Authors’ Response to Reviews of

A Transformer-based agent model of GEOS-Chem v14.2.2 for informative prediction of $PM_{2.5}$ and O_3 levels to future emission scenarios: TGEOS v1.0

Dehao Li, Jianbing Jin*, Guoqiang Wang, Mijie Pang, Hong Liao*

Geoscientific Model Development Discussions, 10.5194/egusphere-2025-2186

RC: ***Reviewers’ Comment***, AR: Authors’ Response, □ Manuscript Text

1. Overview

Response to Referee 1: We would like to express our sincere gratitude to the reviewer for the thorough evaluation of our manuscript and the thoughtful, in-depth comments. Their constructive insights have been invaluable in strengthening the quality and clarity of our work.

2. Major concerns

RC: ***I am not quite sure how the authors transformed the data from a shape of (1, 1045) to (9, 116) for implementing the CNN model. The range of hyperparameter tuning of Random Forests also seems somewhat biased. For example, why is the maximum number of estimators limited to 500, and why does the tree depth start from 10? In addition, it is unclear whether Table 1 in the response to review refers to the results for PM2.5 or O3, but the performance appears to be clearly worse than that shown in Figure 2 (also in the response to review).***

AR: Thank you for the comment. As illustrated in Fig. 1, the original input of TGEOS is a flattened vector that concatenates the 116 features from a 3×3 neighborhood (9 grid cells), along with the month indicator, resulting in a total length of 1045. For the CNN implementation, we reconstructed this vector back into its spatial layout by reshaping it into a 3×3×116 feature tensor. The month index was extracted separately and embedded into the model during training. This reshaping step does not alter the data content; it merely restores the spatial structure needed for convolutional processing.

On the other hand, we acknowledge the reviewer’s concerns for RF model. During hyperparameter tuning, we used only 40% of the training data to accelerate model selection. Therefore, the tuning performance and the final full-data performance may differ. Furthermore, we admit that our previous statements have caused some misunderstandings. In the previous response the R^2 values shown in Table 1 referred to the squared correlation coefficient (R^2), whereas Figure 2 presented the correlation coefficient (R). This difference in metrics may cause the performance to appear clearly inconsistent. Table 1 (also in previous response) reported the average R^2 and MAE across all 12 prediction targets ($PM_{2.5}$ and O_3 combined), rather than for a single target. Therefore, we have corrected this inconsistency and the renewed hyperparameter tuning results of RF model are listed in Table 1, and we have checked this elsewhere in the article.

Furthermore, our preliminary experiments showed that the RF model is relatively insensitive to the number of estimators since the error tends to converge as the number of trees increases (Probst et al., 2019). Around 100

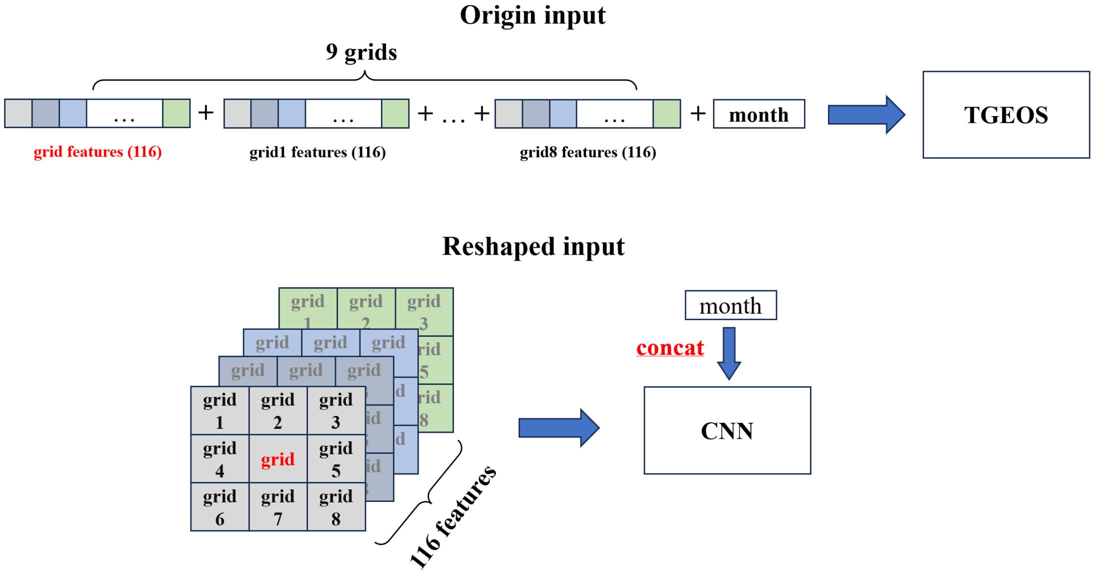

Figure 1: Overview of model inputs for TGEOS and CNN.

trees already achieved stable performance, and increasing beyond 500 considerably raised computational cost without meaningful improvement. For this reason, we set the upper bound to 500 to balance accuracy and efficiency. Since our dataset is high-dimensional, shallow trees (e.g., max_depth < 10) exhibited clear underfitting and could not capture the nonlinear relationships among features, as shown in Table 1. Therefore, we restricted the search range to deeper trees (max_depth > 10) to ensure adequate model capacity.

