AeroMosaic: Transport-Aware Spatiotemporal Multimodal Learning for Atmospheric Pollution Risk

Muyang Zheng*
Hunan University
Changsha, China
miyazmy@hnu.edu.cn

Jiaming Ma*
University of Science and Technology of China
Hefei, China
JiamingMa@mail.ustc.edu.cn

Zongyu Zhang
Hunan University
Changsha, China
zzy187@hnu.edu.cn

Qingsong Wen†
Squirrel Ai Learning
Bellevue, USA
qingsongedu@gmail.com

Abstract

Atmospheric pollution risk inference serves as a core enabling technology for air-quality warning and environmental management. Unlike generic time-series extrapolation, target-day risk depends jointly on the local pollutant state, the exogenous meteorological regime, and directed cross-city transport. Existing time-series models are strong sequence learners but treat these signals as one homogeneous tensor, graph-based air-quality models typically rely on static or weakly directed spatial priors, and multimodal methods often merge heterogeneous inputs without assigning it a functional role. In this paper, we propose AeroMosaic, a transport-aware spatiotemporal multimodal learning framework that routes the numerical state, compact precipitation text, and wind-aligned graph structure through distinct functional paths. AeroMosaic encodes the historical numerical state as prefix tokens of a language-model backbone, summarizes recent precipitation as prompt-side context, constructs a daily wind-driven directed acyclic graph (DAG) to organize upstream-to-downstream evidence, and refines city embeddings through post-pooling DAG residual propagation. On six $PM_{2.5}$ risk-inference benchmarks with a year-based split and thirteen baselines, AeroMosaic achieves the best Macro-F1 and Micro-F1 in all six regions under the oracle parent-label protocol, which supplies target-date labels of upstream cities. Matched no-oracle protocols further show that the parent-information interface also works when target-day labels are replaced by historical upstream observations.

1 Introduction

Airborne fine-particulate matter (such as $PM_{2.5}$) is one of the leading environmental risk factors for global mortality and morbidity [1–4]. Accurate air-quality prediction therefore serves as a core enabling technology for public-health warnings, emission-control interventions, and environmental management [5, 6]. For decision-oriented management, the relevant output is often not a raw concentration trajectory but a risk category that triggers warnings or interventions. We refer to this task as atmospheric pollution risk inference: given the recent history of a region, the model infers the pollution risk class of every city on a target date.

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To this end, researchers have developed a series of data-driven methods. One line of work treats the task as multivariate time-series forecasting and leverages the strong capability of linear, decomposition-based, and Transformer architectures to model temporal dependencies [7–15]; recent studies further capture periodic patterns, period heterogeneity, and long-horizon evolution [16–18]. A second line models spatiotemporal dependencies across locations with spatiotemporal graph neural networks [19–27], and air-quality-specific models incorporate domain structure such as meteorology, physical diffusion, and causal covariates [28–37]. A third line adapts pretrained language models and time-series foundation models to temporal data through reprogramming, prompting, or large-scale pretraining [38–47], and extends them to urban spatiotemporal prediction [48–50].

Despite the substantial progress, three limitations remain for atmospheric pollution risk inference. (i) **Homogeneous inputs.** The informative signal is distributed across the local pollutant history, external meteorological forcing, and upstream transport, and these signals play different functional roles [6, 51]. Classical time-series and causal analyses distinguish autoregressive state, exogenous forcing, and directional predictive dependence [52–54], whereas most existing models stack all signals into one homogeneous input tensor, which blurs the mechanism that determines target-day risk. (ii) **Static spatial structure.** Cross-city influence is directional because pollutants are transported along the wind field, and the transport direction changes from day to day. Most spatiotemporal and air-quality models rely on static distance graphs, partition-based structures, or learned connectivity shared across dates, which do not explicitly represent day-varying upstream-to-downstream transport. (iii) **Undifferentiated multimodal fusion.** Language-model interfaces provide a flexible way to combine numerical and textual inputs [39, 55], but they typically either serialize numerical values into long prompts or merge all inputs into a single sequence, without assigning numerical state, semantic context, and transport structure to distinct paths. Therefore, it is crucial to keep heterogeneous atmospheric signals role-separated and fuse them under a day-varying directed transport prior.

In this paper, we propose `AeroMosaic`, a transport-aware spatiotemporal multimodal learning framework for atmospheric pollution risk inference that assembles heterogeneous atmospheric signals into a structured mosaic. Specifically, `AeroMosaic` keeps the historical numerical state of each city in numerical form through learnable prefix tokens of a language-model backbone, so that numerical values are not serialized into long text. Second, recent precipitation is summarized by deterministic rules into compact prompt text, which provides exogenous environmental context. Third, an offline builder constructs a daily wind-driven directed acyclic graph (DAG) from geographic candidates and the daily wind field; the DAG determines the upstream context in the prompt and drives a post-pooling residual block that propagates city embeddings from upstream to downstream cities. Furthermore, root-sensitive modules compensate for cities without upstream parents. Each modality thus has a defined role before fusion. We evaluate `AeroMosaic` under an oracle parent-label protocol, in which the prompt of each non-root city contains the target-date labels of its DAG parents, and complement it with oracle-access and matched no-oracle analyses. Our contributions are as follows:

- **Daily wind-driven transport prior.** To capture day-varying directional transport that static spatial graphs miss, we construct a daily wind-driven DAG from geographic candidates and the daily wind field. The DAG determines the upstream context of each city and drives post-pooling residual propagation from upstream to downstream cities.
- **Role-separated multimodal learning.** To avoid stacking heterogeneous signals into one homogeneous tensor, we route the numerical state through prefix tokens of a language-model backbone and recent precipitation through compact prompt text, and fuse them with transport structure after city-level pooling. Upstream information enters through a dedicated prompt field, so the same model operates under the oracle parent-label protocol and under no-oracle protocols.
- **Empirical validation.** On six regional benchmarks against thirteen time-series and air-quality baselines, `AeroMosaic` achieves the best Macro-F1 and Micro-F1 in every region, with an average relative improvement of 23.7% over the strongest baseline. Removing the DAG residual or LoRA lowers the medium-to-heavy-class recall by 36.5% and 54.2%.

2 Related Work

Time-series forecasting. Transformer-based, linear, and multiscale models capture long-range dependencies, trend, seasonal, and periodic structure in multivariate time series [7–14, 56], and recent

models further exploit periodic patterns and period heterogeneity [16–18]. These models operate on homogeneous numerical tensors without distinct roles for numerical, textual, and graph inputs.

Spatiotemporal and air-quality forecasting. Spatiotemporal graph neural networks model dependencies across locations with predefined or adaptive graphs [19–23, 57], and lightweight designs improve efficiency and robustness [26, 27, 58]. Air-quality models introduce fixed distance graphs, group structures, physical diffusion, or causal covariates [28, 30, 32–37, 59–61]. Their spatial structure is shared across dates, whereas `AeroMosaic` rebuilds a directed transport graph daily.

Language models for time series. Pretrained language models and time-series foundation models are adapted to temporal data through reprogramming, prompting, tokenization, or large-scale pretraining [38, 39, 41–44, 46, 55], and are extended to urban spatiotemporal prediction [48–50]. `AeroMosaic` builds on this interface but separates numerically preserved state, compact semantic context, and explicit transport structure into distinct paths that are fused under a daily directed prior.

3 Method

3.1 Problem Formulation and Overview

We formulate atmospheric pollution risk inference as city-level classification over a region. For each target date t, `AeroMosaic` processes all N cities of a region jointly as one region-date group. For city i, the input contains a historical numerical tensor $\mathbf{X}^{(i)} \in \mathbb{R}^{L\times D}$ over L historical steps with $D = d_p + d_m + d_t$ pollutant, meteorological, and time channels, prompt ingredients (region, city, date, recent precipitation, and DAG-derived upstream context), and the daily DAG $\mathbf{A}^{(t)}$ shared by all cities of the group. The model outputs the $PM_{2.5}$ risk class $y_i \in \{0, \dots, C-1\}$ of every city on date t; the future $PM_{2.5}$ window of the target city is used only to derive y_i.

The core design of `AeroMosaic` is to assign each type of atmospheric evidence to the path that matches its role. As shown in Figure 1, the local pollutant state enters as numerical prefix tokens, recent precipitation enters as compact prompt text, and transport structure enters through a daily wind-driven DAG that both selects the upstream context of the prompt and propagates city embeddings after pooling. The following subsections describe the DAG construction (Section 3.2), the role-separated encoder and transport refinement (Section 3.3), the training objective (Section 3.4), and the resulting information flow (Section 3.5). Notation is summarized in Table 3 (Appendix A.1), and Algorithm 1 (Appendix A.2) gives the complete forward pass.

3.2 Daily Wind-Driven DAG Construction

Cross-city pollution transport is directional and changes with the daily wind field, so a static distance graph cannot express which city is upstream of which on a given day. `AeroMosaic` therefore builds a new directed graph for every target date. An offline graph builder constructs a daily DAG $\mathbf{A}^{(t)} \in \mathbb{R}^{N\times N}_{\geq 0}$ in destination-by-source form, where $\mathbf{A}^{(t)}_{ji} > 0$ denotes an edge $i \to j$.

The builder first selects geographically plausible candidate pairs $\mathcal{E}_{\text{cand}}$ from city coordinates, with optional terrain filtering. For a candidate edge $i \to j$ with estimated target-day wind vector $\mathbf{w}_i^{(t)}$ at the source city and geographic displacement $\mathbf{d}_{i\to j}$, it measures how well the wind carries air from i towards j and scores the edge as

$$\cos\theta_t(i,j) = \frac{\mathbf{w}_i^{(t)} \cdot \mathbf{d}_{i\to j}}{\|\mathbf{w}_i^{(t)}\|\,\|\mathbf{d}_{i\to j}\|}, \qquad s_{i\to j}^{(t)} = \frac{\max(\cos\theta_t(i,j), 0)\,\|\mathbf{w}_i^{(t)}\|}{\max(\text{dist}(i,j), \epsilon)}. \tag{1}$$

The score increases with directional alignment and wind strength and decreases with distance, which reflects that strong, well-aligned wind over a short distance is the most plausible transport channel. Edges are retained when the distance, wind magnitude, and $\theta_t(i,j)$ pass the thresholds of the dataset, and the parent set of each destination is capped by $s_{i\to j}^{(t)}$ when a cap is set, which keeps the prompt-side upstream context compact. Finally, cycles are removed by deleting selected cycle-internal edges, so that every city has well-defined upstream ancestors and a topological order.

The builder outputs $\mathbf{A}^{(t)}$, its row-normalized form $\hat{\mathbf{A}}^{(t)} = \text{RowNorm}(\mathbf{A}^{(t)})$, root indicators r_i, parent sets $\mathcal{P}_i^{(t)}$, and DAG levels $\ell_i^{(t)}$. Rows with zero in-degree remain all-zero, so root cities, which

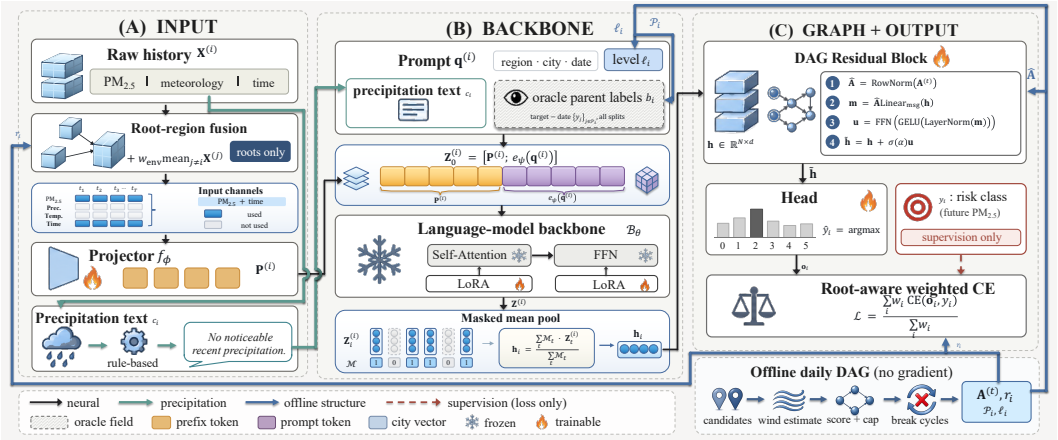


Figure 1: Framework of AeroMosaic. Solid black arrows denote the neural computation, teal arrows denote deterministic precipitation context derived from the historical input, blue arrows denote structure derived offline from coordinates and the daily wind field, and dashed red arrows denote supervision used only by the loss. Parent-label prompt context provides target-date upstream labels under the oracle parent-label protocol and is replaced or removed under the no-oracle protocols. No future $PM_{2.5}$ observation of the target city enters the model input.

have no upstream parent on that day, receive no incoming message. Each output has a distinct role in the model: $\hat{\mathbf{A}}^{(t)}$ drives residual message passing, $\mathcal{P}_i^{(t)}$ determines the upstream context inserted into the prompt, r_i activates root-region fusion and root-aware loss weighting, and $\ell_i^{(t)}$ enters the prompt as structural context. The builder has no trainable parameters; the residual block learns the message transformation and its strength. Appendix A.3 details the construction and the per-dataset settings.

