

Air Pollution Nowcasting via Multimodal Learning with Temporal and Visual Data

Sojeong Kim^{1*}, Jaewan Chun^{1*}, Jongha Lee^{1*}, Sung-Chul Hong²,
Jae-Bum Lee², Seung-Hee Eun², Yonghee Lee², Kijung Shin¹

¹Kim Jaechul Graduate School of AI, KAIST

²Air Quality Forecasting Center, National Institute of Environmental Research

¹{kimsojeong, jjwpalace, jhsk777, kijungs}@kaist.ac.kr, ²{schong, gercljb, eunshee, lyhee94}@korea.kr

Abstract—Air pollution, especially fine particulate matter ($PM_{2.5}$), poses a serious threat to global public health. Nowcasting (i.e., real-time prediction) is essential to mitigate exposure risks. However, effectively applying deep learning to $PM_{2.5}$ nowcasting requires addressing several challenges: (C1) irregular spatial distribution of ground-based observations, (C2) data imbalance due to the rarity of extreme pollution levels, and (C3) increasing difficulty with longer lead times.

In this paper, we propose MULTIAIR (Multimodal learning for Air pollution nowcasting), a novel multimodal deep-learning method for $PM_{2.5}$ nowcasting. MULTIAIR addresses these challenges by: (S1) integrating satellite imagery to complement ground-based observations, (S2) employing learnable normalization to improve extreme-level predictions, and (S3) adopting curriculum learning for stable longer lead-time forecasting. We empirically show that MULTIAIR outperforms state-of-the-art baselines, achieving on average 14.9% and 13.3% improvements compared to the strongest baselines in regression and classification metrics, respectively.

Index Terms—Multimodal Learning, Nowcasting, Time Series, Image Data, Data Imbalance, Curriculum Learning

I. INTRODUCTION

$PM_{2.5}$ (i.e., particulate matter with a diameter of 2.5 micrometers or less) is a major air pollutant linked to serious health problems, including acute and chronic respiratory diseases, cardiovascular diseases, and premature mortality [1], [2]. Therefore, accurately forecasting $PM_{2.5}$, especially nowcasting (i.e., real-time forecasting), is essential for promptly detecting sudden changes in air quality and supporting timely public health interventions. Recent machine learning-based approaches, including CNN-LSTM [3], GCN [4], and Transformer [5], have achieved strong performance by capturing spatial-temporal dependencies in $PM_{2.5}$ forecasting.

Despite these advances, several challenges remain in achieving accurate $PM_{2.5}$ nowcasting:

- **C1. Irregular spatial distribution of ground-based observations.** Ground-based observations are collected from sensors. Their irregular spatial distribution, however, leads to data sparsity in certain regions, making it challenging to accurately capture spatial patterns.
- **C2. Data imbalance due to the rarity of extreme pollution levels.** The primary goal of $PM_{2.5}$ nowcasting is to

predict extreme pollution levels, which pose serious health risks. However, the rarity of such events [6], [7] leads to severe data imbalance (i.e., insufficient training samples) and thus the underestimation of such events [8], [9].

- **C3. Increasing difficulty with longer lead times.** While nowcasting primarily focuses on short lead-time forecasting, slightly longer lead times such as 3-6 hours still fall within this scope. However, longer lead times increase prediction difficulty, requiring machine learning models to learn and capture subtle input signals.

In this paper, we propose MULTIAIR (Multimodal learning for Air pollution nowcasting), a novel multimodal learning approach that integrates time-series and image-sequence data for $PM_{2.5}$ nowcasting. MULTIAIR addresses the above challenges (C1–C3) through the following key components:

- **(S1) Adaptive multimodal fusion:** MULTIAIR addresses Challenge C1 (i.e., the irregular spatial distribution of ground-based observations) by integrating satellite images as a complementary data source. Rather than simply concatenating multimodal representations, MULTIAIR employs a specialized decoder to adaptively integrate temporal information from ground-based observations and spatial information from satellite images.
- **(S2) Learnable normalization and a specialized loss function:** In response to Challenge C2 (i.e., data imbalance), we employ learnable normalization to mitigate discrepancies in time series data for rare (i.e., extreme) cases, along with an error-adaptive loss function that adaptively adjusts weighting to emphasize high-error cases, which often include underpredicted extreme events.
- **(S3) Curriculum learning and auto-regressive prediction:** In response to Challenge C3 (i.e., increased difficulty with longer lead times), we employ curriculum learning (i.e., progressive training from simpler to more complex tasks) to improve long lead-time prediction, which is a complex task. Additionally, we incorporate auto-regressive prediction to capture temporal dependencies essential for accurate long lead-time prediction.

We evaluate MULTIAIR on a multimodal dataset with AOD (aerosol optical depth) satellite imagery and ground-based observations from South Korea (537 stations) and China (583

*Equal Contributions

stations) over 2021–2023, forecasting 1 to 6 hours ahead. We show that MULTIAIR outperforms state-of-the-art methods across various $PM_{2.5}$ levels, achieving on average 14.9% and 13.3% improvements compared to the strongest baselines in regression and classification metrics, respectively. For reproducibility, our source code and datasets are available at <https://anonymous.4open.science/r/MultiAir-20C5>.

The rest of the paper is organized as follows. In Section II, we review related work. In Section III, we introduce basic notations used in this paper and define the air pollution nowcasting problem. In Section IV, we describe our proposed approach, MULTIAIR. In Section V, we provide experimental results. Lastly, in Section VI, we provide conclusions.

II. RELATED WORK

In this section, we discuss prior work relevant to this study.

A. Deep Learning for (Multi-Sensor) Time Series Data

In time series analysis, modeling the temporal patterns of time series data is crucial. TimesNet [10] transforms one-dimensional time series into two-dimensional tensors based on multiple periodicities and captures multi-scale temporal variations through convolution-based operations. TimeMixer [11] is a fully MLP-based time series forecasting model that decomposes and aggregates seasonal and trend components and ensembles multiple predictors. This approach combines multi-scale information for both past extraction and future prediction. Multi-sensor time series require modeling not only temporal patterns within each sensor but also spatial dependencies across sensors. $PM_{2.5}$ forecasting is a representative multi-sensor prediction task, as measurements at different stations are spatially correlated. Prior studies have modeled temporal dependencies using architectures such as BiLSTMs [12] and Transformers [5], [13] while capturing spatial relationships among sensors using CNN-based approaches [3] and graph neural networks (GNNs) [4], [13].

B. Multimodal Forecasting with Time Series and Image Data

Multimodal approaches combine time-series and geo-spatial images for forecasting, particularly for weather and air-quality prediction. One of the key challenges is handling data heterogeneity between grid-based image data and sparsely, irregularly distributed time series data. For the $PM_{2.5}$ forecasting task, GNN-LSTM [14] models monitoring stations as graph nodes, defines edges via a distance-based threshold, and assigns edge weights according to wind direction and speed. In addition, aerosol optical depth (AOD) features are incorporated into the LSTM inputs to capture aloft pollution transport. CrossViViT [15] processes satellite imagery and time-series data through separate modality encoders and integrates them via cross-transformer layers. It then employs a transformer-based decoder for future sequence prediction. ASOC [16] is a classification model for precipitation nowcasting that uses time-series observation data and radar reflectivity images. Specifically, the model utilizes a pre-trained image-based model, DeepRaNE [17], to obtain pixel embeddings. These

TABLE I
FREQUENTLY USED NOTATIONS.

Notation	Description
t	current time (unit: hours)
T_{out}	maximum lead time (length of output sequences)
T_t	length of input time series data excluding current time t
T_i	length of input image sequence data excluding current time t
Δt	lead time which ranges from 1 to T_{out} , i.e., $\Delta t \in \{1, \dots, T_{out}\}$
S_T	set of ground-based stations in the target region
S_C	set of auxiliary ground-based stations in the context region
S	set of all ground-based stations, i.e., $S = S_T \cup S_C$
$\mathbf{P}^{(t)}$	ground-truth $PM_{2.5}$ values at time t from all stations
$\mathbf{P}_s^{(t)}$	ground-truth $PM_{2.5}$ value at time t from station s
$\mathbf{O}^{(t)}$	observation values except $PM_{2.5}$ at time t from all stations
$\mathbf{O}_s^{(t)}$	observation values except $PM_{2.5}$ at time t from station s
$\mathbf{I}^{(t)}$	AOD image at time t centered around the target region
$\mathbf{I}_s^{(t)}$	AOD value at time t at the pixel corresponding to station s
$\hat{\mathbf{P}}_s^{(t+\Delta t)}$	predicted $PM_{2.5}$ values at time $t + \Delta t$ at station s
d_t	number of observed values in a ground observation sensor
d_h	number of hidden dimensions used in the proposed model

pixel embeddings are concatenated with the corresponding ground observation data and auxiliary information (e.g., locations and lead times) as input to the LSTM and attention blocks. Our model shares some encoder-level design choices with ASOC; however, our key contributions lie elsewhere, namely in the key components S1-S3 described in Section I.