We have updated Figure 11 in the manuscript and added Table S9 in the supplement.

RC: ***The definition of RSM remains unclear, especially since "L68-71" mentioned that ML techniques have been employed in RSM techniques to further optimize modeling efficiency and estimation accuracy of RSMs. According to the authors' own definition of RSMs as "statistical surrogates" (L63), does an RSM that incorporates ML still qualify as an RSM? In addition, I find the term "face-to-face" (L99) unclear. Are they referring to surface-to-surface?***

AR: Thank you for the comment. We clarify that Response Surface Methodology (RSM) is a specific class of surrogate modeling techniques, rather than a standalone statistical model. RSM belongs to the broader family of surrogate models because its core purpose is to construct an efficient approximation (a "response surface") of an expensive or complex system. In classical RSM, this surrogate is a low-order polynomial; however, modern surrogate-modeling literature recognizes that the response surface can also be built using machine-learning regressors such as neural networks (Xing et al., 2020). Thus, incorporating ML methods does not change the identity of RSM. Instead, it reflects an evolution within surrogate modeling, in which the RSM workflow (from systematic CTM perturbations to statistical emulator and to pollutant concentration response surface) remains intact, while the surrogate function itself can be improved using more advanced learners.

In addition, as to the "face-to-face" in line 99, it refers to surface-to-surface modeling that uses feature maps as inputs while maps of targets as outputs.

Table 1: The five hyperparameter configurations exhibiting the best performance, ranked according to the correlation coefficient (R) and mean absolute error (MAE), along with three configurations demonstrating the worst performance based on the same evaluation metrics.

N estimators	Max depth	Min samples split	Min samples leaf	Avg R	Avg MAE
300	25	4	2	0.8604	5.059
200	50	5	10	0.8602	5.060
100	25	10	9	0.8592	5.070
100	15	6	6	0.8581	5.104
300	15	7	4	0.8565	5.116
300	5	4	6	0.7649	8.179
300	5	4	6	0.7649	8.179
500	5	5	7	0.7648	8.184

We have revised the corresponding part of the manuscript and added these contents in the revised supplementary, details are shown in below:

> ### 1 Introduction
>
> To overcome the computational challenge and efficiently retrieve the nonlinear relationship between emissions and concentrations, data-driven statistical emulators have been proposed to accelerate numerical simulations (Castruccio et al., 2014). As a simplified-form of CTM, a reliable emulator can effectively capture the intricate relationships between important CTM inputs and concentration outputs, and rapidly estimate "CTM-aligned" concentrations of pollutants. ~~Response Surface Model (RSM), served as statistical surrogates developed by the US EPA (EPA, 2006) to establish the relationships between emission rates and the concentration responses of CTM, has been continuously developed since the past decade.~~ As an effective framework for constructing CTM-based statistical surrogate models, Response Surface Model (RSM) was originally developed by the US EPA (EPA, 2006) to establish the relationships between emission rates and the concentration responses of CTM by constructing pollutant response surfaces using polynomial or parametric regression methods. RSM has been successfully employed in the response modeling of $PM_{2.5}$ (Wang et al., 2011) and O_3 (Xing et al., 2011) to precursor emissions in China for typical regions. To address the inherent computational burden stemmed from considerable advanced CTMsupports for model building (Xing et al., 2011), optimized versions of conventional RSM were developed, such as ERSM (Zhao et al., 2015; Xing et al., 2017) and pf-RSM (Xing et al., 2018). Recently, novel machine learning (ML) techniques, for its well performance in simulating complex non-linear relationships in atmospheric systems (Liu et al., 2021) and dealing with tasks involving multiple variables and objectives (Masmoudi et al., 2020; Huang et al., 2021), ~~have been employed in RSM techniques to further optimize modeling efficiency and estimation accuracy of RSMs~~ have been employed as alternative fitting modules within the RSM framework to further optimize modeling efficiency and estimation accuracy of RSM (Xing et al., 2020; Li et al.,

2022). Based on this advantage, many studies have attempted to build effective emulators using pure ML method (Huang et al., 2021; Zhang et al., 2023a). For example, Zhang et al. (2023a) used ResCNN framewoek to predict annual $PM_{2.5}$ concentration from fossil energy use and reveal the co-benefits of the energy transition, demonstrating the potential of ML method in addressing the emulator modeling task.

