pollutants such as nitrogen dioxide (NO_2), particulate matter (PM_{10} , $\text{PM}_{2.5}$) and ozone (O_3) is strongly associated with respiratory illness [1], cardiovascular disease [2], and premature mortality [3]. Although global PM_{10} concentration tends to decrease, $\text{PM}_{2.5}$, NO_2 and O_3 have increased in 65%, 71% and 89% of cities around the world [4]. Accurate estimation of pollutant concentrations is essential for assessing pollution exposure, addressing associated health impacts [5], and helping policymakers design effective measures [6].

Deterministic modelling with chemistry-transport (CT) models can be used to estimate pollution exposure by providing spatially and temporally resolved concentration fields [7, 8]. However, these estimates remain subject to uncertainties arising from several sources, including emission inventories, meteorological fields, and the representation of atmospheric chemical and physical processes. To reduce these uncertainties, simulated concentrations can be corrected using ground-based observations through spatial interpolation-based methods, such as kriging [9], or data assimilation techniques [8, 10]. These approaches are generally associated with higher computational costs and rely on precise prior knowledge of error distributions and their spatial correlations, which limits their applicability in real-time monitoring scenarios.

To address this, we investigate the reconstruction of spatial fields directly from sensor observations, without relying on CT models or data assimilation during inference. While data from monitoring stations provide us with real-time observation values, these stations are usually sparse and unevenly distributed. High-precision ground stations are expensive to deploy and maintain. By contrast, low-cost sensors that are more scalable often suffer from limited accuracy, higher sensitivity, and require regular calibration [11, 12]. Additionally, pollutant concentration fields can exhibit abrupt transitions, leading to nonlinear and nonstationary spatial patterns that vary across pollutants [13]. Urban air-quality reconstruction remains an intrinsically ill-posed inverse problem [14, 15] without prior simulations. Sparse monitoring observations do not uniquely determine the underlying pollution field because multiple atmospheric states can be consistent with the same set of measurements. Because observations are often sparse and noisy, there is a need for reconstruction methods that remain robust under these conditions.

In recent years, the use of machine learning techniques has shown rapid progress in the reconstruction of air pollution fields [16–18]. Although these methods have higher

reconstruction accuracy than traditional approaches, several important limitations remain unaddressed. First, most studies model pollutants individually or train separate models for each [16], ignoring strong inter-pollutant correlations [19]. Second, studies using methods trained on synthetically generated full-field data rarely evaluate the reconstruction quality on real-world observations, as noisy measurements limit model transfer to new data. Moreover, real-world sensor networks switch on and off over time, leading to temporally incomplete observations. This practical challenge is often overlooked in existing modelling frameworks [20].

In order to address these highlighted gaps, we adopt a strategy that jointly models multiple pollutants and performs an ML-based reconstruction. We work with two different datasets, the *simulation* data to train the model and *observation* data to evaluate on real-world air quality measurements from the Paris metropolitan area. To provide a spatially consistent representation of the sensor network to the ML model, Voronoi tessellations are constructed. This also mitigates the issues arising from temporally missing observations.

The main contribution of this study is the application of a diffusion-based generative model [21] to enhance the reconstruction of air pollution fields. The diffusion models generate multiple solutions guided by sparse observations. This stochastic formulation enables reconstruction of finer details, particularly in the regions of high spatial variability of pollution fields. We compare this approach with various deterministic models. To our knowledge, this study is among the first to reconstruct high resolution spatial air pollution fields using a diffusion- or flow-based generative AI model [22]. Previous studies have utilised diffusion models to perform reconstruction of pollution in the temporal domain [23] (interpreted as signal recovery). Others have performed reconstruction in the spatial domain [24] without training on full-space data. These methods perform a probabilistic inference at unknown sensor locations rather than supervised learning to predict the true field.

In this paper, the evaluation of the reconstruction is conducted on an independent dataset of observed real-world sensor readings. To provide a fair assessment and evaluate the model’s generalisability beyond the training seasons, the models are trained on simulation data for 10 months, from January to October 2014, and tested on out-of-distribution real observational data from November to December 2014. Since the full-field spatial data is not available for the observation dataset, we introduce point-wise cross-validation and spectral plots to evaluate the model’s performance. Our results demonstrate that diffusion models generalise well to real-world data. Combined with their ability to generate reconstructions in real-time after training, these findings highlight the potential of diffusion models for operational urban air quality monitoring and mapping.

2 Results

2.1 Performance of deterministic models on simulation data

The ML models are trained and evaluated using full-field simulation data generated by Polyphemus[25], a chemistry-transport model. The deterministic deep learning models include VUNet [26, 27], ViTAE [28], and CLSTM [29, 30], each selected for its different

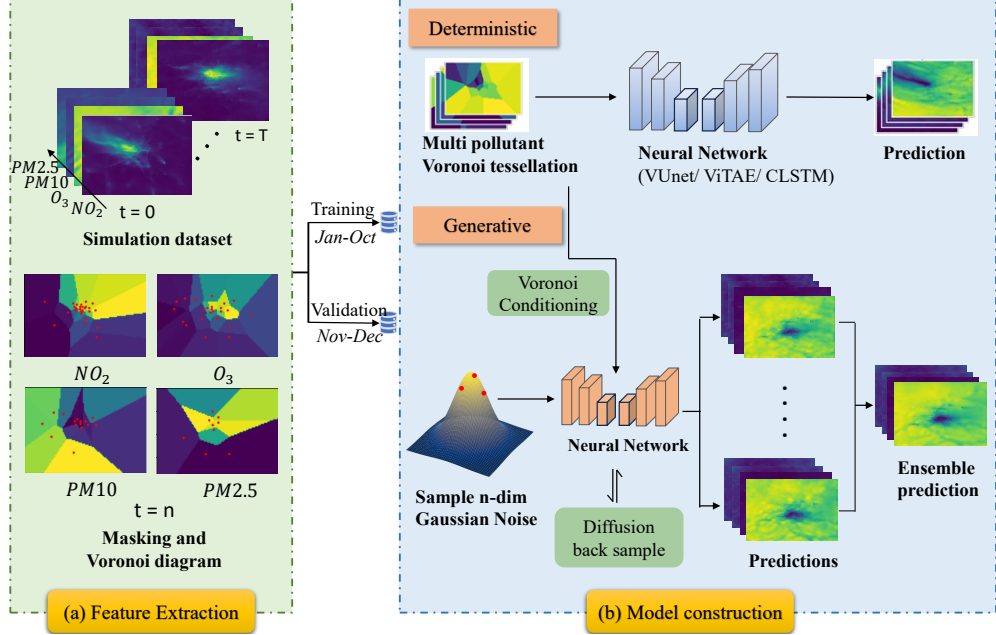


Fig. 1: Illustration of the training procedure of various models using simulation data. Panel (a) illustrates the feature extraction step which creates masks and generates Voronoi tessellations for simulation data at each time step, Panel (b) illustrates the differences in output generation by deterministic and generative models.

strengths in processing complex spatio-temporal datasets (more details in Appendix section G). The results of the Kriging interpolation method [31], widely used in operational scenarios, are provided as a baseline. Details of model construction and data processing are provided in Section 4 (Methods). The previous k hourly observations are used to reconstruct the field at the current time step. This experiment (Table 1) investigates the performance of models when considering time windows of different lengths, $k = \{1, 2, 3, 4, 6, 8, 12\}$.

As the Kriging approach has been mainly designed for static data, it is evaluated on a single time step. As detailed in Table 1, Kriging is significantly outperformed by ML-based models across all metrics. All ML models consistently achieve SSIM values greater than 0.8. This indicates the ML model’s capability to capture key spatial patterns and preserve the visual structures formed by pollution fields. However, this metric is less sensitive to distinguish among various ML models.

Each model has a different optimal window length k . From Table 1, CLSTM achieves the best performance at $k = 8$, whereas VUnet and ViTAE achieve at $k = 4$ and $k = 6$. The variation in performance can be explained by the differing architecture of each model. CLSTM is specifically designed for sequential data. This architecture enables output quality to improve steadily with increasing k . In contrast, the VUnet model processes temporal data as channel-wise concatenation of input timesteps and lacks an explicit mechanism to model sequential dependencies. The ViTAE model,

Model	Metric	Number of input time steps						
		1	2	3	4	6	8	12
CLSTM	MRE ▼	0.130	0.120	0.118	0.117	0.115	0.112	0.114
	SSIM Δ	0.805	0.820	0.823	0.829	0.832	0.838	0.838
	MFE ▼	0.182	0.173	0.172	0.171	0.166	0.160	0.158
	MFB ≈ 0	0.012	0.030	0.029	0.029	0.011	-0.009	-0.004
VUNet	MRE ▼	0.119	0.118	0.116	0.109	0.118	0.120	0.121
	SSIM Δ	0.847	0.848	0.852	0.859	0.846	0.841	0.841
	MFE ▼	0.178	0.181	0.172	0.202	0.176	0.185	0.186
	MFB ≈ 0	-0.009	0.005	-0.004	0.044	-0.009	0.035	0.003
ViTAE	MRE ▼	0.142	0.141	0.138	0.132	0.129	0.133	0.136
	SSIM Δ	0.815	0.843	0.828	0.829	0.840	0.839	0.837
	MFE ▼	0.195	0.195	0.198	0.184	0.182	0.185	0.182
	MFB ≈ 0	0.014	-0.033	-0.062	0.016	-0.019	-0.007	-0.013
Kriging	MRE ▼	0.422						
	SSIM Δ	0.773						
	MFE ▼	0.336						
	MFB ≈ 0	-0.113						

Table 1: Performance of deterministic models when evaluated on the validation hold-out of the synthetic dataset with differing lengths of time windows. The best value for each metric within each model across the tested time-window lengths is bolded. The Δ symbol next to a metric indicates that a higher value is considered better, ▼ indicates that a lower value is better, and ≈ 0 indicates that a value closer to 0 is better.

based on a Vision Transformer architecture, contains a large number of neural network parameters (see Table G4 in the Appendix section). The increased model complexity may make optimisation more challenging under the limited training data available (10 months of hourly snapshots) and cause higher validation errors.

2.2 Generative model and the ensemble effect

Details of the diffusion model are provided in the Methods section 4.3.2. The stochasticity inherent in diffusion models enables them to output diverse samples. By aggregating multiple predictions, the ensemble methods improve the quality of the output. During output generation, the ensemble size E measures the number of samples to be averaged and the inference/sampling steps R measures the number of integration steps used to generate each sample prediction. The numerical experiments evaluate the effect of E and R on the performance of the diffusion model.

