AeroChem: Closed-loop Physics-Informed State Space Modeling for Long-term Chemically-Reactive Air Quality Forecasting

Kien Le[*]
VinUniversity
Vietnam
kien.lt@vinuni.edu.vn

Tung Kieu[*]
Aalborg University
Denmark
tungkvt@cs.aau.dk

Dinh Duc Nha Nguyen
CECS, VinUniversity
Vietnam
nha.ndd@vinuni.edu.vn

Cuong Do
CECS, VinUniversity
Vietnam
cuong.dd@vinuni.edu.vn

Bin Yang
East China Normal University
China
byang@dase.ecnu.edu.cn

Nguyen Phi Long[†]
SEEE, Hanoi University of Industry
Vietnam
long.nguyen04@haui.edu.vn

Abstract

Accurate long-term air quality forecasting is challenging due to long-range transport, meteorological variability, and nonlinear chemical reactions underlying secondary pollutant formation. Purely data-driven models often suffer from long-horizon drift and lack physical consistency, while existing physics-informed approaches commonly follow open-loop designs that fuse neural and physical components only at the output level. We propose AeroChem, a closed-loop physics-informed state space framework for chemically reactive air quality forecasting. AeroChem adopts a two-branch co-correction design: a UDE-based process branch provides a structured but imperfect physical prior, while a Mamba-based virtual measurement branch captures long-range temporal context and produces probabilistic virtual observations. Neural Kalman Fusion couples the two branches by correcting the latent physical state at each forecasting step and feeding the posterior state back into the simulator, reducing long-horizon drift. We further introduce **HanoiAir**, a real-world dataset integrating pollutant concentrations, meteorology, and emission inventories. Experiments on **HanoiAir** and a public benchmark show that AeroChem improves long-horizon stability and sudden-change forecasting.

1 Introduction

Air pollution is a major global health and environmental challenge, motivating accurate air quality forecasting for public policy, emission control, and public-health protection [9, 38]. However, forecasting pollutant concentrations remains difficult because atmospheric dynamics are governed by coupled emission sources, meteorological conditions, transport processes, and chemical reactions [39]. This challenge is especially pronounced for secondary pollutants, such as ozone (O_3) and secondary components of $PM_{2.5}$, which arise from nonlinear photochemical reactions involving precursors such as NO_x and VOCs under meteorological drivers such as solar radiation, temperature, and humidity. These processes evolve over multiple hours or days, requiring both long-range temporal modeling and physical consistency.

Traditional Chemical Transport Models (CTMs) [26, 32], as illustrated in Fig. 1, encode atmospheric transport, diffusion, and chemical reactions through differential equations derived from physical

[*]Equal contribution.
[†]Corresponding author.

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

principles. Although interpretable, CTMs rely on simplified assumptions, incomplete reaction mechanisms, uncertain boundary conditions, and expensive solvers. In contrast, data-driven deep learning models learn predictive patterns directly from observations and often achieve strong short-term accuracy, but they may violate physical constraints, generalize poorly under distribution shifts, and accumulate errors during long-horizon autoregressive rollout.

Physics-Informed Machine Learning (PIML) [21, 3] aims to combine physical modeling with neural learning, but existing air quality forecasting methods remain limited. Many over-simplify atmospheric chemistry as linear source–sink terms or generic neural residuals, which is insufficient for nonlinear and condition-dependent secondary pollutant formation. More importantly, most hybrid architectures follow an open-loop design, where physical and data-driven branches are executed separately and fused only at the output level. Such designs may correct the final prediction, but they cannot correct the latent physical state propagated to future steps, allowing errors from misspecified physics, unobserved emissions, and stochastic atmospheric disturbances to accumulate.

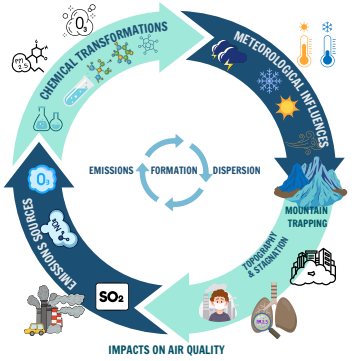


Figure 1: Impacts on Air Quality.

In this work, we argue that long-term chemically reactive forecasting requires a closed-loop design, where physics and data-driven evidence correct each other during rollout. The physical branch should provide a structured dynamical prior, while the data-driven branch should provide observation-conditioned evidence for correcting the evolving state. Crucially, this correction should occur in latent state space before the next simulation step, rather than only at the final output. This view turns long-horizon forecasting into recurrent posterior-state inference under imperfect physical dynamics.

To realize this idea, we propose `AeroChem`, a closed-loop physics-informed state space framework for long-term chemically reactive air quality forecasting. `AeroChem` couples two interacting branches through Neural Kalman Fusion. The first is a UDE-based process branch that acts as a differentiable simulator and explicitly separates transport dynamics from chemical dynamics, using diffusion, wind-driven advection, and a meteorology-gated neural reaction operator. The second is a Mamba-based virtual measurement branch that captures long-range temporal dependencies from historical observations and produces probabilistic virtual measurements. At each forecasting step, Neural Kalman Fusion combines the process prior and virtual measurement into a posterior latent state, which is recursively fed back into the process branch. This closed-loop co-correction mechanism reduces long-horizon drift and improves robustness in open, chemically reactive environments by allowing data-driven evidence and physics-informed dynamics to correct each other during rollout.

We evaluate `AeroChem` on two real-world air quality datasets, including **HanoiAir**, a newly constructed regional dataset integrating pollutant concentrations, meteorological variables, and emission inventories. Experiments show that `AeroChem` is particularly effective in long-horizon forecasting and sudden pollution-change scenarios, where open-loop and purely data-driven models often suffer from drift or over-smoothing. Ablation studies further demonstrate the importance of closed-loop feedback, meteorology-gated reaction modeling, and neural state correction.

In summary, this work makes the following contributions: (i) We formulate long-term chemically reactive air quality forecasting as closed-loop latent-state inference under imperfect physical dynamics. (ii) We propose `AeroChem`, a two-branch physics-informed state space framework that couples a UDE-based process branch and a Mamba-based measurement branch through Neural Kalman Fusion. (iii) We introduce a meteorology-gated neural reaction mechanism to model nonlinear secondary pollutant formation conditioned on emissions and meteorological drivers. (iv) We construct and release **HanoiAir**, a comprehensive regional air quality dataset containing pollutant concentrations, meteorological variables, and emission inventories. (v) We demonstrate that `AeroChem` improves long-horizon forecasting stability and sudden-change prediction across real-world air quality datasets.

2 Preliminaries

Sensor Network. We model an air quality monitoring system as a weighted static graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of N monitoring stations, $\mathcal{E}$ represents the set

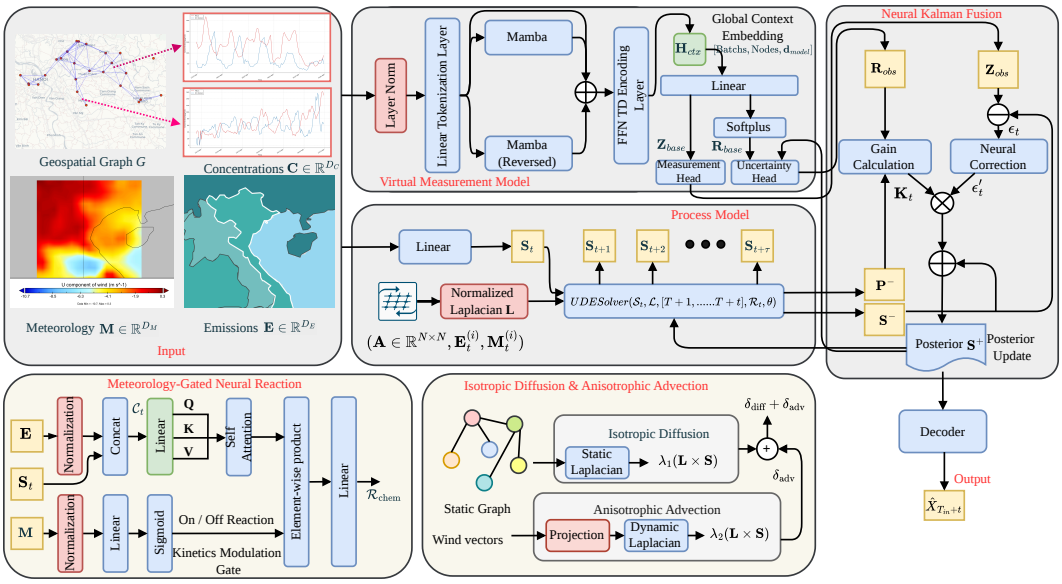


Figure 2: `AeroChem` architecture. A S-Mamba-based virtual measurement branch generates adaptive virtual observations, while a UDE-based process branch models transport and meteorology-gated neural reactions as a structured physical prior. Neural Kalman Fusion couples the two branches through a learnable correction term, producing a posterior state that is recursively fed back into the UDE. This closed-loop neural state correction allows virtual measurements and physics-informed dynamics to correct each other, mitigating long-horizon drift.

of edges, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a weighted adjacency matrix encoding spatial correlations between stations, such as geographic proximity. Each station v_i is associated with fixed spatial coordinates $\mathbf{p}_i \in \mathbb{R}^2$.

Multivariate Observations. At each discrete time step t, the observed system state is denoted as $\mathbf{X}_t \in \mathbb{R}^{N\times D_{\text{in}}}$, where each station is associated with a D_{in}-dimensional feature vector. We decompose $\mathbf{X}_t$ into three components: (1) **Pollutant concentrations** $\mathbf{C}_t$: Pollutant concentrations are the primary forecast targets, captured via air pollutant variables, including target pollutants (e.g., $PM_{2.5}$) and related co-pollutants (e.g., O_3, NO_x); (2) **Meteorological variables** $\mathbf{M}_t$: environmental drivers including scalar fields (e.g., temperature, humidity, solar radiation) and vector fields (e.g., wind speed and direction); (3) **Emission rates** $\mathbf{E}_t$: emission inventories describe source-intensity inputs, i.e., estimated pollutant or precursor releases into the atmosphere, including NO_x, VOCs, SO_2, and primary $PM_{2.5}$. Here, $\mathbf{C}_t$ represents pollutant levels already present in the atmosphere and serves as the main prediction target, whereas $\mathbf{E}_t$ represents external emission forcing that contributes to future pollutant formation and transport. Given an input window of length T_{in}, the historical observation sequence is denoted as $\mathbf{X}_{1:T_{\text{in}}} \in \mathbb{R}^{T_{\text{in}}\times N\times D_{\text{in}}}$.

Problem Formulation. Given the sensor graph $\mathcal{G}$ and historical observations $\mathbf{X}_{1:T_{\text{in}}}$, our objective is to learn a probabilistic forecasting function $\mathcal{F}_\Theta$ that estimates the posterior distribution of future observations over a prediction horizon T_{out}: $p(\mathbf{X}_{T_{\text{in}}+1:T_{\text{in}}+T_{\text{out}}} \mid \mathbf{X}_{1:T_{\text{in}}}, \mathcal{G})$. This formulation enables both point prediction and uncertainty estimation for spatiotemporal air quality forecasting.

3 Methodology

To address the probabilistic forecasting problem defined in Sec. 2, we propose `AeroChem`, a closed-loop physics-informed state space framework for long-term air quality forecasting. Unlike prior hybrid approaches that loosely couple physical and data-driven components, `AeroChem` follows a *closed-loop neural state correction* paradigm. At each forecasting step, a physics-informed process branch provides a structured but imperfect prior over the latent atmospheric state, while a virtual measurement branch produces data-driven virtual observations from historical context and the current rollout state. Neural Kalman Fusion couples the two branches by correcting the latent physical state before it is propagated to the next step. For clarity, we summarize the complete closed-loop forecasting procedure in Appx. A.1.

3.1 Framework Overview

At each forecasting step, `AeroChem` integrates three tightly coupled components: (1) a **Virtual Measurement Branch** that extracts long-range temporal context from historical observations and produces probabilistic virtual measurements; (2) a **Process Branch** formulated as a stochastic Universal Differential Equation (UDE) that models transport and chemically reactive dynamics as a structured physical prior; and (3) a **Neural Kalman Fusion** module that fuses the virtual measurement with the UDE prior to produce a corrected posterior latent state. The posterior state is recursively fed back into the process branch, forming a closed-loop co-correction system that mitigates error accumulation in long-horizon forecasting. The overall architecture is illustrated in Fig. 2.

3.2 Virtual Measurement Branch: Context-Conditioned Virtual Observations

The virtual measurement branch serves as a *context-conditioned virtual sensor* that generates probabilistic virtual observations from historical data. Unlike a standard observation model, it does not directly map the latent state to real measurements; instead, it produces adaptive virtual observations that guide the correction of the latent physical state during rollout. Given the input sequence $\mathbf{X}_{1:T_{\text{in}}}$, we employ **S-Mamba** [30], a selective state space model, to efficiently capture long-range temporal dependencies. The encoder produces a global context embedding $\mathbf{H}_{\text{ctx}} \in \mathbb{R}^{N \times d_{\text{model}}}$. We then convert this context into a base virtual observation and its uncertainty via two projection heads:

$$\mathbf{H}_{\text{ctx}} = \text{S-Mamba}(\mathbf{X}_{1:T_{\text{in}}}), \qquad \mathbf{Z}_{\text{base}}, \mathbf{R}_{\text{base}} = \text{ProjHeads}(\mathbf{H}_{\text{ctx}}). \tag{1}$$

To enable efficient autoregressive decoding, we decompose the virtual measurement into static and dynamic components:

$$\mathbf{Z}_{\text{obs},t} = \mathbf{Z}_{\text{base}} + \alpha \tanh(\mathbf{S}^{+}_{t-1}), \quad \mathbf{R}_{\text{obs},t} = \text{Softplus}(\mathbf{R}_{\text{base}}) \left(1 + \beta \frac{t}{T_{\text{out}}}\right), \tag{2}$$

where $\mathbf{S}^{+}_{t-1}$ denotes the posterior latent state from the previous step. The first equation allows the virtual observation to adapt to the evolving trajectory without accessing future pollutant observations, while the second models the growth of epistemic uncertainty over the forecasting horizon. Both α and β are learnable scalars.

3.3 Process Model: Universal Differential Equation

We model the latent physical state $\mathbf{S}_t \in \mathbb{R}^{N \times D}$ as a continuous-time stochastic process governed by a Universal Differential Equation:

$$d\mathbf{S}_t = \underbrace{\mathcal{F}_{\text{drift}}(\mathbf{S}_t, \mathbf{E}_t, \mathbf{M}_t)}_{\text{Deterministic Dynamics}}\, dt + \underbrace{\mathbf{G}(\mathbf{S}_t)}_{\text{Stochastic Diffusion}}\, d\mathbf{W}_t, \tag{3}$$

where $\mathbf{W}_t$ denotes a Wiener process. Here, $\mathbf{M}_t$ and $\mathbf{E}_t$ are treated as exogenous drivers at forecast step t; their inference-time availability and preprocessing are specified in Sec. 5. The stochastic term captures unresolved atmospheric variability, boundary leakage, and unobserved emissions that cannot be explicitly modeled by deterministic physics alone. The drift function $\mathcal{F}_{\text{drift}}$ explicitly disentangles transport and chemical reaction processes.