TextS2. Rationality of selecting 3x3 neighborhood as research domain

Since the main purpose of TGEOS framework is to provide rapid monthly-scale concentration estimation of air pollutants under future emission scenarios, it means that estimations for each month is evaluated independently rather than as part of a temporal sequence, and the explicit spatiotemporal evolution is not involved. Unlike short-term air quality forecasting where long-range transport of pollutants and emissions cannot be ignored, large time scale prediction as monthly or annually substantially weakens such long-range physical transport effects (Jung et al., 2022). Under monthly averaging, the pollutant concentration at a target grid cell is predominantly influenced by its immediate surroundings (Hu et al., 2025), and the relatively coarse spatial resolution encompasses the dominant transport footprint. At 0.5° × 0.625° resolution, a 3 × 3 grid domain already covers a geographic extent of approximately 26,000 to 33,000 km^2, within which emissions typically exert the most pronounced influence on local pollutant concentrations (Liu et al., 2019). Therefore, our analysis focuses on a 3×3 neighborhood.

RC: ***The authors emphasize "Representing these as full spatial fields and training a CNN-based model in a field-to-field form was not feasible under our available computational resources. Therefore, instead of modeling spatial fields directly, we adopted a high-dimensional sequential modeling strategy" (L227-229) and "This approach offers a more scalable solution while preserving the ability to capture complex relationships among variables across grid points" (L231-232) without providing sufficient evidence. They should justify their claims based on computational complexity. For example, while the computational complexity of the Transformer is greater than O(N^2), what is it for the CNN in this case? The authors also claim that "Transformers are better suited for capturing long-range dependencies." However, since the current study domain is limited to a 3×3 grid (i.e., only 9 spatial points), there are essentially no long-range spatial dependencies to capture. In this setting, the advantage of Transformer architectures over convolutional models may not be justified. In addition, since the Vision Transformer (ViT) can balance both spatial and temporal representations, I do not understand why the authors did not consider testing it.***

AR: Thank you very much for your thoughtful comment. First, we would like to clarify that the present study does not involve explicit spatiotemporal evolution. Our framework is designed for rapid monthly-scale concentration estimation of air pollutants under future emission scenarios, where each month is evaluated independently rather than as part of a temporal sequence. Unlike short-term air quality forecasting where long-range transport of pollutants and emissions cannot be ignored, large time scale prediction as monthly or annually substantially weakens such long-range physical transport effects (Jung et al., 2022). Under monthly averaging, the pollutant concentration at a target grid cell is predominantly influenced by its immediate surroundings (Hu et al., 2025), and the relatively coarse spatial resolution encompasses the dominant transport footprint. At 0.5° × 0.625° resolution, a 3 × 3 grid domain already covers a geographic extent of approximately 26,000 to 33,000 km^2, within which emissions typically exert the most pronounced influence on local pollutant concentrations (Liu et al., 2019). Therefore, our analysis focuses on a 3×3 neighborhood.

Given the limited number of spatial grid points (9 grids for each sample), the available spatial information is inherently insufficient for architectures that rely heavily on rich spatial structure, such as CNNs or Vision

Transformers (ViTs). We have limited study domain but multiple input channels. To better capture the correlations between these variables, we first aggregate the inputs of the 9 grid cells into a high-dimensional feature space and then allow the Transformer's self-attention mechanism to learn effective interactions among these features. Accordingly, the "long-range dependencies" mentioned in the manuscript refer to relationships in the high-dimensional feature space, rather than long-range physical spatial dependencies.

Furthermore, the reason why we emphasized it is infeasible to conduct surface-to-surface modeling based on the entire study area is primarily due to insufficient training data. The training dataset comprises only 36 scenarios, yielding fewer than 500 samples totally, which significantly falls short of the requirements for robust model development. In addition, to clarify the computational advantages of our sequence-based Transformer design, we provide a direct comparison with a hypothetical global CNN that operates on the full spatial field (surface-to-surface mapping).

Theoretically, a global CNN takes as input a spatial tensor of size $H \times W$ with C$_{in}$ channels. For a single convolutional layer with C$_{out}$ filters of size $k \times k$, the computational complexity $\mathcal{O}$ is:

$$\mathcal{O}(NC_{in}C_{out}k^2) \tag{1}$$

where N equals $H \times W$ and is the total number of pixels in the feature map. C_{in} and C_{out} refer to input and output channels, respectively. k is the size of convolutional kernel.

For a Transformer encoder with hidden dimension d and sequence length N, the per-layer computational cost $\mathcal{O}$ is dominated by self-attention:

$$\mathcal{O}(N^2d) \tag{2}$$

Including the feed-forward network ($\mathcal{O}(Nd^2)$), the total per-layer cost is:

$$\mathcal{O}(N^2d + Nd^2) \tag{3}$$

We find that the computational complexity of CNNs scales linearly with N (like $\mathcal{O}(N)$), whereas the self-attention mechanism in Transformers exhibits quadratic growth with respect to the number of input sequences ($\mathcal{O}(N^2)$). It seems that computational complexity of Transformer is greater than that of CNN for each layer, especially when N is large. However, the actual computational complexity is determined by the specific model.