Figure 2 plots the evaluation metrics for the average of all pollutants. It shows that the prediction quality improves with increasing ensemble size and inference steps. Even with fewer sampling steps, $R \approx 10$, the model demonstrates strong performance,

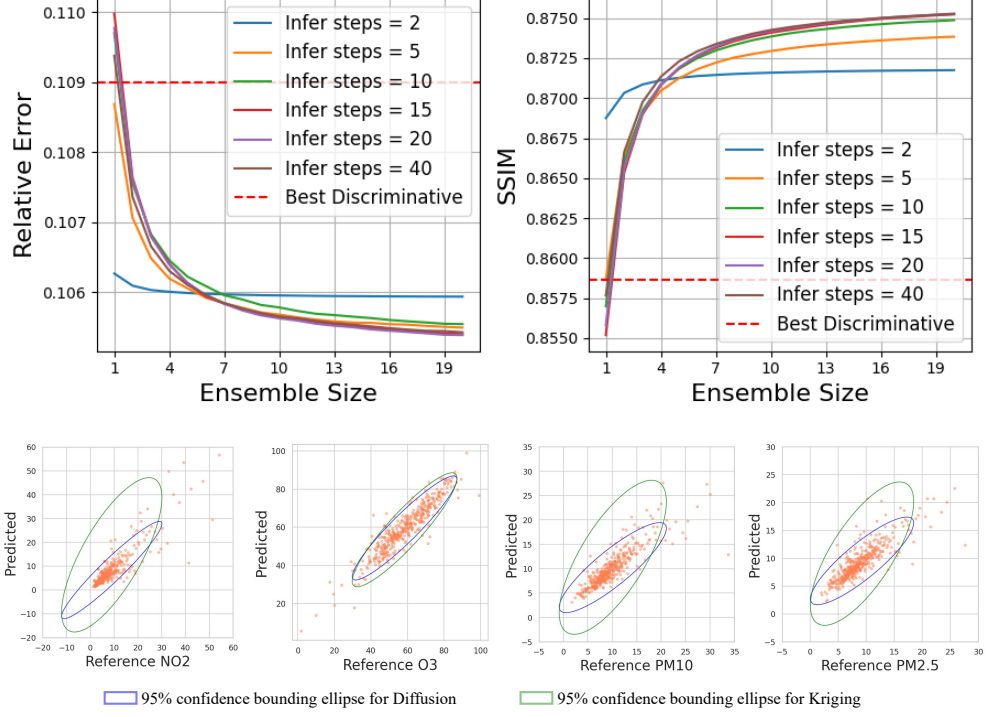


Fig. 2: Effect of the parameters Ensemble Size (E) and Inference steps (R) on the diffusion model in MRE and SSIM metrics averaged over the four pollutants. The dashed line in each plot represents the best-performing deterministic model from Table 1. The lower graph plots a 95% confidence ellipse bounding the points for ground truth vs prediction. The blue ellipse is for the diffusion model ($E = 20$) and the green ellipse represents the Kriging model.

highlighting the efficiency of DPM Solver++ [32] for sampling. Compared to the best-performing deterministic model from Table 1, ensembling allows the diffusion model to achieve higher performance. It helps reduce stochastic variation in outputs, but after the variance is sufficiently reduced and the model bias remains, there is no further measurable improvement. This is observed in Figure 2, as the performance gain plateaus at $E \approx 20$. To reduce the computational inference time without compromising the quality of the result, we choose $E = 20$ and $R = 10$ for the diffusion model.

Additionally, Figure 2 plots the reference simulation against the predicted values for all grid points across samples and overlays a 95% confidence bounding ellipse for the diffusion model ($E = 20$, $R = 10$) and the Kriging baseline. For NO₂, PM₁₀ and PM_{2.5}, the diffusion model has a tighter and more compact ellipse, indicating a reduced dispersion of prediction error and a significant improvement from the Kriging model.

Diffusion models show better performance as they progressively denoise the data, allowing a coarse-to-fine generation process [33]. To quantify this behaviour, we conduct spectral analysis on intermediate outputs during the reverse sampling process (Figure 3). Using a 50-step denoising process ($R = 50$), we observe that the generated fields progressively match the reference simulation, with low-frequency components (corresponding to coarse structure) recovered first, followed by higher-frequency values (corresponding to fine details). This result is a strong indicator of the advantage that diffusion models provide. Rather than purely increasing the model size or capacity, diffusion models implicitly perform a more complex process of aligning the spectral structure.

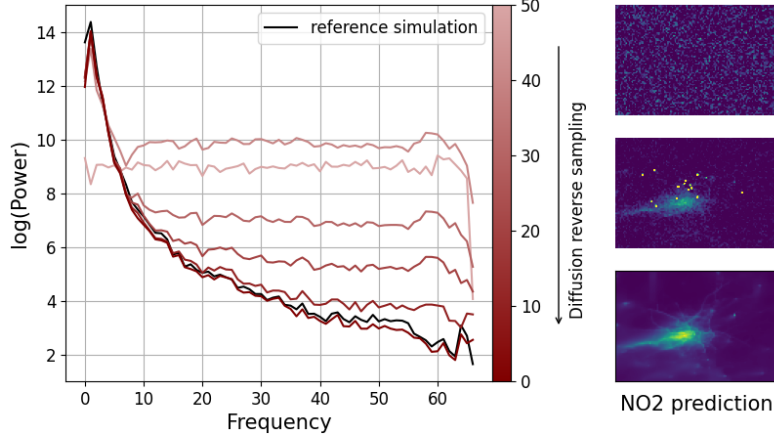


Fig. 3: From pure Gaussian noise to generated output, progressive denoising mechanism in the diffusion model of a single NO₂ sample from the simulation dataset. In a 50-step denoising ($R = 50$), the power spectrum plot shows the logarithm of the power values of intermediate steps until the final output and compare it to the reference simulation.

Appendix Figure C5 compares the power spectral plots for deterministic and generative outputs for each pollutant in the validation set of the simulation data. Since NO₂ and O₃ have shorter lifetimes and greater spatial variability, they exhibit higher power at high frequencies compared to PM₁₀ and PM_{2.5}. Importantly, the diffusion model is better at capturing the high-frequency structure of NO₂ and O₃, while the deterministic models perform better for particulate matter. This behaviour contrasts with point-wise metrics (Appendix Table C2), where diffusion achieves superior performance for PM_{2.5}. The discrepancy arises because particulate matter fields are smoother and dominated by low-frequency components, reducing the advantage of diffusion models in reconstructing fine-scale details.

2.3 Validation on a real-world dataset

To validate the performance of trained models under realistic conditions, we investigate the model’s robustness using real-world observation data obtained from the Central Air Quality Monitoring Laboratory (LCSQA) through the Geod’air (Management of Air Quality Observation Data) platform [34]. From the experiments presented in Section 2.1 and Section 2.2, we select the best-performing hyperparameters for each model and evaluate their performance on the observation dataset using multiple metrics.

As detailed in section 4.2, we apply augmentations to the simulation dataset to reduce the distribution mismatch with real-world observations, allowing the synthetic data to mimic the characteristics of the real-world dataset. We compare the performance on real-world data across different models with various augmentation methods such as Gaussian, Perlin, Correlated, and Time-aware Gaussian noise (details in Appendix section F). Since the full-field values for real data are unknown, we determine the accuracy of the models using a specially defined Mean Relative Error (Section 4.2 and Appendix section E).

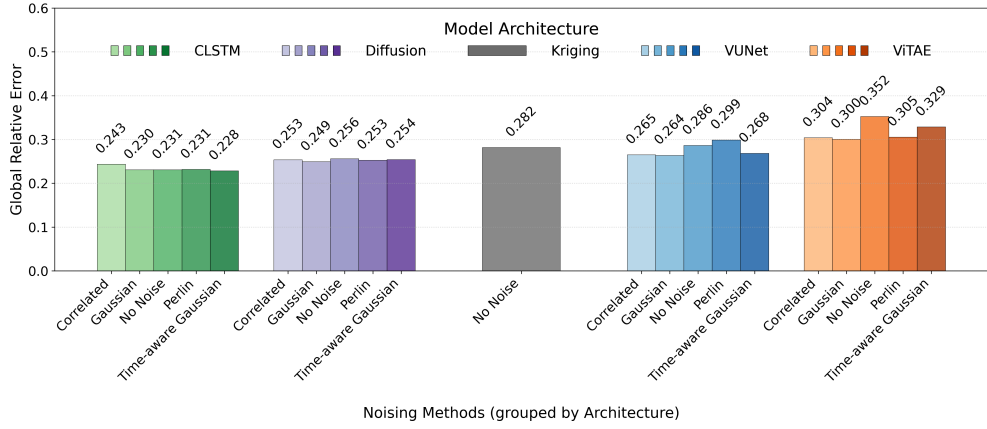


Fig. 4: Comparison of various models with different noising augmentations of (a) Gaussian, (b) Perlin, (c) Correlated, (d) Time-aware Gaussian, and (e) No noise. From Table 1, we select the best-performing time step for each model. The Mean Relative Error metric is used on the real observation dataset.

Figure 4 shows that ViTAE, Kriging, and VUNet achieve MREs of 0.300, 0.282, and 0.264, respectively, while the Diffusion model has an MRE of 0.249, and the lowest error is achieved by the CLSTM model with an MRE of 0.228. The introduction of data augmentation during training consistently reduces MRE for all models, indicating improved robustness to noise. The largest improvements are observed for ViTAE (14.77%) and VUNet (7.69%), suggesting that these models strongly benefit from exposure to more diverse training data. By contrast, the smaller gains for diffusion (2.73%) and CLSTM (1.29%) indicate that their internal mechanisms of stochastic

sampling and temporal modelling, respectively, enable robust performance on noisy out-of-distribution data and thus reduce the reliance on explicit augmentations.

While MRE provides a point-wise accuracy measure, we also compare the structure of predicted pollution fields using the power spectrum plot to analyse their frequency content (Metric details in Appendix section E). Figure 5a shows the log mean power, averaged across samples and pollutants as a function of frequency. We plot the reference simulation data (considered as the ground truth when using synthetic observations) against the outputs of the generative and deterministic models evaluated on synthetic and real observations.

