Supporting Information for "Spatiotemporal Machine Learning Approaches for Atmospheric Composition Emulation in NASA GISS ModelE"

Mohammad H. Erfani[1,2,*], Kara D. Lamb[3], Susanne E. Bauer[2], Kostas Tsigaridis[1,2], Marcus van Lier-Walqui[1,2], Gavin Schmidt[2]

[1]Center for Climate Systems Research, Columbia University, New York, NY 10027

[2]NASA Goddard Institute for Space Studies, New York, NY 10027

[3]Department of Earth and Environmental Engineering, Columbia University, New York, NY 10027

[*]Now at Center for Western Weather and Water Extremes, Scripps Institution of Oceanography, University of California San Diego, San Diego, CA, USA

Contents of this file

1. Dataset Description
2. Model Architectures
3. Model Dynamics and Interpretation
4. Model Parameter Summary and Performance Analysis
5. Comparison of OMA and Non-Interactive (NINT) Schemes
6. Analysis of Model Performance at the Grid Level
7. Experiment Setup for Training SFNO

Corresponding author: Mohammad H. Erfani (moerfani@ucsd.edu)

In the Supplementary Information, we provide a comprehensive overview of the architecture and components of the machine learning models developed and employed in this study. We describe the datasets used, detail the model architectures and their underlying dynamics, and analyze their interpretability using attention visualization during inference. We further summarize model parameters, assess their performance, and compare the OMA and Non-Interactive (NINT) schemes. Next, we analyze model performance at the grid level and evaluate the computational efficiency and inference time of each model. Finally, we describe the experimental setup and training details of the baseline model (SFNO).

S.1. Dataset Description

Figure S1 shows the annual averages of BCB emission sources, concentrations, and their global means over the training period (1950). This figure corresponds to Figure 2 in the main manuscript, enabling comparison of the spatial distribution of BCB emissions and concentrations between these two consecutive years. Based on Figure S1a, the active source regions and the resulting surface-level concentrations (Figure S1b) remain nearly identical to those in 1951. However, the peak global average values in 1950 are slightly lower than those in 1951—for instance, the maximum global average in 1950 is approximately 0.24 μg/m^3, compared to about 0.32 μg/m^3 in 1951 (see Figure S1c).

Figure S2 compares the statistics of BCB concentrations between 1950 and 1951. As shown in Figure S2a, the quartiles, minimum, and maximum values are nearly identical for both years. However, Figure S2b indicates that the extreme outlier values are more frequent in 1951. This is further evident in Figure S2c, where the exceedance probability

plot shows a higher occurrence of extreme BCB concentration events in 1951 compared to 1950.

S.2. Model Architectures

S.2.1. ConvLSTM Architecture

The ConvLSTM was configured with three cells: the first cell applied a convolutional operation to map the five inputs into 64 cell states using a kernel size of 7; the second cell mapped the 64 cell states into 32 using a kernel size of 5; and the final cell reduced the 32 cell states to 16 with a kernel size of 3 before a single convolutional layer mapping 16 cell states to 1 output layer with a kernel size of 1. The last convolution operation acted as a regression layer. Figure S3 illustrates the different components of the model architecture. The hidden state of the last sequence of the third cell, $H^3_{t_{48}}$, with dimensions Batch Size (BS) $\times$ 16 $\times$ 100 $\times$ 154, is fed into the last convolution layer with a kernel size of one to project it to the target values.

Stacking additional convolutional layers generally helps to capture spatial dependencies more effectively, as deeper networks can exponentially increase their capacity to model both spatial and temporal dependencies (Lin et al., 2020). However, because this model is intended to perform emulation in a computationally efficient manner, we aimed to keep it as lightweight as possible. While we have not conducted a formal ablation study, we tested different numbers of cells and hidden-state sizes for each cell to avoid inefficient architectures. In terms of temporal dependencies, we also experimented with various time-step lengths, both longer and shorter than 48, based on the longevity of aerosol species considered in the main model, but using 48 preceding timesteps showed the best performance.