3.3 Role-Separated Encoding and Transport Refinement

The encoder realizes the role separation in three stages: it encodes the numerical state and the prompt context of each city into one city embedding, and then refines all city embeddings of the group along the daily DAG. We omit the superscript t when the target date is clear.

Numerical prefix tokens. Serializing a multivariate history into text produces long prompts and discards the precision of numerical values. AeroMosaic instead maps the numerical history into a small number of continuous tokens that are placed before the prompt. Because root cities have no incoming DAG messages, they lack the regional context that non-root cities later receive from their parents; AeroMosaic compensates for this by a bounded root-region fusion that adds the regional mean history to the input of each root city. A numerical projector f_ϕ (LayerNorm, linear layer, SiLU, and linear layer) then maps the flattened $PM_{2.5}$ and time channels of the fused history into K prefix tokens:

$$\mathbf{X}_{\text{pre}}^{(i)} = \mathbf{X}^{(i)} + r_i w_{\text{env}} \frac{1}{N-1} \sum_{j\neq i} \mathbf{X}^{(j)}, \qquad \mathbf{P}^{(i)} = f_\phi\big(\text{vec}(\mathbf{X}_{\text{pre}}^{(i)})\big) \in \mathbb{R}^{K\times d}, \tag{2}$$

where $r_i = 1$ for root cities and $r_i = 0$ otherwise, w_{env} controls the strength of the fusion, and d is the hidden size of the backbone. The prefix tokens keep the pollutant state in numerical form within a sequence of only K positions, while meteorological information enters through the precipitation text and the wind-driven DAG. Appendix A.4 describes the root-sensitive modules.

Prompt context and city embedding. Recent precipitation acts as an episodic washout signal whose timing matters more than its exact value, and upstream conditions are naturally expressed as categorical labels of named cities. AeroMosaic therefore provides both as compact text. The prompt $\mathbf{q}^{(i)} = \text{Prompt}(g, i, t, \ell_i, \mathbf{b}_i, \mathbf{c}_i)$ contains the region identifier g, city, date, DAG level ℓ_i, the parent-label block $\mathbf{b}_i$, and the recent-precipitation summary $\mathbf{c}_i$. The summary is generated by fixed rules from the historical window: it states the overall precipitation category and the categories of the early, middle, and late segments of the window. Root cities receive a no-parent marker in $\mathbf{b}_i$, and non-root cities receive the labels of their parents in $\mathcal{P}_i$ under the oracle parent-label protocol. With

token embedding map e_ψ and language-model backbone $\mathcal{B}_\theta$ [55] adapted by LoRA [62], the city embedding is

$$\mathbf{Z}_0^{(i)} = [\mathbf{P}^{(i)}; e_\psi(\mathbf{q}^{(i)})], \qquad \mathbf{Z}^{(i)} = \mathcal{B}_\theta(\mathbf{Z}_0^{(i)}), \qquad \mathbf{h}_i = \mathrm{Pool}(\mathbf{Z}^{(i)}), \tag{3}$$

where $[\cdot;\cdot]$ denotes token-axis concatenation and Pool is masked mean pooling over valid positions. Inside $\mathcal{B}_\theta$, self-attention lets the text tokens attend to the numerical prefix, so the numerical state is interpreted in the context of the precipitation episode and the upstream labels. Pooling then separates this within-city fusion from the cross-city fusion that follows. Prompt examples, the precipitation-text procedure, and an input visualization are given in Appendices A.7, A.8, and A.6.

Post-pooling DAG residual. The prompt conveys upstream labels, but it does not let a city access the learned representations of its upstream neighbors. `AeroMosaic` therefore propagates city embeddings along the daily DAG after pooling. Stacking the city embeddings gives $\mathbf{h} \in \mathbb{R}^{N\times d}$, which is refined as

$$\mathbf{m} = \hat{\mathbf{A}}\,\mathrm{Linear}_{\mathrm{msg}}(\mathbf{h}), \quad \mathbf{u} = \mathrm{FFN}\big(\mathrm{GELU}(\mathrm{LayerNorm}(\mathbf{m}))\big), \quad \tilde{\mathbf{h}} = \mathbf{h} + \sigma(\alpha)\,\mathbf{u}. \tag{4}$$

Because $\hat{\mathbf{A}}$ is destination-by-source and row-normalized, $\mathbf{m}_i$ is the weighted average of the transformed embeddings of the parents of city i, and the update $\mathbf{u}_i$ of each root city is set to zero. The learnable gate $\sigma(\alpha)$ is initialized at $\sigma(-3) \approx 0.05$, so training starts close to the backbone embedding and gradually increases the contribution of upstream messages. Operating on pooled embeddings keeps the graph step independent of the token length. The output head yields the class logits $\mathbf{o}_i = \mathrm{Head}(\tilde{\mathbf{h}}_i) \in \mathbb{R}^C$ and the prediction $\hat{y}_i = \arg\max_c \mathbf{o}_{i,c}$.

3.4 Training Objective

Root cities are structurally harder because they receive no upstream messages, and heavy-pollution classes are rare but decision-critical. `AeroMosaic` therefore trains each region-date group with a bounded root-aware weighted cross-entropy:

$$\mathcal{L} = \frac{\sum_{i=1}^N w_i\,\mathrm{CE}(\mathbf{o}_i, y_i)}{\sum_{i=1}^N w_i}, \qquad w_i = \min\big((1 + (r_{\mathrm{base}} - 1)r_i)\,e_{y_i}^{r_i},\, w_{\max}\big), \tag{5}$$

where $r_{\mathrm{base}} > 1$ up-weights root cities, $e_y \geq 1$ further up-weights root cities whose label belongs to a heavy-pollution class, and $w_{\max}$ bounds the weight for stable optimization. Normalizing by $\sum_i w_i$ keeps the loss scale comparable across groups with different numbers of roots. We optimize $\Theta = \{\phi, \theta_{\mathrm{adp}}, \omega_{\mathrm{msg}}, \omega_{\mathrm{ffn}}, \alpha, \omega_{\mathrm{head}}\}$, where ϕ parameterizes the numerical projector, θ_{adp} denotes the LoRA adapter parameters inside $\mathcal{B}_\theta$, and the remaining parameters belong to the DAG residual block and the prediction head; the pretrained backbone weights stay frozen. For model selection, checkpoints whose validation Macro-F1 is within 0.003 of the best value observed so far are ranked by the validation Macro-F1 on root cities. The prompt fields are instantiated deterministically from stored examples in all splits. Table 7 (Appendix B.1) lists the hyperparameters.

3.5 Information Flow

The role separation of `AeroMosaic` is reflected in where each signal enters the computation and where the signals are fused. The numerical path evolves as $\mathbf{X}^{(i)} \to \mathbf{X}_{\mathrm{pre}}^{(i)} \to \mathbf{P}^{(i)}$, the prompt path contributes $e_\psi(\mathbf{q}^{(i)})$, and the graph path contributes $\hat{\mathbf{A}}^{(t)}$. These paths meet at two fusion points. The first is token-level fusion inside $\mathcal{B}_\theta$, where the numerical prefix and the prompt tokens of one city interact through self-attention and are pooled into $\mathbf{h}_i$. The second is city-level fusion in the DAG residual block, where $\hat{\mathbf{A}}^{(t)}$ combines the embeddings of different cities into $\tilde{\mathbf{h}}_i$.

The outputs of the graph builder enter at different points. The parent sets $\mathcal{P}_i^{(t)}$ affect only the prompt, through the parent-label block $\mathbf{b}_i$; the normalized adjacency $\hat{\mathbf{A}}^{(t)}$ affects only residual message passing; and the root indicators r_i affect root-region fusion, the zero update of root cities, and the loss weights, but never the labels y_i. The target-date parent labels in $\mathbf{b}_i$ are the only oracle information, and they enter exclusively through this prompt field (Figure 5, Appendix A.7); replacing this field with no parent context or with historical upstream observations yields the no-oracle protocols of Section 4.4 without changing any other part of the model. Training, validation, and test share the same forward map, and only training compares the logits $\mathbf{o}_i$ with the labels y_i to update Θ. Appendix A.5 (Table 6) lists the source window, model visibility, and role of every signal.

3.6 Branch-Wise Approximation View

The role-separated design admits a branch-wise approximation view. Let the class-score vector of city i be generated by an unknown compositional mechanism $\mathbf{s}_i^\star = h^\star(\mathbf{z}_{i,\mathrm{state}}^\star, \mathbf{z}_{i,\mathrm{ctx}}^\star, \mathbf{z}_{i,\mathrm{transport}}^\star)$ of local numerical state, exogenous environmental context, and transport-aware upstream evidence. If the learned branches satisfy $\|\hat{\mathbf{z}}_{i,k} - \mathbf{z}_{i,k}^\star\| \le \varepsilon_k$ for $k \in \{\mathrm{state}, \mathrm{ctx}, \mathrm{transport}\}$, the fusion map is L_h-Lipschitz, and the learned head contributes an estimation error $\varepsilon_{\mathrm{fit}}$, then

$$\|\hat{\mathbf{s}}_i - \mathbf{s}_i^\star\| \le L_h(\varepsilon_{\mathrm{state}} + \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{transport}}) + \varepsilon_{\mathrm{fit}}, \tag{6}$$

and the predicted class is unchanged whenever the margin $m_i^\star = s_{i,y_i}^\star - \max_{c \neq y_i} s_{i,c}^\star$ exceeds twice the right-hand side. This decomposition motivates the design: prefix tokens target the state term, precipitation text the context term, and the wind-driven DAG the transport term.

4 Experiments

We conduct experiments to answer the following research questions. **RQ1:** Does `AeroMosaic` outperform time-series and air-quality baselines across regions (Section 4.3)? **RQ2:** Is the improvement attributable solely to access to target-day parent labels (Section 4.4)? **RQ3:** How do the transport-aware refinement, backbone adaptation, and input modalities contribute (Section 4.5)? **RQ4:** How does `AeroMosaic` organize transport evidence in a concrete pollution episode (Section 4.6)?

4.1 Datasets

We analyze six regional $PM_{2.5}$ risk-inference datasets derived from the processed KnowAir release introduced with $PM_{2.5}$-GNN [30]. KnowAir combines $PM_{2.5}$ observations from air-quality monitoring stations operated by China's Ministry of Ecology and Environment with meteorological variables derived from the ECMWF ERA5 reanalysis [63]. City coordinates are computed by averaging the coordinates of the monitoring stations associated with each city. The resulting data cover 184 cities from January 2015 to December 2018; we partition them into BTH, YRD, SDP, ZHD, CJD, and CYB. Each target date defines one city group, and each city-day receives a six-class label derived from the target-date future window. All regions use 2015–2016 for training, 2017 for validation, and 2018 for testing, so that evaluation follows a chronological split in the spirit of out-of-distribution spatiotemporal evaluation [58]. The concrete setting is $L = 8$, $d_p = 1$, $d_m = 10$, $d_t = 2$, and $D = 13$; dataset statistics are reported in Table 1.

The optional terrain filter uses the quantized elevation raster distributed with the public $PM_{2.5}$-GNN implementation; because that release does not document the upstream elevation product or its license, we cite the implementation itself. We do not redistribute the processed KnowAir data, the monitoring-station metadata, or the terrain raster, which should be obtained from the cited source under the original providers' terms. ERA5 is available from the Copernicus Climate Data Store.

Our implementation, validation utilities, and reproduction instructions are available at `https://github.com/miya1102/aeromosaic_release_261018`. The repository includes a bundle for the BTH no-oracle setting; reproducing the six-region main results requires the archived regional bundles and the external ChatTS checkpoint described in Appendix B.1.

Table 1: Statistics of the six regional $PM_{2.5}$ risk-inference datasets. Each sample corresponds to one city-day prediction instance. Labels L0–L5 denote increasing $PM_{2.5}$ pollution categories, and proportions are computed over the combined train/validation/test splits.