C. Data Imbalance Mitigation Techniques

Data imbalance often causes machine learning models to underperform on minority classes. For example, in $PM_{2.5}$ nowcasting, the scarcity of extreme pollution events may lead to systematic underestimation of high pollution levels. Learnable normalization [18], [19] alleviates this issue by adaptively rescaling feature representations, thereby enhancing sensitivity to rare events. RevIN [18] normalizes input time series through learnable affine transformations, followed by output denormalization for adaptive prediction. Dish-TS [19] further accounts for differences in the occurrence frequencies of rare cases in the input and output spaces. Another approach to mitigating data imbalance is to incorporate sample weighting into loss functions, placing greater emphasis on underrepresented rare events. Class-balanced loss [20] adjusts weights by considering the effective number of samples in each class, and focal loss [21] mitigates majority class dominance by adaptively weighting minority samples. In our work, we introduce a unified strategy blending adaptive normalization with a redesigned form of focal-R loss [22], a regression variant of focal loss, to better handle extreme pollution cases.

D. Curriculum Learning

Curriculum learning [23] is a training strategy inspired by the way humans learn, where simpler tasks are introduced first, and more complex tasks are gradually added as the learning progresses. The primary objective of curriculum learning is to guide the model to build a strong foundation by learning easier patterns before tackling more challenging ones. It can guide the machine learning process, thereby improving the generalizability and stability of the model. Curriculum learning has also been applied in multimodal learning. However, while existing methods account for difficulty variations across

samples [24] or modalities [25], we focus on the increasing difficulty associated with longer lead times.

III. PROBLEM DESCRIPTION

In this section, we introduce the problem setting and notation, and then formulate the air pollution nowcasting problem.

A. Problem Settings

We consider two spatial domains: the *target region*, where air pollution nowcasting is performed, and the *context region*, which provides contextual information. Each region contains stations (i.e., sensors) that periodically collect d_t ground-based observation values, including $PM_{2.5}$. We denote the set of stations in the target and context regions as S_T and S_C , respectively. In addition to the ground-based station measurements, we incorporate aerosol optical depth (AOD) images collected periodically from a geostationary satellite. These images are centered on the target region but also capture surrounding areas. We assume that time series data from the stations and image sequence data from the satellite are collected every hour.

B. Basic Notations

We introduce commonly used notations, which are listed in Table I. We use $\mathbf{P}^{(t)} \in \mathbb{R}^{|S_T \cup S_C|}$ as ground-truth $PM_{2.5}$ values at time t from all stations and use $\mathbf{P}_s^{(t)} \in \mathbb{R}$ as $PM_{2.5}$ value at time t from each station s . Then, we use $\mathbf{O}^{(t)} \in \mathbb{R}^{|S_T \cup S_C| \times (d_t - 1)}$ as other observation values at time t from all stations and use $\mathbf{O}_s^{(t)} \in \mathbb{R}^{d_t - 1}$ as other observation values at time t from each station s . Lastly, $\mathbf{I}^{(t)} \in \mathbb{R}^{H \times W}$ denotes the AOD image at time t centered around the target region, where H and W denote the height and the width of the image, respectively, and $\mathbf{I}_s^{(t)} \in \mathbb{R}$ denotes the AOD value at the pixel corresponding to the location of each station s in $\mathbf{I}^{(t)}$.

C. Air Pollution Nowcasting Problem Definition

The goal of air pollution nowcasting in general is to predict future $PM_{2.5}$ values (units: $\mu g/m^3$) at the locations of all stations within the target region (defined in Section III-A) at short lead times. In our task, $PM_{2.5}$ values are predicted at one-hour intervals from $t+1$ to $t+T_{out}$, where t is the current time (units: hours) and $T_{out} = 6$ (hours) is the maximum lead time. We denote a lead time by Δt , i.e., $\Delta t \in \{1, \dots, T_{out}\}$.

To predict the future $PM_{2.5}$ values, we use past $PM_{2.5}$ values and other observation values measured at ground-based observation stations (i.e., sensors) in the target and context regions, along with past AOD images collected from a geostationary satellite. Formally, we define the air pollution nowcasting problem at the current time t as:

- **Input:** (1) past $PM_{2.5}$ values $\mathbf{P}^{(t-T_1)}, \dots, \mathbf{P}^{(t)}$, (2) past other observation values $\mathbf{O}^{(t-T_1)}, \dots, \mathbf{O}^{(t)}$, (3) past AOD images $\mathbf{I}^{(t-T_1)}, \dots, \mathbf{I}^{(t)}$.
- **Objective:** to accurately predict future $PM_{2.5}$ values $\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T}^{T_{out}}_{\Delta t=1}$ from $t+1$ to $t+T_{out}$ at the locations of all stations in the target region.

Here, T_t denotes the input length of time series data excluding the current time t , and T_i denotes the input length of image

sequence data excluding t . Note that our task is regression, while we use both regression and classification metrics for evaluation under varying pollution levels (Section V-A).

IV. PROPOSED METHOD

In this section, we propose MULTIAIR, a novel multimodal learning approach for $PM_{2.5}$ nowcasting. Considering the three challenges in Section I (i.e., C1-C3), MULTIAIR is designed to achieve the following goals:

- **Fusing complementary data modalities:** In response to the irregular spatial distribution of ground-based observation data (i.e., Challenge C1), MULTIAIR integrates complementary satellite images, especially through specialized encoders (Section IV-B) and decoders (Section IV-C).
- **Addressing data imbalance for extreme-level prediction:** Extreme $PM_{2.5}$ values are rare, posing a challenge for accurate prediction (Challenge C2). To address this, we employ a specialized (de)normalization scheme (Sections IV-A and IV-C) and a specialized loss function (Section IV-D).
- **Taming the difficulty of long-term prediction:** To address the increased difficulty with longer lead times (Challenge C3), curriculum learning is employed to train an auto-regressive model (Section IV-C), which captures temporal dependencies useful in long-term prediction (Section IV-D).

Refer to Figure 1 and Algorithm 1 for the visual overview and overall process of MULTIAIR, respectively. Below, we detail the (1) data preprocessing, (2) encoding, (3) decoding, and (4) training procedures of MULTIAIR.

A. Data Preprocessing via Learnable Normalization

For the time series inputs $(\mathbf{P}^{(t')})_{t'=t-T_t}^t$ and $(\mathbf{O}^{(t')})_{t'=t-T_t}^t$ at time t , data preprocessing is applied to address data scale-related challenges.¹ Here, we assume that missing values have already been processed as described in Appendix IX [26].

One challenge stems from the different ranges of observation values. For example, SO_2 values typically range from 0 to 0.2 (ppm), while $PM_{2.5}$ values typically range from 0 to 150 ($\mu g/m^3$). Since such disparities can hinder the effective use of smaller-scale values during model training, we apply Z-score normalization to all non- $PM_{2.5}$ observations as follows:

$$\tilde{\mathbf{O}}_s^{(t')} = (\mathbf{O}_s^{(t')} - \boldsymbol{\mu}) / \boldsymbol{\sigma}, \quad (1)$$

where station $s \in S_T \cup S_C$; time $t' \in \{t-T_t, \dots, t\}$; and $\boldsymbol{\mu} \in \mathbb{R}^{d_t-1}$ and $\boldsymbol{\sigma} \in \mathbb{R}^{d_t-1}$ denote the means and standard deviations of observation values, computed across all ground-based stations over the entire period in the training data.

Another complex challenge is data imbalance due to the rarity of extreme pollution levels (Challenge C2 in Section I). To address this, for input $PM_{2.5}$ values, we employ learnable normalization (discussed in Section II-C). The key idea is to model scale information separately, allowing for scale-insensitive predictions. This improves prediction accuracy across various scales, including rare extreme $PM_{2.5}$ levels.

¹For input images (i.e., $(\mathbf{I}^{(t')})_{t'=t-T_i}^t$), raw AOD values are used without normalization as they are encoded separately from ground-based observation data (see Section IV-B).