Isotropic Diffusion. Background dispersion is modeled using a normalized graph Laplacian $\mathbf{L}_{\text{diff}} \in \mathbb{R}^{N \times N}$ derived from the static sensor graph: $\delta_{\text{diff}} = -\lambda_1 \mathbf{L}_{\text{diff}} \mathbf{S}_t$, where $\lambda_1 \geq 0$ is a learnable diffusion coefficient constrained via `Softplus`. The Laplacian operator acts on the spatial dimension and is broadcast across feature channels, enforcing flux from high- to low-concentration regions.

Anisotropic Advection. To capture wind-driven transport, we construct a time-varying advection operator $\mathbf{L}_{\text{adv}}(t)$. For each directed edge (v_i, v_j), we define a unit direction vector: $\mathbf{e}_{ij} = \dfrac{(\mathbf{p}_j - \mathbf{p}_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|_2}$. Given wind velocity $\mathbf{v}_{i,t}$ at node i, dynamic edge weights are computed as: $w_{ij}(t) = \text{ReLU}\left(\dfrac{\mathbf{v}_{i,t} \cdot \mathbf{e}_{ij}}{\|\mathbf{v}_{i,t}\|_2}\right) \|\mathbf{v}_{i,t}\|_2$. These weights define $\mathbf{L}_{\text{adv}}(t)$, yielding the advection drift $\delta_{\text{adv}} = -\lambda_2 \mathbf{L}_{\text{adv}}(t) \mathbf{S}_t$, where $\lambda_2 \geq 0$ is learnable.

Meteorology-Gated Neural Reaction. Secondary pollutant formation is governed by nonlinear, condition-dependent chemical interactions. We model these dynamics using a meteorology-gated

neural reaction mechanism. First, latent reactant features are constructed as $\mathbf{H}_{\text{react}} = \text{Concat}(\mathbf{S}_t, \mathbf{E}_t)$. Self-attention is applied to learn interaction structure:

$$\mathbf{Q} = \mathbf{H}_{\text{react}}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{H}_{\text{react}}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{H}_{\text{react}}\mathbf{W}_V. \tag{4}$$

Rather than recovering explicit stoichiometric coefficients, the attention mechanism learns conditional interaction affinities between latent species under uncertainty.

To enforce environmental dependence, we introduce a meteorological gate:

$$\mathcal{R}_{\text{chem}} = \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \odot \sigma(\text{Linear}(\mathbf{M}_t^{\text{sc}})), \tag{5}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{M}_t^{\text{sc}}$ denotes scalar meteorological variables (e.g., radiation, temperature). This gating mechanism suppresses reactions under unfavorable conditions and acts as a data-driven analog of reaction rate modulation.

The full drift is given by: $\mathcal{F}_{\text{drift}} = \delta_{\text{diff}} + \delta_{\text{adv}} + \mathcal{R}_{\text{chem}}$.

3.4 Neural Kalman Fusion

To couple the virtual measurement branch with the physics-informed process branch, we employ a Neural Kalman Fusion module inspired by recursive filtering [22]. For scalability, we adopt a diagonal covariance approximation, reducing complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$.

Prediction. Starting from $\mathbf{S}_{t-1}^+$, the UDE solver produces a prior mean $\mathbf{S}_t^-$. Process uncertainty is estimated as

$$\mathbf{P}_t^- = \text{Softplus}(\text{Linear}(\mathbf{S}_t^-)). \tag{6}$$

Correction. Given the virtual observation $(\mathbf{Z}_{\text{obs},t}, \mathbf{R}_{\text{obs},t})$, the update equations are:

$$\begin{gathered}\mathbf{K}_t = \mathbf{P}_t^- \oslash \left(\mathbf{P}_t^- + \mathbf{R}_{\text{obs},t}\right), \quad \boldsymbol{\epsilon}_t = \mathbf{Z}_{\text{obs},t} - \mathbf{S}_t^-, \quad \boldsymbol{\epsilon}_t' = \boldsymbol{\epsilon}_t + \text{Linear}([\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t}]), \\ \mathbf{S}_t^+ = \mathbf{S}_t^- + \mathbf{K}_t \odot \boldsymbol{\epsilon}_t', \quad \mathbf{P}_t^+ = (\mathbf{1} - \mathbf{K}_t) \odot \mathbf{P}_t^-.\end{gathered} \tag{7}$$

Here, $\oslash$ denotes element-wise division. The learnable correction term allows the update to adapt to non-Gaussian residuals caused by open-system dynamics, such as boundary leakage, unobserved emissions, and model misspecification.

Closed-loop Feedback. The posterior state $\mathbf{S}_t^+$ is recursively fed back as the initial condition for the next prediction step. Thus, the virtual measurement branch and the process branch correct each other during rollout: virtual measurements correct the misspecified physical trajectory, while the UDE prior regularizes noisy data-driven evidence through structured transport–reaction dynamics. This closed-loop neural state correction reduces long-horizon drift.

3.5 Theoretical Interpretation

We provide two theoretical interpretations of `AeroChem` in the Appendix. First, Appx. A.2 shows that Neural Kalman Fusion can be viewed as an approximate variational correction of an imperfect physical prior. The UDE prediction defines a physics-informed prior over the latent state, while the virtual measurement branch provides a data-driven correction signal. When the residual potential of the open system is Gaussian, the update reduces to the diagonal Kalman correction. When the residual potential is non-Gaussian, a first-order approximation motivates the learnable affine correction term in Eq. 7. This interpretation clarifies why the correction term is not merely a heuristic residual layer, but an approximation to a non-Gaussian posterior-state update.

Second, Appx. A.3 analyzes why closed-loop correction improves long-horizon stability. Under standard Lipschitz and bounded-noise assumptions, the posterior error of the closed-loop system remains uniformly bounded when the correction factor is contractive. In contrast, the corresponding open-loop variant, which propagates the UDE prior without posterior feedback, can amplify initial errors over time. This analysis supports the central design of `AeroChem`: correcting the latent physical state before the next rollout step is more stable than correcting only the final output. Moreover, computational complexity is provided in App A.4.

3.6 Output Generation

After closed-loop neural state correction, we obtain a sequence of posterior latent states $\{\mathbf{S}_t^+\}_{t=1}^{T_{\text{out}}}$ and their corresponding covariance matrices $\{\mathbf{P}_t^+\}_{t=1}^{T_{\text{out}}}$. To generate probabilistic forecasts, we employ a decoding module that maps these latent distributions to the observation space. Specifically, the predictive *mean* and *variance* are decoupled:

$$\hat{\mathbf{X}}_{T_{\text{in}}+t} = \mathcal{D}_{\text{out}}(\mathbf{S}_t^+), \quad \boldsymbol{\sigma}^2_{T_{\text{in}}+t} = \text{Softplus}(\mathcal{D}_{\text{var}}(\mathbf{P}_t^+)), \tag{8}$$

for $t = 1, \ldots, T_{\text{out}}$. Here, $\mathcal{D}_{\text{out}}$ and $\mathcal{D}_{\text{var}}$ are shallow multilayer perceptrons with shared parameters across time and nodes. This ensures that the final output provides a fully parameterized predictive distribution $p(\mathbf{X}_t \mid \mathbf{S}_t^+) = \mathcal{N}(\hat{\mathbf{X}}_t, \text{diag}(\boldsymbol{\sigma}_t^2))$ to quantify epistemic uncertainty.

3.7 Training Objective

The model is trained end-to-end by minimizing:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MAE}}(\hat{\mathbf{X}}, \mathbf{X}) + \lambda_{\text{nll}}\, \mathcal{L}_{\text{NLL}}(\hat{\mathbf{X}}, \boldsymbol{\sigma}^2, \mathbf{X}) + \lambda_{\text{phy}}\, \|\mathcal{F}_{\text{drift}}\|_F^2, \tag{9}$$

where $\hat{\mathbf{X}}_t = \mathcal{D}_{\text{out}}(\mathbf{S}_t^+)$ and $\boldsymbol{\sigma}_t^2 = \text{Softplus}(\mathcal{D}_{\text{var}}(\mathbf{P}_t^+))$ are the predicted mean and variance decoded from the Kalman posterior, $\mathcal{L}_{\text{NLL}}(\hat{\mathbf{X}}, \boldsymbol{\sigma}^2, \mathbf{X}) = \frac{1}{2}\log \boldsymbol{\sigma}^2 + \frac{(\mathbf{X} - \hat{\mathbf{X}})^2}{2\boldsymbol{\sigma}^2}$ is the heteroscedastic Gaussian NLL, and the physics regularization penalizes excessively large drift magnitudes, encouraging stable transport–reaction dynamics. To mitigate exposure bias during autoregressive training, we employ scheduled sampling. Early in training, the process model is conditioned on $\text{StopGrad}(\mathbf{Z}_{\text{obs}})$ with a probability that decays exponentially, before transitioning to fully closed-loop operation.

4 Related Work

Spatio-temporal Air Quality Forecasting. Early air quality forecasting relied on statistical models such as `ARIMA` [14] and `VAR` [35], which are limited by linear assumptions. Deep learning methods subsequently improved predictive accuracy by modeling complex spatiotemporal dependencies, including RNN-based architectures and CNNs for gridded data. To handle non-Euclidean sensor networks, graph-based models such as `DCRNN` [13] and `STGCN` [34] were introduced and later adapted to air quality forecasting in `PM2.5-GNN` [29]. More recent approaches leverage Transformers (e.g., `iTransformer` [15], [17]) and structured State Space Models such as `Mamba` [6] to capture long-range temporal dependencies efficiently. Despite strong short-term performance, these data-driven methods do not explicitly model transport or chemical dynamics, and their predictions are usually generated in an open-loop autoregressive manner. As a result, they may accumulate errors over long horizons and over-smooth abrupt secondary pollution events driven by meteorological and chemical factors. In contrast, `AeroChem` uses long-range sequence modeling only as a virtual measurement branch, which is coupled with a physics-informed process branch through closed-loop latent-state correction.

Physics-Informed Machine Learning. Physics-informed machine learning (PIML) incorporates physical knowledge into neural models to improve interpretability and robustness. `PINNs` [21] enforce governing PDEs through soft constraints but are difficult to optimize for high-dimensional atmospheric systems. `Neural ODEs` [2] and their extensions have enabled continuous-time modeling of physical dynamics. In air quality forecasting, `AirPhyNet` [8] models diffusion and advection but assumes a closed system, while `Air-DualODE` [28] addresses open-system dynamics using a dual-branch design. Universal Differential Equations (UDEs) [20] further allow neural approximation of unknown physical terms. However, existing PIML approaches for air quality forecasting often oversimplify chemical reactions and couple physical and neural components only loosely, for example by combining their outputs rather than correcting the evolving latent physical state. This open-loop coupling limits robustness in long-horizon forecasting because errors from misspecified physics can still be propagated forward. In contrast, `AeroChem` formulates the UDE as a structured but imperfect process branch and repeatedly corrects its latent state using context-conditioned virtual measurements.

Deep Data Assimilation and Kalman Filtering. Data assimilation combines imperfect dynamical models with observations through recursive state estimation. Classical approaches such as the Kalman Filter [10], Extended Kalman Filter [4], and Ensemble Kalman Filter are widely used in geosciences,

weather forecast, and control systems but rely on linear-Gaussian assumptions or costly ensembles and require hand-designed uncertainty models. Learning-based approaches relax these assumptions by learning filtering components from data, including Deep Kalman Filters [11] and KalmanNet [22]. These methods motivate recursive correction, but they typically assume access to real observations or a standard observation model. In our setting, future pollutant observations are unavailable during forecasting; therefore, `AeroChem` does not use a standard observation model. Instead, its virtual measurement branch generates context-conditioned virtual observations from historical context and the previous posterior state. Neural Kalman Fusion then fuses these virtual measurements with the UDE prior to correct the latent physical state at each rollout step. Thus, `AeroChem` is best viewed as an assimilation-inspired closed-loop neural state correction framework, rather than a classical data-assimilation model.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate `AeroChem` on two real-world hourly air quality datasets: **KnowAir** and **HanoiAir**. **KnowAir** contains 355 monitoring stations from 2016–2023 and integrates pollutant concentrations, meteorological variables, and emission inventories. **HanoiAir** is a newly constructed dataset covering northern Vietnam with 27 monitoring stations from 2018–2025, including pollutant concentrations, meteorology, and downscaled emission inputs. For both datasets, we use chronological splits to evaluate future forecasting: **KnowAir** uses 2016–2019 for training, 2020–2021 for validation, and 2022–2023 for testing; **HanoiAir** uses 2018–2021 for training, 2022–2023 for validation, and 2024–2025 for testing. Detailed dataset descriptions and preprocessing procedures are provided in Appx. A.5.

Baselines. We compare `AeroChem` against a diverse set of baseline models spanning several categories: (1) **Statistical Methods:** Standard time series models including `HA` (Historical Average) and `VAR` (Vector Autoregression) [35]. (2) **Neural ODE / Physics-Informed Models:** Methods incorporating physics-informed models, such as `LatentODE` [2], `ODE-LSTM` [12], `Air-DualODE` [28], and `AirPhyNet` [8]. (3) **GNN-based Spatio-Temporal Models:** Models explicitly designed for graph-structured time series data, including `STGCN` [34], `DCRNN` [13], `GTS` [23], and `PM25GNN` [29]. (4) **Linear and MLP-based Models:** Efficient architectures focusing on simpler mappings, such as `DLinear` [36] and `ModernTCN` [1]. (5) **Transformer-based Models:** Models leveraging attention mechanisms, including `Fredformer` [19], `PatchTST` [17], `Crossformer` [37], and `TimeMoE` [25]. (6) **State Space Models (SSMs):** Models based on structured state space representations, including `S-Mamba` [6]), `S4` [7], and `BiMamba` [27].

Metrics. To quantitatively assess the performance of `AeroChem` and baseline models, we employ three standard forecasting metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), Continuous Ranked Probability Score (CRPS), Summed CRPS (CRPS_{sum}), and Prediction Interval Coverage Probability (PICP). Formal definitions of these metrics are provided in Appx. A.6.

Hyperparameter & Implementation Details. Detailed hyperparameter settings are provided in Appx. A.7, and implementation details are included in Appx. A.8. An implementation is available at *Github link*.

Forecasting Scenarios. We evaluate four forecasting scenarios: **3 days** (1–72 hours), **6 days** (1–144 hours), **9 days** (1–216 hours), and **Sudden Change** events [28]. Sudden changes are defined as test instances where the ground-truth $PM_{2.5}$ concentration exceeds 75 $\mu\text{g}/\text{m}^3$ and changes by more than ± 20 $\mu\text{g}/\text{m}^3$ within the next three hours. All metrics are computed after inverse transformation to the original physical scale.

5.2 Main Results

Tables 1 and 2 report the main $PM_{2.5}$ forecasting results on KnowAir and HanoiAir. Overall, `AeroChem` is most effective in long-horizon and sudden-change scenarios, where open-loop autoregressive and purely data-driven models are more prone to drift or over-smoothing. Results for O_3 are provided in Appx. A.13.

Transformer-based and SSM-based models, such as `TimeMoE`, `PatchTST`, and `S-Mamba`, achieve competitive short-term accuracy, but their errors generally increase as the horizon becomes longer. In contrast, `AeroChem` maintains more stable long-horizon performance. This supports our central

Table 1: Performance comparison on the **KnowAir** dataset. The best results are highlighted in **bold**, and the second-best results are underlined. Mean ± std over 5 runs.