ID	Full Name	# Cities	Samples			Label Distribution (%)					
			Train	Val	Test	L0	L1	L2	L3	L4	L5
BTH	Beijing–Tianjin–Hebei	13	9,490	4,745	4,745	31.7	39.5	15.6	5.8	5.6	1.8
YRD	Yangtze River Delta	16	11,680	5,840	5,840	40.8	44.0	10.9	3.0	1.3	0.0
SDP	Shandong Peninsula	15	10,950	5,475	5,475	28.6	43.3	16.7	6.0	4.6	0.7
ZHD	Central Plains–Huaihe	17	12,410	6,205	6,205	22.5	46.2	17.3	6.8	5.9	1.3
CJD	Middle Yangtze	21	15,330	7,665	7,665	40.1	43.6	11.5	3.0	1.6	0.1
CYB	Chengdu–Chongqing Basin	17	12,410	6,205	6,205	44.4	39.1	11.4	3.4	1.7	0.1

4.2 Evaluation Setup and Baselines

Each training batch is one region-date group, and all cities in the group are processed jointly with the group DAG. Model selection uses validation Macro-F1 with root-city Macro-F1 as a secondary criterion (Section 3.4). Backbone, training, and protocol details are given in Appendices B.1 and A.5. We compare AeroMosaic with thirteen representative time-series and air-quality models: DLinear [10], CATS [64], CrossGNN [65], DSFormer [66], SOFTS [67], SparseTSF [68], TimeMixer [14], TimeXer [15], UMixer [69], AirFormer [32], AirPhyNet [34], GAGNN [33], and MGSFormer [35]. All methods share the same train/validation/test split and raw data source, and the baselines receive all 13 numerical channels. Time-series baselines model inter-city correlations through the city axis, and air-quality baselines use static, partition-based, or flow-weighted spatial priors. AeroMosaic additionally uses compact precipitation text, a daily wind-driven DAG, and parent-label prompt context under the oracle parent-label protocol; Appendix B.2 summarizes these differences and discusses further air-quality models [30, 36, 37, 61].

4.3 Main Results

To answer RQ1, Table 2 compares AeroMosaic with thirteen baselines on the six regions. AeroMosaic achieves the best Macro-F1 and Micro-F1 in every region, with an average relative improvement of 23.7% over the strongest baseline (26.3% in Macro-F1 and 21.0% in Micro-F1). Averaged over the six regions, AeroMosaic reaches 0.637 Macro-F1 and 0.822 Micro-F1.

The baseline results reveal the importance of spatial structure. Air-quality models with explicit spatial priors obtain an average Macro-F1 of 0.436 across regions, compared with 0.294 for generic time-series models, and they provide the strongest baseline Macro-F1 in five of the six regions (MGSFormer on BTH and CJD, GAGNN on YRD, and AirPhyNet on SDP and CYB). Time-series models such as TimeMixer and TimeXer remain competitive in Micro-F1 on several regions, but their Macro-F1 is much lower, which indicates that they mainly predict the frequent clean and lightly polluted classes; DLinear, which treats each channel independently, obtains a Macro-F1 between 0.19 and 0.27. AeroMosaic narrows this gap between the two metrics: its Macro-F1 is 77% of its Micro-F1 on average, compared with 68% for the air-quality models and 53% for the time-series models, so its improvement comes largely from the less frequent, higher-risk classes. The per-class results in Figure 9 confirm this: on BTH, the precision of AeroMosaic on lightly and moderately polluted days is about 0.74 and 0.59, versus at most 0.40 and 0.24 for the baselines.

The size of the improvement follows the role of cross-city transport in each region. The largest improvements appear on SDP and BTH, where AeroMosaic improves Macro-F1 by 40.2% over AirPhyNet and 37.8% over MGSFormer, respectively. On BTH, SDP, and ZHD, where medium-to-heavy pollution days account for 27.0%–30.0% of the samples (Table 1), the average relative Macro-F1 improvement is 33.7%, compared with 19.0% on YRD, CJD, and CYB. Pollution episodes in these regions are shaped by regional transport, which the daily wind-driven DAG and the post-pooling residual propagation model explicitly, whereas the baselines rely on homogeneous input tensors or static spatial priors that are shared across dates. On CJD, MGSFormer and GAGNN are strong baselines with Macro-F1 0.530 and Micro-F1 0.778, respectively, and AeroMosaic still improves them by 3.9% and 11.5%. The strongest baseline varies across regions, while AeroMosaic consistently ranks first, which shows that the design generalizes across regional pollution regimes.

4.4 Oracle and No-Oracle Analysis

To answer RQ2, we conduct two BTH analyses of the parent-information interface. First, we provide MGSFormer with the same target-day parent labels as numerical covariates. This raises its Micro-F1/Macro-F1 from $0.6561/0.5518$ to $0.6894/0.5732$, while AeroMosaic obtains $0.8387/0.7239$; target-day parent information is therefore not the sole source of the improvement. Second, because the parent labels enter through a dedicated prompt field, the interface can be instantiated without target-day labels. We evaluate AeroMosaic, MGSFormer, and TimeMixer under three matched no-oracle protocols: no parent context, model-predicted parent classes, and the last observed parent-city $PM_{2.5}$ in the input window. With historical upstream observations, AeroMosaic attains $0.5825/0.3462$ with the same DAG and architecture. Appendix B.4 reports the full results and inference times.

Table 2: Macro-F1 and Micro-F1 on six $PM_{2.5}$ risk-inference benchmarks (mean $\pm$ standard deviation over three random seeds). `AeroMosaic` is evaluated under the oracle parent-label protocol. The best results are in red and the second-best results are underlined in blue; "N/A" marks non-convergence.

	Datasets	BTH		YRD		SDP	
	Methods	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Time Series	CATS [64]	0.2410 ± 0.0116	0.4891 ± 0.0173	0.1923 ± 0.0021	0.5463 ± 0.0063	0.2109 ± 0.0004	0.4903 ± 0.0003
	CrossGNN [65]	0.3565 ± 0.0261	0.5626 ± 0.0180	0.3357 ± 0.0127	0.6270 ± 0.0129	0.2837 ± 0.0188	0.5676 ± 0.0187
	DLinear [10]	0.2454 ± 0.0439	0.4835 ± 0.0822	0.2397 ± 0.0497	0.4900 ± 0.1092	0.1906 ± 0.0005	0.4780 ± 0.0004
	DSFormer [66]	0.2914 ± 0.0225	0.5059 ± 0.0180	0.2760 ± 0.0228	0.5301 ± 0.0245	0.2818 ± 0.0004	0.5097 ± 0.0001
	SOFTS [67]	0.2727 ± 0.0108	0.4921 ± 0.0116	0.3532 ± 0.0435	0.6292 ± 0.0499	0.3402 ± 0.0006	0.5795 ± 0.0003
	SparseTSF [68]	0.2182 ± 0.0360	0.3890 ± 0.0832	0.2376 ± 0.0435	0.4872 ± 0.1119	0.1750 ± 0.0104	0.3208 ± 0.0319
	TimeMixer [14]	0.4047 ± 0.0144	0.6176 ± 0.0072	0.4356 ± 0.0445	0.6815 ± 0.0059	0.3311 ± 0.0003	0.5937 ± 0.0000
	TimeXer [15]	0.3418 ± 0.0512	0.5673 ± 0.0825	0.4320 ± 0.1011	0.6697 ± 0.0125	0.2933 ± 0.0015	0.5909 ± 0.0001
	Umixer [69]	0.2409 ± 0.1000	0.5202 ± 0.0105	0.2308 ± 0.0027	0.5299 ± 0.0486	0.2225 ± 0.0000	0.5070 ± 0.0000
Air Quality	AirFormer [32]	0.2581 ± 0.1037	0.4615 ± 0.1133	0.4132 ± 0.0158	0.6721 ± 0.0125	0.4309 ± 0.0175	0.5760 ± 0.0223
	AirPhyNet [34]	0.4104 ± 0.0133	0.6061 ± 0.0088	0.3560 ± 0.0056	0.6767 ± 0.0050	0.4435 ± 0.0130	0.6103 ± 0.0096
	GAGNN [33]	0.5218 ± 0.0001	0.6260 ± 0.0001	0.4833 ± 0.0241	0.6645 ± 0.0199	N/A	N/A
	MGSFormer [35]	0.5255 ± 0.0002	0.6393 ± 0.0003	0.2335 ± 0.0025	0.5526 ± 0.0057	0.2100 ± 0.0144	0.4798 ± 0.0279
	AeroMosaic	**0.7239** ± 0.0292	**0.8387** ± 0.0061	**0.5976** ± 0.0033	**0.7671** ± 0.0035	**0.6218** ± 0.0084	**0.8166** ± 0.0025

	Datasets	ZHD		CJD		CYB	
	Methods	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Time Series	CATS [64]	0.4544 ± 0.0095	0.5786 ± 0.0070	0.1895 ± 0.0039	0.5380 ± 0.0109	0.2442 ± 0.0090	0.5600 ± 0.0099
	CrossGNN [65]	0.3239 ± 0.0166	0.5538 ± 0.0090	0.3360 ± 0.0155	0.6540 ± 0.0111	0.3959 ± 0.0286	0.6773 ± 0.0132
	DLinear [10]	0.2120 ± 0.0390	0.4336 ± 0.0726	0.1937 ± 0.0470	0.4839 ± 0.1124	0.2723 ± 0.0426	0.5823 ± 0.0760
	DSFormer [66]	0.2331 ± 0.0347	0.4782 ± 0.0089	0.2008 ± 0.0223	0.4346 ± 0.0412	0.3151 ± 0.0082	0.6118 ± 0.0099
	SOFTS [67]	0.1991 ± 0.0140	0.4763 ± 0.0094	0.2005 ± 0.0065	0.5598 ± 0.0078	0.3090 ± 0.0081	0.5980 ± 0.0193
	SparseTSF [68]	0.4243 ± 0.0055	0.5184 ± 0.0093	0.1634 ± 0.0024	0.4769 ± 0.0065	0.2176 ± 0.0054	0.5129 ± 0.0103
	TimeMixer [14]	0.4241 ± 0.0150	0.6183 ± 0.0047	0.3446 ± 0.0070	0.7459 ± 0.0035	0.2520 ± 0.0077	0.5748 ± 0.0058
	TimeXer [15]	0.5249 ± 0.0093	0.6019 ± 0.0065	0.2927 ± 0.0064	0.6876 ± 0.0047	0.4129 ± 0.0825	0.7264 ± 0.0662
	Umixer [69]	0.4464 ± 0.0238	0.5705 ± 0.0095	0.4536 ± 0.0189	0.7313 ± 0.0086	0.2263 ± 0.0148	0.5324 ± 0.0350
Air Quality	AirFormer [32]	0.4848 ± 0.0167	0.5709 ± 0.0157	0.4854 ± 0.0163	0.7261 ± 0.0195	0.4389 ± 0.0057	0.7091 ± 0.0087
	AirPhyNet [34]	0.4942 ± 0.0096	0.6461 ± 0.0074	0.4669 ± 0.0071	0.7464 ± 0.0056	0.5258 ± 0.0076	0.7410 ± 0.0112
	GAGNN [33]	0.5198 ± 0.0088	0.5962 ± 0.0087	0.5264 ± 0.0055	0.7777 ± 0.0053	0.4668 ± 0.0378	0.7128 ± 0.0102
	MGSFormer [35]	0.4760 ± 0.0076	0.5681 ± 0.0130	0.5295 ± 0.0048	0.7745 ± 0.0048	0.3182 ± 0.0367	0.5403 ± 0.0241
	AeroMosaic	**0.6466** ± 0.0154	**0.7861** ± 0.0064	**0.5499** ± 0.0331	**0.8674** ± 0.0053	**0.6802** ± 0.0096	**0.8551** ± 0.0088

4.5 Ablation Study

To answer RQ3, Figure 2 summarizes the ablations on the medium-to-heavy pollution classes (labels 2–4), which are the most relevant for warnings, and Table 9 (Appendix B.3) reports the region-wise recall. All variants are trained and evaluated with the same protocol and seeds.

Transport-aware refinement is the main source of the improvement on heavy-pollution classes. Removing the post-pooling DAG residual lowers the average recall by 36.5% (from 0.561 to 0.356), with a decrease in every region. The decrease is largest on BTH (46.3%, from 0.685 to 0.368) and SDP (38.3%), the regions where `AeroMosaic` also obtains its largest improvements in Table 2, and smallest on YRD (29.6%). Without the residual block, precision also falls to about 0.35–0.42 in all regions, so the model loses both coverage and reliability on polluted days. Since the parent-label prompt is retained in this variant, the decrease measures the contribution of propagating upstream city representations beyond the prompt-side upstream labels: the labels give the risk level of the parents, while the messages carry their learned states, including numerical history and precipitation context.