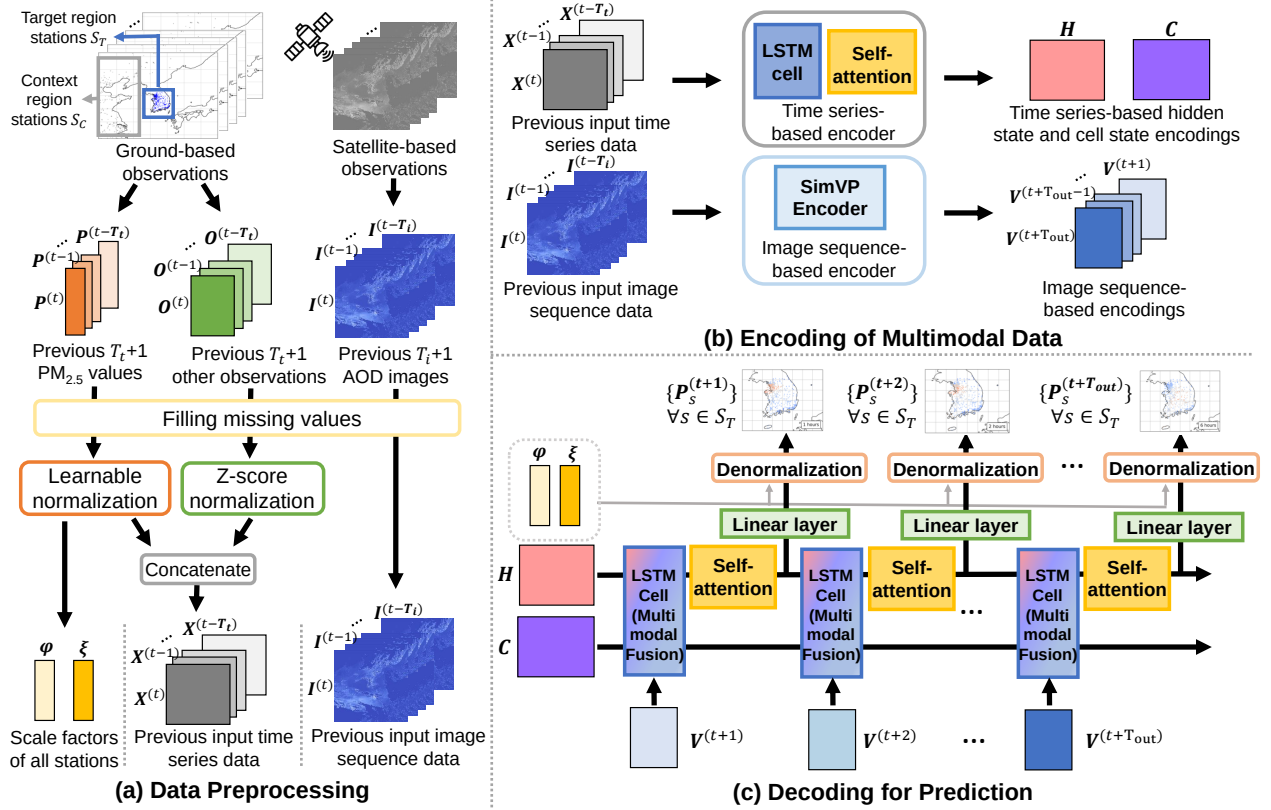


Fig. 1. Overview of MULTIAIR, which consists of (1) data preprocessing, (2) encoding of multimodal data, and (3) decoding for prediction.

Algorithm 1: Inference step of MULTIAIR $f_{\Phi, \psi}$

Input: (1) Current time t , (2) Maximum lead time T_{out} , (3) Previous snapshots of $PM_{2.5}$ values $(\mathbf{P}^{(t')})_{t'=t-T_t}^t$, (4) Previous snapshots of other observations $(\mathbf{O}^{(t')})_{t'=t-T_t}^t$, (5) Previous AOD images $(\mathbf{I}^{(t')})_{t'=t-T_t}^t$.

Output: Predicted $PM_{2.5}$ values $(\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T})_{\Delta t=1}^{T_{out}}$ at all stations in the target region from $t+1$ to $t+T_{out}$.

```

1 foreach  $s \in S_T \cup S_C$  do
2    $[\varphi_s, \xi_s] \leftarrow f_{\theta}([\mathbf{P}_s^{(t-T_t)}] \parallel \dots \parallel \mathbf{P}_s^{(t)})$ 
3   for  $t' \leftarrow t - T_t$  to  $t$  do
4      $\mathbf{X}^{(t')} \leftarrow \text{PREPROCESS}(\mathbf{P}^{(t')}, \mathbf{O}^{(t')}, \{\varphi_s\}_{s \in S_T \cup S_C}, \{\xi_s\}_{s \in S_T \cup S_C})$ 
5    $\mathbf{H}, \mathbf{C} \leftarrow \text{ENC}_T((\mathbf{X}^{(t')})_{t'=t-T_t}^t)$  // Time-series encoder
6    $(\mathbf{V}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}} \leftarrow \text{ENC}_I(t, T_{out}, (\mathbf{I}^{(t')})_{t'=t-T_t}^t)$  // Image-sequence encoder
7    $(\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T})_{\Delta t=1}^{T_{out}} \leftarrow \text{DEC}(t, T_{out}, \mathbf{H}, \mathbf{C}, (\mathbf{V}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}, \{\varphi_s\}_{s \in S_T}, \{\xi_s\}_{s \in S_T})$  // Multimodal decoder for prediction
8 return  $(\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T})_{\Delta t=1}^{T_{out}}$ 

```

Specifically, input $PM_{2.5}$ values are normalized as follows:

$$\tilde{\mathbf{P}}_s^{(t')} = (\mathbf{P}_s^{(t')} - \varphi_s) / (\sqrt{\xi_s^2 + \epsilon}). \quad (2)$$

Here, station $s \in S_T \cup S_C$; time $t' \in \{t - T_t, \dots, t\}$; $\epsilon > 0$ is a small fixed constant introduced for numerical stability; and $\varphi_s \in \mathbb{R}$ and $\xi_s \in \mathbb{R}$ are scale coefficients computed using input $PM_{2.5}$ values from the station s as follows:

$$\varphi_s, \xi_s = f_{\theta}([\mathbf{P}_s^{(t-T_t)}] \parallel \dots \parallel \mathbf{P}_s^{(t)}), \quad (3)$$

where f_{θ} is a learnable function, such as a linear mapping. The scale coefficients φ_s and ξ_s are expected to be learned to represent the overall scale and fluctuation scale of input $PM_{2.5}$ values from station s , respectively. Note that the scale coefficients are stored for use in the denormalization of predicted $PM_{2.5}$ values in Section IV-C.

After normalization, the final input time series data from each station $s \in S_T \cup S_C$ at time t is obtained as follows:

$$\mathbf{X}_s^{(t')} = [\tilde{\mathbf{P}}_s^{(t')} \parallel \tilde{\mathbf{O}}_s^{(t')} \parallel \mathbf{m}_s^{(t')}], \quad (4)$$

where we concatenate (1) the normalized $PM_{2.5}$ value, (2) the normalized other observation values, and (3) the missing value indicator vector $\mathbf{m}_s^{(t')} \in \{0, 1\}^{d_t}$ (see Appendix IX [26] for details of $\mathbf{m}_s^{(t')}$). These preprocessed time series inputs $(\mathbf{X}^{(t')})_{t'=t-T_t}^t$ are used for the next encoding step, together with the input AOD images $(\mathbf{I}^{(t')})_{t'=t-T_t}^t$.

Complexity analysis. This data processing step takes $O(|S| \cdot T_t \cdot d_t)$ time by linearly scanning every station across the past T_t time steps and d_t features to apply the learnable normalization.

B. Encoding of Multimodal Data

In this section, we introduce two specialized encoders in MULTIAIR, designed to process input time series data (i.e., ground-based observations) and image sequence data (i.e., AOD images from a satellite), both obtained from the preprocessing step.

1) *Time-series encoder (Algorithm 2):* The first encoder is designed to capture temporal and spatial dynamics of the input time series data $(\mathbf{X}^{(t')})_{t'=t-T_t}^t$, i.e., ground-based observations from multiple stations. To achieve this, we integrate

Algorithm 2: Time-series encoder ENC_T

Input: Previous time series values from all stations
 $(\mathbf{X}^{(t')})_{t'=t-T_t}^t$

Output: Time series-based hidden state encodings $\mathbf{H}$ and cell state encodings $\mathbf{C}$ at all stations

```

1  $\mathbf{H} \leftarrow \{\emptyset\}_{s \in S_T \cup S_C}$  // Initial hidden state
2  $\mathbf{C} \leftarrow \{\emptyset\}_{s \in S_T \cup S_C}$  // Initial cell state
3 for  $t' \leftarrow t - T_t$  to  $t$  do
4   foreach  $s \in S_T \cup S_C$  do
5      $\hat{\mathbf{X}}_s^{(t')} \leftarrow [\mathbf{X}_s^{(t')} \parallel \text{PE}(s_x) \parallel \text{PE}(s_y) \parallel \text{PE}(t'_m) \parallel \text{PE}(t'_d) \parallel \text{PE}(t'_h)]$ 
6      $\mathbf{H}_s, \mathbf{C}_s \leftarrow \text{LSTM}_E(\hat{\mathbf{X}}_s^{(t')}, \mathbf{H}_s, \mathbf{C}_s)$ 
7    $\{\mathbf{H}_s\}_{s \in S_T \cup S_C} \leftarrow \text{MHA}(\{\mathbf{H}_s\}_{s \in S_T \cup S_C})$ 
8 return  $\mathbf{H}, \mathbf{C}$ 

```

an RNN-based model (spec., LSTM), which captures temporal dependencies, and a self-attention mechanism, which captures spatial dependencies between observation stations.