Figure S3 illustrates how the ConvLSTM architecture processes spatiotemporal data over a sequence of 48 timesteps, denoted by $X_{t1}, X_{t2}, \ldots, X_{t48}$, each with dimensions $5 \times 100 \times 154$. These five inputs include the velocity field in 3 dimensions, precipitation, and emissions. The overall input shape, $BS \times 48 \times 5 \times 100 \times 154$, represents the batch size (BS), the sequence length (48), the number of channels (5), and the spatial dimensions (lat = 100, lon = 154) considering 5 pixels of cyclic padding along the longitude on each side and 5 pixels of reflective padding along the latitude on each side. The model consists of three stacked ConvLSTM cells, labeled 'Cell-1,' 'Cell-2,' and 'Cell-3,' each operating across the entire sequence of 48 timesteps. 'Cell-1' applies a 5×5 convolutional filter with 64 channels, producing hidden states H^1 and cell states C^1 at each timestep. 'Cell-2' utilizes a 3×3 convolutional filter with 32 channels, outputting states H^2 and C^2, while 'Cell-3' applies a 3×3 convolutional filter with 16 channels, outputting states H^3 and C^3. At each timestep t, the hidden and cell states from each ConvLSTM cell are updated and passed to the next timestep. After processing all timesteps, the final output H^3_{48} represents the hidden states at the last timestep t_{48}. This is then fed into a 1×1 convolutional layer with a single output channel (i.e., bottleneck layer), producing the final output $\hat{Y}_{t_{48}}$ with dimensions $1 \times 100 \times 154$. This output represents the model's prediction for the given timestep.

S.2.2. Gated Conv3D Architecture

Figure S4 illustrates how the Gated Conv3D architecture processes spatiotemporal data over a sequence of 48 timesteps. In the spatial encoder, the sequential input data is reshaped from $B \times T \times C \times H \times W$ to $(B \times T) \times C \times H \times W$ so that only spatial

correlations are captured, where B is the batch size, T is the number of timesteps, C is the number of feature channels, H is the spatial height, and W is the spatial width.

The output features from the spatial encoder are then reshaped into $(B \times C) \times T \times H \times W$ and fed into six layers of Gated Conv3D. In this step, the model simultaneously extracts spatiotemporal features and reduces the number of feature channels to one. The result is passed through a 1×1 convolutional layer with a single output channel—*temporal readout*—producing the final output $\hat{Y}_{t_{48}}$ with dimensions $B \times 1 \times 1 \times 100 \times 154$.

The number of Gated Conv3D layers is determined based on Equation 1, where b is the dilation base. Using dilation is an effective way to increase the receptive field size while keeping the number of layers relatively small. In this study, the kernel size k is set to 2, and the sequence length l is 48. According to this formula, the model requires 6 Conv3D layers to cover the entire 48 timesteps and effectively predict the current timestep.

$$n = \lceil log_b \frac{(l-1)(b-1)}{k-1} + 1 \rceil \tag{1}$$

Equation 2 shows how to compute the number of zero-padding entries required at each layer. Given a dilation base b, a kernel size k, and a number of i layers below the current layer, the number of zero-padding entries p needed for the current layer is computed as follows:

$$p = b^i(k-1) \tag{2}$$

S.3. Model Dynamics and Interpretation

As described in the *Materials and Methods* section, the model uses a sequence of 48 consecutive timesteps to predict the BCB concentration at the current timestep. As shown in Figure S3, the initial hidden state is initialized to zero at the first timestep. The model then recurrently updates the hidden state across the remaining 47 timesteps, implicitly capturing the temporal dynamics of the input sequence.

A portion of these 48 steps acts as a "spin-up" period, allowing the model to adjust from the initial zero state and stabilize its internal representation. To investigate this behavior, we extracted the hidden state output after each timestep and fed it into the bottleneck layer during the inference phase. These outputs were scaled to match the ground truth concentrations (i.e., $\times 10^{-11} kg/kg_{air}$) and then averaged across the spatial domain (latitude and longitude), resulting in a 48-element vector for each instance. These vectors were then plotted alongside the corresponding ground truth values for comparison.

Figure S5 shows four representative instances sampled at different stages of the test period, spanning low, medium, and high BCB concentration regimes. For low-concentration cases, the model converges rapidly within the first 8–10 timesteps (Figures S5a and S5b). In contrast, for high-concentration instances, convergence occurs later (around timestep 35), indicating that the model initially trends toward lower values and requires additional steps to adjust to higher concentrations (Figures S5c and S5d). It is worth noting that, because the ModelE is coupled with a fire model, BCB concentrations exhibit diurnal oscillations; consequently, depending on the time of day, the BCB trajectory may follow either a rising or falling trend.