Backbone adaptation is necessary for aligning the numerical prefix and the prompt with the risk classes. Removing LoRA lowers the average recall by 54.2% (to 0.257). The effect depends on the class balance of the region: on YRD, CJD, and CYB, where medium-to-heavy days account for only 15.2%–16.5% of the samples, recall drops to 0.020, 0.161, and 0.159, and precision falls to about 0.09, 0.21, and 0.15, whereas on BTH, SDP, and ZHD the model without LoRA retains a precision of 0.43–0.58. Without adaptation, the frozen backbone cannot map the numerical prefix tokens to the rare classes, and the model reverts to the dominant classes.

In contrast, the three input-modality variants remain within 0.538–0.568 in average recall. The improvement on heavy-pollution classes therefore stems mainly from the transport-aware refinement and the backbone adaptation rather than from any single input modality, which agrees with the design of `AeroMosaic`: the structural components define how evidence is combined, and each modality contributes region-dependent context. Figure 9 (Appendix C) reports per-class results.

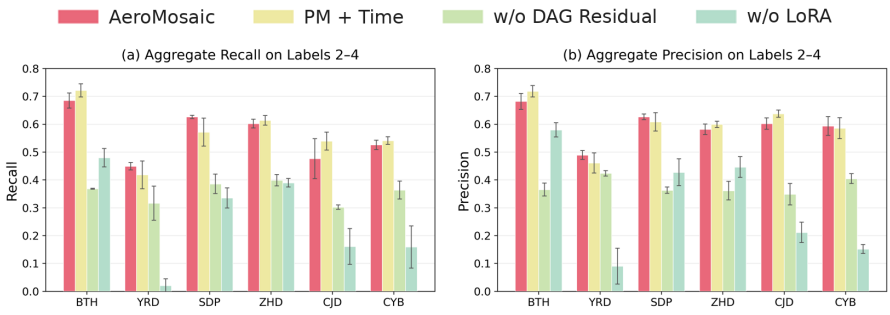


Figure 2: Six-region ablation summary on medium-to-heavy pollution classes. The 'w/o DAG Residual' variant removes the post-pooling DAG residual and keeps the parent-label prompt.

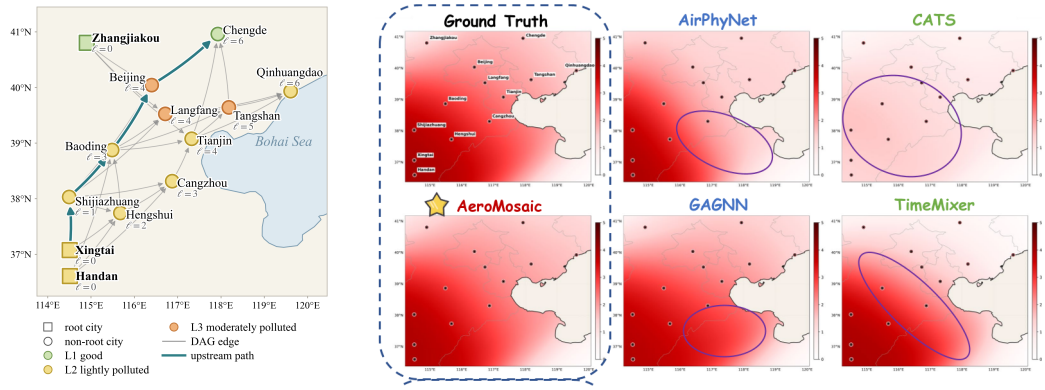


Figure 3: BTH case studies. Left: daily wind-driven DAG on 2018-03-26 from the BTH test data, with root cities (squares), ground-truth risk levels, DAG levels ℓ, and a highlighted upstream path. Right: Gaussian heatmaps of the ground truth, `AeroMosaic`, and selected baselines on 2018-01-18; purple ellipses mark regions where baselines deviate from the ground truth.

4.6 BTH Case Study

To answer RQ4, Figure 3 visualizes `AeroMosaic` on two BTH test dates. The left panel shows the daily DAG of 2018-03-26. Edges point from the southern and western cities towards Beijing, Tianjin, and the north-eastern coast, which yields three root cities (Handan, Xingtai, and Zhangjiakou) and a maximum DAG level of six. Each non-root city receives at most three upstream parents; for example, the highlighted path propagates upstream context from Xingtai through Shijiazhuang and Baoding to Beijing and Chengde. The moderately polluted cities Beijing, Langfang, and Tangshan lie at levels four and five, so their embeddings are refined with the states of their upstream cities.

The right panel compares the predicted regional risk fields on 2018-01-18, a date with a strong pollution gradient from the heavily polluted south-west (Shijiazhuang, Xingtai, and Handan) to the cleaner north-east. `AeroMosaic` reproduces both the location of the high-risk core and the gradual decrease towards the coast. The baselines fail in characteristic ways. CATS, a time-series model without spatial structure, returns an almost uniform light field and misses the polluted core. TimeMixer recovers the core but extends high risk too far along the diagonal towards Tianjin. The air-quality models capture the south-west core but misplace the transition zone in the central plain: AirPhyNet underestimates the risk around Hengshui and Cangzhou, whereas GAGNN overestimates it. Both errors occur in the transition zone downwind of the polluted south-west, where the risk level depends on daily transport, whereas the graph topologies of these models are fixed across dates. In the other five regions, `AeroMosaic` likewise follows the large-scale pattern of the ground truth (Appendix C.1); the complete BTH and YRD figure is Figure 6.

5 Discussion

The results show that transport-aligned residual propagation and adapter-based representation alignment are the main sources of the improvement, while precipitation text provides region-dependent auxiliary context. Equipping a strong baseline with the same target-day parent labels narrows but does not close the performance gap, which indicates that the role-separated interface contributes beyond access to upstream information. The main limitation is the oracle parent-label protocol: target-date parent labels are privileged information at forecast time, so the oracle results serve as an upper bound for the parent-information interface. The parent-information interface is replaceable, and the matched no-oracle protocols in Appendix B.4 provide a starting point for deployment-oriented evaluation. Recursive rollout of predicted upstream labels along the DAG, learnable transport graphs, and support for rare pollution classes are natural directions for future work.

6 Conclusion

We presented `AeroMosaic`, a transport-aware spatiotemporal multimodal learning framework for atmospheric pollution risk inference. `AeroMosaic` constructs a daily wind-driven DAG and routes the numerical state, compact precipitation context, and graph-structured upstream evidence through role-specific paths. On six regional benchmarks, `AeroMosaic` achieves the best Macro-F1 and Micro-F1 in every region under the oracle parent-label protocol, and ablations identify post-pooling DAG residual propagation and backbone adaptation as its key components. The replaceable parent-information interface also supports historical upstream observations.

Acknowledgments and Disclosure of Funding

This work was supported in part by Squirrel Ai Learning.

References

[1] World Health Organization et al. *WHO global air quality guidelines: particulate matter (PM2. 5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. World Health Organization, 2021.

[2] Aaron J. Cohen, Michael Brauer, Richard Burnett, H. Ross Anderson, Joseph Frostad, Kara Estep, et al. Estimates and 25-year trends of the global burden of disease attributable to ambient air pollution: An analysis of data from the Global Burden of Diseases Study 2015. *The Lancet*, 389(10082):1907–1918, 2017. doi: 10.1016/S0140-6736(17)30505-6.

[3] Christopher J. L. Murray, Aleksandr Y. Aravkin, Peng Zheng, Cristiana Abbafati, Kaja M. Abbas, et al. Global burden of 87 risk factors in 204 countries and territories, 1990–2019: A systematic analysis for the Global Burden of Disease Study 2019. *The Lancet*, 396(10258): 1223–1249, 2020. doi: 10.1016/S0140-6736(20)30752-2.

[4] J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer. The contribution of outdoor air pollution sources to premature mortality on a global scale. *Nature*, 525(7569):367–371, 2015. doi: 10.1038/nature15371.

[5] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2267–2276, 2015. doi: 10.1145/2783258.2788573.

[6] Daniel A Vallero. *Fundamentals of air pollution*. Academic press, 2014.

[7] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proc. of AAAI*, 2021.

[8] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Proc. of NeurIPS*, 2021.

[9] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.

[10] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proc. of AAAI*, 2023.

[11] Y Nie. A time series is worth 64words: Long-term forecasting with transformers. *The 11th International Conference on Learning Representations (ICLR)*, 2023.

[12] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[13] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[14] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

[15] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *arXiv preprint arXiv:2402.19072*, 2024.

[16] Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[17] Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Xu Wang, Binwu Wang, and Yang Wang. Phat: Modeling period heterogeneity for multivariate time series forecasting. In *International Conference on Learning Representations*, volume 2026, pages 135659–135685, 2026.

[18] Jiaming Ma, Siyuan Mu, Ruilin Tang, Haofeng Ma, Qihe Huang, Zhengyang Zhou, Pengkun Wang, Binwu Wang, and Yang Wang. To see far, look close: Evolutionary forecasting for long-term time series. *arXiv preprint arXiv:2601.23114*, 2026.

[19] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[20] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[21] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[22] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 753–763, 2020. doi: 10.1145/3394486.3403118.

[23] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Proc. of NeurIPS*, 2020.

[24] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proc. of CIKM*, 2022.

[25] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proc. of CIKM*, 2023.

[26] Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. *Advances in Neural Information Processing Systems*, 2025.

[27] Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. Bist: A lightweight and efficient bi-directional model for spatiotemporal prediction. *Proceedings of the VLDB Endowment*, 18(6):1663–1676, 2025.

[28] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *Proc. of IJCAI*, 2018.

[29] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. Deep distributed fusion network for air quality prediction. In *Proc. of KDD*, 2018.

[30] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*, 2020.

[31] Jindong Han, Hao Liu, Hengshu Zhu, Hui Xiong, and Dejing Dou. Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4081–4089, 2021. doi: 10.1609/aaai.v35i5.16529.

[32] Yuxuan Liang, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. Airformer: Predicting nationwide air quality in china with transformers. In *Proc. of AAAI*, 2023.

[33] Ling Chen, Jiahui Xu, Binqing Wu, and Jianlong Huang. Group-aware graph neural network for nationwide city air quality forecasting. *ACM Transactions on Knowledge Discovery from Data*, 2023.

[34] Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. Airphynet: Harnessing physics-guided neural networks for air quality prediction. *arXiv preprint arXiv:2402.03784*, 2024.

[35] Chengqing Yu, Fei Wang, Yilun Wang, Zezhi Shao, Tao Sun, and Yongjun Xu. Mgsfformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction. *Information Fusion*, 2025.

[36] Jiaming Ma, Guanjun Wang, Sheng Huang, Kuo Yang, Binwu Wang, Pengkun Wang, and Yang Wang. Spatiotemporal causal decoupling model for air quality forecasting. *arXiv preprint arXiv:2505.20119*, 2025. doi: 10.48550/arXiv.2505.20119.

[37] Jiaming Ma, Zhiqing Cui, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Zhe Zhao, and Yang Wang. Causal learning meet covariates: Empowering lightweight and effective nationwide air quality forecasting. In *International Joint Conference on Artificial Intelligence*, 2025.

[38] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, 2023.

[39] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[40] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, 2023.

[41] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

[42] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[43] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[44] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53140–53164, 2024.

[45] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 16115–16152, 2024.

[46] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32369–32399, 2024.

[47] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: A unified multi-task time series model. In *Advances in Neural Information Processing Systems*, 2024.

[48] Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. Urbangpt: Spatio-temporal large language models. *arXiv preprint arXiv:2403.00813*, 2024.

[49] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. Unist: A prompt-empowered universal model for urban spatio-temporal prediction. *arXiv preprint arXiv:2402.11838*, 2024.

[50] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*, 2024.

[51] Henian Zhang, Yuhang Wang, Tae-Won Park, and Yi Deng. Quantifying the relationship between extreme air pollution events and extreme weather events. *Atmospheric Research*, 2017.

[52] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[53] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[54] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 2021.

[55] Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. ChatTS: Aligning time series with LLMs via synthetic data for enhanced understanding and reasoning. In *Proceedings of the VLDB Endowment*, 2025.

[56] A Vaswani. Attention is all you need. *Proc. of NeurIPS*, 2017.

[57] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*, 2023.

[58] Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. Robust spatio-temporal centralized interaction for ood learning. In *Forty-second International Conference on Machine Learning*, 2025.

[59] Chunyang Wang, Yanmin Zhu, Tianzi Zang, Haobing Liu, and Jiadi Yu. Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction. In *Proc. of WSDM*, 2021.

[60] Jiaming Ma, Guanjun Wang, Sheng Huang, Kuo Yang, Binwu Wang, Pengkun Wang, and Yang Wang. Spatiotemporal causal decoupling model for air quality forecasting. *arXiv preprint arXiv:2505.20119*, 2025.