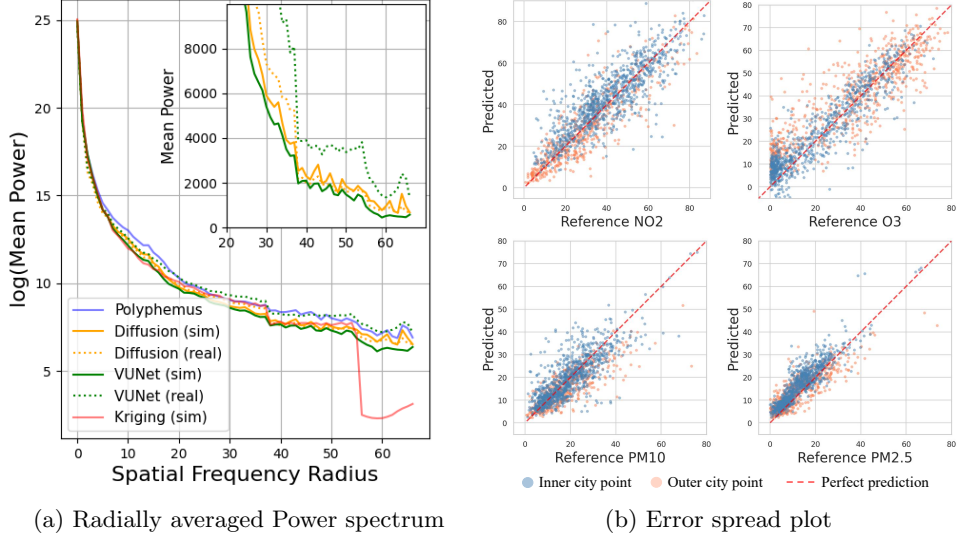


Fig. 5: Panel (a) compares the log of power values of reference simulation data and predicted results from the diffusion model ($E = 20$), VUNet and the Kriging model on real and simulated data. The values are averaged over all four pollutants. Panel (b) is plotted for the real-world data at the held-out evaluation points. Orange points indicate sensor locations in the outer parts of the city and blue points indicate locations in the inner part of the city.

On the simulation data (solid colour lines) the diffusion model closely matches the reference spectrum (Polyphemus model) across frequencies. This implies that both large-scale structure and fine-scale details are preserved by the diffusion model. By contrast, the Kriging model decays in power values, reflecting oversmoothing of output. VUNet (deterministic) relies on a compressed latent representation, which introduces an information bottleneck and leads to a loss of finer details, as reflected by reduced high-frequency components.

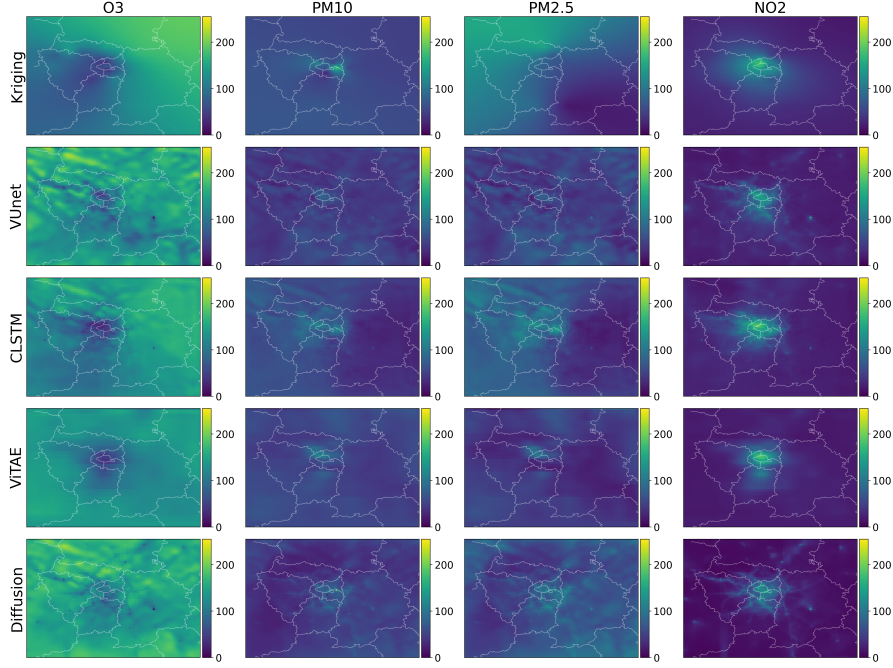


Fig. 6: Visual comparison of (a) Kriging (b) VUNet (with Gaussian noise) (c) CLSTM (with Time-aware Gaussian noise) (d) ViTAE (with Gaussian noise) (e) Ensemble Diffusion (with Gaussian noise) models on O_3 , PM_{10} , $PM_{2.5}$, NO_2 pollutants on December 01 00:00:00 2014

On real-world data (dashed lines), the deterministic model deviates from its simulation spectrum and shows increased high-frequency power, suggesting sensitivity to input noise and hallucination of fine textures. By contrast, the diffusion model remains closely aligned with the simulation spectrum, highlighting its greater robustness to noisy observations and its ability to preserve realistic spatial structure.

Additionally, Figure 5b plots the held-out sensor observations against the predicted values of the diffusion model for the real dataset on the held-out grid points. As described in Section 4.2, the sensor coverage is denser in inner-city regions than in outer-city locations. Despite this, a similar spread of inner- and outer-city points highlights the superior generalisation ability of the ML models under sparse observational coverage.

Alongside the quantitative results, we assess the quality of reconstruction through visual comparisons (Figure 6). From Figure 4, we select the best-performing augmentation setting for each model and visualise the results in Figure 6. Kriging and ViTAE produce distorted fields, whereas VUNet, CLSTM, and Diffusion models produce outputs with clear structural patterns that closely resemble air pollution fields. The low performance of ViTAE is likely due to overfitting, as its larger model capacity limits the generalisation to noisy real-world data, while the other ML models demonstrate robustness to noise.

3 Discussion

This study demonstrates the effectiveness of machine learning models for the reconstruction of pollution fields. Despite being trained for a relatively short period of 10 months, the ML-based models outperform the Kriging baseline and produce realistic reconstructions. The maximum number of active sensors (when no station is under maintenance) for each pollutant is 28, 24, 15, 9, respectively, for NO_2 , O_3 , PM_{10} , and $\text{PM}_{2.5}$, which corresponds to a sparsity of 99.661%, 99.714%, 99.819%, and 99.891% against the number of grid points in the field. Our method highlights the ability of ML models to generate full fields while observing highly sparse data.

Integration of previous time steps into deterministic models (Table 1) reveals the influence of historical data, as accuracy improves significantly. In general, increasing the size of the time window leads to improved performance with diminishing returns after a certain length. The optimal window length is likely influenced by the variability of weather patterns in the reconstructed region. Under highly volatile weather conditions, long temporal windows may encompass multiple distinct weather events, some of which may be unrelated to the target period. Although deterministic models perform better than Kriging, they are limited by their formulation, as they are trained to predict a single outcome. This one-to-one mapping restricts their ability to be deployed in complex physical environments where uncertainty quantification and robustness to variability are required.

We construct one of the first spatially-aware generative AI models for the reconstruction of air pollution. The diffusion model provides two unique advantages over deterministic models. First, it models the distribution of the pollution fields rather than predicting a single output. This enables uncertainty exploration by allowing us to sample multiple outputs from the predicted distribution. Due to an increase in *extreme pollution* events, these fields often exhibit anomalies and highly dynamic spatio-temporal behaviour [35]. Diffusion models help capture the uncertainty that is usually observed in atmospheric modelling. Second, diffusion models are able to preserve realistic spatial structure as they implicitly follow a frequency-structured generation process. For pollutants with shorter atmospheric lifetimes (NO_2 , O_3), we observe pollution fields that are sharper with high variability. The gradual denoising process of the diffusion model facilitates the recovery of fine-scale details (Figure 3), consistent with our analysis in the spectral domain. More broadly, the results suggest that sparse-monitor reconstruction should not be viewed only as an interpolation problem, but as a probabilistic inference problem constrained by atmospheric structure.

This study provides findings that advance the application of ML techniques in environmental science and support real-world deployment solutions. First, ML algorithms show a remarkable ability to leverage inter-pollutant relationships under extremely sparse monitoring conditions to improve output quality. Despite the relatively low monitoring coverage for PM_{10} and $\text{PM}_{2.5}$, we achieve high reconstruction quality. Second, the transition to real-world observations is an important aspect for assessing the model’s generalisation capability. By incorporating noise-based augmentations during training, these models can be deployed in real-world environments without requiring retraining. This represents a significant result, offering a cost-effective and scalable

solution for reconstruction tasks while enabling the extension of this research to other weather and climate modelling applications. In particular, the construction of precise air quality concentration fields is of significant importance for analysing the spatial relationships between healthcare outcomes and air quality data.

The main limitations of the current study are its dependence on training with simulation data and its limited generalisability to different cities. The model has to be retrained to adapt to a particular region. The limited availability of simulation data constrains the use of high-capacity ML models. To address this limitation, future research could incorporate auxiliary data sources—such as topographic information, emission inventories, climate variables, and satellite observations—as model inputs or conditioning variables for generative models. By leveraging the ability of ML models to process multimodal data, the interdependencies among these factors could provide a more detailed analysis. The effect of previous time step data on the reconstruction should be studied in more detail, especially for generative models.

4 Methods

4.1 Air-quality simulations

The air quality of the Paris metropolitan region is simulated using the three-dimensional Eulerian chemistry–transport model Polyphemus/Polair3D [36] with the configuration described in Lugon et al. [25]. The model accounts for the main atmospheric processes, including emissions from anthropogenic and biogenic sources, transport (advection and diffusion), chemical transformations, and removal by dry and wet deposition. The formation of secondary gases and aerosols is represented through coupling with the SSH-aerosol module [37]. As detailed in Lugon et al. [25], the simulations are performed over a one-year period in 2014 on a $2 \text{ km} \times 2 \text{ km}$ horizontal grid with 14 vertical levels. The initial and boundary conditions are prescribed from larger-scale Polair3D simulations over Europe and metropolitan France, following the setup described in Andre et al. [38].

4.2 Data processing and masking techniques

This study uses two different datasets for training and evaluating the models. The *simulation* dataset of all four pollutants (NO_2 , O_3 , $\text{PM}_{2.5}$, PM_{10}) is obtained as described in Section 4.1. The ML models are trained on this data from January to October and validated on an independent period from November to December. To evaluate real-world performance, we use the *observation* dataset obtained from the Central Air Quality Monitoring Laboratory (LCSQA) through the Geod’air (Management of Air Quality Observation Data) platform. This dataset contains hourly recordings of the concentrations of the four regulated pollutants mentioned above from the Paris metro area from January 1 to December 30, 2014.