Specifically, for each station $s \in S_T \cup S_C$, the time series-based hidden state encoding vector $\mathbf{H}_s \in \mathbb{R}^{d_h}$ and the time series-based cell state encoding vector $\mathbf{C}_s \in \mathbb{R}^{d_h}$ are initialized, where d_h denotes the dimension of the encodings. Then, the time-series encoder sequentially updates these encodings by alternating (1) the LSTM cell and (2) the self-attention block, as outlined in Algorithm 2.

- **LSTM cell for temporal dependencies:** In addition to the input time series data $(\mathbf{X}_s^{(t')})_{t'=t-T_t}^t$, we incorporate auxiliary information on the location of each station s and time t' as extra input to the LSTM cell [27], denoted as LSTM_E . Specifically, at each time step $t' \in \{t - T_t, \dots, t\}$, the input from the station s to LSTM_E is:

$$\hat{\mathbf{X}}_s^{(t')} = [\mathbf{X}_s^{(t')} \parallel \text{PE}(s_x) \parallel \text{PE}(s_y) \parallel \text{PE}(t'_m) \parallel \text{PE}(t'_d) \parallel \text{PE}(t'_h)], \quad (5)$$

where $\text{PE}(\cdot) \in \mathbb{R}^{d_p}$ is a sinusoidal encoding vector from the original Transformer [28], with a dimension d_p ; s_x and s_y denote global location information of station s (s_x : longitude, s_y : latitude); and t'_m , t'_d , and t'_h denote the global temporal information (t'_m : month, t'_d : day, t'_h : hour) of t' . Using the input $\hat{\mathbf{X}}_s^{(t')}$, LSTM_E updates the hidden and cell state encodings for each station s at t' , i.e.,

$$\mathbf{H}_s, \mathbf{C}_s \leftarrow \text{LSTM}_E(\hat{\mathbf{X}}_s^{(t')}, \mathbf{H}_s, \mathbf{C}_s). \quad (6)$$

- **Self-attention block for spatial dependencies:** Then, we further update the state encodings for each station, considering the spatial dependencies between observation stations. To this end, we compute self-attention among the hidden state encodings $\{\mathbf{H}_s\}_{s \in S_T \cup S_C}$ using the multi-head self-attention block [28], denoted as MHA, incorporating information from relevant stations, i.e.,

$$\{\mathbf{H}_s\}_{s \in S_T \cup S_C} \leftarrow \text{MHA}(\{\mathbf{H}_s\}_{s \in S_T \cup S_C}), \quad (7)$$

where $\{\mathbf{H}_s\}_{s \in S_T \cup S_C}$ serves as the key, query, and value input for MHA.

The time-series encoder alternates the LSTM cell and self-attention module iteratively for the length of the input time series (i.e., $T_t + 1$ times). Lastly, it produces the final hidden-state encodings and cell-state encodings for all stations, denoted as $\mathbf{H}$ and $\mathbf{C}$, respectively. The detailed process of the time-series encoder is described in Algorithm 2.

Algorithm 3: Image-sequence encoder ENC_I

Input: (1) Current time t , (2) Maximum lead time T_{out} ,
 (3) Previous AOD image sequence $(\mathbf{I}^{(t')})_{t'=t-T_i}^t$

Output: Image-sequence encodings at all stations
 $(\mathbf{V}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$

```

1  $(\hat{\mathbf{I}}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}} \leftarrow f_\psi([\mathbf{I}^{(t-T_i)} \parallel \dots \parallel \mathbf{I}^{(t)}])$ 
2 for  $\Delta t \leftarrow 1$  to  $T_{out}$  do
3    $\mathbf{V}^{(t+\Delta t)} \leftarrow \{\emptyset\}_{s \in S_T \cup S_C}$ 
4    $\mu_I^{(t+\Delta t)} \leftarrow \sum_{s \in S_T \cup S_C} \hat{\mathbf{I}}_s^{(t+\Delta t)} / |S_T \cup S_C|$ 
5    $(\sigma_I^{(t+\Delta t)})^2 \leftarrow \sum_{s \in S_T \cup S_C} (\hat{\mathbf{I}}_s^{(t+\Delta t)} - \mu_I^{(t+\Delta t)})^2 / |S_T \cup S_C|$ 
6   foreach  $s \in S_T \cup S_C$  do
7      $\mathbf{V}_s^{(t+\Delta t)} \leftarrow [\mathbf{I}_s^{(t-T_i)} \parallel \dots \parallel \mathbf{I}_s^{(t)} \parallel \hat{\mathbf{I}}_s^{(t+\Delta t)} \parallel \mu_I^{(t+\Delta t)} \parallel (\sigma_I^{(t+\Delta t)})^2]$ 
8 return  $(\mathbf{V}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$ 

```

Complexity analysis. For the time-series encoder, each of the T_t time steps involves an LSTM update with a computational cost of $O(|S| \cdot d_h^2)$, followed by a full-station self-attention operation with a cost of $O(|S| \cdot d_h^2 + |S|^2 \cdot d_h)$. Hence, the total time complexity of the time-series encoder is $O(T_t \cdot (|S| \cdot d_h^2 + |S|^2 \cdot d_h))$.

2) *Image-sequence encoder (Algorithm 3):* The image-sequence encoder is designed to capture spatiotemporal characteristics of the input image sequence data. To accomplish this, we leverage a video-prediction model f_ψ , specifically SimVP [29], which is both effective and computationally efficient for this task. Prior to use, we pre-train f_ψ to predict future AOD image sequences, as described in Section IV-D.

Specifically, at the current time t , the input AOD image sequence $(\mathbf{I}^{(t')})_{t'=t-T_i}^t$ is fed into the video prediction model f_ψ to predict the future AOD image sequence $(\hat{\mathbf{I}}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$ from $t+1$ to $t+T_{out}$. Then, we construct the image-sequence encoding vector $\mathbf{V}_s^{(t+\Delta t)} \in \mathbb{R}^{T_i+4}$ for each future time $t+\Delta t$ and each station s by concatenating (1) **AOD prediction:** the predicted AOD value $\hat{\mathbf{I}}_s^{(t+\Delta t)}$ at the pixel corresponding to the station s from the predicted AOD image at $t+\Delta t$, (2) **temporal context:** the previous AOD values $\mathbf{I}_s^{(t-T_i)}, \dots, \mathbf{I}_s^{(t)}$ from the pixel corresponding to the location of the station s in the input AOD image sequence, and (3) **spatial context:** mean $\mu_I^{(t+\Delta t)}$ and variance $(\sigma_I^{(t+\Delta t)})^2$ of the AOD values across all pixels corresponding to any station from the predicted AOD image at $t+\Delta t$. We denote the image sequence-based encoding vectors for all stations at all lead times as $(\mathbf{V}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$. The details of the image-sequence encoder are provided in Algorithm 3.

Note that these image-sequence encodings complement time-series encodings. Specifically, they capture temporal and spatial contextual information that time series from irregularly-located observation stations alone cannot effectively provide (see Challenge C1 in Section I).

Complexity analysis. The image-sequence encoder takes $O(H \cdot W(T_i + T_{out}))$ time for applying a CNN to T_i input frames and T_{out} predicted frames, where each frame consists

of an $H \times W$ map, and an additional operation for per-station pixel extraction and statistical calculation with a cost of $O(|S| \cdot T_i \cdot T_{out})$.² As a result, the total complexity of the image-sequence encoder is $O(H \cdot W(T_i + T_{out}) + |S| \cdot T_i \cdot T_{out})$.

C. Multimodal Decoding for Prediction (Algorithm 4)

In the decoding step, MULTIAIR produces final $PM_{2.5}$ predictions (i.e., $\hat{\mathbf{P}}_s^{(t+\Delta t)}$ for each station s and lead time Δt) based on time-series encodings $\mathbf{H}$ and $\mathbf{C}$ (Section IV-B1) from ground-based observation data and image-sequence encodings $(\mathbf{V}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$ (Section IV-B2) from satellite image data, which are complementary. To effectively decode these multimodal encodings, especially by addressing Challenges C1–C3 (Section I), we employ a specialized auto-regressive structure and a learnable denormalization scheme.

1) *Recurrent Auto-regressive structure with multimodal fusion*: The proposed decoder has an auto-regressive structure composed of the LSTM cell [27], denoted as LSTM_D , and the multi-head self-attention module [28] MHA defined in IV-B1.³ These components capture temporal and spatial dependencies, respectively, to enhance prediction accuracy. Refer to Figure 1 for a visual description of the decoder.

Instead of relying on general multimodal fusion methods (e.g., MLPs with concatenated multimodal embeddings), MULTIAIR achieves adaptive multimodal fusion through the gating mechanism of an LSTM. The forget gate filters out temporal patterns in ground-based time-series data, while the input gate selectively incorporates new spatial information from satellite images. Through this dynamic balancing, MULTIAIR can preserve trends of time-series data while flexibly capturing sudden spatial variations.