[61] Shuo Wang, Yun Cheng, Qingye Meng, Olga Saukh, Jiang Zhang, Jingfang Fan, Yuanting Zhang, Xingyuan Yuan, and Lothar Thiele. Pcdcnet: A surrogate model for air quality forecasting with physical-chemical dynamics and constraints. *arXiv preprint arXiv:2505.19842*, 2025. doi: 10.48550/arXiv.2505.19842.

[62] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[63] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

[64] Jiecheng Lu, Xu Han, Yan Sun, and Shihao Yang. Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. *arXiv preprint arXiv:2403.01673*, 2024.

[65] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *In Proc. of NeurIPS*, 36:46885–46902, 2023.

[66] Chengqing Yu, Fei Wang, Zezhi Shao, Lin Wu, and Yongjun Xu. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proc. of CIKM*, 2023.

[67] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *arXiv preprint arXiv:2404.14197*, 2024.

[68] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024.

[69] Xiang Ma, Xuemei Li, Lexin Fang, Tianlong Zhao, and Caiming Zhang. U-mixer: An unet-mixer architecture with stationarity correction for time series forecasting. In *Proc. of AAAI*, 2024.

Appendix Contents

A Methodology Details **16**

A.1 Notation Table 16

A.2 Forward-Pass Pseudocode 17

A.3 DAG Construction Details 17

A.4 Root-Sensitive Modules 17

A.5 Information Boundary 18

A.6 Input Visualization 18

A.7 Prompt Instantiation 19

A.8 Precipitation Text Construction 20

B Experiment Details **20**

B.1 Experimental Setup 20

B.2 Baseline Configurations 21

B.3 Ablation Tables 21

B.4 Additional Analyses 22

C Case Study **23**

C.1 Spatial Comparisons 23

C.2 Per-Class Precision and Recall 26

D Broader Impact and Responsible Use **27**

A Methodology Details

A.1 Notation Table

Table 3: Main notation used in the sequential description of `AeroMosaic`. The table follows the input–encoding–graph–output order used in Algorithm 1.

Symbol	Shape / Type	Meaning
N	scalar	Number of cities in one region-date group.
i, t	indices	City index and target-date index.
C	scalar	Number of $PM_{2.5}$ risk classes.
L	scalar	Historical input length.
K	scalar	Number of numerical prefix tokens.
d	scalar	Hidden dimension of token and city representations.
g	identifier	Region identifier used in prompt construction.
$\mathbf{X}^{(i)}$	$\mathbb{R}^{L\times D}$	Historical numerical tensor for city i.
$D = d_p + d_m + d_t$	scalar	Raw feature dimension split into pollutant-history, meteorological, and time-derived channels.
$\mathbf{X}^{(i)}_{\text{pre}}$	$\mathbb{R}^{L\times D}$	Root-adjusted numerical tensor after optional root-region fusion.
$\mathbf{q}^{(i)}$	token field	Compact prompt field containing city/date metadata, recent precipitation text, and upstream context.
$\ell_i^{(t)}$	scalar / level	DAG level of city i on target date t used as compact structural prompt metadata.
$\mathbf{b}_i^{(t)}$	text block	Parent-label block inserted into the prompt under the fixed oracle parent-label protocol.
$\mathbf{c}_i$	text summary	Recent-precipitation summary derived from the historical input window.
$\mathcal{B}_\theta$	backbone	Language-model backbone that maps prefix–prompt embeddings to hidden states.
$e_\psi(\cdot)$	embedding map	Prompt-token embedding function.
$f_\phi(\cdot)$	projector	Numerical prefix projector.
$\mathbf{P}^{(i)}$	$\mathbb{R}^{K\times d}$	Learnable numerical prefix tokens for city i.
$\mathbf{Z}_0^{(i)}$	token sequence	Concatenated prefix and prompt-token embeddings.
$\mathbf{Z}^{(i)}$	token states	Hidden states produced by $\mathcal{B}_\theta$.
$\mathbf{h}_i$	$\mathbb{R}^d$	Pooled city embedding before graph propagation.
$\mathbf{h}$	$\mathbb{R}^{N\times d}$	Stacked city embeddings for the region-date group.
$\mathbf{A}^{(t)}$	$\{0,1\}^{N\times N}$	Daily DAG adjacency in destination-by-source form.
$\hat{\mathbf{A}}^{(t)}$	$\mathbb{R}^{N\times N}$	Row-normalized daily adjacency.
$\mathcal{P}_i^{(t)}$	set	Parent-city set of city i on date t.
$\mathcal{E}_{\text{cand}}$	set	Candidate city-pair set before wind-directed filtering.
$\mathbf{w}_i^{(t)}$	vector	Estimated target-day wind vector of city i used by the DAG builder.
$\mathbf{d}_{i\to j}$	vector	Geographic displacement from city i to city j.
$\theta_t(i,j)$	scalar	Wind-direction alignment angle for candidate edge $i \to j$.
$s_{i\to j}^{(t)}$	scalar	Transport score used to rank retained candidate edges.
$\text{dist}(i,j)$	scalar	Geographic distance between cities i and j.
ϵ	scalar	Small denominator constant used in edge scoring.
$\text{Pool}(\cdot)$	operator	Masked mean pooling from token states to a city representation.
$\text{Linear}_{\text{msg}}$	map	Message projection used before DAG aggregation.
FFN	network	Feed-forward transformation in the DAG residual block.
GELU, LayerNorm	operators	Nonlinear activation and normalization used inside the DAG residual block.
r_i	$\{0,1\}$	Root indicator.
$\mathbf{m}, \mathbf{u}$	$\mathbb{R}^{N\times d}$	Aggregated DAG messages and transformed residual update.
α	scalar	Learnable residual gate parameter for DAG propagation.
$\sigma(\cdot)$	function	Sigmoid function used by the residual gate.
$\tilde{\mathbf{h}}_i$	$\mathbb{R}^d$	Refined city embedding after DAG residual propagation.
$\mathbf{o}_i$	$\mathbb{R}^C$	Class-logit vector for city i.
$y_i, \hat{y}_i$	class labels	Supervision label and predicted class for city i.
w_i	scalar	Root-aware sample weight used in the training loss.
w_{env}	scalar	Root-region fusion weight.
$r_{\text{base}}, e_{y_i}, w_{\max}$	scalars	Base root factor, class-dependent factor, and maximum sample weight.
CE	loss	Cross-entropy term in the group-level objective.
Θ	parameter set	Trainable parameters optimized by the training objective.
$\theta_{\text{adp}}, \omega_{\text{msg}}, \omega_{\text{ffn}}, \omega_{\text{head}}$	parameters	Adapter, message-projection, feed-forward, and output-head parameters included in Θ.
$\mathcal{L}$	scalar	Group-level weighted cross-entropy loss.

A.2 Forward-Pass Pseudocode

Algorithm 1 `AeroMosaic` inference for one region-date city group

Require: City histories $\{\mathbf{X}^{(i)}\}_{i=1}^{N}$, prompt ingredients for each city, daily DAG adjacency $\mathbf{A}$, DAG metadata including root indicators $\{r_i\}_{i=1}^{N}$, DAG levels $\{\ell_i^{(t)}\}_{i=1}^{N}$, and parent sets $\{\mathcal{P}_i^{(t)}\}_{i=1}^{N}$
Ensure: Predicted $PM_{2.5}$ class tokens $\{\hat{y}_i\}_{i=1}^{N}$

Row-normalize the daily adjacency: $\hat{\mathbf{A}} \leftarrow \mathrm{RowNorm}(\mathbf{A})$, keeping zero-in-degree rows all-zero
for each city $i = 1, \ldots, N$ **do**
  Set $\mathbf{X}_{\mathrm{pre}}^{(i)} \leftarrow \mathbf{X}^{(i)}$
  if $r_i = 1$ **then**
    Apply root-region fusion to update $\mathbf{X}_{\mathrm{pre}}^{(i)}$ with coarse regional context
  end if
  Project the preprocessed history into prefix tokens: $\mathbf{P}^{(i)} \leftarrow f_\phi(\mathrm{vec}(\mathbf{X}_{\mathrm{pre}}^{(i)}))$
  Instantiate the compact prompt: $\mathbf{q}^{(i)} \leftarrow \mathrm{Prompt}(g, i, t, \ell_i^{(t)}, \mathbf{b}_i^{(t)}, \mathbf{c}_i)$
  Form the mixed sequence: $\mathbf{Z}_0^{(i)} \leftarrow [\mathbf{P}^{(i)}; e_\psi(\mathbf{q}^{(i)})]$
  Encode token states: $\mathbf{Z}^{(i)} \leftarrow \mathcal{B}_\theta(\mathbf{Z}_0^{(i)})$
  Pool valid token states: $\mathbf{h}_i \leftarrow \mathrm{Pool}(\mathbf{Z}^{(i)})$
end for
Stack city embeddings into $\mathbf{h} \in \mathbb{R}^{N \times d}$
Aggregate upstream messages: $\mathbf{m} \leftarrow \hat{\mathbf{A}}\,\mathrm{Linear}_{\mathrm{msg}}(\mathbf{h})$
Refine messages: $\mathbf{u} \leftarrow \mathrm{FFN}(\mathrm{GELU}(\mathrm{LayerNorm}(\mathbf{m})))$
Set $\mathbf{u}_i \leftarrow \mathbf{0}$ for every root city i ($r_i = 1$)
Apply residual DAG update: $\tilde{\mathbf{h}} \leftarrow \mathbf{h} + \sigma(\alpha)\mathbf{u}$
Compute logits: $\mathbf{o}_i \leftarrow \mathrm{Head}(\tilde{\mathbf{h}}_i)$ for all cities i
Infer city risk labels: $\hat{y}_i \leftarrow \arg\max_c \mathbf{o}_{i,c}$ for all cities i
return $\{\hat{y}_i\}_{i=1}^{N}$

A.3 DAG Construction Details

The DAG builder takes city coordinates, an altitude map, estimated target-day wind vectors, a distance threshold, a wind-magnitude threshold, an angular threshold, and a parent cap as inputs. It builds geographically plausible candidate city pairs, removes terrain-blocked pairs when terrain filtering is enabled, retains directed edges whose wind magnitude and angular alignment pass the configured thresholds, scores surviving edges by alignment, wind magnitude, and distance, caps the number of parents per destination city, and breaks cycles by removing selected cycle-internal edges. The output is the daily DAG adjacency together with root indicators, topological order, DAG levels, and parent counts.

The daily DAG is a transport-aligned structural prior; its construction choices determine which upstream cities send graph messages and which cities become roots. Table 4 lists the builder settings of each dataset. The target-day wind is estimated either by persistence of the last historical observation or by the target-day mean of the meteorological input, which serves as forecast information in the KnowAir setting [30]. Learned adaptive graphs [26] provide a complementary way to capture spatial dependencies and can be combined with the wind-driven prior.

Table 4: Daily DAG builder settings for each dataset.

Dataset	Wind estimate	Radius	Alignment	Parent cap	Edge weight
BTH	Persistence	600 km	$\cos\theta \geq 0.25$	3	$s_{i\to j}^{(t)}$
YRD, SDP, ZHD, CJD, CYB	Target-day mean	$\approx$300 km	$\theta \leq 60^\circ$	None	Binary

A.4 Root-Sensitive Modules

Root cities are structurally different from non-root cities because they have no upstream parents in the daily DAG. As a result, the post-pooling DAG residual block cannot deliver incoming parent messages to these nodes. The root-region fusion module provides a coarse group-level input compensation for

root nodes, while the root-aware loss increases the optimization emphasis on structurally harder root samples within a bounded range. Both mechanisms compensate for the graph-induced asymmetry between root and non-root cities.

Both root modules are retained in all ablation variants of Table 9.

A.5 Information Boundary

Under the oracle parent-label protocol, the model-visible inputs for each target city and date are the historical $PM_{2.5}$ and time channels, the recent-precipitation text generated from the historical input window, the daily DAG structure and metadata, and the prompt-side parent-label context. The supervision label is derived from the target-date future $PM_{2.5}$ window and is used only as the training and evaluation target. The daily DAG is built offline from geographic priors and the target-day wind estimate described in Appendix A.3, without access to the target city's realized $PM_{2.5}$ values in the target horizon. Table 5 summarizes these roles, and Table 6 lists the source window and model visibility of each signal.

Table 5: Protocol card for the oracle parent-label protocol: model-visible inputs, supervision, offline structure, and oracle parent-label context.

Signal	**Use**	**Role**
Historical $PM_{2.5}$ and time	Numerical prefix	Model input
Recent precipitation	Prompt text	Auxiliary context from history
Daily DAG	Graph structure and metadata	Offline structural prior
Parent labels	Prompt field	Oracle upstream parent-label context
Prompt reconstruction	Not performed	Stored oracle prompts are read directly
Target-city future $PM_{2.5}$	Label construction only	Supervision only

Table 6: Source window, model visibility, and role of each signal.