For all pollutants, the concentration field is discretised on a grid with a resolution of $n_x = 75$ and $n_y = 110$ pixels. We define the binary mask $\mathbf{\Omega}_t \in \{0, 1\}^{4 \times n_x \times n_y}$, where 1 indicates the presence of a monitoring sensor at time t . Let $\mathbf{x}_t \in \mathbb{R}^{4 \times n_x \times n_y}$ denote the true concentration fields at time t , which are inaccessible in real-world

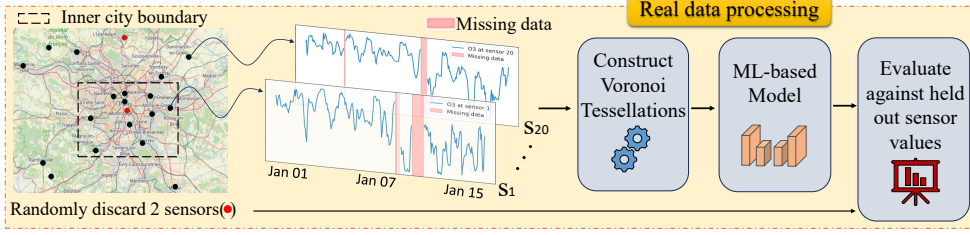


Fig. 7: Flowchart indicating the processing of real-world data. (a) Randomly withhold one active sensor inside and one active sensor outside the city boundary, (b) Record observation values from station number $S_1 \dots S_{20}$, (c) Construct Voronoi tessellations for each time step, (d) Reconstruct pollution fields with the trained ML model, (e) Evaluate model performance against held-out sensor points

applications. We define the tessellated fields $\mathbf{z}_t = f^v(\boldsymbol{\Omega}_t, \mathbf{x}_t) \in \mathbb{R}^{4 \times n_x \times n_y}$ where f^v is the two-dimensional tessellation function. We concatenate the four pollutants to facilitate multi-pollutant analysis (as illustrated in figure 1).

We develop a cross-validation masking technique that randomly selects active sensor locations, as shown in Figure 7. At each time step, one sensor from the inner-city and one from the outer-city regions are classified as evaluation locations: $\boldsymbol{\Omega}_t^{\text{eval}}$. While training, these points are removed from the model’s input mask: $\boldsymbol{\Omega}_t^{\text{train}} = \boldsymbol{\Omega}_t \odot (1 - \boldsymbol{\Omega}_t^{\text{eval}})$. The boundary of the inner-city is defined by the area of the administrative city of Paris. This distinction helps to evaluate the model’s performance in both densely and sparsely monitored regions. The variation in masking encourages the models to learn different configurations that could be used to obtain richer reconstructions for a sparsely observed field.

We apply data augmentation to simulation data during training to diversify the model’s input and reduce the distribution gap between observed and simulated data. Figure 8 panels (a) to (d) compares the distribution of simulation and real data before augmentations and (e) to (h) highlights the shift after augmentations. Experiments are performed using various perturbation techniques such as Gaussian, Time-aware Gaussian, Perlin, and Cross-correlation noise (details can be found in Appendix section F). Each method is parametrised by a mean and standard deviation that determine the magnitude of the applied perturbation. To better align the pollutant-specific distributions, the mean is determined by $\mu_{\text{noise}} = \mu_{\text{obs}} - \mu_{\text{sim}}$, where $\mu_{\text{obs}}, \mu_{\text{sim}}$ denote the average observation and simulation values, respectively, computed across all sensors and time steps used in the training. The standard deviation controls the strength of the perturbation and is selected through a randomised search on the VUNet model. More details of these methods are provided in Appendix section F.

4.3 Model frameworks

In this section, we discuss in detail the ML models employed in this study, namely VUNet, ViTAE, CLSTM, and the diffusion model.

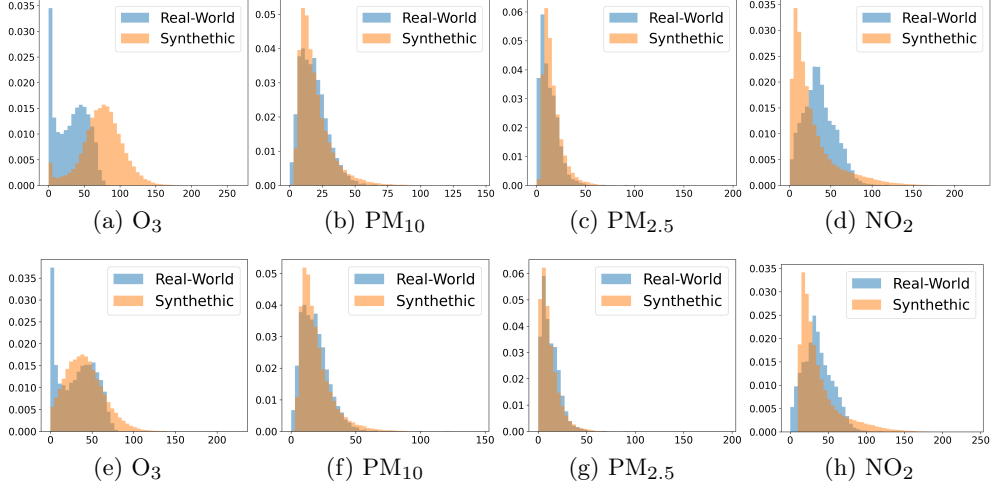


Fig. 8: Comparison of the distribution of the masked field for each pollutant between real and synthetic data. Figures (a) to (d) are before augmentations and (e) to (h) are the transformed distributions after augmentation. The blue histogram indicates real-world data, and the orange histogram indicates synthetic data.

4.3.1 Deterministic models

VUNet (short for Voronoi UNet) is inspired by Fukami et al. [26]. We adopt the UNet architecture, consisting of 4 downsampling and upsampling steps along with self-attention injected into the bottom layer. The input is a concatenation ($\oplus$ operator) of Voronoi tessellation and its corresponding masks ($\mathbf{z}_t$ and $\mathbf{\Omega}_t$). To observe the effect of historical data, previous time step inputs are concatenated too. The concatenation is performed by stacking along the channel dimension:

$$\mathbf{y}_t = f^{\text{UNET}}((\mathbf{z}_{t-k} \oplus (\mathbf{x}_{t-k} \odot \mathbf{\Omega}_{t-k})) \oplus \cdots \oplus (\mathbf{z}_t \oplus (\mathbf{x}_t \odot \mathbf{\Omega}_t))), \quad (1)$$

where $\odot$ is the Schur product and $\mathbf{y}_t$ denotes the output of the neural network which is an estimation of $\mathbf{x}_t$. This is a simple model design that compresses the data into a latent space, which encourages the model to learn the most important features.

Extending the formulation of Fan et al. [28], we modify **ViTAE** by replacing its original decoder (CNN) with a Unet model. The encoder is a Vision Transformer that divides the input image into patches, flattens them, and linearly projects them into a fixed-dimensional embedding space. The resulting embeddings are then rearranged into a multi-channel 2D format and passed through a Unet decoder. In the present work, the ViTAE model receives only sparse observations as input, similar to Fan et al. [28],

$$\mathbf{y}_t = f^{\text{ViTAE}}((\mathbf{x}_{t-k} \odot \mathbf{\Omega}_{t-k}) \oplus \cdots \oplus (\mathbf{x}_t \odot \mathbf{\Omega}_t)). \quad (2)$$

Unlike VUNet, which operates only on local neighbourhood regions, ViTAE is capable of establishing long-distance interactions and finding meaningful patterns.

Originally proposed by Shi et al. [30] for precipitation nowcasting, the Convolutional LSTM extends the traditional LSTM [39] by integrating convolutional operations into its temporal architecture. This hybrid design allows the model to capture the spatio-temporal dependencies inherent in air quality data. This study extends this implementation by building on the improved Peephole LSTM architecture [29]. This model receives the previous k time steps' Voronoi tessellations as input to estimate the concentration fields,

$$\mathbf{y}_t = f^{\text{CLSTM}}(\mathbf{z}_{t-k}, \dots, \mathbf{z}_t). \quad (3)$$

All deterministic models are trained using the mean squared error (MSE) between the model output $\mathbf{y}_t$ and the reference simulations $\mathbf{x}_t$,

$$L_{\text{MSE}} = \frac{1}{4 \times n_x \times n_y \times N_{\text{train}}} \sum_{t=1}^{N_{\text{train}}} \|\mathbf{y}_t - \mathbf{x}_t\|^2, \quad (4)$$

where N_{train} denotes the number of training samples, corresponding to all hourly data from January through October 2014.

4.3.2 Generative model

Conditional diffusion models learn the distribution of the dataset rather than learning a one-to-one mapping. The output is sampled from a learned distribution, which allows the model to generate different outputs for the same inputs. These models are conditioned on sparse observations for guided and controllable generation. We work with a class of diffusion models based on the *score-based stochastic differential equation* [21] and use the Elucidating Diffusion Model (EDM) framework [40] for its implementation. Instead of directly optimising the MSE loss as in deterministic models, diffusion models aim to generate realistic samples that are consistent with the input conditions by progressively denoising (through back sampling) randomly generated Gaussian noise. For back sampling, which is a technique that solves reverse SDEs, we use DPM Solver++ [32] for faster and high-quality sample generation.

The diffusion models operate in a pseudo time space defined by $r \in [0, R]$ such that $\{\mathbf{x}_r^t\}_{r=0}^R$ represents the noise added to the original data at time r with $\mathbf{x}_0^t = \mathbf{x}^t$ denoting the original data and $\mathbf{x}_R^t$ as pure Gaussian noise. The denoiser model D_θ is trained to predict the *score function* (derivative of log probability), which is used in the sampling process (details in Appendix section D).

The denoiser (D_θ) is a UNet model conditioned on sparse observations. These are represented as Voronoi tessellations and projected into latent space using an Encoder. These embeddings are injected into UNet layers through cross-attention. The Voronoi embedding is defined as

$$\mathbf{z}_{\text{emb}}^t = f^{\text{EMB}}(\mathbf{z}^t), \quad (5)$$

where f^{EMB} is a Transformer encoder model.

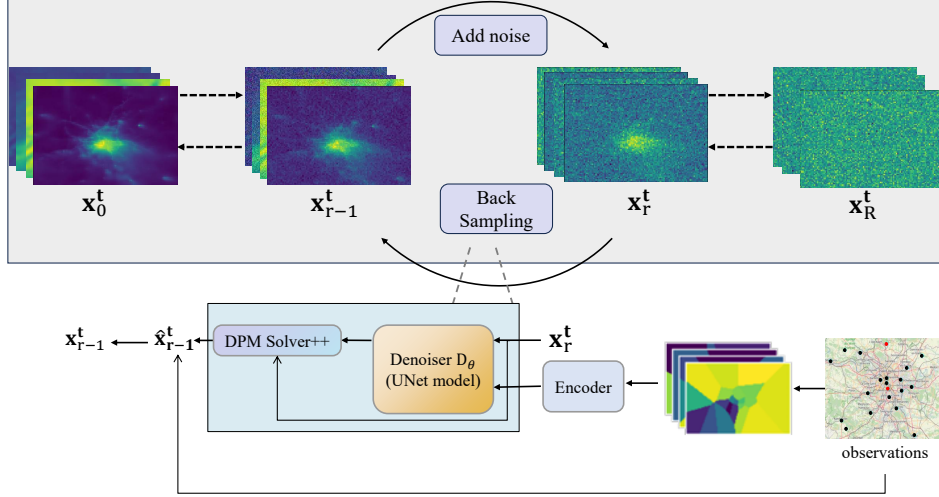


Fig. 9: Diffusion process of the NO₂ pollutant showing the forward and reverse processes.