In this research, for surface-to-surface CNN, N is around 7100 at China's spatial resolution of 0.5° $\times$ 0.625°. C_{in} is 116 since we have 116 features, C_{out} and k are assumed as 128 and 3, respectively, in a typical CNN layer. So the resulting per-layer complexity $\mathcal{O}$ is nearly 9.5 x 10^8 FLOPs.

$$\mathcal{O}(7100 \times 116 \times 128 \times 3^2) \approx 9.5 \times 10^8 FLOPs \tag{4}$$

It should be noticed that a reliable and realistic surface-to-surface CNN used in air quality modeling, such U-Net and ResNet (Xing et al., 2020; Huang et al., 2021), contains at least 19 such layers and larger number of channels (typically increases exponentially like 64, 128, 256, 512), yielding the total CNN cost of approximately 10^{10} to 10^{11} FLOPs.

$$Total\ cost \approx 10^{10} \sim 10^{11}\ FLOPs \tag{5}$$

In contrast, our method converts each grid cell's local 3×3 neighborhood (9 points × 116 features) into a sequence token of length N = 1045. With hidden dim d is 512, the per-layer cost is:

$$\mathcal{O}(1045^2 \times 512 + 1045 \times 512^2) \approx 8.3 \times 10^8 \tag{6}$$

With 6 encoder layers in the model, the total complexity is nearly 5.0×10^9 FLOPs, which is 1 or 2 orders of magnitude smaller than that of surface-to-surface modeling (10^{10} to 10^{11} FLOPs).

$$Total\ cost \approx 5.0 \times 10^9\ FLOPs \tag{7}$$

Overall, the computational complexity of a single CNN layer is generally lower than that of a Transformer, whereas the total complexity of a global surface-to-surface CNN is strikingly larger than that of a sequence-based Transformer in this case, because achieving competitive performance of CNN require deeper architectures with large hidden dimensions (He et al., 2015; Liu et al., 2023), which also makes it require more training epochs to converge. Moreover, in practical deep learning applications, CNNs are predominantly memory-bound (Leitersdorf et al., 2023), constrained by factors such as GPU memory bandwidth, activation graph size, and memory access overhead. Consequently, CNNs continue to face significant computational bottlenecks and substantial memory requirements when performing large-scale spatial modeling tasks, such as those involving the Chinese region at a resolution of 0.5° × 0.625°.

Additionally, we construct a ViT-like model for comparison because of the reviewer's concerns about Vision Transformer. The overall framework of this ViT model is presented in Fig. 2. In this model, the 3×3 grid cells are treated as spatial patches; a lightweight CNN is used to generate patch-level embeddings (Wang et al., 2022; Yao et al., 2024); a month token functions as a global CLS token; and the resulting token sequence is then processed by a multi-layer Transformer encoder. The hyperparameter settings and evaluation results are provided in Table 2 and Fig. 3. The results show that the ViT architecture performs slightly worse than our pure Transformer–based TGEOS model. This is primarily because the spatial domain consists of only a 3×3 grid, which is too limited to form meaningful spatial structures. Consequently, the ViT model cannot fully exploit its advantages in modeling long-range spatial dependencies.

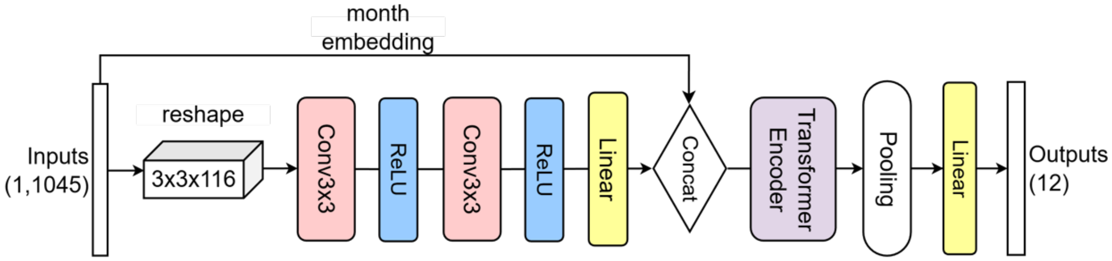

Figure 2: Model architecture of ViT model.

We sincerely appreciate the reviewer's insightful recommendation regarding the use of Vision Transformers. This suggestion is highly relevant and aligns closely with the broader methodological directions in advanced

Table 2: Hyperparameters tuning for ViT model.