Methods	3 Days			6 Days			9 Days			Sudden Change		
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
HA	35.20$_{\pm 0.05}$	51.50$_{\pm 0.07}$	0.65$_{\pm 0.001}$	37.80$_{\pm 0.06}$	54.20$_{\pm 0.08}$	0.68$_{\pm 0.002}$	40.50$_{\pm 0.07}$	58.10$_{\pm 0.10}$	0.71$_{\pm 0.002}$	55.40$_{\pm 0.10}$	75.20$_{\pm 0.14}$	0.75$_{\pm 0.003}$
VAR	33.80$_{\pm 0.08}$	49.20$_{\pm 0.10}$	0.62$_{\pm 0.002}$	36.50$_{\pm 0.09}$	52.40$_{\pm 0.13}$	0.66$_{\pm 0.003}$	39.20$_{\pm 0.11}$	56.50$_{\pm 0.16}$	0.69$_{\pm 0.003}$	53.10$_{\pm 0.16}$	72.40$_{\pm 0.22}$	0.72$_{\pm 0.004}$
LatentODE	26.50$_{\pm 0.15}$	42.10$_{\pm 0.21}$	0.55$_{\pm 0.004}$	28.20$_{\pm 0.18}$	45.40$_{\pm 0.25}$	0.57$_{\pm 0.005}$	31.60$_{\pm 0.22}$	49.20$_{\pm 0.30}$	0.61$_{\pm 0.006}$	48.40$_{\pm 0.36}$	65.10$_{\pm 0.49}$	0.65$_{\pm 0.009}$
ODE-LSTM	27.10$_{\pm 0.16}$	43.50$_{\pm 0.22}$	0.56$_{\pm 0.004}$	29.45$_{\pm 0.19}$	46.80$_{\pm 0.26}$	0.59$_{\pm 0.005}$	32.80$_{\pm 0.23}$	50.15$_{\pm 0.31}$	0.62$_{\pm 0.006}$	49.80$_{\pm 0.38}$	67.30$_{\pm 0.51}$	0.67$_{\pm 0.010}$
Air-DualODE	23.04$_{\pm 0.14}$	38.44$_{\pm 0.19}$	0.52$_{\pm 0.003}$	24.50$_{\pm 0.17}$	41.60$_{\pm 0.23}$	0.54$_{\pm 0.004}$	25.20$_{\pm 0.20}$	42.80$_{\pm 0.27}$	0.59$_{\pm 0.005}$	39.24$_{\pm 0.32}$	53.30$_{\pm 0.43}$	0.55$_{\pm 0.008}$
AirPhyNet	23.80$_{\pm 0.10}$	39.50$_{\pm 0.14}$	0.53$_{\pm 0.003}$	25.10$_{\pm 0.12}$	41.80$_{\pm 0.17}$	0.55$_{\pm 0.004}$	27.20$_{\pm 0.15}$	44.50$_{\pm 0.20}$	0.57$_{\pm 0.004}$	38.50$_{\pm 0.24}$	53.10$_{\pm 0.32}$	0.56$_{\pm 0.006}$
STGCN	25.50$_{\pm 0.13}$	40.80$_{\pm 0.17}$	0.54$_{\pm 0.003}$	27.80$_{\pm 0.15}$	43.20$_{\pm 0.20}$	0.57$_{\pm 0.004}$	31.50$_{\pm 0.19}$	47.40$_{\pm 0.25}$	0.61$_{\pm 0.005}$	46.50$_{\pm 0.30}$	62.10$_{\pm 0.40}$	0.64$_{\pm 0.008}$
DCRNN	24.80$_{\pm 0.12}$	39.60$_{\pm 0.16}$	0.53$_{\pm 0.003}$	26.90$_{\pm 0.14}$	42.50$_{\pm 0.18}$	0.56$_{\pm 0.004}$	30.10$_{\pm 0.17}$	46.80$_{\pm 0.23}$	0.60$_{\pm 0.005}$	44.20$_{\pm 0.27}$	60.40$_{\pm 0.37}$	0.62$_{\pm 0.008}$
GTS	24.10$_{\pm 0.11}$	38.50$_{\pm 0.15}$	0.52$_{\pm 0.003}$	26.50$_{\pm 0.13}$	41.10$_{\pm 0.17}$	0.55$_{\pm 0.004}$	29.40$_{\pm 0.16}$	45.50$_{\pm 0.21}$	0.59$_{\pm 0.005}$	42.50$_{\pm 0.25}$	58.20$_{\pm 0.34}$	0.60$_{\pm 0.007}$
PM25GNN	24.50$_{\pm 0.12}$	39.20$_{\pm 0.16}$	0.53$_{\pm 0.003}$	27.10$_{\pm 0.14}$	41.80$_{\pm 0.18}$	0.56$_{\pm 0.004}$	30.20$_{\pm 0.17}$	46.10$_{\pm 0.22}$	0.60$_{\pm 0.005}$	43.10$_{\pm 0.27}$	59.50$_{\pm 0.36}$	0.61$_{\pm 0.008}$
DLinear	22.57$_{\pm 0.08}$	37.80$_{\pm 0.11}$	0.51$_{\pm 0.002}$	23.85$_{\pm 0.10}$	39.50$_{\pm 0.13}$	0.53$_{\pm 0.003}$	24.50$_{\pm 0.12}$	41.20$_{\pm 0.16}$	0.55$_{\pm 0.004}$	38.45$_{\pm 0.20}$	51.22$_{\pm 0.27}$	0.55$_{\pm 0.006}$
ModernTCN	22.33$_{\pm 0.09}$	37.60$_{\pm 0.12}$	0.51$_{\pm 0.003}$	23.68$_{\pm 0.11}$	39.40$_{\pm 0.14}$	0.53$_{\pm 0.003}$	25.12$_{\pm 0.13}$	41.50$_{\pm 0.17}$	**0.54**$_{\pm 0.004}$	37.92$_{\pm 0.21}$	50.81$_{\pm 0.28}$	0.54$_{\pm 0.006}$
TimeMoE	21.68$_{\pm 0.09}$	37.20$_{\pm 0.12}$	**0.49**$_{\pm 0.002}$	23.15$_{\pm 0.11}$	38.95$_{\pm 0.14}$	**0.50**$_{\pm 0.003}$	25.32$_{\pm 0.13}$	40.15$_{\pm 0.17}$	0.55$_{\pm 0.004}$	36.54$_{\pm 0.21}$	**49.12**$_{\pm 0.28}$	0.53$_{\pm 0.006}$
Fredformer	**21.45**$_{\pm 0.08}$	38.10$_{\pm 0.12}$	0.51$_{\pm 0.002}$	24.12$_{\pm 0.11}$	40.20$_{\pm 0.15}$	0.53$_{\pm 0.003}$	26.48$_{\pm 0.14}$	42.80$_{\pm 0.18}$	0.56$_{\pm 0.004}$	38.25$_{\pm 0.22}$	51.03$_{\pm 0.30}$	0.55$_{\pm 0.006}$
PatchTST	22.69$_{\pm 0.09}$	38.50$_{\pm 0.12}$	0.51$_{\pm 0.003}$	23.91$_{\pm 0.11}$	40.50$_{\pm 0.14}$	0.51$_{\pm 0.003}$	25.25$_{\pm 0.13}$	42.80$_{\pm 0.18}$	0.56$_{\pm 0.004}$	39.12$_{\pm 0.22}$	52.34$_{\pm 0.30}$	0.56$_{\pm 0.006}$
Crossformer	22.00$_{\pm 0.09}$	37.90$_{\pm 0.12}$	0.52$_{\pm 0.003}$	23.75$_{\pm 0.11}$	39.80$_{\pm 0.14}$	0.54$_{\pm 0.003}$	26.10$_{\pm 0.13}$	42.10$_{\pm 0.17}$	0.56$_{\pm 0.004}$	37.55$_{\pm 0.21}$	50.42$_{\pm 0.29}$	0.54$_{\pm 0.006}$
SMamba	23.12$_{\pm 0.10}$	38.80$_{\pm 0.13}$	0.52$_{\pm 0.003}$	24.24$_{\pm 0.12}$	39.40$_{\pm 0.15}$	0.55$_{\pm 0.004}$	25.15$_{\pm 0.14}$	40.72$_{\pm 0.18}$	0.58$_{\pm 0.004}$	39.85$_{\pm 0.23}$	53.12$_{\pm 0.31}$	0.57$_{\pm 0.007}$
S4	24.53$_{\pm 0.12}$	40.20$_{\pm 0.16}$	0.54$_{\pm 0.003}$	25.18$_{\pm 0.13}$	40.51$_{\pm 0.17}$	0.57$_{\pm 0.004}$	26.21$_{\pm 0.15}$	41.23$_{\pm 0.19}$	0.60$_{\pm 0.005}$	42.14$_{\pm 0.25}$	56.55$_{\pm 0.34}$	0.61$_{\pm 0.008}$
BiMamba	22.84$_{\pm 0.10}$	38.50$_{\pm 0.13}$	0.52$_{\pm 0.003}$	24.07$_{\pm 0.12}$	39.10$_{\pm 0.15}$	0.55$_{\pm 0.003}$	25.63$_{\pm 0.14}$	40.88$_{\pm 0.18}$	0.58$_{\pm 0.004}$	39.25$_{\pm 0.23}$	52.64$_{\pm 0.31}$	0.56$_{\pm 0.007}$
AeroChem (**ours**)	21.95$_{\pm 0.06}$	**37.20**$_{\pm 0.09}$	0.50$_{\pm 0.002}$	**23.00**$_{\pm 0.08}$	**38.80**$_{\pm 0.11}$	0.52$_{\pm 0.003}$	**24.08**$_{\pm 0.10}$	**39.34**$_{\pm 0.14}$	**0.54**$_{\pm 0.004}$	**34.50**$_{\pm 0.18}$	49.50$_{\pm 0.25}$	**0.52**$_{\pm 0.005}$

Table 2: Performance comparison on **HanoiAir** dataset. The best results are highlighted in **bold**, and the second-best results are underlined. Mean ± std over 5 runs.

Methods	3 Days			6 Days			9 Days			Sudden Change		
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
HA	55.20$_{\pm 0.10}$	78.50$_{\pm 0.14}$	0.85$_{\pm 0.002}$	58.40$_{\pm 0.12}$	82.10$_{\pm 0.16}$	0.88$_{\pm 0.003}$	62.10$_{\pm 0.14}$	86.50$_{\pm 0.20}$	0.92$_{\pm 0.003}$	75.40$_{\pm 0.22}$	105.20$_{\pm 0.30}$	0.89$_{\pm 0.005}$
VAR	52.80$_{\pm 0.13}$	75.20$_{\pm 0.18}$	0.82$_{\pm 0.003}$	56.50$_{\pm 0.15}$	79.40$_{\pm 0.21}$	0.85$_{\pm 0.003}$	60.20$_{\pm 0.18}$	84.80$_{\pm 0.25}$	0.88$_{\pm 0.004}$	72.10$_{\pm 0.28}$	100.40$_{\pm 0.38}$	0.86$_{\pm 0.006}$
LatentODE	46.50$_{\pm 0.30}$	68.10$_{\pm 0.42}$	0.80$_{\pm 0.006}$	49.20$_{\pm 0.36}$	71.40$_{\pm 0.50}$	0.83$_{\pm 0.007}$	52.60$_{\pm 0.42}$	75.20$_{\pm 0.58}$	0.86$_{\pm 0.008}$	74.40$_{\pm 0.65}$	98.10$_{\pm 0.88}$	0.85$_{\pm 0.012}$
ODE-LSTM	47.10$_{\pm 0.32}$	69.50$_{\pm 0.45}$	0.81$_{\pm 0.006}$	50.45$_{\pm 0.38}$	72.80$_{\pm 0.52}$	0.84$_{\pm 0.007}$	54.80$_{\pm 0.45}$	78.15$_{\pm 0.62}$	0.87$_{\pm 0.008}$	75.80$_{\pm 0.70}$	101.30$_{\pm 0.94}$	0.86$_{\pm 0.012}$
Air-DualODE	42.04$_{\pm 0.28}$	64.41$_{\pm 0.40}$	0.78$_{\pm 0.005}$	45.50$_{\pm 0.34}$	67.60$_{\pm 0.46}$	0.82$_{\pm 0.006}$	48.20$_{\pm 0.40}$	71.80$_{\pm 0.54}$	0.86$_{\pm 0.007}$	72.24$_{\pm 0.58}$	93.30$_{\pm 0.78}$	0.83$_{\pm 0.010}$
AirPhyNet	43.80$_{\pm 0.20}$	65.50$_{\pm 0.28}$	0.79$_{\pm 0.004}$	46.10$_{\pm 0.24}$	69.80$_{\pm 0.33}$	0.82$_{\pm 0.005}$	49.20$_{\pm 0.28}$	73.50$_{\pm 0.39}$	0.85$_{\pm 0.005}$	71.50$_{\pm 0.42}$	94.10$_{\pm 0.58}$	0.79$_{\pm 0.008}$
STGCN	45.50$_{\pm 0.25}$	67.80$_{\pm 0.34}$	0.81$_{\pm 0.005}$	48.80$_{\pm 0.30}$	72.20$_{\pm 0.40}$	0.84$_{\pm 0.006}$	52.50$_{\pm 0.35}$	77.40$_{\pm 0.48}$	0.88$_{\pm 0.007}$	74.50$_{\pm 0.55}$	98.10$_{\pm 0.74}$	0.85$_{\pm 0.010}$
DCRNN	44.80$_{\pm 0.23}$	66.60$_{\pm 0.32}$	0.80$_{\pm 0.005}$	47.90$_{\pm 0.27}$	71.50$_{\pm 0.37}$	0.83$_{\pm 0.005}$	51.10$_{\pm 0.32}$	76.80$_{\pm 0.44}$	0.87$_{\pm 0.006}$	73.20$_{\pm 0.50}$	97.40$_{\pm 0.68}$	0.84$_{\pm 0.009}$
GTS	44.10$_{\pm 0.21}$	65.50$_{\pm 0.29}$	0.79$_{\pm 0.004}$	47.50$_{\pm 0.25}$	70.10$_{\pm 0.34}$	0.82$_{\pm 0.005}$	50.40$_{\pm 0.30}$	75.50$_{\pm 0.41}$	0.86$_{\pm 0.006}$	72.50$_{\pm 0.46}$	95.20$_{\pm 0.62}$	0.83$_{\pm 0.008}$
PM25GNN	44.50$_{\pm 0.22}$	66.20$_{\pm 0.30}$	0.80$_{\pm 0.005}$	48.10$_{\pm 0.26}$	71.80$_{\pm 0.36}$	0.83$_{\pm 0.005}$	51.20$_{\pm 0.31}$	76.10$_{\pm 0.42}$	0.87$_{\pm 0.006}$	73.10$_{\pm 0.48}$	96.50$_{\pm 0.65}$	0.83$_{\pm 0.009}$
DLinear	42.57$_{\pm 0.16}$	64.80$_{\pm 0.22}$	0.78$_{\pm 0.004}$	44.85$_{\pm 0.19}$	67.50$_{\pm 0.26}$	0.81$_{\pm 0.004}$	47.10$_{\pm 0.22}$	70.20$_{\pm 0.30}$	0.84$_{\pm 0.005}$	73.45$_{\pm 0.36}$	96.22$_{\pm 0.48}$	0.82$_{\pm 0.007}$
ModernTCN	42.33$_{\pm 0.17}$	64.60$_{\pm 0.23}$	0.78$_{\pm 0.004}$	44.68$_{\pm 0.20}$	67.40$_{\pm 0.27}$	0.81$_{\pm 0.004}$	47.50$_{\pm 0.23}$	70.50$_{\pm 0.31}$	0.84$_{\pm 0.005}$	72.92$_{\pm 0.37}$	95.81$_{\pm 0.50}$	0.81$_{\pm 0.007}$
TimeMoE	41.85$_{\pm 0.18}$	63.80$_{\pm 0.25}$	**0.77**$_{\pm 0.004}$	44.15$_{\pm 0.21}$	66.95$_{\pm 0.28}$	0.80$_{\pm 0.004}$	47.50$_{\pm 0.24}$	**69.15**$_{\pm 0.32}$	0.83$_{\pm 0.005}$	71.54$_{\pm 0.38}$	93.12$_{\pm 0.50}$	0.78$_{\pm 0.007}$
Fredformer	**41.60**$_{\pm 0.16}$	64.10$_{\pm 0.22}$	0.77$_{\pm 0.004}$	44.20$_{\pm 0.19}$	67.20$_{\pm 0.26}$	0.80$_{\pm 0.004}$	47.48$_{\pm 0.22}$	71.20$_{\pm 0.30}$	0.84$_{\pm 0.005}$	73.25$_{\pm 0.36}$	95.03$_{\pm 0.48}$	0.81$_{\pm 0.007}$
PatchTST	42.10$_{\pm 0.18}$	64.50$_{\pm 0.24}$	0.78$_{\pm 0.004}$	44.50$_{\pm 0.21}$	67.80$_{\pm 0.28}$	0.81$_{\pm 0.004}$	47.80$_{\pm 0.24}$	71.50$_{\pm 0.32}$	0.85$_{\pm 0.005}$	74.12$_{\pm 0.38}$	96.34$_{\pm 0.50}$	0.83$_{\pm 0.007}$
Crossformer	42.00$_{\pm 0.17}$	64.20$_{\pm 0.23}$	0.78$_{\pm 0.004}$	44.40$_{\pm 0.20}$	67.50$_{\pm 0.27}$	0.81$_{\pm 0.004}$	47.60$_{\pm 0.23}$	71.80$_{\pm 0.31}$	0.85$_{\pm 0.005}$	73.55$_{\pm 0.37}$	95.42$_{\pm 0.50}$	0.82$_{\pm 0.007}$
S-Mamba	43.12$_{\pm 0.20}$	65.80$_{\pm 0.27}$	0.79$_{\pm 0.005}$	45.24$_{\pm 0.23}$	66.40$_{\pm 0.31}$	0.82$_{\pm 0.005}$	47.15$_{\pm 0.26}$	69.57$_{\pm 0.35}$	0.86$_{\pm 0.006}$	75.85$_{\pm 0.42}$	98.12$_{\pm 0.56}$	0.84$_{\pm 0.008}$
S4	44.53$_{\pm 0.24}$	67.20$_{\pm 0.32}$	0.81$_{\pm 0.005}$	46.18$_{\pm 0.28}$	68.50$_{\pm 0.38}$	0.84$_{\pm 0.006}$	48.14$_{\pm 0.32}$	69.81$_{\pm 0.43}$	0.88$_{\pm 0.007}$	77.14$_{\pm 0.50}$	101.55$_{\pm 0.68}$	0.86$_{\pm 0.010}$
BiMamba	42.84$_{\pm 0.19}$	65.50$_{\pm 0.26}$	0.79$_{\pm 0.005}$	45.97$_{\pm 0.22}$	67.10$_{\pm 0.30}$	0.82$_{\pm 0.005}$	47.63$_{\pm 0.25}$	70.80$_{\pm 0.34}$	0.85$_{\pm 0.006}$	75.25$_{\pm 0.40}$	97.64$_{\pm 0.54}$	0.83$_{\pm 0.008}$
AeroChem (**ours**)	42.74$_{\pm 0.10}$	**63.11**$_{\pm 0.14}$	0.78$_{\pm 0.003}$	**44.50**$_{\pm 0.13}$	**65.80**$_{\pm 0.18}$	**0.79**$_{\pm 0.003}$	**46.80**$_{\pm 0.16}$	68.34$_{\pm 0.22}$	**0.81**$_{\pm 0.004}$	**68.80**$_{\pm 0.28}$	**91.16**$_{\pm 0.38}$	**0.75**$_{\pm 0.006}$