Saliency maps are a common interpretability tool used to highlight the regions of the input that contribute most strongly to the model's prediction. In our case, the saliency map reflects the normalized cumulative attention of the ConvLSTM model across all spatial locations during the prediction of BCB concentration at the current timestep.

Figure S6 presents three examples from different times during the test period. In each case, the left panel shows the BCB emission source, the middle panel displays the resulting BCB concentration, and the right panel visualizes the corresponding saliency map. Both the emission source and concentration represent averages over the 48 preceding timesteps, while the saliency map captures the spatial distribution of the model's attention at the final prediction step.

S.3.1. The Saliency Map Analysis

As shown in Figure S6a, when BCB concentrations are elevated over northern Africa and parts of the northern mid-latitudes, the model's attention is primarily focused on these regions. In contrast, during periods of higher global BCB concentrations—such as the instance in Figure S6b—the model shifts its focus toward central Africa and the Amazon basin, consistent with the spatial distribution of strong emission sources and concentrations in those areas. Additionally, enhanced attention is observed over the tropical Atlantic and Pacific Oceans, likely due to the influence of advection in the Intertropical Convergence Zone (ITCZ), which transports emitted particles across these regions.

A similar pattern is observed in Figure S6c, where the model attends to areas with both strong local emissions and regions downstream of the emissions, highlighting the model's capacity to learn complex spatiotemporal transport dynamics.

Figure S7a and S7b compare the saliency maps produced by the Gated Conv3D and SimVP models, respectively, at the same timestep as shown in Figure S6b for ConvLSTM. The Gated Conv3D model exhibits a similar spatial pattern of attention to that of ConvLSTM, with notable focus over central Africa and the Amazon basin—regions of intense BCB emission and high concentration, though the overall strength of attention is somewhat lower.

In contrast, the SimVP model produces a markedly different saliency map, with generally low attention values across the entire domain. This divergence can be attributed to the architectural design of SimVP, which utilizes a specialized attention mechanism that captures spatiotemporal dependencies with minimal inductive bias. As a result, it tends to spread attention more uniformly or sparsely, rather than concentrating it on dominant emission or transport regions. While this may offer advantages in modeling flexibility, it may also limit the interpretability of learned physical transport pathways in the way models like ConvLSTM or Gated Conv3D do.

S.4. Model Parameter Summary and Performance Analysis

This section provides a detailed summary of the model's parameters, memory requirements, and computational performance. To measure the average inference time of a model on a GPU (or CPU), an example input tensor is first initialized, and both the model and input are moved to the specified device (e.g., CUDA for GPU). The model is set to evaluation mode, and a warm-up phase is performed for 10 iterations to mitigate cold start effects. During the timing phase, CUDA synchronization ensures accurate measurement on the GPU. The inference is run 100 times with gradient computation disabled to cal-

culate the average inference time per iteration, providing a stable estimate of the model's performance.

Table S1 presents key metrics, including the total number of parameters, estimated memory usage, and execution times on both GPU and CPU, providing insights into the models' efficiency during training and inference. The input size for ConvLSTM and Gated Conv3D is $1 \times 48 \times 5 \times 100 \times 154$, while for SimVP, it is $1 \times 48 \times 5 \times 160 \times 160$. All three models were trained and tested on an NVIDIA A100 80GB PCIe.

S.5. Comparison of OMA and Non-Interactive (NINT) Schemes

Figure S8 presents a comparison between the Interactive OMA scheme and the Non-Interactive (NINT) scheme over the test period. To reproduce the NINT data, the dataset was first resampled in the time domain to a monthly resolution, where the mean value of each month was computed. The monthly averages were then interpolated back to the original temporal resolution (half-hourly) to obtain a continuous time series representing the NINT scheme throughout the year.

S.6. Analysis of Model Performance at the Grid Level

To examine the model's performance at the grid level over the test period, we selected three grid cells located in regions with high, moderate, and low concentrations. Figure S9 shows the locations of these three grids on the globe, along with their corresponding latitude and longitude coordinates. The ground truth BCB concentrations and the ConvLSTM model predictions for each grid are shown in Figure S10 over the test period.