Specifically, for the first lead time, the hidden state of the LSTM cell is initialized with the time series-based hidden state encodings $\mathbf{H}$, and the cell state is initialized with the time series-based cell state encodings $\mathbf{C}$. Then, LSTM_D updates the hidden and cell state encodings $\mathbf{H}_s, \mathbf{C}_s$ using the image-sequence encoding at the first lead time $\mathbf{V}_s^{(t+1)}$ as input for each station $s \in S_T \cup S_C$, i.e.,

$$\mathbf{H}_s, \mathbf{C}_s = \text{LSTM}_D(\mathbf{V}_s^{(t+1)}, \mathbf{H}_s, \mathbf{C}_s). \quad (8)$$

Note that, here, the encodings from multiple modalities are finally fused. After that, the hidden state encodings $\mathbf{H}_s$ are further refined through MHA, incorporating information from other stations, as follows:

$$\{\mathbf{H}_s\}_{s \in S_T \cup S_C} = \text{MHA}(\{\mathbf{H}_s\}_{s \in S_T \cup S_C}), \quad (9)$$

where $\{\mathbf{H}_s\}_{s \in S_T \cup S_C}$ serves as the key, query, and value input for MHA. Lastly, the updated hidden state encodings $\mathbf{H}$ and cell state encodings $\mathbf{C}$ are fed back into LSTM_D for the next lead time, and the above process is repeated with the image-sequence encodings $(\mathbf{V}^{(t+\Delta t)})_{\Delta t=2}^{T_{out}}$ serving as inputs for subsequent lead times, as illustrated in Figure 1.

²Note that architecture-specific constant terms, such as the number of encoder/decoder stacks, the number of translator modules, and the convolution-kernel sizes, are omitted for clarity.

³Here, auto-regressive decoding refers to the recursive propagation of the decoder's hidden and cell states across lead times, rather than the direct feedback of predicted $PM_{2.5}$ values.

Algorithm 4: Multimodal decoder DEC

Input: (1) Current time t , (2) Maximum lead time T_{out} , (3) Image-sequence encodings $\{\mathbf{V}^{(t+\Delta t)}\}_{\Delta t=1}^{T_{out}}$ at all stations, (4) Time series-based hidden state encodings $\mathbf{H}$, (5) Time series-based cell state encodings $\mathbf{C}$, (6) Overall scale coefficients $\{\varphi_s\}_{s \in S_T}$, (7) Fluctuation scale coefficients $\{\xi_s\}_{s \in S_T}$.

Output: Predicted $PM_{2.5}$ values $(\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T})_{\Delta t=1}^{T_{out}}$ at all stations in the target region

```

1 for  $\Delta t \leftarrow 1$  to  $T_{out}$  do
2   foreach  $s \in S_T \cup S_C$  do
3      $\mathbf{H}_s, \mathbf{C}_s \leftarrow \text{LSTM}_D(\mathbf{V}_s^{(t+\Delta t)}, \mathbf{H}_s, \mathbf{C}_s)$ 
4    $\{\mathbf{H}_s\}_{s \in S_T \cup S_C} \leftarrow \text{MHA}(\{\mathbf{H}_s\}_{s \in S_T \cup S_C})$ 
5   foreach  $s \in S_T$  do
6      $\hat{\mathbf{P}}_s^{(t+\Delta t)} \leftarrow (\mathbf{W}\mathbf{H}_s + \mathbf{b})(\sqrt{\xi_s^2 + \epsilon}) + \varphi_s$ 
7 return  $(\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T})_{\Delta t=1}^{T_{out}}$ 

```

This auto-regressive structure enhances prediction accuracy for longer lead times by capturing temporal dependencies. Thus, combined with multimodal fusion, it helps address Challenges C1 and C3 in Section I.

2) *Final prediction with learnable denormalization*: In the proposed decoder, for each lead time, the updated hidden state encoding $\mathbf{H}_s$ for each station s within the target region is fed into a linear layer to produce initial $PM_{2.5}$ predictions. This linear layer, parameterized by the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$, is shared across all stations and lead times.

These initial $PM_{2.5}$ predictions are denormalized to produce final predictions. As the inverse process of learnable normalization (see Section IV-A), denormalization here restores scale information, which was separately modeled during normalization. To this end, we leverage the overall scale coefficient φ_s and fluctuation scale coefficient ξ_s , which were derived for each station s during normalization.

Using both, for each lead time Δt and each station $s \in S_T$, the final prediction of $PM_{2.5}$ value is obtained as follows:

$$\hat{\mathbf{P}}_s^{(t+\Delta t)} \leftarrow (\mathbf{W}\mathbf{H}_s + \mathbf{b})(\sqrt{\xi_s^2 + \epsilon}) + \varphi_s, \quad (10)$$

where $\mathbf{H}_s$ is the updated hidden state encoding corresponding to time $t + \Delta t$. More details about the decoding process are provided in Algorithm 4.

Note again that learnable denormalization, along with learnable normalization (Section IV-A), enables separate modeling of scale information, allowing for scale-insensitive predictions. This enhances accuracy across different scales, including extreme $PM_{2.5}$ levels, which are often rare in training data.

Complexity analysis. The multimodal decoder operates over T_{out} auto-regressive steps. At each step, it performs (1) an LSTM with time complexity $O(|S| \cdot d_h^2)$, (2) a full-station self-attention with time complexity $O(|S|^2 \cdot d_h + |S| \cdot d_h^2)$, and (3) a linear projection followed by the denormalization over all target stations with time complexity $O(|S_T| \cdot d_h)$. Consequently, the total process of the multimodal decoder takes $O(T_{out}(|S|^2 \cdot d_h + |S| \cdot d_h^2))$ time.

Algorithm 5: Training process of MULTIAIR

Input: (1) Training time set $\mathbb{T}$, (2) Minimum time T_{min} ,
(3) Maximum time T_{max} ,
(4) Maximum number of training epochs E_{max} ,
(5) Training $PM_{2.5}$ data $\{\mathbf{P}^{(t')}\}_{t'=T_{min}}^{T_{max}}$,
(6) Training air quality data $\{\mathbf{O}^{(t')}\}_{t'=T_{min}}^{T_{max}}$,
(7) Training AOD images $\{\mathbf{I}^{(t')}\}_{t'=T_{min}}^{T_{max}}$,
(8) Learnable model parameters Φ (i.e., $\theta, \mathbf{W}, \mathbf{b}$, and parameters of $LSTM_E, LSTM_D, MHA$),
(9) Video-prediction model parameters ψ

Output: Trained model parameters Φ , video-prediction model parameters ψ

```

/* Stage 1: Pretraining video-prediction
model parameters  $\psi$  */
1 for epoch  $\leftarrow 1$  to  $E_{max}$  do
2  $\mathcal{L}(\psi) \leftarrow 0$ 
3 foreach  $t \in \mathbb{T}$  do
4  $(\hat{\mathbf{I}}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}} \leftarrow f_{\psi}([\mathbf{I}^{(t-T_i)} || \dots || \mathbf{I}^{(t)}])$ 
5 for  $\Delta t \leftarrow 1$  to  $T_{out}$  do
6  $\mathcal{L}(\psi) \leftarrow \mathcal{L}(\psi) + \mathcal{L}_{MSE}(\mathbf{I}^{(t+\Delta t)}, \hat{\mathbf{I}}^{(t+\Delta t)})$ 
7  $\psi \leftarrow \psi - \eta \nabla \mathcal{L}(\psi)$ 
8 Freeze  $\psi$  during the curriculum-learning stage;
/* Stage 2: Time-ordered curriculum
learning for model parameters  $\Phi$  */
9 for  $T_c \leftarrow 1$  to  $T_{out}$  do
10 for epoch  $\leftarrow 1$  to  $E_{max}$  do
11  $\mathcal{L}(\Phi) \leftarrow 0$ 
12 foreach  $t \in \mathbb{T}$  do
13  $(\{\hat{\mathbf{P}}_s^{(t+\Delta t)}\}_{s \in S_T})_{\Delta t=1}^{T_c} \leftarrow$ 
 $f_{\Phi, \psi}(t, T_c, (\mathbf{P}^{(t')})_{t'=t-T_t}^t, (\mathbf{O}^{(t')})_{t'=t-T_t}^t, (\mathbf{I}^{(t')})_{t'=t-T_i}^t)$ 
14 for  $\Delta t \leftarrow 1$  to  $T_c$  do
15 foreach  $s \in S_T$  do
16  $\mathcal{L}(\Phi) \leftarrow \mathcal{L}(\Phi) + \mathcal{L}_{focal}(\mathbf{P}_s^{(t+\Delta t)}, \hat{\mathbf{P}}_s^{(t+\Delta t)})$ 
17  $\Phi \leftarrow \Phi - \eta \nabla \mathcal{L}(\Phi)$ 
18 return  $\Phi, \psi$ 

```

D. Training Process with Curriculum Learning (Algorithm 5)

In this subsection, we present our novel training strategy for training learnable parameters in MULTIAIR, with a focus on (1) pre-training, (2) curriculum learning, and (3) error-adaptive loss. Note that the training dataset over the training time set $\mathbb{T}$ consists of (1) $PM_{2.5}$ time series data $\{\mathbf{P}^{(t')}\}_{t'=T_{min}}^{T_{max}}$, (2) other observation time series data $\{\mathbf{O}^{(t')}\}_{t'=T_{min}}^{T_{max}}$, and (3) training AOD image sequence data $\{\mathbf{I}^{(t')}\}_{t'=T_{min}}^{T_{max}}$, where $T_{min} = \min(\mathbb{T}) - \max(T_i, T_t)$ and $T_{max} = \max(\mathbb{T}) + T_{out}$.