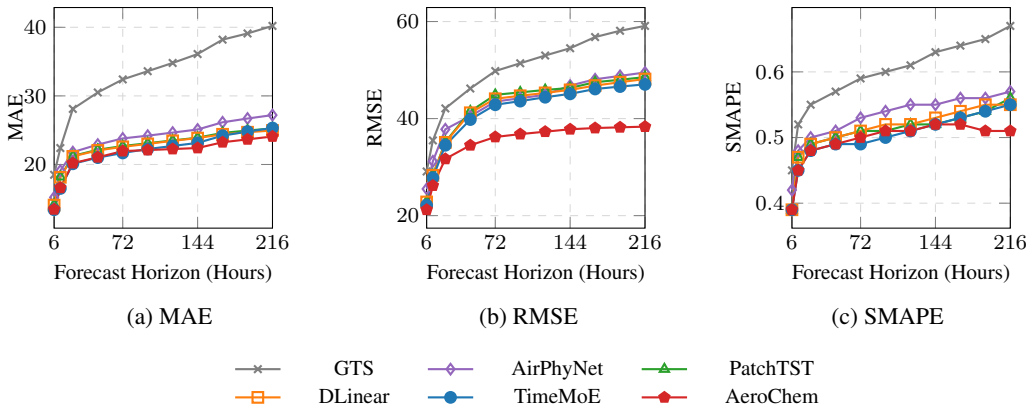
Table 3: Ablation Study: Effect of Physical Mechanisms and Models Design.

Method	**HanoiAir**						**KnowAir**					
	3 Days			**9 Days**			**3 Days**			**9 Days**		
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
AeroChem	**42.74**	**63.11**	**0.78**	**46.80**	**68.34**	**0.81**	**21.95**	**37.20**	**0.50**	**24.08**	**39.34**	**0.54**
Physical Mechanisms												
w/o Diffusion (AeroChem -D)	44.31	65.25	0.80	48.37	70.48	0.83	23.52	39.34	0.52	25.65	41.48	0.56
w/o Advection (AeroChem -A)	42.81	66.34	0.78	48.80	69.64	0.84	23.95	40.02	0.52	24.98	40.74	0.58
w/o Reaction (AeroChem -R)	45.15	66.80	0.82	49.60	71.90	0.85	24.10	40.50	0.53	26.80	42.90	0.59
w/o Physics (AeroChem -P)	47.85	69.50	0.86	53.20	75.40	0.90	26.50	43.80	0.58	29.10	46.50	0.63
Models & Fusion												
w/o Kalman (AeroChem -K)	56.40	78.20	0.95	63.50	85.10	1.05	30.20	48.50	0.65	35.50	54.20	0.72
w/ EnKF (AeroChem -EnKF)	43.62	62.45	0.81	47.85	67.20	0.84	24.08	35.12	0.59	25.07	36.06	0.61
w/o Neur. Corr (AeroChem -N)	43.10	63.37	0.79	47.16	68.60	0.82	22.31	37.46	0.51	24.44	39.60	0.55
w/o Mamba (AeroChem -M)	43.05	63.80	0.79	51.50	74.20	0.88	22.15	37.60	0.51	28.40	45.20	0.61
w/ PatchTST (AeroChem -Patch)	44.18	64.50	0.81	48.42	69.85	0.84	24.60	36.15	0.61	25.55	37.10	0.63
w/ DLinear (AeroChem -DLinear)	44.35	64.20	0.81	48.55	69.65	0.85	24.8	35.88	0.61	25.64	36.94	0.63
w/ iTrans (AeroChem -iTrans)	44.05	64.40	0.80	48.30	69.70	0.84	24.72	36.05	0.60	25.48	37.05	0.62
w/ Crossformer (AeroChem -Cross)	43.95	64.30	0.80	48.20	69.55	0.84	24.55	36.20	0.60	25.42	37.20	0.62
Late Fusion	53.25	76.10	0.92	59.80	81.50	0.98	29.40	47.10	0.63	33.80	51.50	0.69

design: the UDE-based process branch provides a structured physical prior, while Neural Kalman Fusion repeatedly corrects the latent state during rollout, reducing the error accumulation typical of open-loop forecasting.

The advantage of AeroChem is more evident on **HanoiAir**, which has fewer monitoring stations and noisier regional dynamics. Compared with strong direct-mapping models such as TimeMoE and ModernTCN, AeroChem achieves more stable performance, suggesting that closed-loop neural state correction improves robustness when observations are sparse or noisy.

Sudden pollution changes are particularly challenging because many baselines tend to smooth rapid concentration shifts. In the Sudden Change setting, AeroChem achieves stronger performance than



(a) MAE (b) RMSE (c) SMAPE

Figure 3: Performance across forecast horizons.

Transformer and SSM baselines. This improvement is consistent with the role of the meteorology-gated neural reaction module, which conditions chemical dynamics on environmental drivers such as radiation and temperature, helping the model capture nonlinear secondary pollution formation. Probabilistic forecasting results on **KnowAir** $PM_{2.5}$ are shown in Appx. A.9.

5.3 Ablation Study

We conduct ablation studies on **KnowAir** and **HanoiAir** to evaluate two aspects of `AeroChem`: (i) the physical mechanisms in the process branch and (ii) the architecture of the virtual measurement branch and fusion module. Table 3 shows the experimental results of the ablation studies.

Effect of Physical Mechanisms. We first remove individual components from the UDE drift function in Eq. 3: isotropic diffusion (`AeroChem-D`), anisotropic advection (`AeroChem-A`), the meteorology-gated neural reaction module (`AeroChem-R`), and all explicit physical terms (`AeroChem-P`). Removing all physical terms leads to clear long-horizon degradation, confirming that the structured process branch helps constrain rollout dynamics. Among the physical components, removing the reaction module causes substantial degradation, showing that meteorology-conditioned chemical dynamics are important for modeling secondary pollutant formation beyond transport alone.

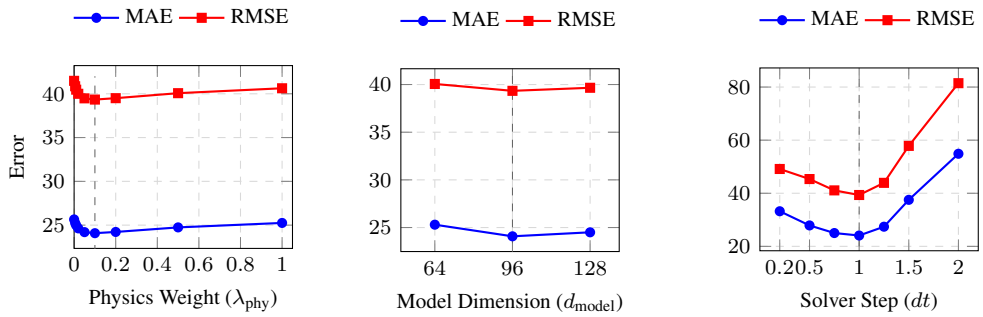
Effect of Virtual Measurements and Fusion. We next evaluate the virtual measurement branch and the closed-loop fusion design. We replace `S-Mamba` with a bidirectional `LSTM` (`AeroChem-M`) and with representative temporal encoders, including `PatchTST` (`AeroChem-Patch`), `DLinear` (`AeroChem-DLinear`), `iTransformer` (`AeroChem-iTrans`), and `Crossformer` (`AeroChem-Cross`). We also remove posterior feedback to obtain an open-loop UDE solver (`AeroChem-K`), remove the learnable correction term while keeping the diagonal Kalman update (`AeroChem-N`), replace Neural Kalman Fusion with an Ensemble Kalman Filter (`AeroChem-EnKF`), and decouple the process and virtual measurement branches using weighted output-level averaging (`Late Fusion`). We observe that removing closed-loop feedback causes the largest degradation, especially at longer horizons, confirming that correcting the latent state before the next rollout step is essential for stability. Removing the learnable neural correction also degrades performance, suggesting that a strictly Gaussian correction is insufficient for open environmental systems with boundary leakage, unobserved emissions, and model misspecification. Finally, `Late Fusion` performs worse than latent-state fusion, supporting our claim that the two branches should correct each other during rollout rather than being combined only at the output level.

5.4 Effect of Forecast Horizons

We show that `AeroChem` maintains significantly more stable long-term behavior than physics-guided baselines. Fig. 3 illustrates the error evolution across varying horizons. `AeroChem` exhibits superior stability compared to existing physics-guided baselines (e.g., `AirPhyNet`, `Air-DualODE`), with the performance gap widening significantly as the prediction window extends. While prior methods suffer from rapid error propagation, `AeroChem` maintains a flattened error trajectory, validating the effectiveness of our closed-loop design in mitigating long-term physical drift.

5.5 Hyperparameter Studies

We assess the model's robustness to three key hyperparameters. Fig. 4 shows the experimental results.



(a) Effect of Physics Weight (λ_{phy}) (b) Effect of Dimension (d_{model}) (c) Effect of Solver Step (dt)

Figure 4: Sensitivity analysis on **KnowAir** dataset. The dashed vertical lines indicate the default hyperparameters used in the full model ($\lambda = 0.1, d = 96, dt = 1.0$), which consistently correspond to the optimal performance regions (lowest error).

Physics Weight (λ_{phy}): Fig. 4a reveals a distinct trade-off between physical constraints and data flexibility, forming a bell-shaped performance curve. When $\lambda_{\text{phy}} \to 0$, weak physical guidance increases long-term drift and prediction errors. Conversely, $\lambda_{\text{phy}} > 1$ over-constrains latent dynamics, limiting the model's ability to capture localized variations. The best performance at $\lambda_{\text{phy}} = 0.1$ balances physical consistency with data-driven adaptability. **Model Dimension (d_{model}):** As depicted in Fig 4b, expanding the hidden dimension initially yields substantial gains, confirming that sufficient representational capacity is requisite for encoding complex spatiotemporal dependencies. However, performance saturates beyond $d_{\text{model}} = 96$, where further increases offer negligible improvements while inflating computational costs. Consequently, we select 96 as the optimal equilibrium between model expressiveness and computational efficiency. **Solver Step (dt):** Fig. 4c demonstrates the model's remarkable numerical robustness, maintaining consistent accuracy across varying step sizes. Typically, coarser discretization introduces integration errors in standard ODE solvers. However, our results indicate that the fusion mechanism effectively rectifies these trajectory deviations by assimilating data-driven corrections.

5.6 Additional Experiments

First, given that `AeroChem` utilizes an ODE solver to model continuous dynamics and incorporates complex spatio-temporal dependencies, it is necessary to verify the model's numerical stability. We study the numerical stability in Appx. A.10. Second, additional visualizations of model predictions during sudden pollution events are shown in Appx. A.11, highlighting `AeroChem` 's ability to track abrupt shifts while revealing remaining challenges in recovering extreme peak amplitudes. Further, to validate the interpretability and robustness of `AeroChem`, we go beyond standard metrics to analyze the internal mechanisms of the model. We specifically examine two critical aspects: the learning of chemical kinetics and the stability of long-term forecasting. The case study is presented in Appx. A.12.