As shown in Figure 9, we start from Gaussian noise at diffusion step R and iteratively denoise using DPM-Solver++ until the final reconstruction is obtained,

$$\hat{\mathbf{x}}_{r-1}^t = \text{DPMSolver}(D_\theta(\mathbf{x}_r^t, \mathbf{z}_{\text{emb}}^t; r)) \quad \text{with} \quad \hat{\mathbf{x}}_R^t = \mathbf{x}_R^t, \quad \mathbf{y}^t = \hat{\mathbf{x}}_0^t. \quad (6)$$

Although conditioning the denoiser provides consistency with the sparse observations, the generated samples may deviate from the observed values during iterative denoising. To prevent this drift, we employ masked back-sampling, which explicitly enforces the observed measurements at every reverse diffusion step. Rather than inserting the clean observations directly into the noisy samples, the observations are first re-noised to match the current noise level,

$$\tilde{\mathbf{x}}_{r-1}^t = \mathbf{x}_0^t + \sigma_{r-1}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (7)$$

and are then reinserted into the generated sample at the observed location

$$\mathbf{x}_{r-1}^t = \hat{\mathbf{x}}_{r-1}^t \odot (1 - \Omega_t) + \tilde{\mathbf{x}}_{r-1}^t \odot \Omega_t \quad (8)$$

This update not only ensures that the reconstructed field matches the observations at sensor locations, but also guides the generation of neighbouring regions, as the corrected sample is subsequently passed through the denoiser in the next reverse diffusion step.

For multiple random noise initialisations $\{\mathbf{x}_R^{t,(e)}\}$ with $e = 1, \dots, E$, an ensemble of output samples is produced $\{\mathbf{y}^{t,(e)}\}$ and is aggregated to produce the average reconstruction,

$$\mathbf{y}_{en}^t = \frac{1}{E} \sum_{e=1}^E \mathbf{y}^{t,(e)}. \quad (9)$$

Ensemble predictions can also be used to estimate predictive uncertainty, which is particularly valuable for data assimilation and risk assessment, as shown in recent studies [41, 42].

Data availability

The simulation data generated and analysed during the current study are available at https://huggingface.co/datasets/potatotopatoo/Air_pollution_simulation. The real-world observation data were obtained from the Geod’air platform operated by LCSQA. These data are open-access at <https://www.geodair.fr/donnees/consultation> and are subject to the terms of use of the data provider.

Code availability

The custom code used to train and evaluate the models and reproduce the results is available at <https://github.com/Miha5092/Air-Quality-Field-Reconstruction>.

Acknowledgments

A.Sabnis, M.Bocquet and S.Cheng acknowledge the support of the French Agence Nationale de la Recherche (ANR) under grant reference ANR-25-CE56-0198-01. This work also benefited from State funding managed by ANR under the France 2030 program, reference ANR-23-IACL-0005. This work has received fundings from the PEPR Villes Durables (French National Research Agency, France, URBHEALTH project ANR-24-PVD0007). CEREa is a member of Institut Pierre-Simon Laplace (IPSL).

Author contributions

AS and MM performed the formal data analysis and implemented the machine learning models. LL and KS carried out the physics-based model simulations to generate the training data. AS, MM, and SC prepared the manuscript with contributions from all co-authors. MB and SC provided the financial support for the project that led to this publication. SC coordinated the research activities. XC and ZS provided technical support. All co-authors reviewed and edited the manuscript.

Competing Interests

The authors declare no financial or non-financial competing interests.

Appendix A Notation Table

Appendix B Simulation data visualisations

Figures B1, B2, B3, B4 compare the predicted output of Kriging, VUNet, CLSTM and the diffusion model against the reference simulation data point from 1st December at 8 AM.

Table A1: Notation Table

Notation	Description
$\mathbf{s}^d$	Set of n distinct locations of sensor points for pollutant d
$\mathbf{z}_t$	Voronoi tessellation at time t
Ω_t	Binary mask of sensor location
Ω_t^{train}	Binary mask used for training model
Ω_t^{eval}	Binary mask used for evaluating model
$\mathbf{x}_t$	Ground truth of pollutants
$\mathbf{y}_t$	Predicted reconstruction from models
R	Number of back sampling steps in diffusion model
E	Ensemble size in diffusion model
$\mathbf{x}_r^t$	Noise added to ground truth image at diffusion time r and pollutant time t
$\mathbf{y}_{en}^t$	Ensemble prediction of diffusion model

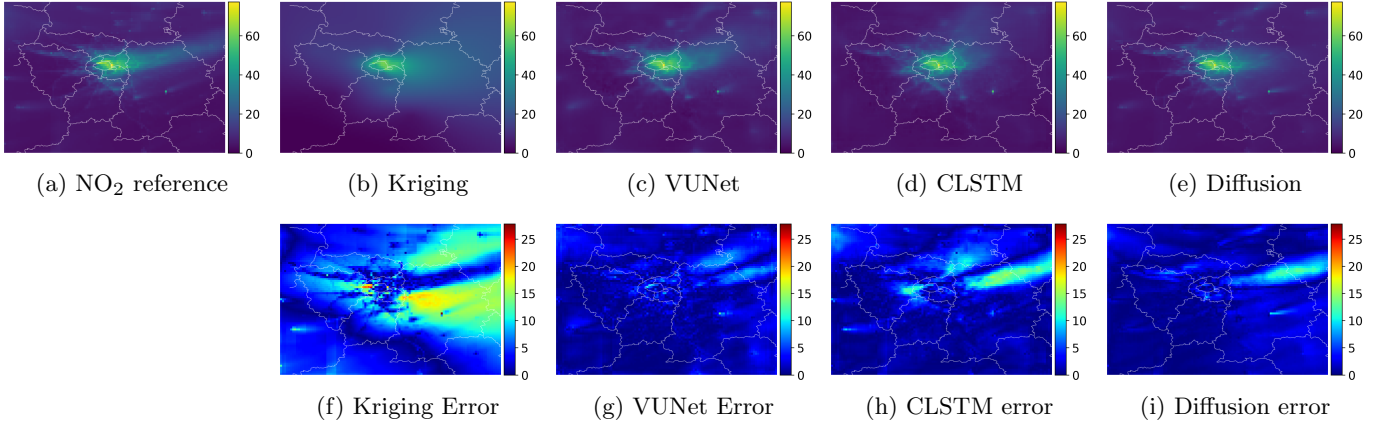


Fig. B1: Side-by-side comparison of the NO_2 predictions at 8 AM on the 1st of December for Kriging, VUNet, CLSTM and the diffusion model

Appendix C Detailed results of each pollutant

Table C2 displays the metrics for each pollutant type on the simulation dataset for models CLSTM, VUNet, ViTAE, and Diffusion.

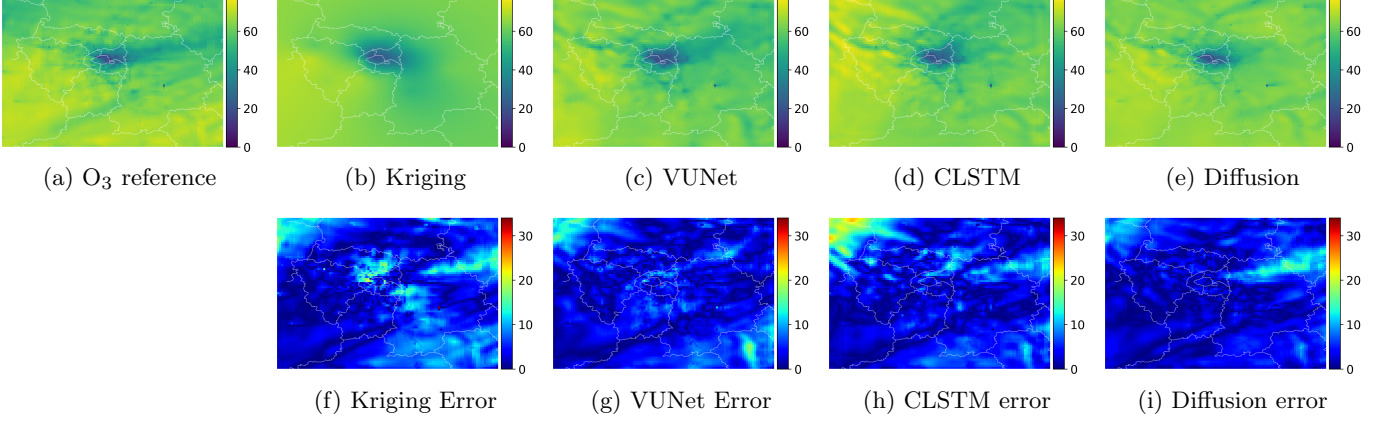


Fig. B2: Side-by-side comparison of the O_3 predictions at 8 AM on the 1st of December for Kriging, VUNet, CLSTM and the diffusion model

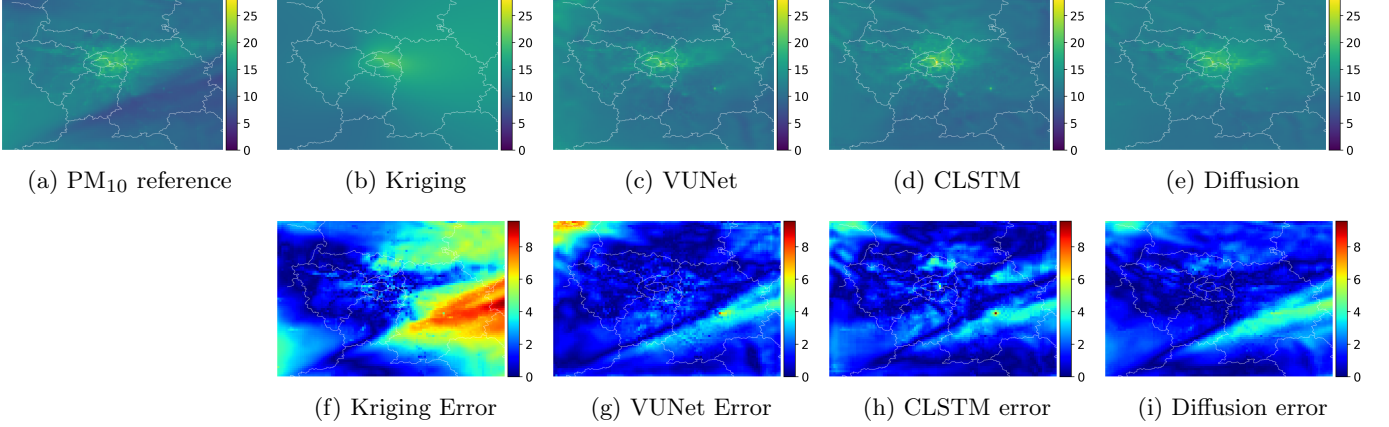


Fig. B3: Side-by-side comparison of the PM_{10} predictions at 8 AM on the 1st of December for Kriging, VUNet, CLSTM and Diffusion model