1) *Pre-training of the video-prediction model:* The first training step is to pre-train the video-prediction model f_{ψ} , which forms the image-sequence encoder of MULTIAIR (Section IV-B2), to enhance its ability to capture spatiotemporal information. As the pretext task, we use the task of predicting future AOD images. Specifically, given the input image sequence $(\mathbf{I}^{(t')})_{t'=t-T_i}^t$ at time t , f_{ψ} predicts the future sequence $(\hat{\mathbf{I}}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$ from $t+1$ to $t+T_{out}$. Then, the mean squared error (MSE) between the ground-truth image

sequences $(\mathbf{I}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$ and the predicted image sequences $(\hat{\mathbf{I}}^{(t+\Delta t)})_{\Delta t=1}^{T_{out}}$ over all time steps t 's in the training set is used as the loss function, denoted as $\mathcal{L}_{MSE}$.

2) *Curriculum learning for longer lead times:* As highlighted in Challenge C3 in Section I, the difficulty of $PM_{2.5}$ prediction increases significantly with longer lead times. To address this, we progressively enhance the capability of MULTIAIR by first training it for short lead times and gradually increasing lead times, following the curriculum learning paradigm (see Section II-D). This approach helps improve the stability of MULTIAIR in predictions for longer lead times.

Specifically, after pretraining the video-prediction model, curriculum learning is applied to train the learnable parameters of MULTIAIR (e.g., $\theta, \mathbf{W}, \mathbf{b}$, parameters of $LSTM_E, LSTM_D, MHA$). At each curriculum stage T_c , which starts from 1 and increases up to T_{out} , MULTIAIR is trained to predict $PM_{2.5}$ values from time $t+1$ to $t+T_c$ at all stations in the target region at every time t in the training set, and to this end, a specialized loss function described in Section IV-D3 is employed. Once training for each stage T_c is completed, T_c is incremented by one, and the parameters of MULTIAIR are further updated to handle the increased difficulty of longer lead times in the same manner. The overall procedure is provided in Algorithm 5.

3) *Error-adaptive loss for extreme level prediction:* As stated in Challenge C2 in Section I, the rarity of extreme $PM_{2.5}$ levels in the training set makes their accurate prediction challenging. To address this, we design a weighted loss function that assigns larger weights to high-error cases, which include such rare cases, adapting focal loss and its variants discussed in Section II-C.⁴

Specifically, the loss function is defined for each station s , current time t , and lead time Δt based on the predicted and ground-truth $PM_{2.5}$ values, $\hat{\mathbf{P}}_s^{(t+\Delta t)}$ and $\mathbf{P}_s^{(t+\Delta t)}$, as follows:

$$\mathcal{L}_{focal}(\mathbf{P}_s^{(t+\Delta t)}, \hat{\mathbf{P}}_s^{(t+\Delta t)}) = (2\sigma((\beta|\mathbf{P}_s^{(t+\Delta t)} - \hat{\mathbf{P}}_s^{(t+\Delta t)}|)^{\gamma}) - 1)|\mathbf{P}_s^{(t+\Delta t)} - \hat{\mathbf{P}}_s^{(t+\Delta t)}|, \quad (11)$$

where σ is a sigmoid function; and β and γ are hyperparameters. Here, the term $2\sigma((\beta|\mathbf{P}_s^{(t+\Delta t)} - \hat{\mathbf{P}}_s^{(t+\Delta t)}|)^{\gamma}) - 1$ serves as the weight for each case (i.e., s, t , and Δt triple), and the weight increases as the prediction error $|\mathbf{P}_s^{(t+\Delta t)} - \hat{\mathbf{P}}_s^{(t+\Delta t)}|$ grows, which can be interpreted as difficulty. The hyperparameters β and γ control the scaling and sensitivity of the weight. The loss function in Eq. (11) is computed and aggregated across all stations and time steps during training for use in the curriculum learning process (Section IV-D2).

While the weight in Eq. (11) is not directly defined based on $PM_{2.5}$ levels, extreme $PM_{2.5}$ values are likely to receive larger weights due to the difficulty of predicting rare cases. Thus, the prediction of such extremes is improved.

V. EXPERIMENTS

In this section, we evaluate MULTIAIR to answer the following questions:

⁴Eq (11) is inspired by focal-R loss, but the weight is zero at zero residual and approaches one as the residual magnitude increases. The focal-R loss is defined as $\sigma((\beta|\mathbf{P}_s^{(t+\Delta t)} - \hat{\mathbf{P}}_s^{(t+\Delta t)}|)^{\gamma})|\mathbf{P}_s^{(t+\Delta t)} - \hat{\mathbf{P}}_s^{(t+\Delta t)}|$.

- **Q1. Effectiveness of MULTIAIR (Section V-B).** How accurate is MULTIAIR in nowcasting for different lead times and pollution levels?
- **Q2. Ablation Study (Section V-C).** How much does each component of MULTIAIR contribute to its performance?
- **Q3. Case study for Extreme Air Pollution (Section V-D).** How accurate is MULTIAIR in predicting extreme air pollution cases ($> 75\mu\text{g}/\text{m}^3$)?

A. Experimental Settings

Below, we describe our experimental settings. All experiments were conducted on a server with 512GB of RAM and eight RTX A6000 GPUs.

1) *Datasets and splits:* We use aerosol optical depth (AOD) images around South Korea collected hourly by GEO-KOMPSAT-2A for satellite data. The size of each AOD image is 900×900 , with $2\text{km} \times 2\text{km}$ resolution. The length T_i of the input image sequence (excluding one for the current time step) is 12, spanning 12 hours. Our target stations S_T are 537 ground-based observation stations in South Korea⁵ and auxiliary stations S_C are 583 stations in China. We use 6 ($= d_t$) air quality measures collected from the stations: $PM_{2.5}$, PM_{10} , O_3 , NO_2 , CO , and SO_2 . The length T_t of input time series data (excluding one for the current time step) is 6, spanning 6 hours. For all experiments, we randomly sample 90% of the dates from 2021 to 2022 for training and use the rest for validation. All data from 2023 is used for evaluation.

2) *Baseline methods:* We use eight baseline methods, including multimodal models [15], [16], advanced time series forecasting models [10], [11], [27], [28], and specialized air quality forecasting models [5], [14], categorized as follows:

- As a baseline method that uses both ground-based observations and satellite images, we adapt (a) ASOC+ [16], the state-of-the-art precipitation nowcasting method, for $PM_{2.5}$ prediction. Specifically, we replace radar images with AOD images, to leverage their correlation with $PM_{2.5}$. Additionally, we modify the model output from precipitation class probabilities to the absolute $PM_{2.5}$ values. We also use (b) CrossViViT-MS, which is an adapted version of CrossViViT [15] for multi-station forecasting.
- For baseline methods using only ground-based observations, we use (c) a vanilla LSTM model with an encoder-decoder structure [27], and (d) a vanilla Transformer model [28]. In addition, we use two state-of-the-art time series forecasting methods: (e) TimesNet [10], and (f) TimeMixer [11].
- Lastly, we use (g) AirFormer [5] and (h) GNN-LSTM [14], which are state-of-the-art $PM_{2.5}$ forecasting models based on transformers and graph neural networks, respectively.

For ASOC+, TimesNet, TimeMixer, and AirFormer, we use the open-source implementation. We equip ASOC+ with DeepRaNE [17] as in [16]. For the input of these baseline methods, we use the sequence of ground-based observations on air quality with positional encoding on location and time for prediction as described in IV-B. CrossViViT-MS and GNN-LSTM were implemented under our station-wise multivariate

⁵<https://www.airkorea.or.kr/web/>

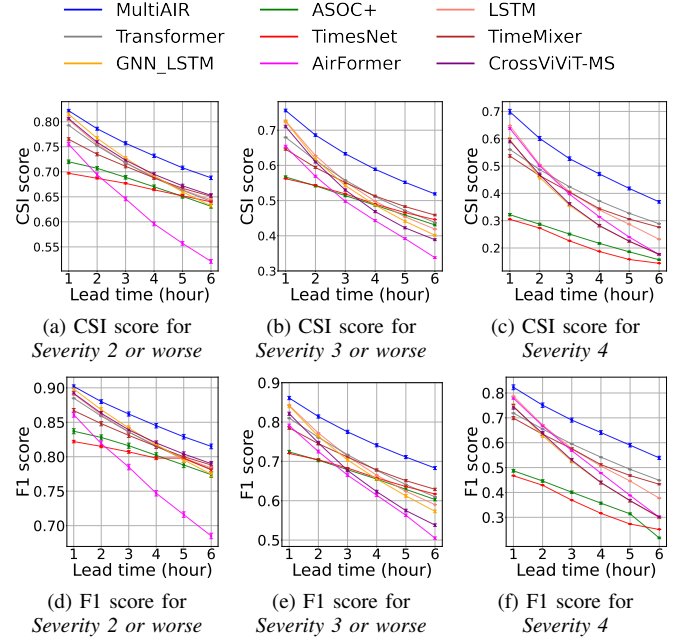


Fig. 2. Predictive performances on the test set evaluated using classification metrics (CSI and F1 scores, where higher values indicate better performance). MULTIAIR performs best across all pollution levels and lead times. Error bars indicate the standard error of the mean across test samples.

forecasting setting across multiple monitoring stations. Detailed experimental settings and hyperparameters are provided in Appendix X [26].