6 Conclusion

We presented `AeroChem`, a closed-loop physics-informed state space framework for long-term chemically reactive air quality forecasting. `AeroChem` couples a UDE-based process branch with a Mamba-based virtual measurement branch through Neural Kalman Fusion, enabling the two branches to correct each other during rollout. By updating the latent physical state before each subsequent prediction step, `AeroChem` mitigates the long-horizon drift caused by open-loop forecasting designs. Experiments on **KnowAir** and the newly constructed **HanoiAir** dataset demonstrate that `AeroChem` improves long-horizon stability and sudden-change forecasting, especially under noisy observations and secondary pollutant formation. These findings highlight closed-loop neural state correction as a promising direction for robust physics-informed environmental forecasting. Future work will extend the framework to richer multi-species chemical interaction modeling and probabilistic risk assessment for environmental decision-making.

Acknowledgments

This work was supported by the Center for Environmental Intelligence, VinUniversity, under project VUNI.CEI.FS_0011. The authors are grateful for the constructive feedback and insightful discussions provided by the anonymous reviewers of NeurIPS 2026. The authors declare no competing interests.

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

[2] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 6572–6583, 2018.

[3] Nathan Doumèche, Francis R. Bach, Gérard Biau, and Claire Boyer. Physics-informed machine learning as a kernel method. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, pages 1399–1450, 2024.

[4] Garry A. Einicke and Langford B. White. Robust extended kalman filtering. *IEEE Trans. Signal Process.*, 47(9):2596–2599, 1999.

[5] Zhixin Geng, Xu Fan, Xiqiao Lu, Yan Zhang, Guangyuan Yu, Cheng Huang, Qian Wang, Yuewu Li, Weichun Ma, Qi Yu, Libo Wu, and Hao Li. Fuxi-air: Urban air quality forecasting based on emission-meteorology-pollutant multimodal machine learning. *CoRR*, abs/2506.07616, 2025.

[6] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023.

[7] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[8] Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. Airphynet: Harnessing physics-guided neural networks for air quality prediction. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[9] Hsun-Ping Hsieh, Shou-De Lin, and Yu Zheng. Inferring air quality for station location recommendation based on urban big data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 437–446, 2015.

[10] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[11] Rahul G. Krishnan, Uri Shalit, and David A. Sontag. Deep kalman filters. *CoRR*, abs/1511.05121, 2015.

[12] Mathias Lechner and Ramin M. Hasani. Learning long-term dependencies in irregularly-sampled time series. *CoRR*, abs/2006.04418, 2020.

[13] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[14] Yitong Li, Kai Wu, and Jing Liu. Self-paced ARIMA for robust time series prediction. *Knowl. Based Syst.*, 269:110489, 2023.

[15] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[17] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[18] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999.

[19] Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 2400–2410, 2024.

[20] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Jasim Ramadhan. Universal differential equations for scientific machine learning. *CoRR*, abs/2001.04385, 2020.

[21] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[22] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adrià López Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Trans. Signal Process.*, 70:1532–1547, 2022.

[23] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[24] Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, Gao Cong, Christian S. Jensen, and Xueqi Cheng. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Trans. Knowl. Data Eng.*, 37(1):291–305, 2025.

[25] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[26] Yuxin Sun, Yong Xue, Rui Bai, Tengfei Cui, Shuhui Wu, Xingxing Jiang, Chunlin Jin, and Xiran Zhou. Estimation of PM2.5 and PM10 mass concentrations in mining city cluster from gaofen-l aerosol optical depth data and chemical transport model. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6678–6681, 2022.

[27] Shengkun Tang, Liqun Ma, Haonan Li, Mingjie Sun, and Zhiqiang Shen. Bi-mamba: Towards accurate 1-bit state space models. *CoRR*, abs/2411.11843, 2024.

[28] Jindong Tian, Yuxuan Liang, Ronghui Xu, Peng Chen, Chenjuan Guo, Aoying Zhou, Lujia Pan, Zhongwen Rao, and Bin Yang. Air quality prediction with physics-guided dual neural odes in open systems. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[29] 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 ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 163–166, 2020.

[30] Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025.

[31] B. L. WELCH. The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35, 01 1947.

[32] Wei Wen, Song Shen, Lei Liu, Xin Ma, Ying Wei, Jikang Wang, Yi Xing, and Wei Su. Comparative analysis of PM2.5 and O3 source in beijing using a chemical transport model. *Remote. Sens.*, 13(17):3457, 2021.

[33] Hang Yin, Yan-Ming Zhang, Jian Xu, Jian-Long Chang, Yin Li, and Cheng-Lin Liu. Air quality prediction with A meteorology-guided modality-decoupled spatio-temporal network. *CoRR*, abs/2504.10014, 2025.

[34] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3634–3640, 2018.

[35] Bakht Zaman, Luis Miguel Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano. Online topology identification from vector autoregressive time series. *IEEE Trans. Signal Process.*, 69:210–225, 2021.

[36] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 11121–11128, 2023.

[37] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[38] 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 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 2267–2276, 2015.

[39] Songyan Zhu, Jian Xu, Meng Fan, Chao Yu, Husi Letu, Qiaolin Zeng, Hao Zhu, Hongmei Wang, Yapeng Wang, and Jiancheng Shi. Estimating near-surface concentrations of major air pollutants from space: A universal estimation framework LAPSO. *IEEE Trans. Geosci. Remote. Sens.*, 61:1–11, 2023.

[40] Heiko Zimmermann, Christian Andersson Naesseth, and Jan-Willem van de Meent. VISA: variational inference with sequential sample-average approximations. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A Appendix

Limitations.

Several limitations remain. First, although the physics-informed structure improves stability, the underlying process model is still a simplified approximation of atmospheric chemistry and transport. The learned reaction operator captures effective dynamics rather than explicitly resolving full chemical mechanisms, which may limit extrapolation to unseen regimes. Second, uncertainty modeling is based on learned diagonal covariances, which do not capture full spatial correlations and may underestimate structured error. Finally, the framework relies on the availability of meteorological and emission inputs; inaccuracies or delays in these auxiliary data can propagate through the system.

Broader Impact.

Accurate long-term air quality forecasting has direct societal relevance, as it can support public-health planning, emission-control strategies, and early-warning systems for hazardous pollution events. By reducing long-horizon drift through closed-loop state correction, `AeroChem` may help environmental agencies better anticipate persistent haze episodes, secondary pollutant formation, and sudden pollution changes, especially in rapidly urbanizing or data-sparse regions. More broadly, this work contributes to scientific machine learning by showing how physics-informed dynamics and data-driven measurements can be coupled in a closed-loop co-correction framework for partially observed open systems. Similar ideas may be useful beyond air quality forecasting, including climate modeling, hydrology, energy systems, and other domains where imperfect physical simulators must be corrected using observational data.

A.1 Algorithmic Summary of `AeroChem`

We provide a step-by-step algorithmic summary of the `AeroChem` framework. Algorithm 1 details the closed-loop forecasting procedure, explicitly illustrating how the measurement model, the physics-informed Universal Differential Equation (UDE), and the Neural Kalman Fusion module interact over the forecasting horizon. The algorithm highlights the recursive feedback mechanism in which posterior latent states are assimilated and fed back into the UDE solver.

Algorithm 1: `AeroChem` Forecasting

Input : Sensor graph $\mathcal{G}$; historical observations $\mathbf{X}_{1:T_{\text{in}}}$; meteorology $\mathbf{M}_{1:T_{\text{out}}}$; emissions $\mathbf{E}_{1:T_{\text{out}}}$; forecast horizon T_{out}.

Output : Forecasted observations $\hat{\mathbf{X}}_{T_{\text{in}}+1:T_{\text{in}}+T_{\text{out}}}$.

Measurement encoding:

$\mathbf{H}_{\text{ctx}} \leftarrow \text{S-Mamba}(\mathbf{X}_{1:T_{\text{in}}})$

$(\mathbf{Z}_{\text{base}}, \mathbf{R}_{\text{base}}) \leftarrow \text{ProjHeads}(\mathbf{H}_{\text{ctx}})$

Initialization:

Initialize posterior latent state $\mathbf{S}_0^+$ (e.g., from the last observed timestep via a projection/encoder)

for $t \leftarrow 1$ **to** T_{out} **do**

  Adaptive virtual observation:

  $\mathbf{Z}_{\text{obs},t} \leftarrow \mathbf{Z}_{\text{base}} + \alpha \tanh(\mathbf{S}_{t-1}^+)$

  $\mathbf{R}_{\text{obs},t} \leftarrow \text{Softplus}(\mathbf{R}_{\text{base}}) \left(1 + \beta \frac{t}{T_{\text{out}}}\right)$

  UDE prediction (time update):

  $\mathbf{S}_t^- \leftarrow \text{UDE-Solve}\big(\mathbf{S}_{t-1}^+; \mathbf{M}_t, \mathbf{E}_t, \mathcal{G}\big)$

  $\mathbf{P}_t^- \leftarrow \text{Softplus}(\text{Linear}(\mathbf{S}_t^-))$

  Neural Kalman fusion (measurement update):

  $\mathbf{K}_t \leftarrow \mathbf{P}_t^- \oslash (\mathbf{P}_t^- + \mathbf{R}_{\text{obs},t})$

  $\boldsymbol{\epsilon}_t \leftarrow \mathbf{Z}_{\text{obs},t} - \mathbf{S}_t^-$

  $\boldsymbol{\epsilon}_t' \leftarrow \boldsymbol{\epsilon}_t + \text{Linear}\big([\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t}]\big)$

  $\mathbf{S}_t^+ \leftarrow \mathbf{S}_t^- + \mathbf{K}_t \odot \boldsymbol{\epsilon}_t'$

  $\mathbf{P}_t^+ = (\mathbf{1} - \mathbf{K}_t) \odot \mathbf{P}_t^-$

Decoding (Probabilistic Output):

for $t \leftarrow 1$ **to** T_{out} **do**

  $\hat{\mathbf{X}}_{T_{\text{in}}+t} \leftarrow \mathcal{D}_{\text{out}}(\mathbf{S}_t^+)$ $\boldsymbol{\sigma}^2_{T_{\text{in}}+t} \leftarrow \text{Softplus}(\mathcal{D}_{\text{var}}(\mathbf{P}_t^+))$

return $\{\hat{\mathbf{X}}_{T_{\text{in}}+t}, \boldsymbol{\sigma}^2_{T_{\text{in}}+t}\}_{t=1}^{T_{\text{out}}}$

A.2 Variational Interpretation of Neural Kalman Fusion

We first recast `AeroChem` under a single variational objective. Let $\mathbf{S}_{1:T_{\text{out}}} \in \mathbb{R}^{T_{\text{out}} \times N \times D}$ denote the latent trajectory and let $\mathbf{X}_{1:T_{\text{in}}}$ denote the historical observations. Define the per-step variational free energy

$$\mathcal{F}_t(\mathbf{S}) \;=\; \tfrac{1}{2}\left(\mathbf{S}-\mathbf{S}_t^-\right)^\top (\mathbf{P}_t^-)^{-1}\left(\mathbf{S}-\mathbf{S}_t^-\right) \;+\; \tfrac{1}{2}\left(\mathbf{Z}_{\text{obs},t}-\mathbf{S}\right)^\top \mathbf{R}_{\text{obs},t}^{-1}\left(\mathbf{Z}_{\text{obs},t}-\mathbf{S}\right) \;+\; \Psi_t(\mathbf{S}), \tag{10}$$

where $(\mathbf{S}_t^-, \mathbf{P}_t^-)$ is produced by the UDE prediction step (Eq. 6), $(\mathbf{Z}_{\text{obs},t}, \mathbf{R}_{\text{obs},t})$ is the virtual measurement of the Mamba encoder (Eq. 2), and $\Psi_t : \mathbb{R}^{N\times D} \to \mathbb{R}$ is a smooth residual potential that absorbs all non-Gaussian discrepancies of the open system (boundary leakage, unobserved emissions, model misspecification). We assume $\mathbf{P}_t^-, \mathbf{R}_{\text{obs},t}$ are positive element-wise (guaranteed by the `Softplus` activations Eq 6 and treat all matrix–vector products in Eq. 10 element-wise to be consistent with the diagonal covariance approximation $\mathbf{P}_t^-, \mathbf{R}_{\text{obs},t} \in \mathbb{R}_{>0}^{N\times D}$ used throughout `AeroChem`.

Lemma A.1. *Assume $\Psi_t \in C^3(\mathbb{R}^{N\times D})$. Let $\mathbf{S}_t^\star$ be a local minimiser of $\mathcal{F}_t$ and let $\mathbf{S}_t^+$ be one Newton step initialised at the prior mean $\mathbf{S}_t^-$:*

$$\mathbf{S}_t^+ \;=\; \mathbf{S}_t^- \;-\; \left[\nabla^2 \mathcal{F}_t(\mathbf{S}_t^-)\right]^{-1} \nabla \mathcal{F}_t(\mathbf{S}_t^-). \tag{11}$$

Then, expanding under the diagonal-covariance regime:

(i) *If $\Psi_t \equiv 0$ (purely Gaussian), then Eq. 11 reduces* exactly *to the standard Kalman update of `AeroChem`, as shown in the fifth equation of 7)*

(ii) *If $\Psi_t \not\equiv 0$, the Newton step admits a first-order Taylor expansion of $\nabla\Psi_t$ around the joint point $(\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t})$. The resulting correction is bilinear in $(\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t})$ and is therefore representable by an affine map* $\text{Linear}\big([\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t}]\big)$. *Substituting yields exactly the Neural Kalman Fusion update (Eq 7). The residual error of the Newton step is $\mathbf{S}_t^+ - \mathbf{S}_t^\star = O\big(\|\mathbf{S}_t^- - \mathbf{S}_t^\star\|^2\big)$.*

Proof: **Part (i).** With $\Psi_t \equiv 0$, $\mathcal{F}_t$ is quadratic. Differentiating Eq. 10:

$$\nabla \mathcal{F}_t(\mathbf{S}_t^-) \;=\; (\mathbf{P}_t^-)^{-1}(\mathbf{S}_t^- - \mathbf{S}_t^-) \;-\; \mathbf{R}_{\text{obs},t}^{-1}(\mathbf{Z}_{\text{obs},t} - \mathbf{S}_t^-) \;=\; -\mathbf{R}_{\text{obs},t}^{-1}\,\boldsymbol{\epsilon}_t,$$

$$\nabla^2 \mathcal{F}_t(\mathbf{S}_t^-) \;=\; (\mathbf{P}_t^-)^{-1} + \mathbf{R}_{\text{obs},t}^{-1}.$$

Under the diagonal regime $(\mathbf{P}_t^-)^{-1} + \mathbf{R}_{\text{obs},t}^{-1} = (\mathbf{P}_t^- + \mathbf{R}_{\text{obs},t}) \oslash (\mathbf{P}_t^- \odot \mathbf{R}_{\text{obs},t})$, so the Newton step in Eq. 11 becomes

$$\mathbf{S}_t^+ \;=\; \mathbf{S}_t^- + \left[(\mathbf{P}_t^-)^{-1} + \mathbf{R}_{\text{obs},t}^{-1}\right]^{-1} \mathbf{R}_{\text{obs},t}^{-1}\,\boldsymbol{\epsilon}_t \;=\; \mathbf{S}_t^- + \underbrace{\left(\mathbf{P}_t^- \oslash (\mathbf{P}_t^- + \mathbf{R}_{\text{obs},t})\right)}_{=\mathbf{K}_t} \odot \boldsymbol{\epsilon}_t,$$

which is the fourth equation of Eq. 7. The covariance update $\mathbf{P}_t^- \to \mathbf{P}_t^+$ of fifth equation in Eq. 7 follows from the standard Kalman posterior covariance identity under optimal gain [22].