Appendix D Diffusion-based generative model

D.1 Mathematical Formulation

We work with a class of diffusion models called *score-based stochastic differential equations* [21], where the data is continuously perturbed over time. In this section, variable r represents the time parameter governing the noise addition process. The forward SDE adds noise to the original image as follows:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, r)dr + g(r)d\mathbf{w}, \quad (\text{D1})$$

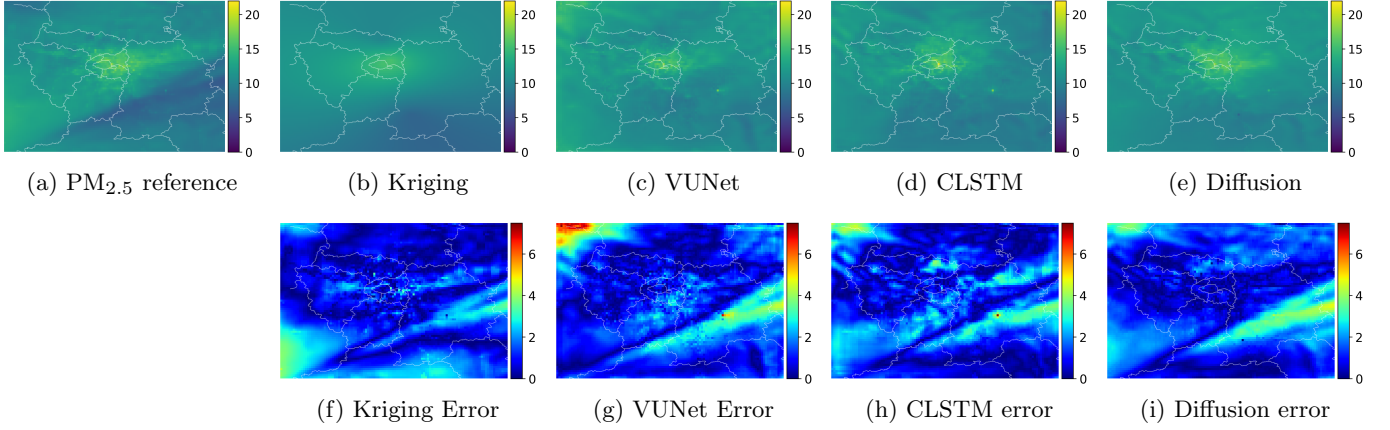


Fig. B4: Side-by-side comparison of the $\text{PM}_{2.5}$ predictions at 8 AM on the 1st of December for Kriging, VUNet, CLSTM and Diffusion model

Model	Metric	Pollutant type			
		NO_2	O_3	PM_{10}	$\text{PM}_{2.5}$
CLSTM (t=8)	MRE ▼	0.266	0.09	0.182	0.188
	SSIM △	0.812	0.551	0.614	0.565
	MFE ▼	0.269	0.08	0.143	0.148
	MFB ≈ 0	0.081	-0.003	-0.046	-0.069
VUNet (t=4)	MRE ▼	0.236	0.089	0.186	0.193
	SSIM △	0.840	0.622	0.659	0.619
	MFE ▼	0.268	0.232	0.152	0.157
	MFB ≈ 0	0.143	0.148	-0.044	-0.07
ViTAE (t=6)	MRE ▼	0.258	0.11	0.228	0.229
	SSIM △	0.837	0.566	0.629	0.586
	MFE ▼	0.256	0.099	0.185	0.188
	MFB ≈ 0	0.088	-0.029	-0.056	-0.079
Diffusion (E=20, R=10)	MRE ▼	0.213	0.091	0.183	0.182
	SSIM △	0.89	0.642	0.699	0.659
	MFE ▼	0.211	0.08	0.144	0.146
	MFB ≈ 0	0.081	-0.031	-0.037	-0.052

Table C2: Evaluation of reconstruction accuracy using MRE, SSIM, MFE, and MFB for each pollutant type. We select the best-performing hyperparameters for each model. The best value for each pollutant is bolded. The Δ symbol next to a metric indicates that a higher value is considered better, ▼ indicates that a lower value is better, and ≈ 0 indicates that a value closer to 0 is better.

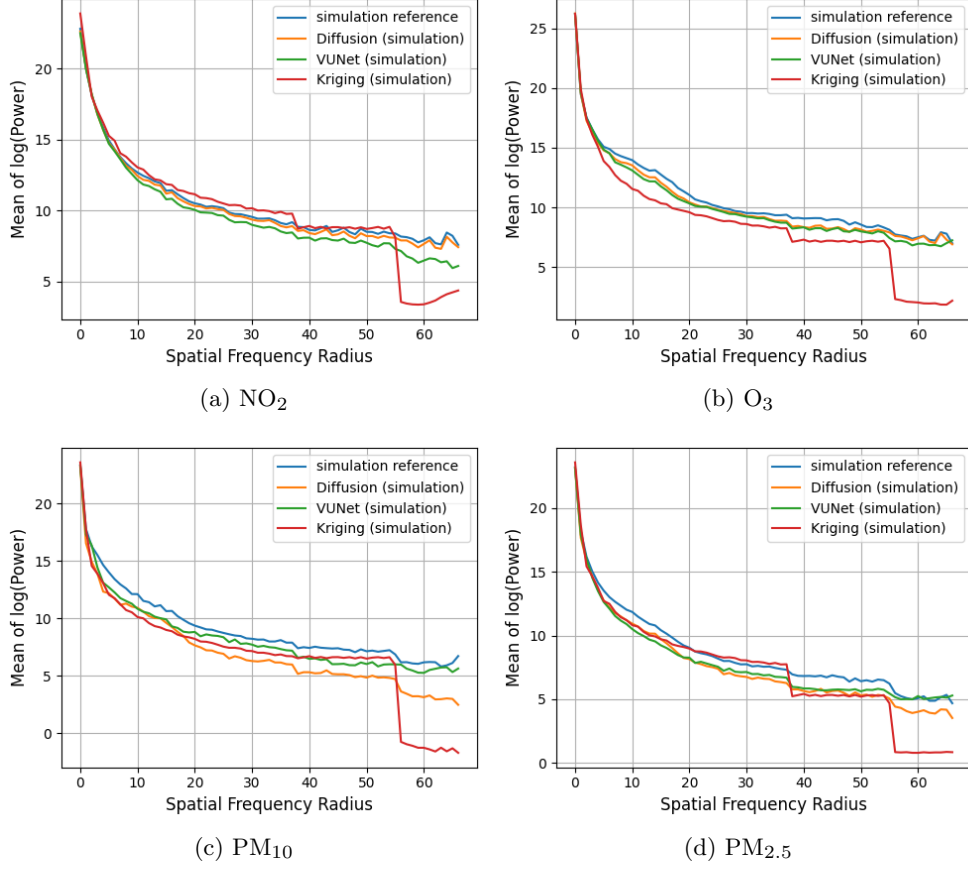


Fig. C5: Comparison of the logarithm of the radially averaged power spectrum for ground truth of simulation data and predicted results from Diffusion, VUNet and the Kriging model.

where $\mathbf{f}(\mathbf{x}, r)$ is the drift coefficient, $g(r)$ is the diffusion coefficient, and $\mathbf{w}$ is the standard Wiener process. When going back in time (denoising), the reverse-time process can also be shown to be a diffusion process given by the equation:

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, r) - g^2(r)\nabla_x \log p_r(\mathbf{x})]dr + g(r)d\mathbf{w}. \quad (\text{D2})$$

The term $\nabla_x \log p_r(\mathbf{x})$ is the *score function* [21], which is estimated through a trained neural network. The choice of coefficients $\mathbf{f}(\mathbf{x}, r)$ and $g(r)$ determines the noise addition process. The Elucidating Diffusion Model (EDM) [40] provides a framework with $\mathbf{f}(r) = 0$ and $g(r) = \sqrt{2\dot{\sigma}(r)\sigma(r)}$, where $\sigma(r)$ is the noise standard deviation added at time r .

Let $D_\theta(\mathbf{x}; \sigma)$ be the neural network parametrised by θ that returns the original image for a noisy image input $\mathbf{x}$. The training objective of the neural net is to minimize

the mean squared error between the model output and the original image:

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \mathbb{E}_{n \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} \|D_{\theta}(\mathbf{x} + \mathbf{n}; \sigma) - \mathbf{x}\|_2^2. \quad (\text{D3})$$

From the above equation, Karras [40] proved that

$$\nabla_x \log p(\mathbf{x}; \sigma) = \frac{D_{\theta}(\mathbf{x}; \sigma) - \mathbf{x}}{\sigma^2}. \quad (\text{D4})$$

Thus, minimising the loss in equation D3 provides a relation between the *score function* and the neural net denoiser D_{θ} . The reverse SDE in equation D2 is solved using DPM Solver++ [32]. This algorithm is more stable and provides faster inference with high-quality generation. Compared to traditional DDIM samplers [43], which require hundreds of steps, DPM Solver++ can achieve high-quality samples within 15 to 20 steps [32].

D.2 Imputation

Diffusion models generate random samples from the learned distribution. Instead, imputation [21] is a technique to guide the back sampling process given observation points. We condition the denoising process with Voronoi tessellations of the observed data from monitoring stations. This is achieved by injecting Voronoi embedding into the layers of the denoiser D_{θ} through a cross attention mechanism.

Furthermore, we apply *masked* back sampling using sparse observation points to guide generation. Since the values from monitoring stations are treated as ground truth, we preserve them during sample generation. Let $\mathbf{x}_t^{\text{real}} \in \mathbb{R}^{4 \times n_x \times n_y}$ be the grid containing sparse observation data, where observed values are placed at their corresponding sensor locations and zero elsewhere. The sparse data information is artificially injected into the output generated from the denoiser D_{θ} :

$$\mathbf{y}_t^i = D_{\theta}(\mathbf{x}_t^i, \mathbf{z}_{\text{emb}}^t; i) \odot (1 - \mathbf{\Omega}_t) + \mathbf{x}_t^{\text{real}} \odot \mathbf{\Omega}_t, \quad (\text{D5})$$

where $i \in (0, R)$ is an intermediate time step of the diffusion process. The DPM Solver++ predicts a one-step denoised image $\mathbf{x}_t^{i-1}$ given $\mathbf{y}_t^i$. The above process is repeated R times to obtain the final generated sample.