3) *Evaluation metrics:* For evaluation, we use both regression metrics and classification metrics, measured for each lead time. As regression metrics, we use root mean square error (RMSE) and mean absolute error (MAE). For classification metrics, according to [30], we divide $PM_{2.5}$ values P (in $\mu\text{g}/\text{m}^3$) into 4 classes: (1) *Severity 1* for $P \leq 15$, (2) *Severity 2* for $15 < P \leq 35$, (3) *Severity 3* for $35 < P \leq 75$, and (4) *Severity 4* for $75 < P$. As previously noted, the $PM_{2.5}$ values observed from stations in Korea are imbalanced across classes, with the following distribution: Severity 1 (53.5%), Severity 2 (36.5%), Severity 3 (9.1%), and Severity 4 (0.9%). Based on these classes, we define three cumulative classes, which are each (a) *Severity 2 or worse* ($P > 15$), (b) *Severity 3 or worse* ($P > 35$), and (c) *Severity 4* ($P > 75$), to evaluate severe air pollution detection, which is the main purpose of air pollution nowcasting.

The classification metrics, the critical success index (CSI) and F1 score, are defined for each cumulative class c as:

$$CSI_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad F1_c = \frac{2 \times TP_c}{2 \times TP_c + FP_c + FN_c},$$

where TP_c, FP_c, FN_c stand for the counts of true positives, false positives, and false negatives, respectively, for each cumulative class c . See Appendix XI [26] for other metrics.

B. Effectiveness of MULTIAIR

We demonstrate the effectiveness of MULTIAIR by comparing its performance against the baseline methods described in Section V-A2. As shown in Figure 2, where classification metrics are used, MULTIAIR outperforms all baseline methods

TABLE II
ABLATION STUDY RESULTS IN TERMS OF AVERAGE CSI SCORE (AVG. CSI), AND RMSE. MULTIAIR PERFORMS BEST ACROSS ALL LEAD TIMES. IN EACH SETTING, THE BEST AND THE SECOND-BEST PERFORMANCES ARE IN **BOLD** AND UNDERLINED.

Lead time	MULTIAIR		w/o Sat.		w/o Dec.		w/o LN		w/o Curr.		w/o HP		w/o HP+Under		w/o HP+Over	
	Avg. CSI	RMSE	Avg. CSI	RMSE	Avg. CSI	RMSE	Avg. CSI	RMSE	Avg. CSI	RMSE	Avg. CSI	RMSE	Avg. CSI	RMSE	Avg. CSI	RMSE
1 hr.	0.759	3.998	0.755	4.038	0.750	4.117	0.752	4.051	0.749	4.154	0.744	4.149	0.744	4.162	0.740	4.167
2 hr.	0.691	5.184	<u>0.685</u>	5.253	0.673	5.401	0.680	<u>5.207</u>	0.677	5.390	0.673	5.359	0.669	5.394	0.665	5.403
3 hr.	0.639	6.108	0.629	6.198	0.620	6.372	0.622	<u>6.138</u>	0.625	6.345	0.619	6.295	0.613	6.340	0.609	6.316
4 hr.	0.597	6.886	<u>0.584</u>	7.003	0.577	7.197	0.573	<u>6.949</u>	0.582	7.172	0.576	7.063	0.567	7.121	0.564	7.054
5 hr.	0.559	7.570	0.544	7.721	0.538	7.957	0.534	<u>7.626</u>	0.543	7.905	0.535	7.723	0.529	7.799	0.526	7.705
6 hr.	0.525	8.160	0.509	8.338	0.505	8.624	0.500	<u>8.201</u>	<u>0.510</u>	8.525	0.498	8.313	0.495	8.435	0.492	8.296

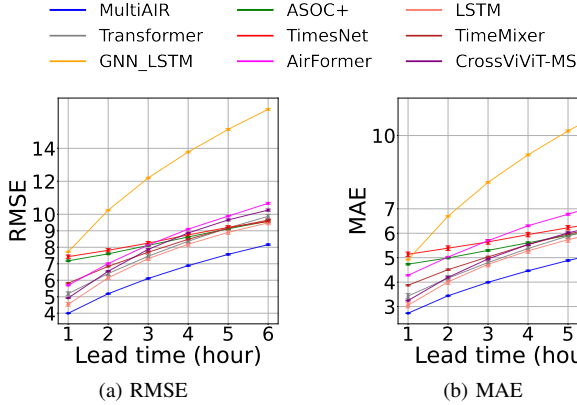


Fig. 3. Predictive performances on the test set evaluated using regression metrics (RMSE and MAE, where lower values indicate better performance). MULTIAIR performs best across all lead times. Error bars indicate the standard error of the mean across test samples. MultiAir also achieves more consistent performance: the station-wise standard deviation of its RMSE is 1.53, lower than those of all competing methods, which are at least 1.74.

across all pollution levels and lead times, achieving an average CSI score that is 13.3% higher than the strongest baseline. Notably, for *Severity 4* ($> 75\mu\text{g}/\text{m}^3$), MULTIAIR achieves, on average, 22.2% higher CSI and 15.1% higher F1 score, demonstrating its strong effectiveness in predicting high pollution levels. As shown in Figure 3, also in terms of regression metrics, MULTIAIR outperforms all baseline methods across all lead times. Specifically, it achieves an average RMSE reduction of 14.9% compared to the strongest baseline. In Appendix VII [26], we provide the detailed numerical values for the figures, and in Appendix XI [26], we show that MULTIAIR achieves the best performance across all additional metrics, including ACC, POD, and NME, while achieving the second-best performance in FAR.

In addition, we evaluate inference time on the test set. As shown in Table III, MULTIAIR takes longer than the baselines due to its image encoding process, but the per-sample inference time remains in the millisecond range. In practical nowcasting scenarios, such latency is negligible compared to the forecast horizon, which typically spans 1–6 hours.

C. Ablation Study

We conduct an ablation study to evaluate the contribution of each component of MULTIAIR. To this end, we consider **w/o Sat.**, which excludes satellite images; **w/o Dec.**, which replaces the proposed decoder with MLP-based satellite-image fusion; **w/o LN**, which removes learnable normalization and

TABLE III
AVERAGE INFERENCE TIME (MS) PER SAMPLE ON THE TEST SET. ALTHOUGH MULTIAIR INCURS ADDITIONAL COST DUE TO IMAGE ENCODING, SUCH LATENCY IS NEGLIGIBLE IN PRACTICAL NOWCASTING SCENARIOS FORECASTING 1–6 HOURS AHEAD.

Model	MULTIAIR	ASOC+	CrossViViT-MS	LSTM
Inference time (ms)	32.82	28.70	28.69	3.08
Transformer	TimesNet	TimeMixer	GNN-LSTM	Airformer
4.94	17.80	5.83	24.77	7.73

denormalization; and **w/o Curr.**, which removes curriculum learning. We also evaluate high-pollution handling variants: **w/o HP**, **w/o HP+Under**, and **w/o HP+Over**, which remove both focal-R loss and learnable normalization and use no sampling, undersampling, and oversampling, respectively. We follow the evaluation protocol in Section V-B, and provide full results with additional metrics in Appendix XI [26].

Table II reports the results in terms of Avg. CSI over the three cumulative pollution classes and RMSE. MULTIAIR achieves the best Avg. CSI and RMSE across all lead times, demonstrating the effectiveness of the full design. The degradation of **w/o Sat.** and **w/o Dec.** confirms the benefit of satellite imagery and the proposed multimodal decoder, while the lower performance of **w/o Curr.** supports the role of curriculum learning for multi-step nowcasting. Moreover, **w/o HP**, **w/o HP+Under**, and **w/o HP+Over** all underperform MULTIAIR, indicating that conventional sampling strategies are less effective than the proposed combination of focal-R loss and learnable normalization for high- $\text{PM}_{2.5}$ prediction.

In addition, we evaluate the sensitivity of MULTIAIR to key hyperparameters; refer to Appendix XII [26] for the results.

D. Case Study for Extreme Air Pollution

Figure 4 presents a case study of extreme air pollution events for *Severity 4*, visualizing predictions across lead times. MULTIAIR shows high prediction accuracy at a short lead time (2 hours) and, notably, continues to outperform competing models even at a longer lead time of 6 hours, accurately capturing extreme air pollution events.