Signal	**Source window**	**Model visibility**	**Role**
Historical $PM_{2.5}$	Input history	Visible through numerical prefix tokens	Pollutant-state evidence
Time-derived features	Input history / calendar index	Visible through numerical prefix tokens	Temporal state evidence
Recent precipitation text	Input history	Visible through prompt text	Compact environmental context
Target-city future $PM_{2.5}$	Target window	Not visible as input	Supervision-label construction only
Wind cue for DAG	Target-day wind estimate (Table 4)	Not directly exposed as a numerical feature	Offline graph construction
Daily DAG metadata	Offline graph builder	Visible through graph structure and prompt fields	Transport-aligned structural prior
Parent-city labels	Target date	Visible in prompts for all splits	Oracle upstream parent-label context

The parent-label field is the only oracle field. Parent labels are precomputed offline and inserted into prompts in the training, validation, and test splits. They do not reveal the target city's own target-window $PM_{2.5}$ value, but they are target-date labels of upstream cities and can therefore correlate with the target label through regional transport. Appendix B.4 reports the oracle-access control and the matched no-oracle protocols.

A.6 Input Visualization

Figure 1 provides the visual audit of model-visible inputs and the supervision-only target window, and also shows the computational order: root-region fusion is applied first, and the resulting history is projected into prefix tokens before encoding and post-pooling DAG residual propagation. Section 3.3 and Algorithm 1 give the exact equations. Figure 4 gives a compact example of the resulting input composition. The detailed prompt template and representative cases appear in Appendix A.7. Figure 4 shows the same information boundary from the input side: the model combines the numerical history,

prompt-side context, and offline DAG-derived structure, while the target-city future $PM_{2.5}$ window is excluded from the input (cf. Table 6).

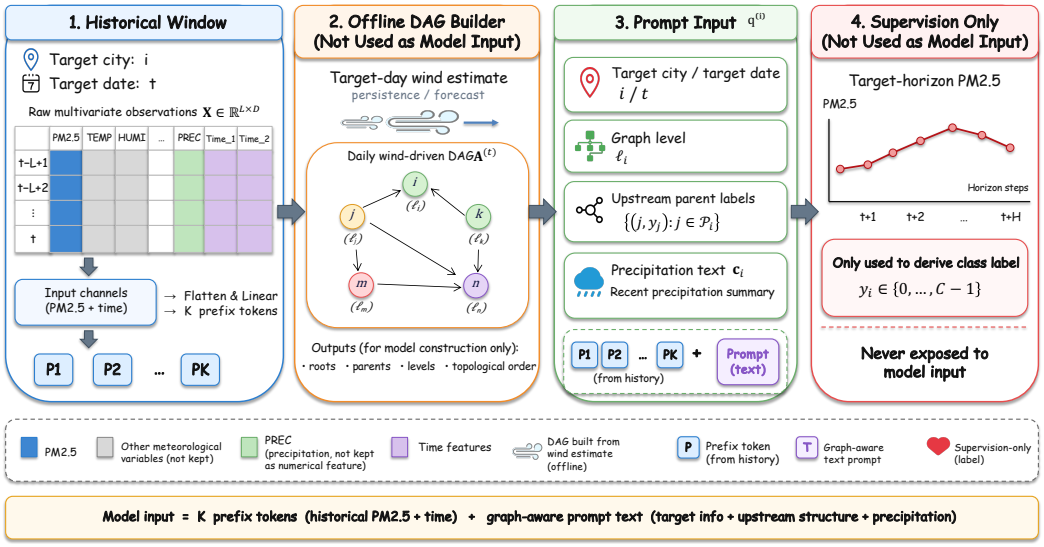


Figure 4: Input composition for a representative date. Historical numerical signals enter through numerical prefix tokens, while recent precipitation and upstream DAG evidence are provided as compact text fields. The daily wind-driven DAG supplies directed transport structure, and the target-city future $PM_{2.5}$ window is reserved for supervision.

A.7 Prompt Instantiation

Prompt Template

You are given multimodal air-quality risk-inference features for region {region}.
The target city's history is provided through the time-series input, and recent precipitation context is provided separately in text form.
Target city: {city} Target date: {date} Graph level: {level}
Use the time-series input as the main evidence for the target city.
Upstream graph context (oracle protocol: target-date parent labels): {parent block}
Recent precipitation context: {recent precipitation text}
Infer the final air-quality risk label for the target city on the target date. Output only one label token: 0, 1, 2, 3, 4, or 5.

Representative prompt-input cases

Root City

City: Chengde
Date: 2018-01-01
Graph level: 0
Upstream: none (root).
No noticeable recent precipitation. Dry conditions persisted.

Single Parent

City: Xingtai
Date: 2018-01-05
Graph level: 1
Upstream:
- Handan: label=2
Recent light rain was observed early in the previous input window, followed by drier conditions later.

Multi-Parent

City: Baoding
Date: 2018-01-18
Graph level: 2
Upstream:
- Shijiazhuang: label=4
- Handan: label=4
- Xingtai: label=4
No noticeable recent precipitation. Dry conditions persisted.

Figure 5: Prompt template and representative root, single-parent, and multi-parent cases. The target-date parent-label field is privileged information used by the oracle parent-label protocol; the no-oracle protocols remove it or replace it with predicted classes or historical observations.

The prompt carries compact recent-precipitation context and prompt-side upstream parent-label context, while the historical numerical features enter as prefix tokens.

A.8 Precipitation Text Construction

The recent-precipitation text is generated by a deterministic rule-based procedure rather than by an LLM or an additional learned model. For each city sample, the procedure reads the total-precipitation channel (column 6 of the 13-channel tensor) over the eight historical time steps and converts the original values from meters to millimeters by multiplying by 1,000. The overall category is determined from the eight-step sum using the following left-closed thresholds in millimeters: no noticeable precipitation (< 0.1), light rain ($[0.1, 10)$), moderate rain ($[10, 25)$), heavy rain ($[25, 50)$), rainstorm ($[50, 100)$), severe rainstorm ($[100, 250)$), and extreme rainstorm (≥ 250).

For the eight-step window, we form early, middle, and late segments from indices $0{:}2$, $2{:}5$, and $5{:}8$, respectively. Each segment sum is rescaled by $8/|S|$ before applying the same category thresholds. A second sentence records the segment categories and, when applicable, a monotone increasing/decreasing trend or the segment with the largest raw sum. Rain occurrence at an individual step is defined as at least 0.1 mm. The text generator itself performs no missing-value imputation; it assumes that preprocessing supplies a finite eight-step sequence. It has no learned parameters or sampling step, so the same precipitation sequence always produces the same text. The procedure does not use the target city's future $PM_{2.5}$ values or any realized target-window pollutant information.

Representative summaries include: "No noticeable recent precipitation. Dry conditions persisted." and "Recent light rain was observed in the previous input window. Early, middle, and late segments: light rain, no noticeable precipitation, no noticeable precipitation. Precipitation decreased across the window. Rain occurred at 2 of eight observations." The text field is thus a compact temporal summary of the input window, and it complements rather than replaces the structured numerical state: the selected $PM_{2.5}$ and time-derived signals still enter through the prefix tokens.

B Experiment Details

B.1 Experimental Setup

The language-model backbone $\mathcal{B}_\theta$ is initialized from the ChatTS-8B checkpoint (release 1103) [55], which builds on Qwen3-8B; the exact checkpoint revision is recorded in the released code. The checkpoint has 36 language-model layers, hidden size 4096, five time-series encoder layers, and patch size 8, and is stored in float16; we fine-tune in bf16. The numerical history is projected into four prefix tokens and concatenated with the prompt-token embeddings before entering the encoder. Masked mean pooling produces one city representation, followed by the DAG residual block and a linear classification head. We train with three random seeds, group-level batches, and all modules enabled (DAG residual propagation, root-region fusion, and root-aware loss), and keep the checkpoint with the best validation score rather than the last epoch.

Numerical input configuration. The numerical history of each city has $D = 13$ channels: $PM_{2.5}$ (index 0), five surface and boundary-layer variables (indices 1–5: 2 m temperature, boundary-layer height, K-index, 950 hPa relative humidity, surface pressure), total precipitation (6), four wind channels (7–10: 950 hPa wind u and v, wind speed, and wind direction), and two time-derived features (11–12). After root-region fusion, the numerical projector receives the $PM_{2.5}$ and time channels (indices 0, 11, and 12); meteorological information enters through the precipitation text and the wind-driven DAG. Ablation variants are named by the numerical channels they retain. The DAG builder reads the wind channels directly, independently of the numerical projector.

The small number of prefix tokens keeps the numerical interface compact, LoRA limits the number of adapted parameters, and the negative initialization of the residual gate makes the graph update conservative early in training.

Table 7: Training hyperparameters shared across regions.

Hyperparameter	Value
Backbone checkpoint	ChatTS-8B (release 1103)
Base model / licenses	Qwen3-8B; Apache-2.0 (model), MIT (code)
Backbone hidden size	4096
LoRA rank / alpha / dropout	16 / 32 / 0.05
LoRA target modules	q, k, v, o, gate, up, down projections
Numerical prefix tokens K	4
Maximum input length	512 tokens (left truncation)
Pooling	masked mean
Learning rate	1×10^{-4}
Epochs	3 (best checkpoint by validation score)
Batch size / gradient accumulation	1 region-date group / 2 steps
Precision	bf16
Random seeds	42, 123, 456
Weight decay / warm-up ratio	0 / 0.03
DAG residual gate init. α_{init} / hidden dimension	-3.0 / 256
Classifier dropout	0.1
Root-region fusion weight w_{env}	0.8
Root-aware loss r_{base} / e_3 / e_4 / $w_{\max}$	1.5 / 2.0 / 3.0 / 4.0

B.2 Baseline Configurations

Table 8 summarizes how the baselines differ from `AeroMosaic`. The baselines receive all 13 numerical channels; `AeroMosaic` uses the $PM_{2.5}$ and time channels together with compact precipitation text, graph-conditioned prompts, and oracle parent-label context. Graph-based baselines use static or partition-based spatial priors, whereas `AeroMosaic` uses a daily wind-driven DAG. Time-series baselines produce city-level outputs jointly through the city axis, whereas `AeroMosaic` introduces inter-city structure through prompt-side parent-label context and the DAG residual block. The ablations in Section 4.5 evaluate the individual components under the same protocol.

Table 8: Baseline configurations. "Static dist." denotes a fixed distance-based graph shared across all dates, and "Wind DAG" denotes a daily wind-driven directed acyclic graph.

Model	Family	Graph type	Multimodal input	Pred. format	Oracle ctx.
DLinear	Time-series	None	No	$[B, T, N, F]$	No
CATS	Time-series	None	No	$[B, T, N, F]$	No
CrossGNN	Time-series	None	No	$[B, T, N, F]$	No
DSFormer	Time-series	None	No	$[B, T, N, F]$	No
SOFTS	Time-series	None	No	$[B, T, N, F]$	No
SparseTSF	Time-series	None	No	$[B, T, N, F]$	No
TimeMixer	Time-series	None	No	$[B, T, N, F]$	No
TimeXer	Time-series	None	No	$[B, T, N, F]$	No
UMixer	Time-series	None	No	$[B, T, N, F]$	No
AirFormer	Air-quality	Dartboard	No	Per-city	No
AirPhyNet	Air-quality	Fixed, flow-weighted	No	Per-city	No
GAGNN	Air-quality	Static dist.	No	Per-city	No
MGSFormer	Air-quality	Static dist.	No	Per-city	No
`AeroMosaic`	Proposed	Wind DAG	Yes	Per-city	Yes

$PM_{2.5}$-GNN [30], AirCade [36], CauAir [37], spatiotemporal causal decoupling [60], and PCDCNet [61] are closely related air-quality models. They operate on structured numerical channels such as AQI histories, meteorological covariates, emissions, and physical constraints, and they do not consume precipitation text or prompt-side parent-label context; PCDCNet additionally relies on emissions-side inputs that are not routinely available in public data.

B.3 Ablation Tables

Table 9 reports the region-wise recall on labels 2–4 underlying Figure 2. Three observations follow.