Appendix E Evaluation Metrics

The model’s performance is evaluated on the simulation data using Mean Relative Error (MRE), Mean Fractional Error (MFE), Mean Fractional Bias (MFB) and Structural Similarity Index Measure (SSIM) [44].

$$\text{MRE} = \frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{y}_t - \mathbf{x}_t\|_2}{\|\mathbf{x}_t\|_2}, \quad (\text{E6a})$$

$$\text{MFE} = \frac{1}{T \times n} \sum_{t=1}^T \sum_{i=1}^n \frac{2|y_{t,i} - x_{t,i}|}{x_{t,i} + y_{t,i}}, \quad (\text{E6b})$$

$$\text{MFB} = \frac{1}{T \times n} \sum_{t=1}^T \sum_{i=1}^n \frac{2(y_{t,i} - x_{t,i})}{x_{t,i} + y_{t,i}} \quad (\text{E6c})$$

where $\mathbf{x}_t \in \mathbb{R}^{4 \times n_x \times n_y}$ is the reference field, $\mathbf{y}_t \in \mathbb{R}^{4 \times n_x \times n_y}$ is the reconstructed image and T is the size of the test dataset. $y_{t,i}, x_{t,i}$ denote the pixel values; for example, $\mathbf{x}_t = [x_{t,1}, \dots, x_{t,n}]$ with $n = 4 \times n_x \times n_y$.

These metrics quantify the error at the pixel level but they do not necessarily reflect the preservation of structural information in reconstructed fields. To address this limitation, SSIM aims to compare patterns between images rather than individual pixel intensities, allowing it to work similarly to human vision.

We lack ground truth data to evaluate real-world observation data. Thus, we apply the binary evaluation mask to the data before calculating the metrics. For example, the MRE metric is computed as follows:

$$\text{MRE} = \frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{y}_t \odot \mathbf{\Omega}_t^{\text{eval}} - \mathbf{x}_t \odot \mathbf{\Omega}_t^{\text{eval}}\|_2}{\|\mathbf{x}_t \odot \mathbf{\Omega}_t^{\text{eval}}\|_2}. \quad (\text{E7})$$

To compute the power spectrum plots, each predicted output is converted to its 2D Fourier representation. It is then shifted such that the zero-frequency component is positioned at the centre. The power spectrum is defined as the squared magnitude of the Fourier coefficients. We then compute the radial average of the spectral image (the mean of values that are located equidistant from the centre). The radial mean spectrum, which is a function of frequency, is averaged over all samples and pollutants, resulting in the power spectrum plot.

Appendix F Data augmentation methods

This study employs a range of data augmentation techniques to reduce the distributional discrepancy between observed and simulated data. These methods were designed to produce realistic augmentations that respect the local correlations present in pollution fields. Each noise generation method has multiple hyperparameters, as well as a mean and standard deviation. Method-specific hyperparameters and their function are described in F3.

F.1 Gaussian perturbation

This perturbation method is the simplest and fastest method used in this study. The main idea is to generate white noise with the desired shape, usually in $\mathbb{R}^{t \times c \times n_x \times n_y}$, and to apply a 2D convolution over the last two dimensions. A kernel with Gaussian weights is used to avoid the generation of sharp transitions [45] while also providing a very rudimentary cloud-like appearance to the noise.

Table F3: Table enumerating and describing noise generation method-specific hyperparameters.

Method	Hyperparameter	Function
Gaussian	sigma	The size of the Gaussian kernel.
Time-aware Gaussian	sigma time sigma	The size of the spatial Gaussian kernel. The size of the temporal Gaussian kernel.
Perlin	base resolution octaves persistence lacunarity	Controls the coarseness of the base noise grid. The number of noise layers used for fractal noise. Controls the amplitude of each successive octave. Controls the frequency of each successive octave.
Cross-correlation	-	-

F.2 Time-aware Gaussian perturbation

This method extends the Gaussian perturbation approach by introducing temporal structure, generating perturbations that are correlated across consecutive time steps in a manner analogous to Gaussian process covariance models [46]. It is the only augmentation strategy considered in this study that explicitly incorporates temporal correlations. This is achieved by generating independent Gaussian noise examples before convolving along the time dimension with a Gaussian kernel.

F.3 Perlin perturbation

Perlin perturbation was initially developed to generate natural-looking textures in computer graphics [47], but over time has evolved into a more general tool for procedural generation. Notably, its use for realistic cloud generation [48] is of particular interest for this study. With its ability to create smooth yet irregular patterns, this method could improve the models’ robustness by introducing realistic cloud-like noise.

Perlin perturbation generates a grid of randomly oriented vectors that define local directional changes. To compute the noise value at a given point, the algorithm calculates contributions from the surrounding grid corners and interpolates them. Fractal Perlin noise extends this concept by summing multiple layers (“octaves”) of Perlin noise at increasing spatial frequencies (“lacunarity”) and decreasing amplitudes (“persistence”).

F.4 Cross-correlation perturbation

This method generates perturbations based on the cross-correlation structure among pixels in the training dataset. By estimating inter-pixel correlations and applying the corresponding covariance transformation, the method produces spatially correlated noise fields that better reflect the statistical properties of air-pollution patterns. As cross-correlation is an image-specific measure, the mean across the time dimension is computed. Using the Cholesky decomposition of the cross-correlation matrix, white noise is converted into cross-correlated noise.

Among the augmentation strategies considered, this method generates the most statistically realistic perturbations, as it explicitly preserves the spatial correlation

structure of the training data. However, this reliance on training-set statistics may limit the diversity of the generated samples, potentially reducing its effectiveness in improving model robustness to out-of-distribution observations.

Appendix G Model architectures

VUNet	ViTAE	CLSTM	Diffusion
15,482,884	18,798,024	2,522,244	27,529,988

Table G4: Table indicating the total number of trainable parameters in each model

VUNet (Figure G6) is a state-of-the-art CNN model that excels at capturing local spatial interactions, whereas ViTAE is a transformer-based model designed to model long-range and global dependencies. It consists of a transformer encoder followed by a Unet decoder. By contrast, CLSTM (Figure G7) specializes in learning temporal sequences of spatial data. Kriging is a traditional interpolation method [31] provided as a baseline for this study. From table G4, ViTAE has the highest number of trainable parameters among the deterministic models, followed by VUNet and CLSTM.

The neural network used in the diffusion model is a simple Unet along with a Voronoi encoder which performs cross-attention between the intermediate layers of the UNet and the latent-variable output of the Voronoi encoder.

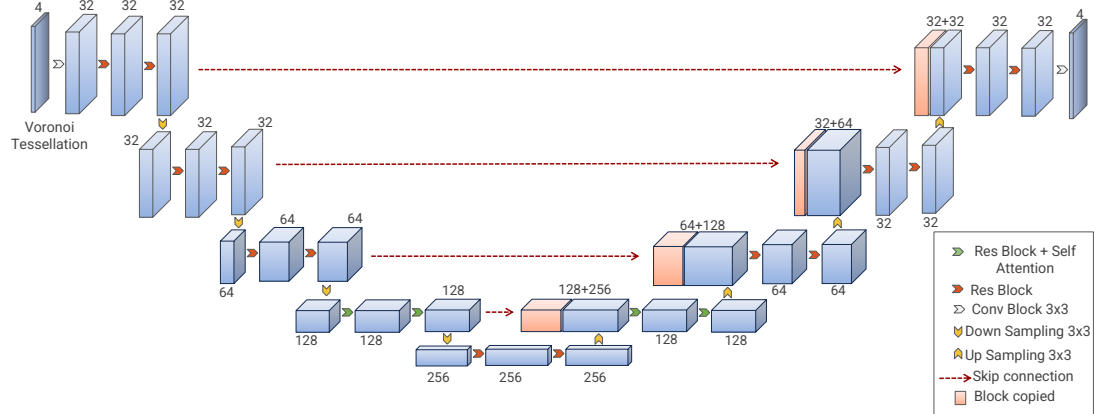


Fig. G6: Model architecture of the VUNet model.

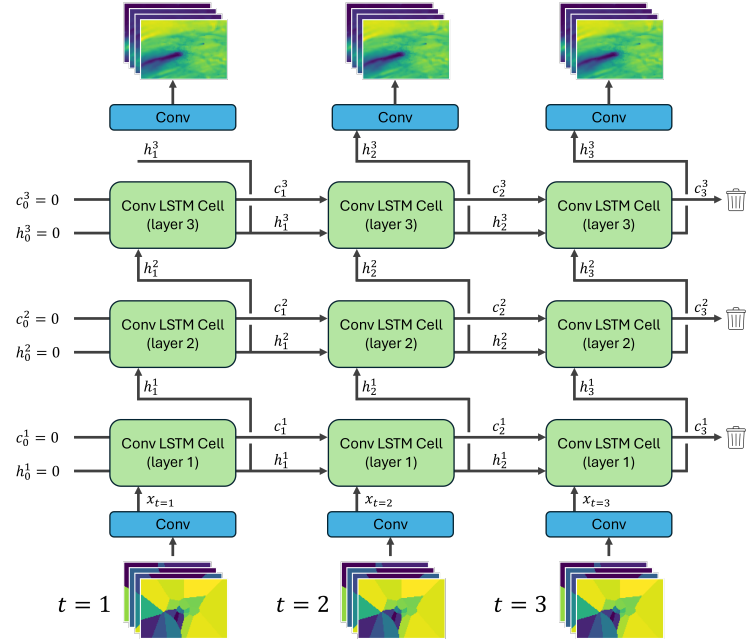


Fig. G7: Model architecture of the CLSTM model with timesteps $k = 3$.