Name	Tuning range	Best value
Hidden dim	128, 256, 512, 1024	512
Number of attention heads	4, 6, 8	8
Number of encoder layers	2, 4, 6	4
CNN output channels	[64, 128, 256]	128

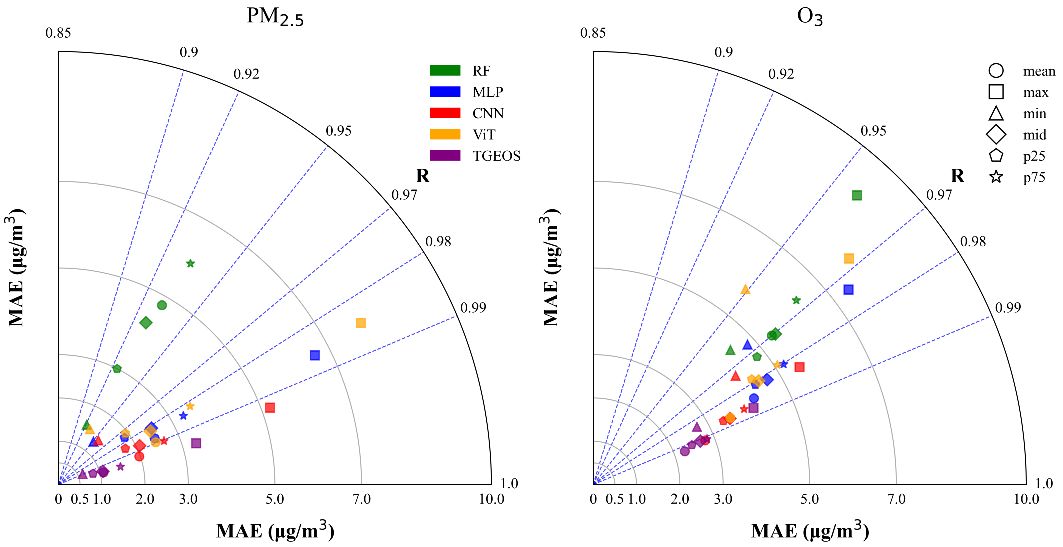

Figure 3: Predictive performance of five models, with green, blue, red, orange and purple represents predictions of RF, MLP, CNN, ViT and TGEOS, respectively. All indicators are averaged in national scale and computed based on the six test scenarios.

air quality modeling. Indeed, ViT architectures are particularly advantageous for surface-to-surface, short-term air quality forecasting, where capturing long-range spatial interactions is essential. The reviewer's comment is therefore not only well-taken but also greatly encouraging for the continued development of our modeling framework. In fact, this line of thinking resonates strongly with our group's recent work. Our latest short-term air pollution forecasting system, Zeeman (Pang et al., 2025), is built primarily on ViT-based architectures, precisely because of their strength in modeling extended spatial dependencies at fine temporal scales. However, due to the specific characteristics of the dataset and study domain (3 × 3 grid) in this work, a ViT architecture cannot fully leverage its strengths and thus becomes less suitable for this specific task.

We have revised corresponding part of the manuscript and added these contents in the supplementary, details are presented below.

2.2.1 Model architecture

In previous emulator modeling, field-to-field modeling using the convolution neural networks (CNN) architecture has been widely used because of the efficient usage of capturing the spatial relationship between features and concentrations (Xing et al., 2020; Huang et al., 2021; Liu et al., 2022). However, both model inputs and outputs are represented as high-resolution 2D matrices in these approaches, which require significant GPU memory and computational resources, especially when the number of input variables increases. As a result, these model were limited to a few kinds of emission species (Xing et al., 2020), or solely average values (Liu et al., 2022). In contrast, our dataset includes over 100 variables, including sectoral emissions and multiple meteorological parameters. Representing these as full spatial fields and training a surface-to-surface model was not feasible under our available computational resources. To clarify this, we provided a direct comparison between a hypothetical global CNN that operates on the full spatial field (surface-to-surface mapping) and Transformer used in this study in Text S3. For importantly, the number of scenarios in our dataset is limited, which could only yield fewer than 500 samples totally for training and significantly fall short of the requirements for robust model development. Therefore, instead of modeling spatial fields directly, we adopted a high-dimensional sequential modeling strategy. Our dataset is not field-based but rather consists of structured multivariate sequences, in which spatial and feature-level information (e.g., emissions, meteorology, and concentrations at 3×3 neighborhood with 9 grid cells) is flattened and treated as a sequence of tokens fed into TGEOS model. This approach offers a more scalable solution while preserving the ability to capture complex relationships among variables across grid points