Structural components. Removing the DAG residual block lowers the recall in every region: from 0.685 to 0.368 on BTH (-0.317), the largest drop, and from 0.449 to 0.316 on YRD (-0.133), the smallest, which corresponds to an average decrease of 0.205. Removing LoRA lowers it further, by 0.304 on average, and the drop is strongly region-dependent: -0.205 on BTH and -0.212 on ZHD, but -0.429 on YRD, where the recall falls to 0.020 and the precision in Figure 2 to about 0.09, so the model essentially stops predicting labels 2–4. The three regions with the smallest share of labels 2–4 (Table 1: YRD, CJD, and CYB with 15.2%–16.5%, against 27.0%–30.0% on BTH, SDP, and ZHD) show the three largest drops. The two ablations also differ in kind: without the DAG residual both precision (about 0.35–0.42) and recall are lowered in all regions, whereas without LoRA the precision stays at 0.43–0.58 on BTH, SDP, and ZHD and collapses on YRD, CJD, and CYB (about 0.09, 0.21, and 0.15).

Input modalities. The three modality variants differ by at most 0.030 in average recall (0.538–0.568). PM + Time is the best variant on BTH, ZHD, CJD, and CYB, and PM + PrecipText + Time on YRD (0.449) and SDP (0.626). Relative to PM + Time, adding precipitation text improves recall on YRD ($+0.031$) and SDP ($+0.054$) and changes it by -0.012 to -0.063 on the other regions, of the same order as the seed standard deviations (0.005–0.088). Precipitation text therefore acts as region-dependent auxiliary context.

All variants retain the parent-label prompt, so the table quantifies the contribution of each component in addition to the prompt-side upstream context.

Table 9: Region-wise ablation results on medium-to-heavy pollution classes. Each value reports aggregate recall over labels 2–4 in that region, shown as mean $\pm$ sample standard deviation across three random seeds. PrecipText denotes recent precipitation text derived from the previous input window. The 'w/o DAG Residual' variant removes post-pooling graph residual propagation while retaining prompt-side oracle parent labels. Higher is better.

Variant	BTH	YRD	CYB	ZHD	SDP	CJD	Avg.
PM + Time	0.721 ± 0.029	0.418 ± 0.061	0.541 ± 0.017	0.614 ± 0.021	0.572 ± 0.062	0.539 ± 0.039	0.568
PM + PrecipText + Time	0.685 ± 0.033	0.449 ± 0.017	0.526 ± 0.021	0.602 ± 0.019	0.626 ± 0.007	0.476 ± 0.088	0.561
PrecipText + Time	0.688 ± 0.027	0.377 ± 0.005	0.525 ± 0.027	0.573 ± 0.021	0.603 ± 0.076	0.463 ± 0.023	0.538
w/o DAG Residual	0.368 ± 0.002	0.316 ± 0.075	0.363 ± 0.039	0.399 ± 0.025	0.386 ± 0.043	0.302 ± 0.010	0.356
w/o LoRA	0.480 ± 0.040	0.020 ± 0.030	0.159 ± 0.093	0.390 ± 0.019	0.335 ± 0.044	0.161 ± 0.079	0.257

B.4 Additional Analyses

This section reports BTH analyses of the parent-information interface, inference time, and precipitation representation. Each result comes from a single run.

Matched No-Oracle Parent-Information Protocols. Target-day parent labels enter `AeroMosaic` through a dedicated prompt field, so removing them changes the parent-information protocol rather than the model definition. The no-parent protocol removes the upstream prompt field, the predicted-only protocol substitutes model-predicted upstream risk classes, and the historical-observation protocol uses the final parent-city $PM_{2.5}$ observation in the input window. None of these protocols uses target-day true parent labels. Each row of Table 10 applies the same protocol and temporal-availability constraint to all three models. For input alignment with the numerical baselines, `AeroMosaic` replaces the precipitation text with numerical precipitation input in this analysis. Historical upstream observations are the most informative no-oracle protocol for `AeroMosaic`: compared with the no-parent protocol, they raise `AeroMosaic` by 0.049 Micro-F1 and 0.041 Macro-F1, whereas MGSFormer and TimeMixer change by at most 0.025. This shows that `AeroMosaic` uses upstream evidence routed along the daily DAG when target-day labels are unavailable, with the same DAG and architecture as in the oracle configuration. Model-predicted parent classes are less informative than historical observations. With precipitation text instead of numerical precipitation, the historical-observation protocol yields $0.5760/0.3374$.

Table 10: Matched BTH no-oracle protocols. Each entry reports Micro-F1/Macro-F1.

Parent-information protocol	`AeroMosaic` (numeric precip.)	MGSFormer	TimeMixer
No parent	0.5334/0.3049	0.6303/0.5091	0.5545/0.4083
Predicted only	0.5128/0.2458	0.6272/0.5070	0.5410/0.4166
Historical upstream $PM_{2.5}$	0.5825/0.3462	0.6318/0.5342	0.5557/0.4052

Oracle-Access Control. Table 11 supplies MGSFormer with the same type of target-day parent labels as numerical covariates within its framework. Oracle covariates raise MGSFormer by 0.0333 Micro-F1 and 0.0214 Macro-F1, while `AeroMosaic` remains higher by 0.1493 Micro-F1 and 0.1507 Macro-F1. Target-day parent information is therefore not the sole explanation for the performance gap.

Table 11: BTH oracle-access control.

Configuration	Oracle parent labels	Micro-F1	Macro-F1
MGSFormer	No	0.6561	0.5518
MGSFormer + oracle-label covariates	Yes	0.6894	0.5732
`AeroMosaic`	Yes	0.8387	0.7239

Inference Time. Table 12 reports single-batch inference time, where one batch is one BTH region-date group of 13 cities. `AeroMosaic` is measured on an NVIDIA H20 with PyTorch 2.6.0 (CUDA 12.4) and BF16 autocast, averaged over five repetitions with CUDA synchronization before and after timing. The baselines are measured in FP32 with synthetic inputs (batch size 1, 13 cities, 8 features, history and prediction lengths $24 \rightarrow 24$), using 20 warm-up and 200 measured iterations. `AeroMosaic` has the largest inference time because of its language-model backbone. At 0.1830 seconds per region-date batch, it amounts to 0.0017% of the 3-hour sampling interval of the dataset, so inference time does not constrain deployment at this cadence.

Table 12: BTH inference time per region-date batch.

	`AeroMosaic`	**TimeXer**	**AirPhyNet**	**MGSFormer**	**DLinear**	**GAGNN**
Time (s/batch)	0.1830	0.0020	0.0320	0.0100	0.0002	0.0220

Precipitation Representation. Table 13 compares the precipitation text with a numerical precipitation scalar ($D = 14$) that adds the same deterministic precipitation category as one additional numerical channel. The two configurations are initialized from different base checkpoints. The numerical precipitation scalar obtains 0.1264 lower Micro-F1 and 0.1677 lower Macro-F1 than the configuration with precipitation text.

Table 13: BTH precipitation representation.

Configuration	Micro-F1	Macro-F1
Precipitation text (main results)	0.8387	0.7239
$D = 14$ numerical precipitation scalar	0.7123	0.5562

C Case Study

C.1 Spatial Comparisons

Figures 6–8 show, for one test date per region, the heatmap of the ground-truth risk class, the output of `AeroMosaic`, and four baselines selected per case; the dates are 2018-01-18 (BTH), 2018-04-02 (YRD), 2018-12-15 (SDP), 2018-03-05 (ZHD), 2018-02-11 (CJD), and 2018-04-30 (CYB). The colour scale is the risk class from 0 to 5, and purple ellipses mark regions where a baseline output deviates visibly from the ground truth. The BTH half of Figure 6 is also shown in Figure 3 of the main text.

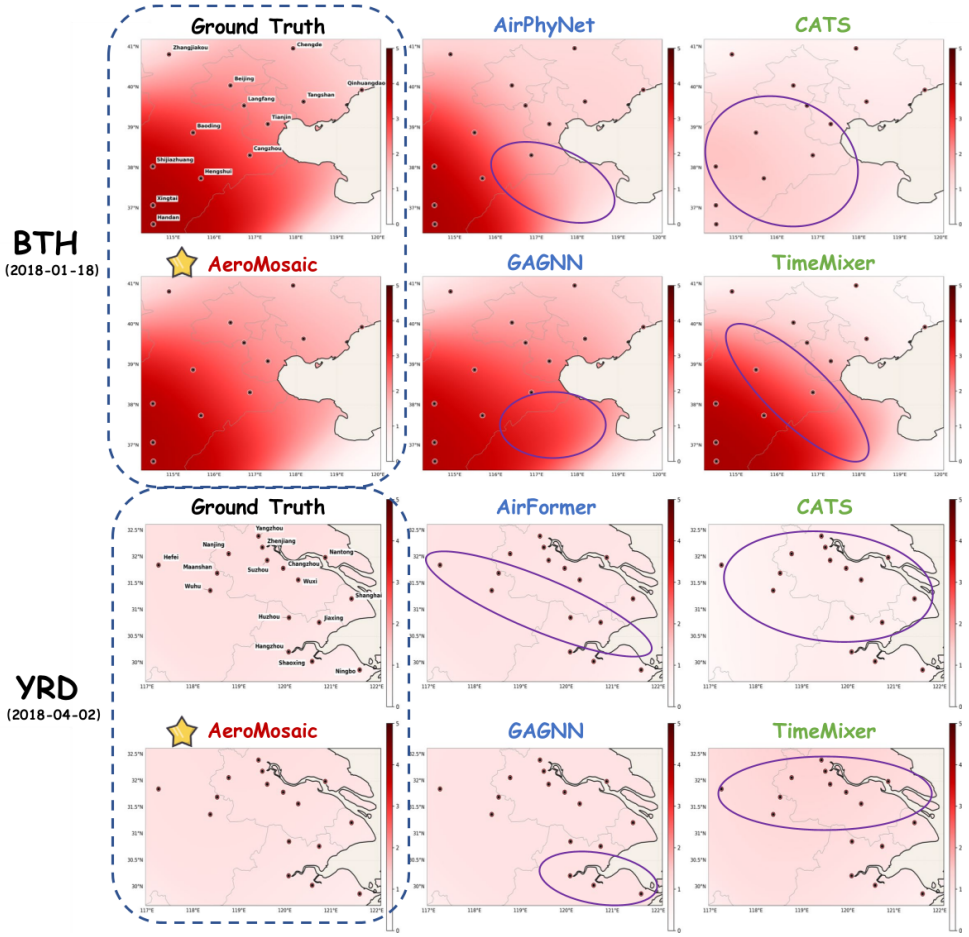


Figure 6: Heatmaps for BTH (2018-01-18) and YRD (2018-04-02): ground truth, `AeroMosaic`, and four baselines per region.

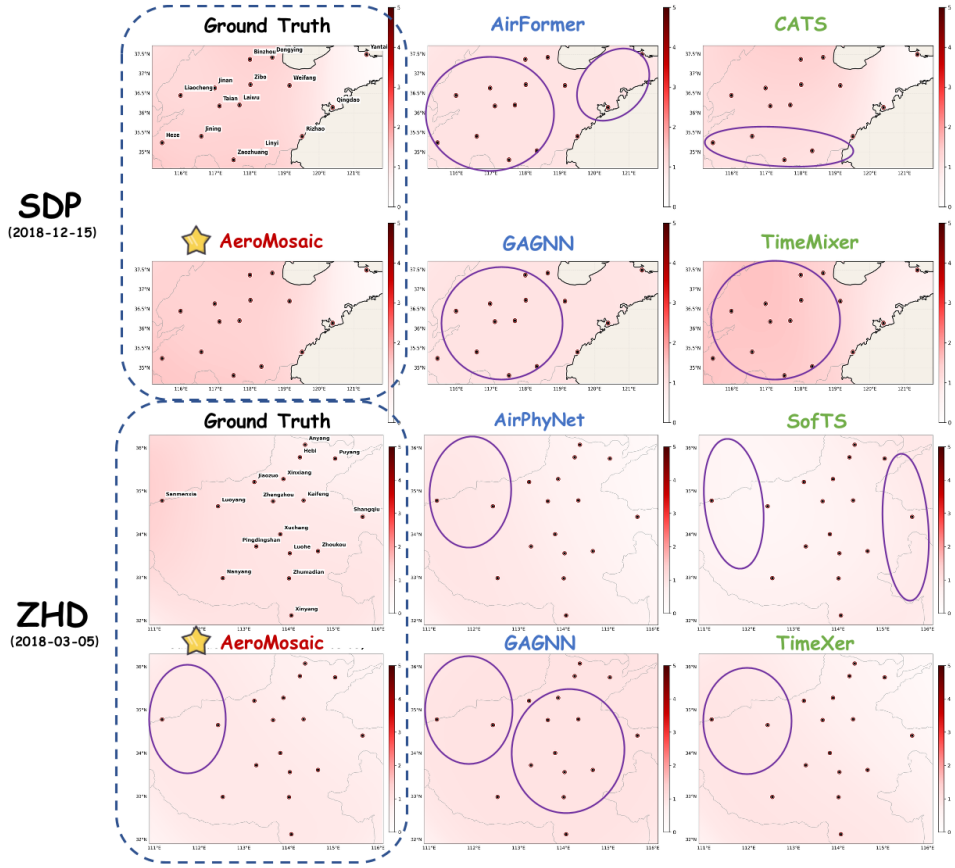


Figure 7: Heatmaps for SDP (2018-12-15) and ZHD (2018-03-05): ground truth, `AeroMosaic`, and four baselines per region.