In regions with high BCB concentrations (on the order of 10–100 $\mu g/m^3$), the model performs very well, achieving an $R^2 = 0.88$. As the concentration decreases (0–1 $\mu g/m^3$),

the model continues to follow the general trend and captures the seasonal variations, though it may not fully reproduce fluctuations at the finer temporal scale. For grids with very low concentrations (0.1–<1 $\mu g/m^3$), the model effectively treats the signal as noise and does not attempt to learn from it, which is an expected and desirable behavior. This outcome aligns with the use of the MSE loss function, which naturally encourages the model to focus on meaningful concentration ranges rather than fitting to noise.

S.7. Experiment Setup for Training SFNO

The training methodology employs a curriculum learning strategy divided into two distinct phases, optimized using the Adam algorithm. To account for spatial distortions inherent in the global grid, a *Latitude Weighted MSE* loss function is utilized. The first 40 epochs constitute a "Teacher Forcing" phase, during which the model is restricted to predicting a single time step ahead ($T = 1$). During this period, the learning rate follows a warmup-cosine-decay schedule: it linearly increases to a peak of 1×10^{-4} over the first 10 epochs (warmup), before annealing via cosine decay to a base of 1×10^{-5} by epoch 40.

At epoch 40, the experiment transitions to an autoregressive curriculum phase. In this stage, the learning rate is held constant at 1×10^{-5} while the prediction horizon expands linearly, increasing by one step per epoch. This forces the model to use its own previous predictions as inputs for subsequent time steps (autoregression), explicitly penalizing the accumulation of errors over longer trajectories. The discontinuity observed in the loss curve at epoch 40 reflects this abrupt increase in task difficulty, as the objective shifts from minimizing immediate single-step error to ensuring multi-step temporal stability.

References

Lin, Z., Li, M., Zheng, Z., Cheng, Y., & Yuan, C. (2020). Self-attention convlstm for spatiotemporal prediction. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 11531–11538).