VI. CONCLUSIONS

In this paper, we introduced MULTIAIR, a novel multimodal learning approach for $\text{PM}_{2.5}$ nowcasting. First, to address the challenge posed by the irregular spatial distribution of ground-based observations, MULTIAIR integrates satellite images as

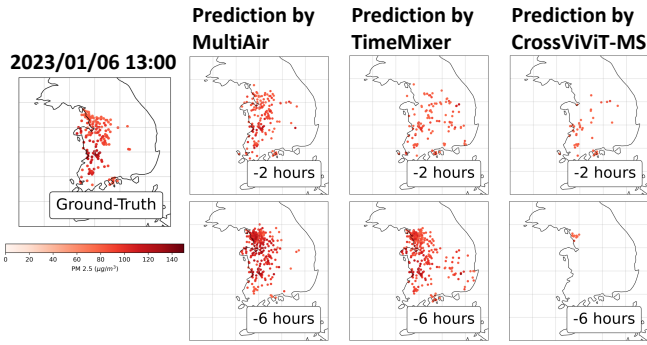


Fig. 4. Prediction results by MULTIAIR, TimeMixer, and CrossViViT-MS for extreme air pollution case (January 6, 2023 at 1 pm). For visibility, we only display observations and predictions of Severity 4 ($> 75\mu\text{g}/\text{m}^3$). The figure on the left shows the ground-truth $\text{PM}_{2.5}$ values, and the other images show the predicted $\text{PM}_{2.5}$ values at different lead times. MULTIAIR predicts the location and intensity of the high-pollution region more accurately than TimeMixer and CrossViViT-MS across lead times.

an external information source for data fusion. Second, to enhance effectiveness in extreme pollution cases, which are challenging to predict due to their rarity, MULTIAIR employs learnable normalization and a specialized loss function. Lastly, to alleviate the difficulty of predicting for a longer lead time, MULTIAIR combines curriculum learning with an autoregressive decoder structure.

Extensive experiments on three years of real-world data show that MULTIAIR consistently outperforms state-of-the-art methods across all lead times (up to 6 hours), achieving 13.3% higher CSI and 14.9% lower RMSE than the strongest baseline. Ablation and case studies further validate its design and effectiveness in predicting extreme pollution events. For reproducibility, our source code and datasets are available at <https://anonymous.4open.science/r/MultiAir-20C5>.

ACKNOWLEDGEMENTS

This study was conducted with the support of the Air Quality Forecasting Center at the National Institute of Environmental Research under the Ministry of Environment (NIER-2025-01-01-017 and NIER-2025-04-02-032). This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2025-02653113, High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale). This research was partly supported by the “Advanced GPU Utilization Support Program” funded by the Government of the Republic of Korea (Ministry of Science and ICT) (05-26-04-0100).

REFERENCES

- [1] R. D. Brook, S. Rajagopalan, C. A. Pope III, J. R. Brook, A. Bhatnagar, A. V. Diez-Roux, F. Holguin, Y. Hong, R. V. Luepker, M. A. Mittleman *et al.*, “Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the american heart association,” *Circulation*, vol. 121, no. 21, pp. 2331–2378, 2010.
- [2] V. A. Southerland, M. Brauer, A. Mohegh, M. S. Hammer, A. Van Donkelaar, R. V. Martin, J. S. Apte, and S. C. Anenberg, “Global urban temporal trends in fine particulate matter ($\text{pm}_{2.5}$) and attributable health burdens: estimates from global datasets,” *The Lancet Planetary Health*, vol. 6, no. 2, pp. e139–e146, 2022.
- [3] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, “Multi-hour and multi-site air quality index forecasting in beijing using cnn, lstm, cnn-lstm, and spatiotemporal clustering,” *Expert Systems with Applications*, vol. 169, p. 114513, 2021.
- [4] L. Chen, J. Xu, B. Wu, and J. Huang, “Group-aware graph neural network for nationwide city air quality forecasting,” *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 3, pp. 1–20, 2023.
- [5] Y. Liang, Y. Xia, S. Ke, Y. Wang, Q. Wen, J. Zhang, Y. Zheng, and R. Zimmermann, “Airformer: Predicting nationwide air quality in china with transformers,” in *AAAI*, 2023.
- [6] J. Liu, F. Weng, and Z. Li, “Ultrahigh-resolution (250m) regional surface $\text{pm}_{2.5}$ concentrations derived first from modis measurements,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2021.
- [7] X. Yan, Z. Zang, Y. Jiang, W. Shi, Y. Guo, D. Li, C. Zhao, and L. Hui, “A spatial-temporal interpretable deep learning model for improving interpretability and predictive accuracy of satellite-based $\text{pm}_{2.5}$,” *Environmental Pollution*, vol. 273, p. 116459, 2021.
- [8] Y. Lu, G. Giuliano, and R. Habre, “Estimating hourly $\text{pm}_{2.5}$ concentrations at the neighborhood scale using a low-cost air sensor network: A los angeles case study,” *Environmental Research*, vol. 195, p. 110653, 2021.
- [9] T. Li, H. Shen, Q. Yuan, X. Zhang, and L. Zhang, “Estimating ground-level $\text{pm}_{2.5}$ by fusing satellite and station observations: a geo-intelligent deep learning approach,” *Geophysical Research Letters*, vol. 44, no. 23, pp. 11–985, 2017.
- [10] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “Timesnet: Temporal 2d-variation modeling for general time series analysis,” in *ICLR*, 2023.
- [11] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. ZHOU, “Timemixer: Decomposable multiscale mixing for time series forecasting,” in *ICLR*, 2024.
- [12] H. Liu, Z. Duan, and C. Chen, “A hybrid multi-resolution multi-objective ensemble model and its application for forecasting of daily $\text{pm}_{2.5}$ concentrations,” *Inf. Sci.*, vol. 516, pp. 266–292, 2020.
- [13] X. Zhou, J. Wang, J. Wang, and Q. Guan, “Predicting air quality using a multi-scale spatiotemporal graph attention network,” *Inf. Sci.*, vol. 680, p. 121072, 2024.
- [14] M. Teng, S. Li, J. Xing, C. Fan, J. Yang, S. Wang, G. Song, Y. Ding, J. Dong, and S. Wang, “72-hour real-time forecasting of ambient $\text{pm}_{2.5}$ by hybrid graph deep neural network with aggregated neighborhood spatiotemporal information,” *Environment International*, vol. 176, p. 107971, 2023.
- [15] O. Boussif, G. Boukachab, D. Assouline, S. Massaroli, T. Yuan, L. Benabbou, and Y. Bengio, “Improving* day-ahead* solar irradiance time series forecasting by leveraging spatio-temporal context,” in *NeurIPS*, 2023.
- [16] J. Ko, K. Lee, H. Hwang, and K. Shin, “Deep-learning-based precipitation nowcasting with ground weather station data and radar data,” in *ICDM Workshop*, 2022.
- [17] J. Ko, K. Lee, H. Hwang, S.-G. Oh, S.-W. Son, and K. Shin, “Effective training strategies for deep-learning-based precipitation nowcasting and estimation,” *Computers & Geosciences*, vol. 161, p. 105072, 2022.
- [18] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, “Reversible instance normalization for accurate time-series forecasting against distribution shift,” in *ICLR*, 2022.
- [19] W. Fan, P. Wang, D. Wang, Y. Zhou, and Y. Fu, “Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting,” in *AAAI*, 2023.
- [20] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *CVPR*, 2019.
- [21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in *ICCV*, 2017.
- [22] Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Katabi, “Delving into deep imbalanced regression,” in *ICML*, 2021.
- [23] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *ICML*, 2009.
- [24] J. Li, S. Zhou, L. Li, H. Wang, J. Bu, and Z. Yu, “Dynamic data-free knowledge distillation by easy-to-hard learning strategy,” *Inf. Sci.*, vol. 642, p. 119202, 2023.
- [25] Y. Zhou, X. Wang, H. Chen, X. Duan, and W. Zhu, “Intra-and inter-modal curriculum for multimodal learning,” in *MM*, 2023.
- [26] S. Kim, J. Chun, J. Lee, S.-C. Hong, J.-B. Lee, S.-H. Eun, Y. Lee, and K. Shin, “Air pollution nowcasting via multimodal learning

- with temporal and visual data: Code, datasets, and appendix,” 2026.
[Online]. Available: <https://anonymous.4open.science/r/MultiAir-20C5>
- [27] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
 - [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.
 - [29] Z. Gao, C. Tan, L. Wu, and S. Z. Li, “Simvp: Simpler yet better video prediction,” in *CVPR*, 2022.
 - [30] C.-H. Ho, I. Park, J. Kim, and J.-B. Lee, “Pm2. 5 forecast in korea using the long short-term memory (lstm) model,” *Asia-Pacific Journal of Atmospheric Sciences*, vol. 59, no. 5, pp. 563–576, 2023.