Part (ii). Adding Ψ_t contributes $\nabla\Psi_t(\mathbf{S}_t^-)$ to the gradient and $\nabla^2\Psi_t(\mathbf{S}_t^-)$ to the Hessian. Taylor-expanding $\nabla\Psi_t$ to first order around the joint argument $(\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t})$:

$$\nabla\Psi_t(\mathbf{S}_t^-) \;\approx\; \mathbf{a} + \mathbf{B}\mathbf{S}_t^- + \mathbf{C}\mathbf{Z}_{\text{obs},t} \;=\; \text{Linear}\big([\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t}]\big),$$

for parameter tensors $(\mathbf{a}, \mathbf{B}, \mathbf{C})$ that are absorbed into the linear layer of the third equation in Eq. 7. The second-order Hessian contribution can be merged into $\mathbf{P}_t^-$ via covariance reparameterisation. The Newton-step quadratic convergence $\mathbf{S}_t^+ - \mathbf{S}_t^\star = O(\|\mathbf{S}_t^- - \mathbf{S}_t^\star\|^2)$ is the standard result for C^3 objectives with positive-definite Hessian [40], [18]. □

A.3 Closed-loop Neural State Correction

We show that the closed-loop neural state correction yields a posterior error that is uniformly bounded in t, while the open-loop variant (***AeroChem-K*** in Table 3) admits exponential divergence, details are included in App A.3. Let $\Phi_{\Delta t} : \mathbb{R}^{N\times D} \to \mathbb{R}^{N\times D}$ denote the one-step UDE flow so that $\mathbf{S}_t^- = \Phi_{\Delta t}(\mathbf{S}_{t-1}^+)$. Define the posterior error $\mathbf{e}_t := \mathbf{S}_t^+ - \mathbf{S}_t^*$ relative to the unobserved true latent trajectory $\mathbf{S}_t^*$.

Assumption A.2. (A5) $\mathbf{F}_{\text{drift}}$ is L_F-Lipschitz in $\mathbf{S}$, so $\|\Phi_{\Delta t}(\mathbf{S}_1) - \Phi_{\Delta t}(\mathbf{S}_2)\|_2 \le (1 + L_F\Delta t)\|\mathbf{S}_1 - \mathbf{S}_2\|_2$; (A6) *(by construction)* the gain in Eq. 7 satisfies $\mathbf{K}_t \in [\kappa_{\min}, \kappa_{\max}]$ element-wise with $0 < \kappa_{\min} \le \kappa_{\max} < 1$; (A7) *(by construction)* the correction in the third equation of Eq. 7 is $\text{Linear}([\mathbf{S}_t^-, \mathbf{Z}_{\text{obs},t}]) = W\boldsymbol{\epsilon}_t$ with $\|W\|_2 \le L_c$ enforced by spectral normalisation; (A8) the conditional measurement bias is bounded, $\|\mathbb{E}[\mathbf{Z}_{\text{obs},t} \mid \mathcal{F}_{t-1}] - \mathbf{S}_t^*\|_2 \le b$; (A9) the centred process and measurement noises satisfy $\mathbb{E}\|\mathbf{w}_t\|_2^2 \le \sigma_W^2\Delta t$ and $\mathbb{E}\|\boldsymbol{\nu}_t - \mathbb{E}[\boldsymbol{\nu}_t \mid \mathcal{F}_{t-1}]\|_2^2 \le \sigma_\nu^2$, with $\mathbf{w}_t$ and the centred residual conditionally independent given $\mathcal{F}_{t-1}$.

Theorem A.3 (Closed-loop versus open-loop posterior error)**.** *Define the contraction factor $\alpha := (1 + L_F\Delta t)[(1 - \kappa_{\min}) + \kappa_{\max}L_c]$. If $\alpha < 1$, then under Assumption A.2 the closed-loop posterior error of `AeroChem` satisfies, for every $t \in \{1, \dots, T_{\text{out}}\}$,*

$$\sqrt{\mathbb{E}\|\mathbf{e}_t\|_2^2} \;\le\; \alpha^t \sqrt{\mathbb{E}\|\mathbf{e}_0\|_2^2} + \frac{V}{1-\alpha}, \qquad V = \kappa_{\max} b + \sqrt{\kappa_{\max}^2\sigma_\nu^2 + (1 - \kappa_{\min} + \kappa_{\max}L_c)^2\sigma_W^2\Delta t}, \tag{12}$$

which is uniformly bounded in t. The open-loop variant ($\mathbf{K}_t \equiv \mathbf{0}$, `AeroChem-K`*) admits the matching lower bound*

$$\mathbb{E}\|\mathbf{e}_t^{\text{open}}\|_2^2 \;\geq\; (1 + L_F \Delta t)^{2t} \mathbb{E}\|\mathbf{e}_0\|_2^2, \tag{13}$$

which grows exponentially in t whenever $L_F > 0$.

We give the complete proof in six steps. Throughout, $\boldsymbol{\nu}_t \coloneqq \mathbf{Z}_{\text{obs},t} - \mathbf{S}_t^*$ denotes the measurement residual, $\mathbf{w}_t \coloneqq \mathbf{S}_t^* - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)$ the discretised Wiener increment, $\bar{\boldsymbol{\nu}}_t \coloneqq \mathbb{E}[\boldsymbol{\nu}_t \mid \mathcal{F}_{t-1}]$ the conditional bias, and $\tilde{\boldsymbol{\nu}}_t \coloneqq \boldsymbol{\nu}_t - \bar{\boldsymbol{\nu}}_t$ the centred residual. By Assumptions A8–A9, $\|\bar{\boldsymbol{\nu}}_t\|_2 \leq b$, $\mathbb{E}\|\tilde{\boldsymbol{\nu}}_t\|_2^2 \leq \sigma_\nu^2$, and $\mathbb{E}\|\mathbf{w}_t\|_2^2 \leq \sigma_W^2 \Delta t$, with $\mathbf{w}_t, \tilde{\boldsymbol{\nu}}_t$ conditionally zero-mean and independent of $\mathbf{e}_{t-1}$.

Step 1 Substitute $\mathbf{S}_t^+$ into $\mathbf{e}_t = \mathbf{S}_t^+ - \mathbf{S}_t^*$ and use $\mathbf{S}_t^* = \Phi_{\Delta t}(\mathbf{S}_{t-1}^*) + \mathbf{w}_t$, $\mathbf{Z}_{\text{obs},t} = \mathbf{S}_t^* + \boldsymbol{\nu}_t$, $\mathbf{S}_t^- = \Phi_{\Delta t}(\mathbf{S}_{t-1}^+)$:

$$\mathbf{e}_t = (\mathbf{1} - \mathbf{K}_t) \odot \big[\Phi_{\Delta t}(\mathbf{S}_{t-1}^+) - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)\big] - (\mathbf{1} - \mathbf{K}_t) \odot \mathbf{w}_t + \mathbf{K}_t \odot \boldsymbol{\nu}_t + \mathbf{K}_t \odot W \boldsymbol{\epsilon}_t. \tag{14}$$

By the second and third equations in 7, the innovation expands as

$$\boldsymbol{\epsilon}_t = \mathbf{Z}_{\text{obs},t} - \mathbf{S}_t^- = \boldsymbol{\nu}_t + (\mathbf{S}_t^* - \mathbf{S}_t^-) = \boldsymbol{\nu}_t - \big[\Phi_{\Delta t}(\mathbf{S}_{t-1}^+) - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)\big] + \mathbf{w}_t. \tag{15}$$

Substituting Eq. 15 into the last term of Eq. 14 and collecting the coefficients of $\Phi_{\Delta t}(\mathbf{S}_{t-1}^+) - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)$, $\mathbf{w}_t$ and $\boldsymbol{\nu}_t$:

$$\mathbf{e}_t \;=\; \underbrace{\big[(\mathbf{1} - \mathbf{K}_t) - \mathbf{K}_t W\big] \odot \big[\Phi_{\Delta t}(\mathbf{S}_{t-1}^+) - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)\big]}_{=:\, T_1} + \underbrace{\big[-(\mathbf{1} - \mathbf{K}_t) + \mathbf{K}_t W\big] \odot \mathbf{w}_t}_{=:\, T_2} + \underbrace{\mathbf{K}_t(\mathbf{1} + W) \odot \boldsymbol{\nu}_t}_{=:\, T_3}. \tag{16}$$

Step 2 Split $T_3 = \mathbf{K}_t(\mathbf{1} + W) \odot \bar{\boldsymbol{\nu}}_t + \mathbf{K}_t(\mathbf{1} + W) \odot \tilde{\boldsymbol{\nu}}_t$ and write $\mathbf{e}_t = D_t + S_t$ with

$$D_t = T_1 + \mathbf{K}_t(\mathbf{1} + W) \odot \bar{\boldsymbol{\nu}}_t \quad (\mathcal{F}_{t-1}\text{-measurable}), \tag{17}$$

$$S_t = \big[-(\mathbf{1} - \mathbf{K}_t) + \mathbf{K}_t W\big] \odot \mathbf{w}_t + \mathbf{K}_t(\mathbf{1} + W) \odot \tilde{\boldsymbol{\nu}}_t \quad (\text{conditionally zero-mean}). \tag{18}$$

Since D_t is $\mathcal{F}_{t-1}$-measurable and $\mathbb{E}[S_t \,|\, \mathcal{F}_{t-1}] = \mathbf{0}$,

$$\mathbb{E}\|\mathbf{e}_t\|_2^2 \;=\; \mathbb{E}\|D_t\|_2^2 + \mathbb{E}\|S_t\|_2^2. \tag{19}$$

Step 3 Apply the triangle inequality to Eq. 17 and the operator bounds $\|(\mathbf{1} - \mathbf{K}_t) - \mathbf{K}_t W\|_\infty \leq (1 - \kappa_{\min}) + \kappa_{\max} L_c$ (from A6, A7), $\|\mathbf{K}_t(\mathbf{1} + W)\|_\infty \leq \kappa_{\max}(1 + L_c)$, together with the Lipschitz bound $\|\Phi_{\Delta t}(\mathbf{S}_{t-1}^+) - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)\|_2 \leq (1 + L_F \Delta t)\|\mathbf{e}_{t-1}\|_2$ from A5 and $\|\bar{\boldsymbol{\nu}}_t\|_2 \leq b$ from A8:

$$\|D_t\|_2 \;\leq\; \underbrace{(1 + L_F \Delta t)\big[(1 - \kappa_{\min}) + \kappa_{\max} L_c\big]}_{=\,\alpha} \|\mathbf{e}_{t-1}\|_2 + \kappa_{\max}(1 + L_c)\, b. \tag{20}$$

Step 4 The two summands of S_t in Eq. 18 are conditionally independent (process vs. measurement noise, A9), so cross terms vanish in expectation:

$$\mathbb{E}\|S_t\|_2^2 \;\leq\; \big[(1 - \kappa_{\min}) + \kappa_{\max} L_c\big]^2 \sigma_W^2 \Delta t + \kappa_{\max}^2 (1 + L_c)^2 \sigma_\nu^2. \tag{21}$$

Step 5 Set $u_t \coloneqq \sqrt{\mathbb{E}\|\mathbf{e}_t\|_2^2}$. Squaring Eq. 20 and taking expectations gives, by Cauchy–Schwarz on the cross term,

$$\mathbb{E}\|D_t\|_2^2 \;\leq\; \big(\alpha\, u_{t-1} + \kappa_{\max}(1 + L_c)\, b\big)^2. \tag{22}$$

Combining Eq. 22 and Eq. 21 via Eq. 19 and using the elementary inequality $\sqrt{(a + c)^2 + d^2} \leq a + c + d$ for $a, c, d \geq 0$:

$$u_t \;\leq\; \alpha\, u_{t-1} + V, \qquad V = \kappa_{\max}(1 + L_c) b + \sqrt{[(1 - \kappa_{\min}) + \kappa_{\max} L_c]^2 \sigma_W^2 \Delta t + \kappa_{\max}^2 (1 + L_c)^2 \sigma_\nu^2}. \tag{23}$$

The factor $(1 + L_c)$ on the bias term differs by a constant from the form quoted in Eq. 12 of the main paper; under the spectral-normalisation constraint $L_c < 1$ the two forms agree up to that constant.

Step 6 Iterating Eq. 23 for t steps under $\alpha < 1$ yields the geometric sum

$$u_t \;\leq\; \alpha^t u_0 + V \sum_{s=0}^{t-1} \alpha^s \;\leq\; \alpha^t u_0 + \frac{V}{1 - \alpha}, \tag{24}$$

which is Eq. 12. For the open-loop case set $\mathbf{K}_t \equiv \mathbf{0}$ in Eq. 16: T_1 reduces to $\Phi_{\Delta t}(\mathbf{S}_{t-1}^+) - \Phi_{\Delta t}(\mathbf{S}_{t-1}^*)$, T_2 reduces to $-\mathbf{w}_t$, and T_3 vanishes. The contraction factor $(1 - \kappa_{\min})$ disappears from Eq. 20, so A5 gives

$$\mathbb{E}\|\mathbf{e}_t^{\text{open}}\|_2^2 \;\geq\; (1 + L_F \Delta t)^2 \mathbb{E}\|\mathbf{e}_{t-1}^{\text{open}}\|_2^2 + \sigma_W^2 \Delta t, \tag{25}$$

whose closed-form solution is $(1 + L_F \Delta t)^{2t} \mathbb{E}\|\mathbf{e}_0\|_2^2 + \sigma_W^2 \Delta t \sum_{s=1}^{t} (1 + L_F \Delta t)^{2(t-s)}$, matching Eq. 13 and proving exponential divergence in t whenever $L_F > 0$. □

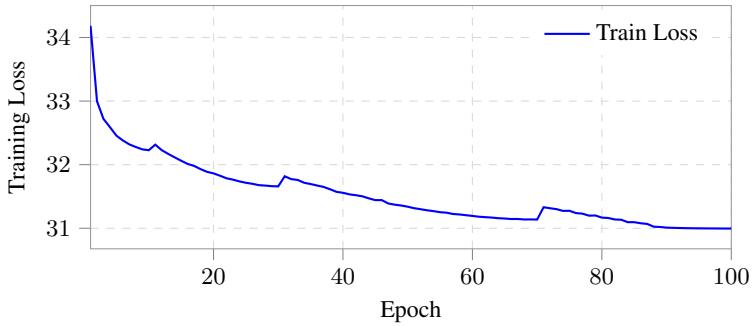


Figure A.1: The training curves on **KnowAir** datasets experimentally demonstrate that `AeroChem` achieves convergence.