3.4 Comparison of different machine learning models

To validate the performance of the TGEOS model in "emission-concentration" modeling against other machine learning models, four widely used machine learning frameworks, namely Multilayer Perceptrons (MLP), Random Forests (RF), Convolutional Neural Network (CNN), and Vision Transformer (ViT) employed in previous studies (Xing et al., 2020; Huang et al., 2021), were simultaneously employed based on the multi-scenario dataset mentioned in Section 2.1. ~~For each ML model, we identified the model with the best combination of hyperparameters after fine-tuning process based on Optuna tool. The MLP model uses 4 hidden layers with 2048, 1024, 512, and 256 neurons, applying ReLU activation and Dropout to prevent overfitting. The optimizer is Adam with a learning rate of $1e^{-3}$, and the loss function is Mean Squared Error (MSE). Training uses a batch size of 1024 and 100 epochs, with a learning rate scheduler to adjust the learning rate dynamically. The RF model uses 300 trees with a maximum depth of 25, a minimum sample split of 4, and a minimum sample per leaf of 2. It uses parallel computation with all CPU cores and performs feature selection by choosing the top 500 important features. The model consists of two convolutional layers followed by fully connected layers, with an additional embedding layer to incorporate month information. The first convolutional layer applies 32 filters of size 3×3 (padding = 1) to the single-channel input, followed by a second convolutional layer with 128 filters of the same size. Both convolutional layers use ReLU activations. The output feature maps are then processed by an adaptive average pooling layer to reduce the spatial resolution to 29×3. To integrate temporal information, a month embedding layer maps month indices (1–12) to a 4-dimensional vector. The pooled convolutional features are flattened and concatenated with the month embedding, forming the input to a three-layer fully connected network: the first linear layer maps the concatenated vector to 256 units, the second reduces it to 64 units, and the final output layer produces 12 regression targets. ReLU activation functions are applied after the first and second fully connected layers. For each model, hyperparameters were obtained after fine-tuning based on Optuna tool.~~ The MLP model uses 4 hidden layers with 2048, 1024, 512, and 256 neurons, applying

ReLU activation and Dropout to prevent overfitting. The RF model uses 300 trees with a maximum depth of 25, a minimum sample split of 4, and a minimum sample per leaf of 2. It uses parallel computation with all CPU cores and performs feature selection by choosing the top 500 important features. The CNN model uses two 3×3 convolutional layers with ReLU activation, followed by adaptive pooling to 29×3. A month embedding is concatenated with the flattened pooled features and passed through three fully connected layers with ReLU applied to the first two. In the ViT model, the 3×3 grid cells are treated as spatial patches; a lightweight CNN is used to generate patch-level embeddings (Wang et al., 2022; Yao et al., 2024); a month token functions as a global CLS token; and the resulting token sequence is then processed by a multi-layer Transformer encoder. It should be noticed that the model inputs for CNN and ViT were reshaped from (1,1045) to (3×3×116) to cater to model architecture, with specific description shown in Fig. S25.

Table S2 and S3 summarize the performance of the three models on the entire test set. We found that TGEOS outperforms the other two models in both R^2 and MAE metrics. To clearly illustrate the predictive performance of different models, we presented a modified Taylor diagram (Taylor, 2005; Fang et al., 2023) in Fig. 11. This diagram simultaneously displays the Mean Absolute Error (MAE) and correlation coefficient (R) for predictions of $PM_{2.5}$ and O_3 indicators from four models in China domain. Our findings indicate that the Random Forest (RF) model performs the poorest. This is primarily due to its reliance on feature importance assessments during feature selection, which overlooks potential underlying features in the data, adversely affecting the model's fitting capability. Additionally, the RF model is sensitive to the distribution of training data, leading to limited extrapolation abilities and poor predictive performance for extreme values. In contrast, the Multi-Layer Perceptron (MLP) shows a significant improvement in predictive performance relative to the RF model. Leveraging its multi-layer neural network structure, the MLP can more effectively learn complex relationships between multiple features. But this layered structure can struggle when dealing with high-dimensional feature spaces, especially for highly stochastic indicators such as maximum values, where the MLP still exhibits considerable prediction errors. ~~Compared to the previously selected MLP and RF models, the CNN-based model demonstrates superior performance, characterized by higher R values as well as lower MAE. This advantage can be attributed to the CNN's local convolution kernel, which is capable of capturing patterns among adjacent data points.~~ Compared to MLP and RF models, models based on CNN and ViT frameworks demonstrate better performance, characterized by higher R values as well as lower MAE. However, these models still perform badly for the prediction of indicators reflecting extreme pollutant events such as 75-percentile and maximum, which is mainly because the available spatial information (3×3 grid) is inherently insufficient for these architectures relying heavily on rich spatial structure.