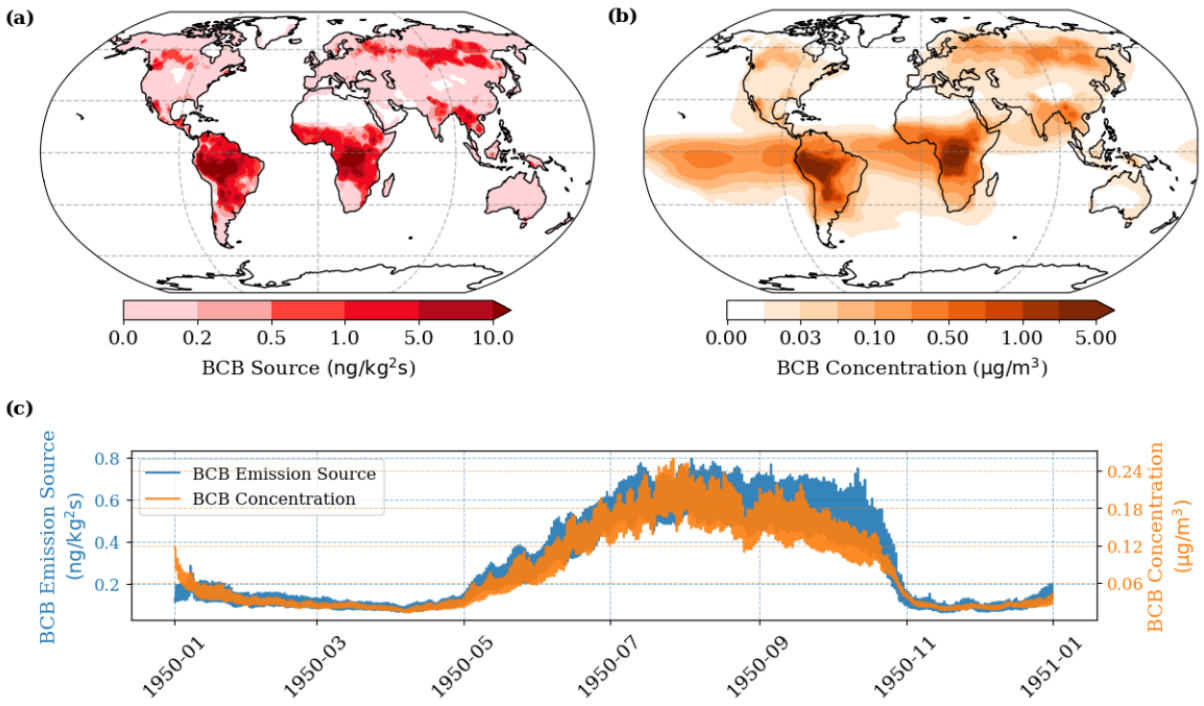


Figure S1. (a) The annual average of BCB emission source over the test period (1950). (b) The annual average of BCB concentrations over the test period (1950). (c) The global average of BCB emission source (left y-axis) and BCB concentrations (right y-axis) over the entire year of 1950.

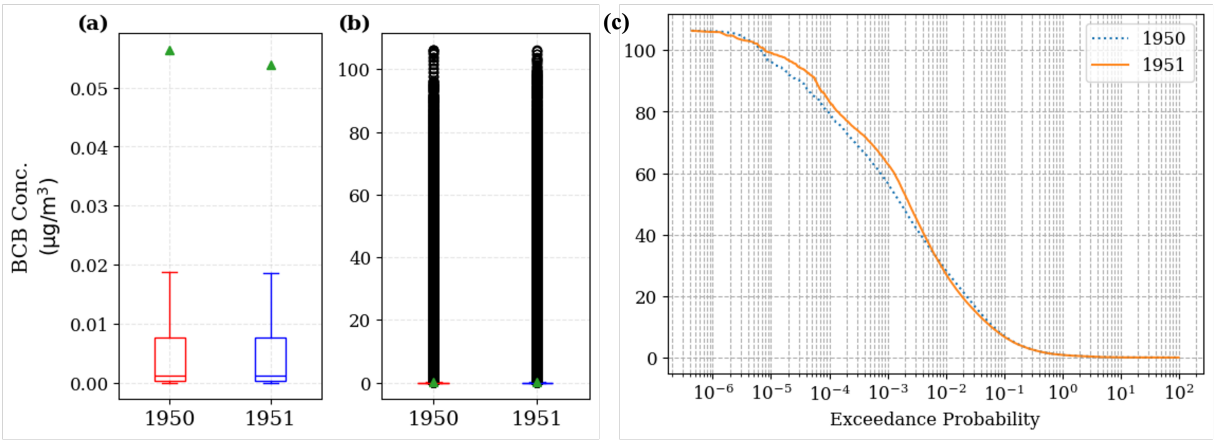


Figure S2. Comparison of data statistics for 1950 and 1951. (a) Boxplot without outliers. (b) Boxplot with outliers. (c) Exceedance probability plot comparing the return period of extreme BCB concentration events in 1950 and 1951.