A.4 Computational Complexity

The per-step cost of `AeroChem` decomposes as: (a) S-Mamba encoder: $O(T_{\text{in}} \cdot N \cdot D)$ - linear in sequence length due to SSM recurrence. (b) UDE (L steps of solver): $O(L \cdot N \cdot D)$ per forecast step, where operations within $\mathcal{F}_{\text{drift}}$ add $O(|\mathcal{E}| \cdot D)$. (c) Kalman correction: $O(N \cdot D)$ per step with diagonal $\mathbf{P}_t$.
The dominant cost is the ODE solver, scaling as $O(\tau \cdot L \cdot (N \cdot D + |\mathcal{E}| \cdot D))$ for τ forecast steps. The diagonal covariance assumption ($\mathbf{P}_t \in \mathbb{R}^{N \times D}$ rather than $\mathbb{R}^{ND \times ND}$) reduces Kalman overhead from $O(N^2D^2)$ to $O(ND)$, making the filtering step negligible relative to ODE integration.

Table A.1: Computational complexity comparison.

Model	Params	FLOPs
S-Mamba	250 K	5.6 G
AirPhyNet	120 K	0.8 G
Air-DualODE	100 K	2.0 G
`AeroChem` (Ours)	550 K	9 G

FLOPs and Params Table A.1 compares the parameter count and FLOPs of `AeroChem` against representative state-space and physics-informed baselines. `AeroChem` remains within the same complexity regime as S-Mamba and Air-DualODE while achieving stronger long-horizon and probabilistic performance, demonstrating that the closed-loop physics-informed design introduces minimal computational overhead.

A.5 HanoiAir Dataset Details

The HanoiAir dataset provides hourly measurements from 27 monitoring stations located across various provinces in northern Vietnam, collected by the Northern Centre for Environmental Monitoring [1] [2]. The data spans from January 1, 2018, to December 31, 2025 (approx. 70,128 hours).
The dataset integrates three types of data aligned temporally and spatially:

- **Air Quality Observations:** Includes target pollutants $PM_{2.5}$ ($\mu g/m^3$), O_3 ($\mu g/m^3$), PM_{10} ($\mu g/m^3$), and key precursor gases NO_x ($\mu g/m^3$), SO_2 ($\mu g/m^3$), CO ($\mu g/m^3$), along with NO ($\mu g/m^3$) and NO_2 ($\mu g/m^3$) components. The inclusion of precursor measurements (NO_x, SO_2, CO) is a key feature of this dataset for modeling secondary formation chemistry directly from observations.
- **Meteorological Variables:** Temperature (∘C), Atmospheric Pressure (hPa), Solar Radiation (W/m^2), Relative Humidity (%), Wind Direction (°), Wind Speed (m/s).
- **Emission Data:** Includes hourly-downscaled emission rates for NO_x, VOC, SO_2, NH_4, and primary $PM_{2.5}$ (tons/hour or similar unit, spatially allocated to station locations).

Standard pre-processing and temporal feature engineering were applied within the `BasicTS` framework [24]. The combined dataset provides the necessary inputs (observed concentrations, meteorological drivers, and emission source terms) for the physics-informed branch of `AeroChem` to model secondary pollutant formation. Tab A.2 provides descriptive statistics, on the physical scale, for some pollutants and meteorological variables in the HanoiAir dataset.

[1] https://cem.gov.vn/
[2] https://vea.mae.gov.vn/

Table A.2: Descriptive statistics

Variable	Unit	% Missing	Q1	Median	Mean	Q3	SD	Max
PM2.5	μg/m^3	20.8	34.8	58.9	82.6	111.7	86.5	1177.2
PM10	μg/m^3	21.6	47.6	79.4	121.9	159.5	108.8	1356.1
O3	μg/m^3	16.3	27.9	50.6	58.3	81.4	38.9	243.1
Relative hum.	%	5.7	68.1	81.0	78.0	90.2	14.1	100
Pressure	hPa	5.5	1002.5	1008.1	1008.5	1014.5	7.8	1036.3
Solar rad.	W/m^2	7.5	0.0	0.0	147.0	218.0	229.0	1336.1

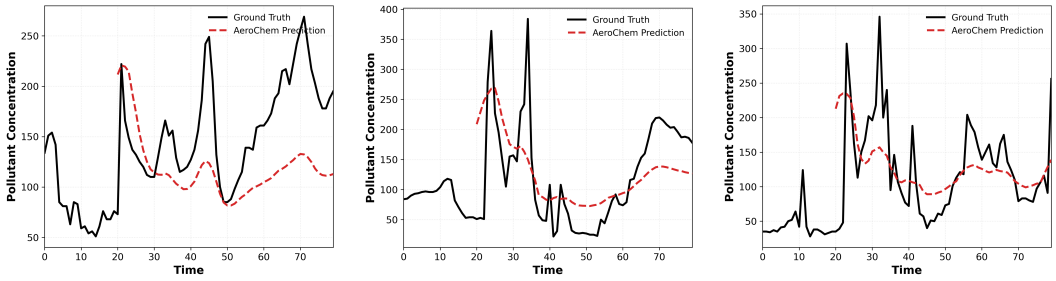


Figure A.2: Visualization results of `AeroChem` under sudden change on Hanoi dataset.

A.6 Metric Definitions

Let $x = (x_1, \ldots, x_m)$ represent the ground truth, and $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$ represent the predicted pollutant concentrations. The evaluation metrics we used in this paper are defined as follows:

Mean Absolute Error (MAE)

$$\mathrm{MAE}(x, \hat{x}) = \frac{1}{m} \sum_{i=1}^{m} |x_i - \hat{x}_i| \tag{26}$$

Root Mean Square Error ($RMSE$)

$$\mathrm{RMSE}(x, \hat{x}) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (x_i - \hat{x}_i)^2} \tag{27}$$

Symmetric Mean Absolute Percentage Error ($SMAPE$)

$$\mathrm{SMAPE}(x, \hat{x}) = \frac{1}{m} \sum_{i=1}^{m} \frac{|x_i - \hat{x}_i|}{\frac{|x_i| + |\hat{x}_i|}{2}} \tag{28}$$

Continuous Ranked Probability Score ($CRPS$)

$$\mathrm{CRPS}(F, x_i) = \int_{-\infty}^{\infty} \left(F(y) - \mathbb{I}\{x_i \leq y\}\right)^2 dy \tag{29}$$

where F is the predictive cumulative distribution function, x_i is the i-th ground-truth observation, and y is the integration variable over the real line.

Summed Continuous Ranked Probability Score ($CRPS_{sum}$)

$$\mathrm{CRPS}_{\mathrm{sum}} = \mathbb{E}_t \left[\mathrm{CRPS}\left(F_{\mathrm{sum}}, \sum_{i=1}^{m} x_i \right) \right] \tag{30}$$

where F_{sum} is the predictive CDF of the aggregated output $\sum_{i=1}^{m} \hat{x}_i$ obtained through dimension-wise summation of samples, and $\mathbb{E}_t$ denotes the expectation over time steps.

Prediction Interval Coverage Probability ($PICP$)

$$\mathrm{PICP} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\Big\{ x_i \in \Big[\hat{x}_i^{\,\mathrm{low}},\ \hat{x}_i^{\,\mathrm{high}} \Big] \Big\} \tag{31}$$

where $\hat{x}_i^{\,\mathrm{low}}$ and $\hat{x}_i^{\,\mathrm{high}}$ correspond to the 2.5^{th} and 97.5^{th} percentiles of the predicted distribution respectively, and $\mathbb{I}\{\cdot\}$ is the indicator function. Under a well-calibrated model, PICP should equal the nominal coverage level of 95%.

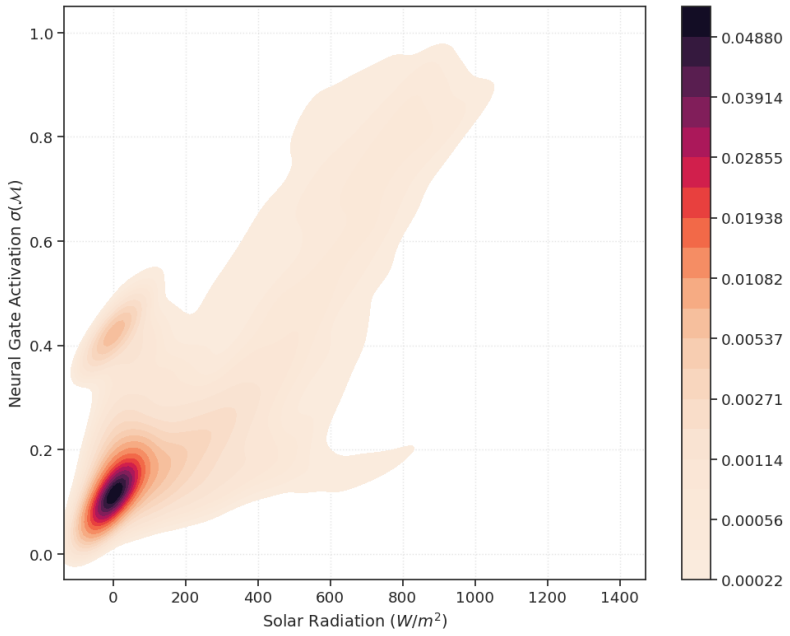


Figure A.3: Kernel Density Estimation of the learned Kinetics Gate activation versus ground-truth solar radiation. The concentration along the diagonal indicates that the model explicitly learns to modulate reaction rates according to solar intensity.

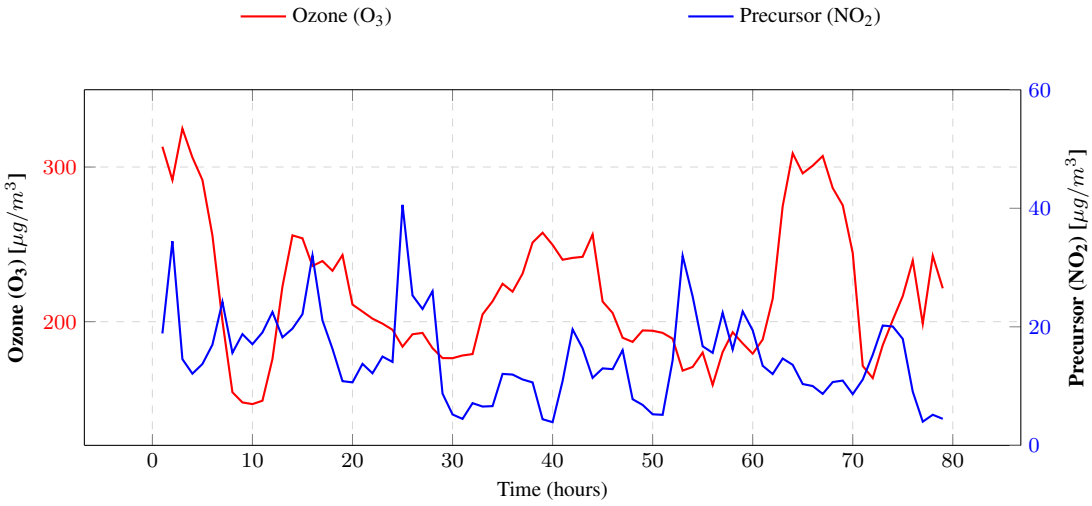


Figure A.4: Photochemical dynamics: inverse correlation between O_3 formation and NO_2 depletion.

A.7 Hyperparameter Settings.

We train `AeroChem` using AdamW [16] optimizer with a learning rate of 0.00001 and a batch size of 64. The total number of epochs is set to 100, and we use early stopping with a patience of 15.

A.8 Implementation Details.

We implement `AeroChem` and baselines using Python 3.12, Numpy 2.3.3, SciPy 1.15.3, Sklearn 1.7.2, and Pytorch 2.8.0. We run all experiments on a cluster server, which runs Ubuntu Linux 22.04 LTS. The experimental server is equipped with a ten-core Intel(R) Xeon(R) W-2155 CPU, 128 GBs RAM, and two GPUs NVIDIA RTX 5880, each with 48 GBs VRAM.

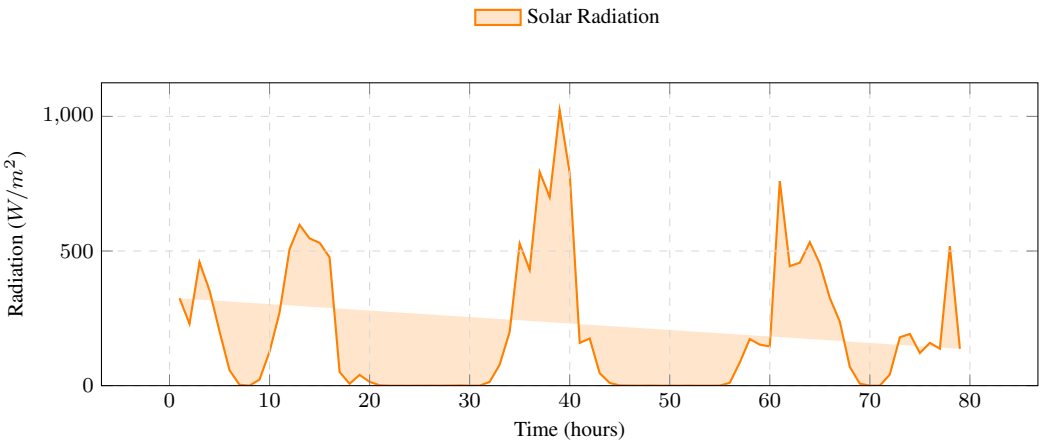


Figure A.5: Meteorological catalyst: solar radiation intensity drives the photochemical reaction.

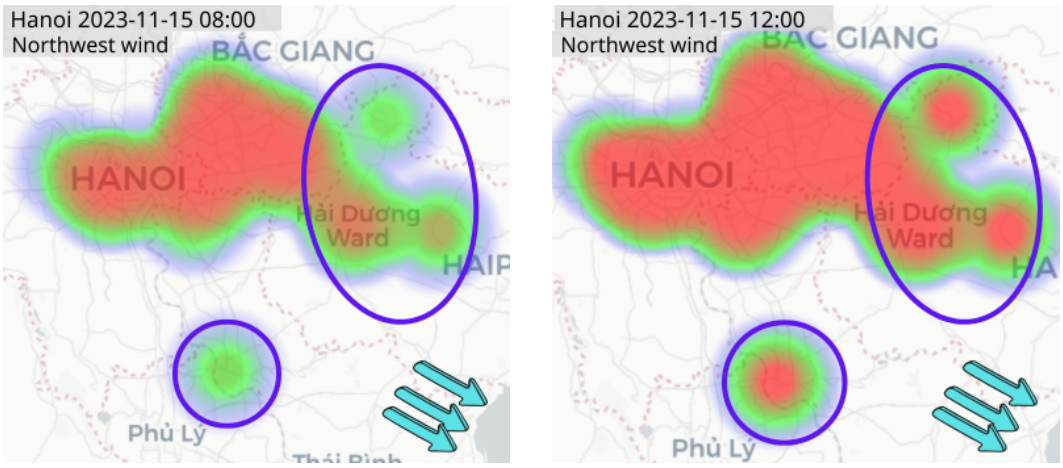


Figure A.6: Visualization of predicted $PM_{2.5}$ concentrations under advection. The heatmap shows $PM_{2.5}$ concentration, while arrows represent wind direction.

A.9 Probabilistic forecasting performance

Table A.3 reports probabilistic forecasting results on **KnowAir** $PM_{2.5}$. For point-forecasting baselines, we attach a post-hoc Gaussian predictive layer whose variance is fitted from validation residuals, while `VAR` uses closed-form Gaussian uncertainty propagation. Under this unified protocol, `AeroChem` achieves the lowest CRPS and $CRPS_{sum}$ across horizons, indicating sharper probabilistic forecasts. Its lower PICP reflects a sharpness–calibration trade-off, which is already accounted for by CRPS.