Conversely, the Transformer-based TGEOS model demonstrates superior performance compared to the other models, exhibiting higher R values (exceeding 0.98 and 0.97) and lower MAE values (less than 4.0 g/m^3 for the majority indicators). These results suggest a higher degree of reliability and accuracy in its predictions. For several indicators where MLP performs poorly, TGEOS demonstrates substantial improvements. ~~The superiority of the Transformer model can be attributed to its greater number of parameters and more complex architecture, which leverage powerful feature extraction capabilities and self-attention mechanisms, allowing it to adapt to intricate patterns and relationships. In contrast to CNN, which are constrained by fixed convolution kernels and limited network depth, TGEOS exhibits a stronger capacity for capturing complex relationships in high-dimensional data. we first aggregate the inputs of the 9 grid cells into a high-dimensional feature space and then allow the Transformer's self-attention mechanism to learn effective interactions among~~

~~these features.~~ The superiority of the Transformer model can be attributed to its greater number of parameters and more complex architecture, which leverage powerful feature extraction capabilities and self-attention mechanisms, allowing it to capture complex relationships in the high-dimensional feature space. It is worth emphasizing that although the capacity of the ViT model in our study was inherently constrained by the limited spatial information available from the compact 3 × 3 domain, as well as the long-term, monthly timescale that reduces meaningful spatial variability—it still achieved strong predictive performance, with R values exceeding 0.97. This demonstrates the promising representational power of ViT architectures even under suboptimal spatial conditions. Given that ViTs typically rely on richer spatial structures to fully realize their advantages in capturing long-range spatial dependencies, there remains substantial room for further performance gains in settings designed for surface-to-surface, short-term prediction (Pang et al., 2025), where such spatial relationships become more pronounced.

4 Conclusions

The TGEOS model still have some limitations to be improved. Firstly, it should be noted that the predictions generated by TGEOS remain incapable of accurately representing actual air pollutant concentrations, even though TGEOS is highly consistent with GEOS-Chem, since systematic biases have been demonstrated to exist within GEOS-Chem itself (Travis and Jacob, 2019; Miao et al., 2020). Therefore, correcting errors in TEGOS based on near-real observations or reanalysis data is of paramount importance and constitutes a priority for our subsequent research. Additionally, in order to isolate the effects of emission changes on future air quality, the meteorology used in all GC simulations was fixed to the year 2017, following the approach of previous studies (Shi et al., 2021; Wang et al., 2023a). This methodology constrains TGEOS's ability to provide robust predictions under cross-meteorology conditions and prevents it from capturing meteorology–emission interactions and potential "emission–climate" feedbacks. Similar limitations have also been observed in other CTM emulators (Xing et al., 2020; Huang et al., 2021; Liu et al., 2022). Therefore, incorporating diverse climate scenarios that account for meteorological variability will be essential to enhance TGEOS's predictive capability for future air quality under more complex conditions where both emissions and meteorology evolve simultaneously.

Finally, the framework established in this study also reveals promising opportunities for air quality modeling based on ViT models, which may provide substantial performance gains when applied to surface-to-surface and short-term prediction tasks where richer spatial information is available. Although the current study design limited the research domain and temporal scale, future extensions of TGEOS that incorporate fully gridded inputs or higher temporal resolution could make it possible to integrate ViT architectures more effectively. Such developments would enable TGEOS to evolve from a point-based, long-term emulator into a short-term air quality prediction system operating over a larger spatial domain.

TextS3. Computational complexity of Transformer and global CNN in this case

Theoretically, a global CNN takes as input a spatial tensor of size $H \times W$ with C$_{in}$ channels. For a single convolutional layer with C$_{out}$ filters of size $k \times k$, the computational complexity $\mathcal{O}$ is: $\mathcal{O}(NC_{in}C_{out}k^2)$(8) where N equals $H \times W$ and is the total number of pixels in the feature map. C_{in} and C_{out} refer to input and output channels, respectively. k is the size of convolutional kernel. For a Transformer encoder with hidden dimension d and sequence length N, the per-layer computational cost $\mathcal{O}$ is dominated by self-attention: $\mathcal{O}(N^2d)$(9) Including the feed-forward network ($\mathcal{O}(Nd^2)$), the total per-layer cost is: $\mathcal{O}(N^2d+Nd^2)$(10) We find that the computational complexity of CNNs scales linearly with N (like $\mathcal{O}(N)$), whereas the self-attention mechanism in Transformers exhibits quadratic growth with respect to the number of input sequences ($\mathcal{O}(N^2)$). It seems that computational complexity of Transformer is greater than that of CNN for each layer, especially when N is large. However, the actual computational complexity is determined by the specific model. In this research, for surface-to-surface CNN, N is around 7100 at China's spatial resolution of 0.5° × 0.625°. C_{in} is 116 since we have 116 features, C_{out} and k are assumed as 128 and 3, respectively, in a typical CNN layer. So the resulting per-layer complexity $\mathcal{O}$ is nearly 9.5 × 10^8 FLOPs. $\mathcal{O}(7100 \times 116 \times 128 \times 3^2) \approx 9.5 \times 10^8 FLOPs$(11) It should be noticed that a reliable and realistic surface-to-surface CNN used in air quality modeling, such U-Net and ResNet (Xing et al., 2020; Xing et al., 2021), contains at least 19 such layers and larger number of channels (typically increases exponentially like 64, 128, 256, 512), yielding the total CNN cost of approximately 10^{10} to 10^{11} FLOPs. $Total\ cost \approx 10^{10} \sim 10^{11}\ FLOPs$(12) In contrast, our method converts each grid cell's local 3×3 neighborhood (9 points × 116 features) into a sequence token of length N = 1045. With hidden dim d is 512, the per-layer cost is: $\mathcal{O}(1045^2 \times 512 + 1045 \times 512^2) \approx 8.3 \times 10^8$(13) With 6 encoder layers in the model, the total complexity is nearly 5.0 × 10^9 FLOPs, which is 1 or 2 orders of magnitude smaller than that of surface-to-surface modeling (10^{10} to 10^{11} FLOPs). $Total\ cost \approx 5.0 \times 10^9\ FLOPs$(14) Overall, the computational complexity of a single CNN layer is generally lower than that of a Transformer, whereas the total complexity of a global surface-to-surface CNN is strikingly larger than that of a sequence-based Transformer in this case, because achieving competitive performance of CNN require deeper architectures with large hidden dimensions, which also makes it require longer training epochs to converge. Moreover, in practical deep learning applications, CNNs are predominantly memory-bound, constrained by factors such as GPU memory bandwidth, activation graph size, and memory access overhead. Consequently, CNNs continue to face significant computational bottlenecks and substantial memory requirements when performing large-scale spatial modeling tasks, such as those involving the Chinese region at a resolution of 0.5° × 0.625°.