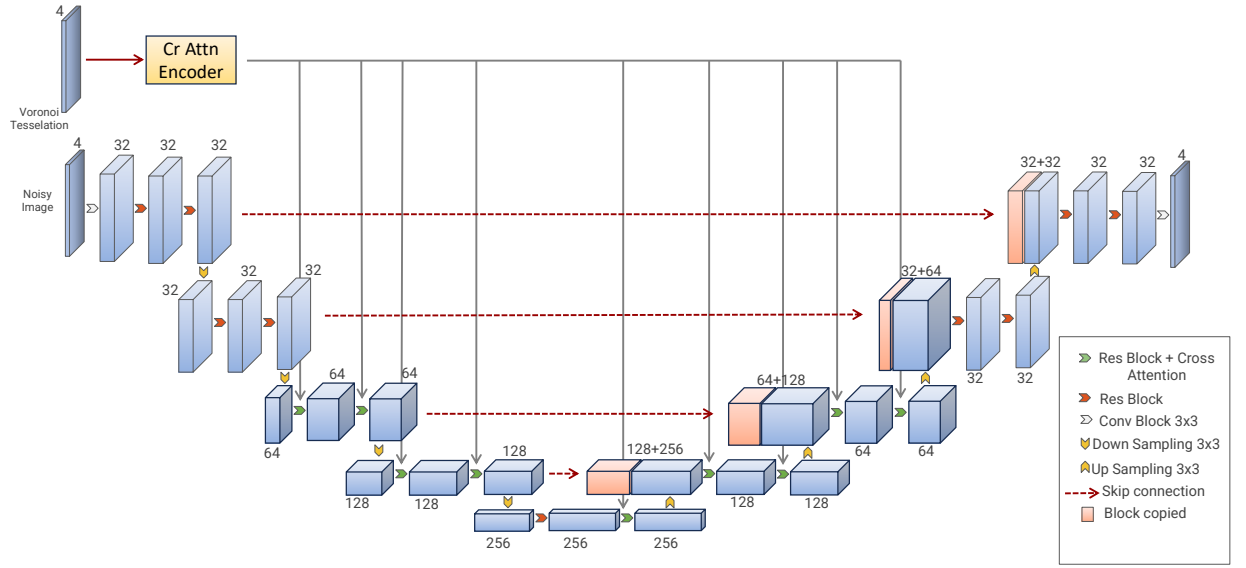


Fig. G8: Model architecture of Denoiser in diffusion model

References

- [1] Zheng, X.-y., Ding, H., Jiang, L.-n., Chen, S.-w., Zheng, J.-p., Qiu, M., Zhou, Y.-x., Chen, Q., Guan, W.-j.: Association between air pollutants and asthma emergency room visits and hospital admissions in time series studies: a systematic review and meta-analysis. *PloS one* **10**(9), 0138146 (2015)
- [2] Alexeeff, S.E., Liao, N.S., Liu, X., Van Den Eeden, S.K., Sidney, S.: Long-term pm_{2.5} exposure and risks of ischemic heart disease and stroke events: review and meta-analysis. *Journal of the American Heart Association* **10**(1), 016890 (2021)
- [3] Di, Q., Wang, Y., Zanobetti, A., Wang, Y., Koutrakis, P., Choirat, C., Dominici, F., Schwartz, J.D.: Air pollution and mortality in the medicare population. *New England Journal of Medicine* **376**(26), 2513–2522 (2017) <https://doi.org/10.1056/NEJMoa1702747> <https://www.nejm.org/doi/pdf/10.1056/NEJMoa1702747>
- [4] Sicard, P., Agathokleous, E., Anenberg, S.C., De Marco, A., Paoletti, E., Calatayud, V.: Trends in urban air pollution over the last two decades: A global perspective. *Science of The Total Environment* **858**, 160064 (2023) <https://doi.org/10.1016/j.scitotenv.2022.160064>

- [5] Brauer, M., Freedman, G., Frostad, J., Van Donkelaar, A., Martin, R.V., Dentener, F., Dingenen, R.v., Estep, K., Amini, H., Apte, J.S., *et al.*: Ambient air pollution exposure estimation for the global burden of disease 2013. *Environmental science & technology* **50**(1), 79–88 (2016)
- [6] Barwick, P.J., Li, S., Lin, L., Zou, E.Y.: From fog to smog: The value of pollution information. *American Economic Review* **114**(5), 1338–1381 (2024)
- [7] Pascal, M., de Crouy Chanel, P., Wagner, V., Corso, M., Tillier, C., Bentayeb, M., Blanchard, M., Cochet, A., Pascal, L., Host, S., Gorla, S., Le Tertre, A., Chatignoux, E., Ung, A., Beaudeau, P., Medina, S.: The mortality impacts of fine particles in France. *Science of The Total Environment* **571**, 416–425 (2016) <https://doi.org/10.1016/j.scitotenv.2016.06.213>
- [8] Sartelet, K., Kerckhoffs, J., Athanasopoulou, E., Lugon, L., Vasilescu, J., Zhong, J., Hoek, G., Joly, C., Park, S.-J., Talianu, C., van den Elshout, S., Dugay, F., Gerasopoulos, E., Ilie, A., Kim, Y., Nicolae, D., Harrison, R.M., Petaja, T.: Air pollution mapping and variability over five European cities. *Environment International* **199**, 109474 (2025) <https://doi.org/10.1016/j.envint.2025.109474>
- [9] Shukla, K., Kumar, P., Mann, G.S., Khare, M.: Mapping spatial distribution of particulate matter using Kriging and Inverse Distance Weighting at supersites of megacity Delhi. *Sustainable Cities and Society* **54**, 101997 (2020) <https://doi.org/10.1016/j.scs.2019.101997>
- [10] Tilloy, A., Mallet, V., Poulet, D., Pesin, C., Brocheton, F.: BLUE-based NO₂ data assimilation at urban scale. *Journal of Geophysical Research: Atmospheres* **118**(4), 2031–2040 (2013) <https://doi.org/10.1002/jgrd.50233>
- [11] Hayward, I., Martin, N.A., Ferracci, V., Kazemimanesh, M., Kumar, P.: Low-cost air quality sensors: biases, corrections and challenges in their comparability. *Atmosphere* **15**(12), 1523 (2024)
- [12] Stowe, S., Bohra, R., Vilcassim, M.: Assessing the efficacy of a low-cost air pollution monitoring device for environmental and occupational exposure assessments. *Environmental Monitoring and Assessment* **198**(1), 40 (2026)
- [13] He, H., Schäfer, B., Beck, C.: Spatial heterogeneity of air pollution statistics in europe. *Scientific Reports* **12**(1), 12215 (2022)
- [14] Hase, N., Miller, S.M., Maaß, P., Notholt, J., Palm, M., Warneke, T.: Atmospheric inverse modeling via sparse reconstruction. *Geoscientific Model Development* **10**(10), 3695–3713 (2017)
- [15] Rodgers, C.D.: *Inverse Methods for Atmospheric Sounding: Theory and Practice* vol. 2. World scientific, Singapore (2000)

- [16] Le, V.-D., Bui, T.-C., Cha, S.-K.: Spatiotemporal deep learning model for citywide air pollution interpolation and prediction. In: 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 55–62 (2020). IEEE
- [17] Sasaki, Y., Harada, K., Yamasaki, S., Onizuka, M.: Airex: Neural network-based approach for air quality inference in unmonitored cities. In: 2022 23rd IEEE International Conference on Mobile Data Management (MDM), pp. 119–124 (2022). IEEE
- [18] Feng, Y., Wang, Q., Xia, Y., Huang, J., Zhong, S., Liang, Y.: Spatio-temporal field neural networks for air quality inference. arXiv preprint arXiv:2403.02354 (2024)
- [19] Seinfeld, J.H., Pandis, S.N.: Atmospheric Chemistry and Physics: from Air Pollution to Climate Change, pp. 131–165. John Wiley & Sons, Hoboken, NJ (2016)
- [20] Santos, J.E., Fox, Z.R., Mohan, A., O’Malley, D., Viswanathan, H., Lubbers, N.: Development of the senseiver for efficient field reconstruction from sparse observations. *Nature Machine Intelligence* **5**(11), 1317–1325 (2023)
- [21] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
- [22] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.-H.: Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys* **56**(4), 1–39 (2023)
- [23] Liu, M., Huang, H., Feng, H., Sun, L., Du, B., Fu, Y.: Pristi: A conditional diffusion framework for spatiotemporal imputation. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE), pp. 1927–1939 (2023). IEEE
- [24] Zheng, G., Wan, Y., Zhang, X., Liu, X.: Noise-aware diffusion for city-scale air-quality reconstruction from sparse monitoring stations. *ISPRS International Journal of Geo-Information* **15**(4), 171 (2026)
- [25] Lugon, L., Kim, Y., Vigneron, J., Chrétien, O., André, M., André, J.-M., Moukhtar, S., Redaelli, M., Sartelet, K.: Effect of vehicle fleet composition and mobility on outdoor population exposure: A street resolution analysis in paris. *Atmospheric Pollution Research* **13**(5), 101365 (2022)
- [26] Fukami, K., Maulik, R., Ramachandra, N., Fukagata, K., Taira, K.: Global field reconstruction from sparse sensors with voronoi tessellation-assisted deep learning. *Nature Machine Intelligence* **3**(11) (2021)
- [27] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for

- biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241 (2015). Springer
- [28] Fan, H., Cheng, S., Nazelle, A.J., Arcucci, R.: Vitae-sl: A vision transformer-based autoencoder and spatial interpolation learner for field reconstruction. *Computer Physics Communications* **308** (2025)
 - [29] Gers, F.A., Schmidhuber, J.: Recurrent nets that time and count. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol. 3, pp. 189–194 (2000). <https://doi.org/10.1109/IJCNN.2000.861302>
 - [30] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* **28** (2015)
 - [31] Kleijnen, J.P.: Kriging metamodeling in simulation: A review. *European Journal of Operational Research* **192**(3) (2009)
 - [32] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 1–22 (2025)
 - [33] Rissanen, S., Heinonen, M., Solin, A.: Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397* (2022)
 - [34] Laboratoire central de surveillance de la qualité de l’air: GEOD’AIR. <https://www.geodair.fr/donnees/consultation>
 - [35] Bashan, N.F., Li, W., Wang, Q.R.: Dynamics of pm_{2.5} and network activity during extreme pollution events. *npj Climate and Atmospheric Science* **7**(1), 171 (2024)
 - [36] Sartelet, K.N., Debry, E., Fahey, K., Roustan, Y., Tombette, M., Sportisse, B.: Simulation of aerosols and gas-phase species over Europe with the Polyphemus system: Part I - Model-to-data comparison for 2001. *Atmospheric Environment* **41**(29), 6116–6131 (2007) <https://doi.org/10.1016/j.atmosenv.2007.04.024>
 - [37] Sartelet, K., Couvidat, F., Wang, Z., Flageul, C., Kim, Y.: SSH-Aerosol v1.1: A Modular Box Model to Simulate the Evolution of Primary and Secondary Aerosols. *Atmosphere* **11**(5) (2020) <https://doi.org/10.3390/atmos11050525>
 - [38] Andre, M., Sartelet, K., Moukhtar, S., Andre, J.M., Redaelli, M.: Diesel, petrol or electric vehicles: What choices to improve urban air quality in the Ile-de-France region? A simulation platform and case study. *Atmospheric Environment* **241**, 117752 (2020) <https://doi.org/10.1016/j.atmosenv.2020.117752>

- [39] Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
- [40] Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* **35**, 26565–26577 (2022)
- [41] Zhuang, Y., Cheng, S., Duraisamy, K.: Spatially-aware diffusion models with cross-attention for global field reconstruction with sparse observations. *Computer Methods in Applied Mechanics and Engineering* **435**, 117623 (2025)
- [42] Leinonen, J., Hamann, U., Nerini, D., Germann, U., Franch, G.: Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. *arXiv preprint arXiv:2304.12891* (2023)
- [43] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020)
- [44] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
- [45] Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 3rd edn. Pearson Education, Upper Saddle River, NJ (2008)
- [46] Williams, C.K., Rasmussen, C.E.: *Gaussian Processes for Machine Learning* vol. 2. MIT press Cambridge, MA, Cambridge, MA (2006)
- [47] Perlin, K.: An image synthesizer. *ACM Siggraph Computer Graphics* **19**(3), 287–296 (1985)
- [48] Man, P.: Generating and real-time rendering of clouds. In: *Central European Seminar on Computer Graphics*, vol. 1 (2006). Citeseer Castá-Papiernicka, Slovakia