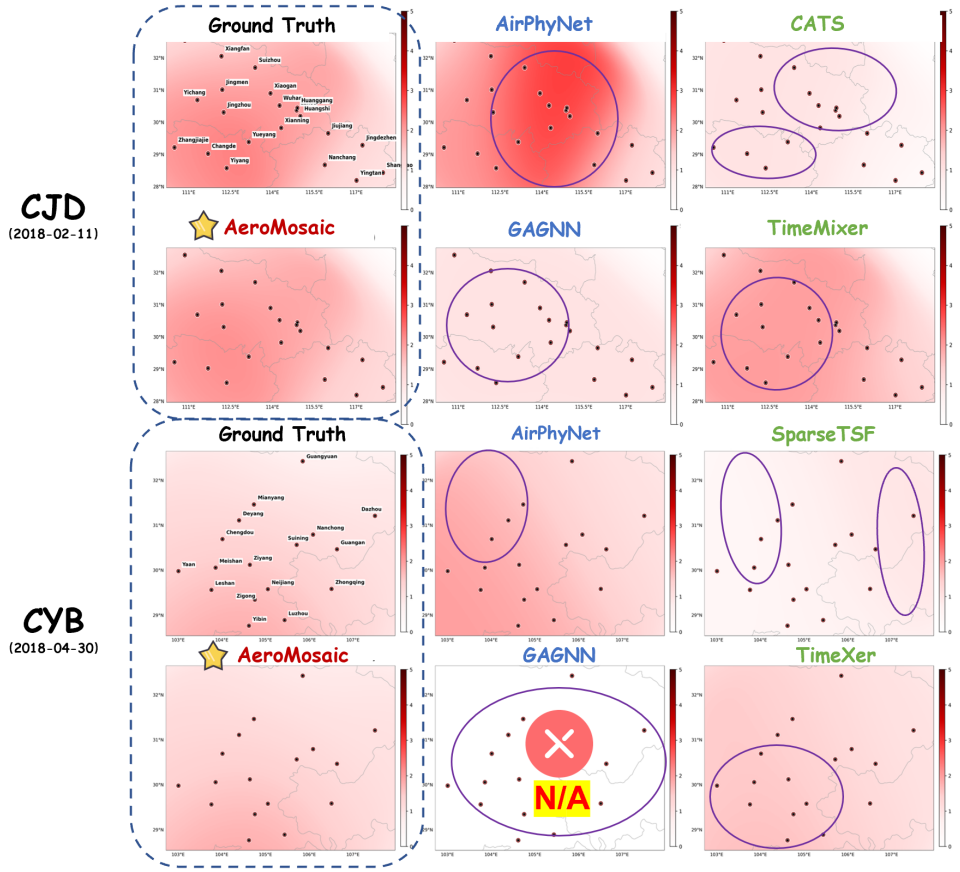


Figure 8: Heatmaps for CJD (2018-02-11) and CYB (2018-04-30): ground truth, AeroMosaic, and four baselines per region. "N/A" marks a baseline without a valid output for this date.

On all six dates the field of AeroMosaic follows the large-scale pattern of the ground truth: the dark south-west and lighter east of BTH, the decrease from west to east on SDP and CJD, and the nearly uniform low-risk fields of YRD and CYB. The only AeroMosaic panel with a highlighted region is ZHD, where the north-western corner is paler than the ground truth; AirPhyNet, SOFTS, GAGNN, and TimeXer deviate in the same corner, so it is hard for all models on this date. The baseline deviations take two forms. Some outputs are too pale: CATS on BTH and CJD returns an almost uniform light field where the ground truth is red, and SparseTSF on CYB lies below the ground truth on both flanks. Others are too intense or misplaced: AirPhyNet on CJD puts a compact high-risk core in the centre of the region, and TimeMixer on SDP raises the level over a broad area in the centre. On the CYB date, GAGNN has no valid output. These maps show single dates and complement the quantitative comparison in Table 2.

C.2 Per-Class Precision and Recall

Figure 9 reports precision and recall for each risk class (mean and standard deviation over three seeds) for AeroMosaic and four baselines in every region; the highest class (above 250) is empty in YRD, CJD, and CYB, where it has almost no test samples. AeroMosaic has the highest precision for nearly every class in all six regions, and the gap is largest on the intermediate classes: on BTH, the precision for lightly and moderately polluted days is about 0.74 and 0.59, against at most about 0.40 and 0.24 for the baselines. Its recall is also the highest for most classes. The exceptions are on the extreme classes, where the baselines trade precision for recall: on BTH, GAGNN reaches a recall of about 0.67 on the highest class at a precision of about 0.39, against 0.41 and 0.80 for AeroMosaic; on SDP, AirFormer and AirPhyNet exceed a recall of 0.8 on the highest class with a precision of about 0.15, whereas AeroMosaic has a recall of about 0.10 and a precision of about 0.31; on YRD, AirPhyNet has a higher recall on heavily polluted days (about 0.73 against 0.55). AeroMosaic

thus predicts the rarest classes more conservatively, and the Macro-F1 gains in Table 2 come mainly from precision on the intermediate and heavy classes rather than from recall on the rarest class. The error bars of the baselines on these rare classes are large, which reflects the small number of test samples.

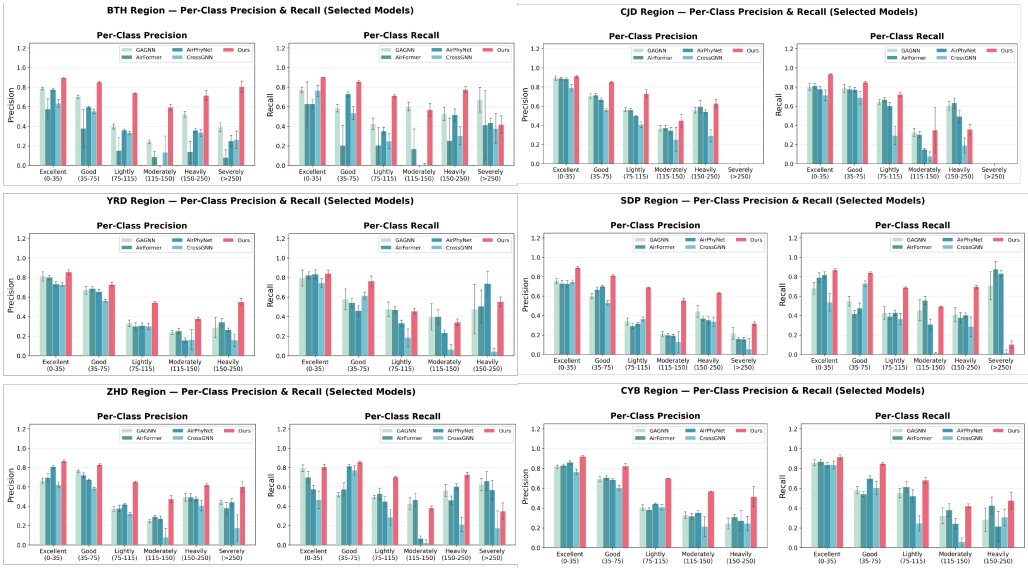


Figure 9: Per-class precision and recall in the six regions (mean and standard deviation over three seeds).

D Broader Impact and Responsible Use

This work studies $PM_{2.5}$ risk inference under a controlled evaluation protocol. It can support research on transport-aware air-quality analysis and improve the interpretability of regional pollution-risk mechanisms. Results obtained under the oracle parent-label protocol are not deployment-ready forecasts; using them as such could create false confidence in public warnings or policy decisions. The matched no-oracle diagnostics indicate the performance to expect without privileged upstream labels, which remains below the strongest tested numerical baseline.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Introduction state the task, the oracle parent-label protocol under which the main results are obtained, the matched no-oracle analyses, and the contributions on task formulation, daily wind-driven DAG construction, role-separated multimodal fusion, and empirical validation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section 5 discusses the oracle parent-label protocol as the main limitation and identifies recursive rollout of predicted upstream labels, learnable transport graphs, and rare-class support as future work; Appendix B.4 reports the no-oracle, oracle-access, inference-time, and precipitation-representation analyses.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not present formal theorems. Section 3.6 provides a branch-wise approximation view that motivates the design.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material.
- Any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper specifies the task formulation, year-based splits, model pipeline, forward-pass pseudocode, notation table, protocol and information-boundary audit, matched no-oracle controls, baseline families, random-seed reporting, and key implementation hyperparameters in the main text and appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: Our implementation, validation utilities, environment specification, and reproduction instructions are publicly available at `https://github.com/miya1102/aeromosaic_release_261018`. The repository includes a self-contained BTH no-oracle prepared bundle and documents public data pointers and all external requirements. We answer No because the six archived frozen regional bundles used for the exact main-table protocol and third-party baseline implementations are not redistributed; therefore the public package supports the documented software path but does not by itself faithfully regenerate every published main-table result.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

Justification: The Experiments section reports the six regional datasets, 2015–2016/2017/2018 train/validation/test split, group-level evaluation setup, model-selection metric, baselines, and metrics. Appendix B.1 reports the backbone instantiation, prefix-token count, numerical input channels, learning rate, epochs, precision, DAG-residual settings, and root-sensitive hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The main comparison and ablation tables report mean $\pm$ standard deviation over three random seeds, and their captions define the variability. The additional BTH analyses in Appendix B.4 are single runs, as stated in that appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests.
   - The factors of variability that the error bars are capturing should be clearly stated.
   - The method for calculating the error bars should be explained.
   - The assumptions made should be given when applicable.
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - If error bars are reported in tables or plots, the authors should explain how they were calculated and reference the corresponding figures or tables.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: The paper reports training precision, batch organization, epochs, key hyperparameters, and per-batch inference times of `AeroMosaic` on an NVIDIA H20 and of five baselines (Appendix B.4). The CPU model, system memory, training wall-clock time, and total project compute were not recorded.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each individual experimental run and estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

Justification: The work uses environmental time-series and regional air-quality data, does not involve human subjects, and discloses the oracle parent-label protocol used in the evaluation.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes].

Justification: The paper is motivated by air-quality risk inference, which has potential positive societal value for environmental analysis and public-health warnings. Appendix D discusses responsible use, including the role of the oracle parent-label protocol.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications.
- The authors should consider possible harms that could arise when the technology is used as intended and functioning correctly, when it gives incorrect results, and from misuse.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper does not release a high-risk general-purpose model, scraped personal dataset, image generator, or similarly dual-use asset. The studied model is a task-specific controlled air-quality risk-inference framework.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards.
- Datasets scraped from the Internet could pose safety risks.
- We recognize that providing effective safeguards is challenging, but encourage authors to make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No].

Justification: The paper credits the KnowAir/$PM_{2.5}$-GNN release, the Ministry of Ecology and Environment monitoring observations, ECMWF ERA5, the baseline methods, and the pretrained backbone. ERA5 and the backbone/code licenses are identified, and the

manuscript does not redistribute the processed KnowAir data, station metadata, or legacy terrain raster. We answer No because the legacy KnowAir release does not state an explicit redistribution license for all processed data and station metadata, and its quantized terrain raster does not document the upstream elevation product or license.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license should be included for each asset.
- For scraped data from a particular source, the copyright and terms of service should be provided.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset should be provided.
- If this information is not available online, authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The submission does not release a new public dataset, model checkpoint, or software asset as a primary contribution. The paper studies a method and provides documentation of the experimental protocol and implementation details.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate details of the dataset/code/model as part of their submissions via structured templates.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets if applicable.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing, human-subject experiments, user studies, or participant compensation.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

Justification: The paper does not involve human subjects, crowdsourcing, participant data collection, or interventions requiring IRB or equivalent review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval or equivalent may be required for human subjects research.
- For initial submissions, do not include information that would break anonymity.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes].

Justification: The method section describes the language-model backbone $\mathcal{B}_\theta$ used in the prefix–prompt pathway, and Appendix B.1 identifies the ChatTS-8B checkpoint (release 1103), its Qwen3-8B base, and the LoRA adaptation used in the implementation; the exact checkpoint revision is recorded in the released code.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as important, original, or non-standard components.
- Please refer to the NeurIPS LLM policy for what should or should not be described.