3. Minor concerns

RC: ***Text S1 is still titled "Fine-tuning experiments."***

AR: Thank you for pointing this out. We have corrected "Fine-tuning experiments" to "Methodology for generating perturbation scenarios" in Text S1.

References

EPA, U.: Technical Support Document for the Proposed PM NAAQS Rule: Response Surface Modeling, 2006.

He, K., Zhang, X., Ren, S., and Sun, J.: Deep Residual Learning for Image Recognition, URL `https://arxiv.org/abs/1512.03385`, 2015.

Hu, K., Liao, H., Liu, D., Jin, J., Chen, L., Li, S., Wu, Y., Wu, C., Zhao, S., Jiang, X., et al.: A novel method for quantifying the contribution of regional transport to PM2. 5 in Beijing (2013–2020): combining machine learning with concentration-weighted trajectory analysis, Geoscientific Model Development, 18, 3623–3634, 2025.

Huang, L., Liu, S., Yang, Z., Xing, J., Zhang, J., Bian, J., Li, S., Sahu, S. K., Wang, S., and Liu, T.-Y.: Exploring deep learning for air pollutant emission estimation, Geoscientific Model Development Discussions, 2021, 1–22, 2021.

Jung, J., Choi, Y., Souri, A. H., Mousavinezhad, S., Sayeed, A., and Lee, K.: The impact of springtime-transported air pollutants on local air quality with satellite-constrained NOx emission adjustments over East Asia, Journal of Geophysical Research: Atmospheres, 127, e2021JD035 251, 2022.

Leitersdorf, O., Ronen, R., and Kvatinsky, S.: ConvPIM: Evaluating Digital Processing-in-Memory through Convolutional Neural Network Acceleration, URL `https://arxiv.org/abs/2305.04122`, 2023.

Liu, S., Chen, T., Chen, X., Chen, X., Xiao, Q., Wu, B., Kärkkäinen, T., Pechenizkiy, M., Mocanu, D., and Wang, Z.: More ConvNets in the 2020s: Scaling up Kernels Beyond 51x51 using Sparsity, URL `https://arxiv.org/abs/2207.03620`, 2023.

Liu, Y., Zheng, M., Yu, M., Cai, X., Du, H., Li, J., Zhou, T., Yan, C., Wang, X., Shi, Z., et al.: High-time-resolution source apportionment of PM2.5 in Beijing with multiple models, Atmospheric Chemistry and Physics, 19, 6595–6609, 2019.

Pang, M., Jin, J., Segers, A., Lin, H. X., Wang, G., Liao, H., and Han, W.: Zeeman: A Deep Learning Regional Atmospheric Chemistry Transport Model, arXiv preprint arXiv:2510.06140, 2025.

Probst, P., Wright, M. N., and Boulesteix, A.: Hyperparameters and tuning strategies for random forest, WIREs Data Mining and Knowledge Discovery, 9, , 2019.

Wang, C., Xu, H., Zhang, X., Wang, L., Zheng, Z., and Liu, H.: Convolutional embedding makes hierarchical vision transformer stronger, in: European conference on computer vision, pp. 739–756, Springer, 2022.

Xing, J., Zheng, S., Ding, D., Kelly, J. T., Wang, S., Li, S., Qin, T., Ma, M., Dong, Z., Jang, C., et al.: Deep learning for prediction of the air quality response to emission changes, Environmental science & technology, 54, 8589–8600, 2020.

Yao, T., Li, Y., Pan, Y., and Mei, T.: Hiri-vit: Scaling vision transformer with high resolution inputs, IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 6431–6442, 2024.