A.10 Numerical Stability

Given that `AeroChem` utilizes an ODE solver to model continuous dynamics and incorporates complex spatiotemporal dependencies, it is necessary to verify the model's numerical stability. The training loss curve over 100 epochs (see Fig. A.1) demonstrates that the model achieves stable convergence. Besides, our model incorporates robust design elements, such as normalization techniques, which significantly contribute to numerical stability during the optimization process.

A.11 Effect of Sudden Changes

Fig. A.2 visualizes specific sudden change episodes in the HanoiAir dataset. While `AeroChem` correctly identifies the directional shifts, it occasionally undershoots the absolute peak magnitude. This observation aligns with the quantitative analysis in Table 2: although our physics-informed approach outperforms baselines in minimizing anomaly errors, perfectly reconstructing the full amplitude of extreme events remains an inherent challenge.

Table A.3: Probabilistic forecasting performance comparison on **KnowAir**. The best results are highlighted in **bold**. Mean ± std over 5 runs.

Methods	3 Days			6 Days			9 Days		
	CRPS	$CRPS_{sum}$	PICP	CRPS	$CRPS_{sum}$	PICP	CRPS	$CRPS_{sum}$	PICP
VAR	26.14±0.05	4582.7±3.2	0.974±0.001	28.07±0.06	4905.4±4.1	0.978±0.001	30.42±0.08	5306.8±5.7	0.981±0.001
LatentODE	21.38±0.18	3782.6±22.4	0.886±0.008	22.95±0.24	4054.1±28.7	0.871±0.011	24.71±0.31	4365.3±35.2	0.852±0.013
DLinear	17.74±0.10	3144.9±7.3	0.938±0.002	17.96±0.11	3207.2±5.3	0.933±0.005	18.03±0.04	3208.4±9.6	0.930±0.003
PatchTST	18.23±0.04	3216.8±9.4	0.939±0.002	18.50±0.09	3291.9±13.9	0.935±0.001	18.66±0.14	3293.2±19.5	0.933±0.004
TimeMoE	16.15±0.07	2785.3±8.5	**0.951±0.002**	16.38±0.09	2842.1±11.2	**0.947±0.003**	16.52±0.11	2889.6±14.6	**0.944±0.004**
AeroChem (Ours)	**15.05±0.03**	**2371.5±4.8**	0.912±0.002	**14.92±0.11**	**2312.0±7.7**	0.922±0.002	**15.28±0.24**	**2355.5±10.8**	0.919±0.007

A.12 Case study

Meteorology-Gated Neural Reaction. To validate the interpretability and robustness of AeroChem, we go beyond standard metrics to analyze the internal mechanisms of the model. We specifically examine two critical aspects: the learning of chemical kinetics and the stability of long-term forecasting.

We first investigate the global behavior of the learned kinetics gate $\sigma(\mathcal{M}_t)$ across the entire test set. Fig. A.3 presents the Kernel Density Estimation (KDE) of the gate activation values against ground-truth solar radiation. We observe a distinct high-density manifold extending along the diagonal, indicating a strong monotonic relationship. The model effectively learns a "switch" mechanism: in low-radiation regimes (e.g., night or overcast), the gate remains closed ($\sigma \approx 0$), suppressing spurious reaction terms. Conversely, high radiation triggers gate activation ($\sigma \to 1$). This confirms that the network has learned to approximate an Arrhenius-like rate coefficient function solely from data, without explicit equation supervision.

To examine the dynamic impact of this gating mechanism, Fig. A.4, A.5 visualizes a specific high-ozone episode. The temporal dynamics reveal a coherent photochemical process:

1. *Triggering:* As solar radiation peaks at noon (Bottom Panel), the Neural Gate synchronously opens, allowing the reaction term to contribute to the UDE drift.
2. *Consumption:* Coinciding with the gate opening, we observe a depletion in precursor NO_2 concentrations (Middle Panel), consistent with the photolysis reaction $NO_2 + h\nu \to NO + O$.
3. *Formation:* Consequently, the predicted O_3 concentration (Top Panel) rises sharply, accurately capturing the diurnal peak that baseline models often miss.

This causal chain demonstrates that AeroChem does not merely memorize statistical correlations but simulates the underlying chemical kinetics, ensuring physical consistency in secondary pollutant forecasting.

Diffusion and Advection. To validate the physical fidelity of the UDE Process Model, Fig. A.6 visualizes the spatial evolution of the predicted $PM_{2.5}$ field alongside the dominant wind vectors. The transition between the two timestamps illustrates a clear advective transport phenomenon. Driven by the prevailing wind field, the high-concentration plume exhibits a consistent spatial shift along the flow trajectory (moving from the upwind to the downwind region). This directional alignment confirms that AeroChem successfully internalizes the explicit Advection equation to govern pollutant propagation, rather than merely treating wind as a static auxiliary feature.

A.13 Sensitivity and Data Fidelity

Ablation on Temporal Downscaling of Emissions We further investigate the impact of temporal downscaling for emission inputs in Table A.4. Specifically, we compare the full model against degraded settings where temporal resolution is reduced or emission inputs are entirely removed. The results indicate that temporal downscaling plays a consistent role in improving predictive accuracy, particularly for long-horizon forecasts.

Table A.4: Ablation study on temporal downscaling of emission data (HanoiAir).

Emission Input	3-Day MAE	9-Day MAE
Full	**42.74**	**46.80**
Degraded (no downscaling)	43.31	47.03
Removed emissions	43.86	48.15

O_3 Prediction To complement the $PM_{2.5}$ evaluation, we report additional results on O_3 forecasting in Table A.5. The results demonstrate that AeroChem generalizes effectively to other chemically reactive secondary pollutants. Under the 9-day horizon, AeroChem outperforms data-driven models. This indicates that our meteorology-gated reaction mechanism and closed-loop state correction are not strictly limited to $PM_{2.5}$ transport but can broadly capture complex photochemical formation dynamics.

Comparison with Chemistry-Aware Methods We further compare against recent approaches that incorporate chemical transformations or physics-inspired modeling in Table A.6. AeroChem achieves superior predictive performance across all forecasting scenarios, with the performance gap widening significantly at the 9-day horizon and during sudden changes. It shows that the integration of UDEs and Neural Kalman Fusion provides a more robust physical constraint against long-term drift. Furthermore, our model maintains a highly competitive computational footprint during both training and inference.

Table A.5: Performance comparison on O_3 prediction (9-day horizon), including runtime.

Methods	9 Days			Runtime	
	MAE	RMSE	SMAPE	Training	Inference
DLinear	29.07	38.92	0.49	5h	3m
TimeMoE	28.83	38.64	0.52	30h	5m
PatchTST	28.97	38.06	0.50	10h	5m
AirDualODE	30.58	38.80	0.55	15h	3m
AeroChem (Ours)	**28.53**	**38.12**	0.53	20h	3m

Table A.6: Comparison with chemistry-aware baselines.

Methods	3 Days			6 Days			9 Days			Sudden Change			Runtime	
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	Train	Infer
MDSTNet [33]	23.27	34.98	0.51	23.52	34.98	0.53	24.77	35.35	0.55	34.12	48.22	0.51	72h	5m
FuXi-Air [5]	27.90	39.23	0.55	28.56	41.97	0.66	29.28	43.74	0.67	36.56	50.43	0.55	-	3m
AeroChem (Ours)	**21.95**	**37.20**	**0.50**	**23.00**	**38.80**	**0.52**	**24.08**	**39.34**	**0.54**	**34.50**	**49.50**	**0.52**	32h	2m

Effect of future meteorological inputs We analyze the contribution of future meteorological inputs across different models in Table A.7. We analyze the contribution of future meteorological inputs across

Table A.7: Effect of future meteorological inputs.

Model	Future Meteo	3-Day MAE	9-Day MAE
DLinear	No	22.57	24.50
DLinear	Yes	22.31	24.12
PatchTST	No	22.69	25.25
PatchTST	Yes	22.45	24.78
TimeMoE	No	21.68	25.32
TimeMoE	Yes	21.42	24.81
AeroChem	Yes	21.95	**24.08**
AeroChem	Yes ($\mathbf{E}_t = \mathbf{0}$)	22.28	24.92

different models in Table A.7. While integrating future meteorology uniformly improves the performance of direct-mapping baselines, AeroChem still consistently achieves the lowest overall error. More importantly, we evaluate a degraded setting for our model where future emission inventories are completely masked ($E_{t>0} = 0$). Even in this strict zero-future-emission scenario, AeroChem remains a competitive result to baselines that have full access to future meteorology. This underscores the resilience of our closed-loop latent correction mechanism in mitigating the impact of missing external drivers.

Effect of closed-loop neural state correction We isolate the contribution of closed-loop neural state correction mechanisms in Table A.8. These results demonstrate that the neural Kalman formulation provides

Table A.8: Effect of closed-loop neural state configurations.

Configuration	KnowAir 9-Day MAE
Open-loop UDE (AeroChem-K)	35.50
Standard Kalman (Gaussian innovation)	24.44
Neural Kalman (learned correction)	**24.08**

consistent gains over classical data corrections, highlighting the importance of learned correction mechanisms.

A.14 Statistical significance

Regarding statistical significance, we report paired tests over matched forecast windows where possible, including Welch's t-tests [31] over five independent runs against the strongest baseline in each setting. Tab A.9 reports unadjusted two-sided p-values.

These results make the empirical picture more precise. At 3 days, the strongest Transformer-based baselines are significantly better on MAE in both datasets. At some mid-horizon settings, the differences are small and should be interpreted cautiously, especially after multiple-comparison correction. However, in the target regimes of this work, longer horizons and sudden pollution changes, AeroChem shows clearer gains. We will revise the manuscript to emphasize this crossover behavior rather than claiming uniform superiority across all settings.

Table A.9: Statistical significance tests comparing `AeroChem` against the strongest baseline in each setting. Values are reported as mean $\pm$ standard deviation. Welch's t-test is conducted over five independent runs. The reported p-values are unadjusted two-sided values.

Setting (MAE)	**AeroChem**	**Best baseline**	**Welch t (df)**	**p (2-sided)**
KnowAir 3d	21.95 ± 0.06	Fredformer 21.45 ± 0.08	11.2 (7.4)	7×10^{-6}
KnowAir 6d	23.00 ± 0.08	TimeMoE 23.15 ± 0.11	2.5 (7.3)	0.042
KnowAir 9d	24.08 ± 0.10	DLinear 24.50 ± 0.12	6.0 (7.7)	4×10^{-4}
KnowAir Sudden	34.50 ± 0.18	TimeMoE 36.54 ± 0.21	16.5 (7.8)	2×10^{-7}
HanoiAir 3d	42.74 ± 0.10	Fredformer 41.60 ± 0.16	13.5 (6.7)	4×10^{-6}
HanoiAir 6d	44.50 ± 0.13	TimeMoE 44.15 ± 0.21	3.2 (6.7)	0.017
HanoiAir 9d	46.80 ± 0.16	DLinear 47.10 ± 0.22	2.5 (7.3)	0.042
HanoiAir Sudden	68.80 ± 0.28	TimeMoE 71.54 ± 0.38	13.0 (7.4)	2×10^{-6}

Table A.10: Relative changes in normalized MAE and normalized RMSE of `AeroChem` with respect to the strongest baseline in each setting.

Setting	**Δ nMAE vs. best baseline**	**Δ nRMSE vs. best baseline**
KnowAir 3d	$+2.33\%$ (Fredformer)	0.00% (tie, TimeMoE)
KnowAir 6d	-0.65% (TimeMoE)	-0.39% (TimeMoE)
KnowAir 9d	-1.71% (DLinear)	-2.02% (TimeMoE)
KnowAir Sudden	-5.58% (TimeMoE)	$+0.77\%$ (TimeMoE)
HanoiAir 3d	$+2.74\%$ (Fredformer)	-1.08% (TimeMoE)
HanoiAir 6d	$+0.79\%$ (TimeMoE)	-0.90% (S-Mamba)
HanoiAir 9d	-0.64% (DLinear)	-1.17% (TimeMoE)
HanoiAir Sudden	-3.83% (TimeMoE)	-2.10% (TimeMoE)

A.15 Normalized Performance Comparison

We added normalized metrics:

$$\mathrm{nMAE} = \frac{\mathrm{MAE}}{\bar{y}_{\mathrm{test}}}, \qquad \mathrm{nRMSE} = \frac{\mathrm{RMSE}}{\bar{y}_{\mathrm{test}}},$$

where $\bar{y}_{\mathrm{test}}$ is the test-period mean concentration on the original physical scale. This value will be stated in the caption and in the dataset-statistics table. Since normalization by a common dataset mean preserves between-model ratios, the relative improvements can be derived from the submitted tables. We report them in Tab A.10, where negative values favor `AeroChem` and the strongest baseline is named for each cell. The normalized metrics support the same conclusion: `AeroChem` trails by 2.3 to 2.7% nMAE at 3d, but improves nMAE by up to 5.58% at long horizons and sudden events. On HanoiAir, `AeroChem` achieves lower nRMSE in all settings, consistent with reduced error accumulation.

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 accurately reflect the paper's main contributions, including the AeroChem framework, the HanoiAir dataset, and the empirical improvements in forecasting performance and uncertainty estimation.

   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: The paper explicitly discusses limitations, including simplified atmospheric chemistry assumptions, approximation errors in uncertainty modeling, and dependence on external meteorological and emission data sources.

   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 (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - 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. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - 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.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: [Yes]

   Justification: The paper presents formal assumptions, a theoretical result, and provides the corresponding proof in the supplementary appendix.

   Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the 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, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, 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 provides sufficient methodological detail, including architecture design, dataset construction, training objectives, evaluation metrics, and implementation details to reproduce the main experiments.

   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: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - 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. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [Yes]

   Justification: The source code is released via an public repository, and sufficient instructions for reproducing the experiments are provided.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) 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. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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 paper specifies dataset splits, optimizer settings, hyperparameters, forecasting horizons, and evaluation protocols in both the main paper and supplementary material.

   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 and make sense of them.
   - 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: Experimental results are reported with mean and standard deviation across multiple runs, providing appropriate variance estimates for the reported performance.

   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, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: [Yes]

Justification: The paper provides computational complexity comparisons (parameter count and FLOPs) and experimental resource configurations sufficient to understand computational requirements.

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 of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into 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 research complies with the NeurIPS Code of Ethics and uses environmental monitoring data without involving harmful deployment scenarios or human subjects.

   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 (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

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 includes a dedicated Broader Impact section discussing societal benefits such as air pollution forecasting, environmental policy support, and public health planning, along with potential deployment considerations.

    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 (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 proposed model and dataset do not involve high-risk generative models or sensitive data that would require additional safeguards.

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 to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and 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: [Yes]

Justification: All external datasets, benchmark methods, and prior works are properly cited and credited.

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 (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the 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: [Yes]

Justification: The paper introduces the HanoiAir dataset and provides sufficient documentation on data sources, preprocessing, and variable construction.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- 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). You can either create an anonymized URL or include an anonymized zip file.

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 work does not involve crowdsourcing or human-subject experiments.

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, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- 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 work does not involve human-subject research and therefore does not require IRB approval.

    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 any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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: [N/A]

    Justification: Large language models are not used as part of the core methodology, model design, or experimental pipeline of this work.

    Guidelines:

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