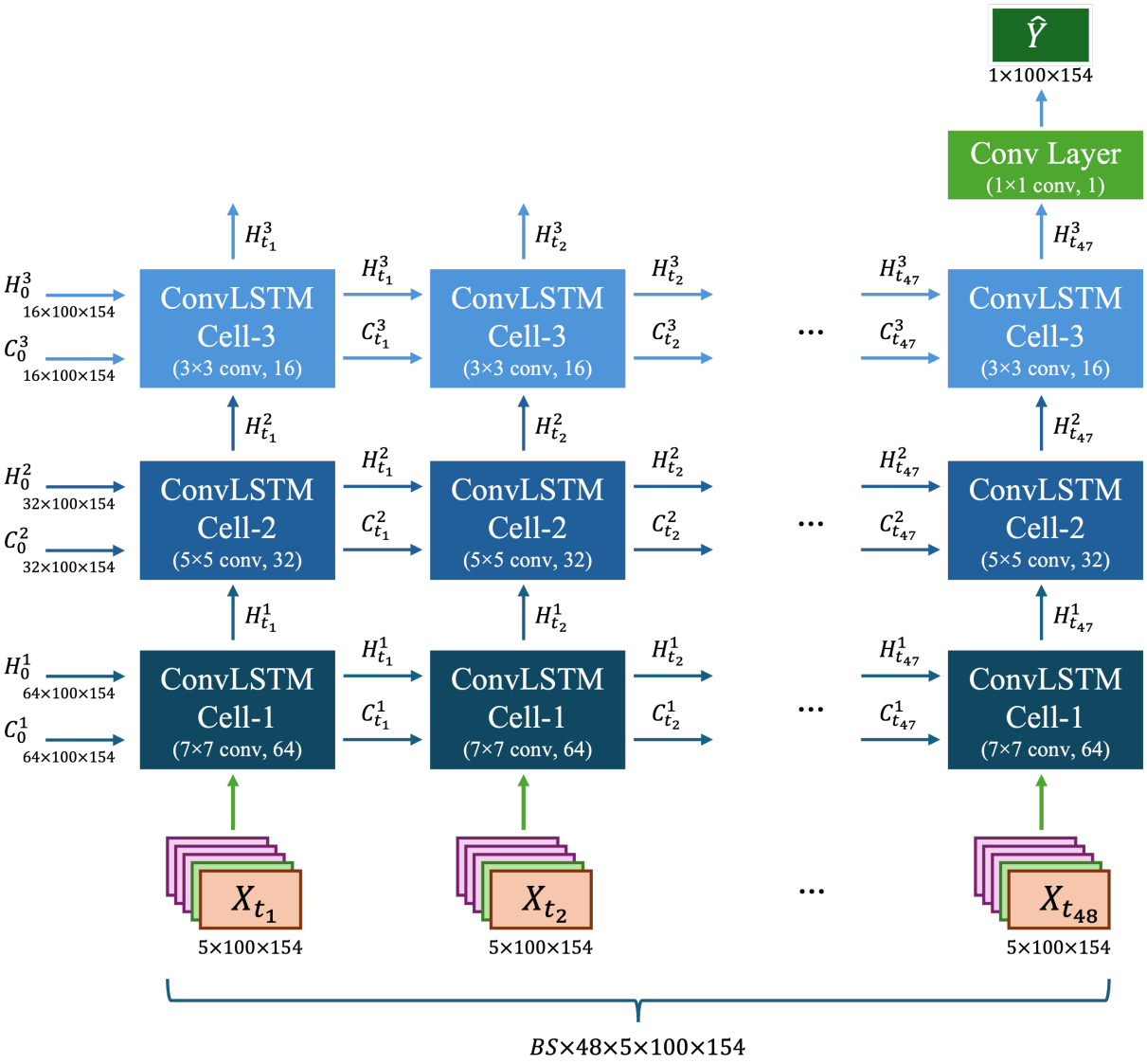


Figure S3. ConvLSTM model architecture. To capture temporal dependency in each timestep, the ConvLSTM received as input the concentration value at a given timestep, along with the ModelE diagnostics from the 48 preceding timesteps.

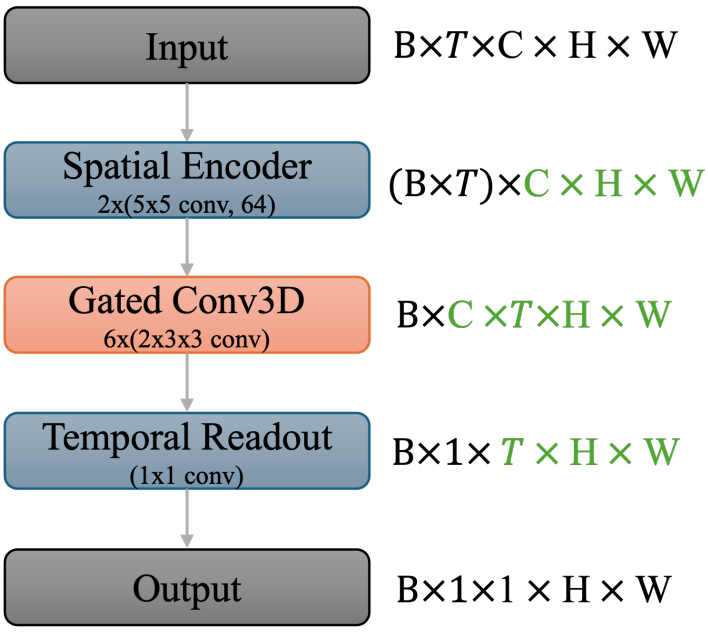


Figure S4. Gated Conv3D model architecture.

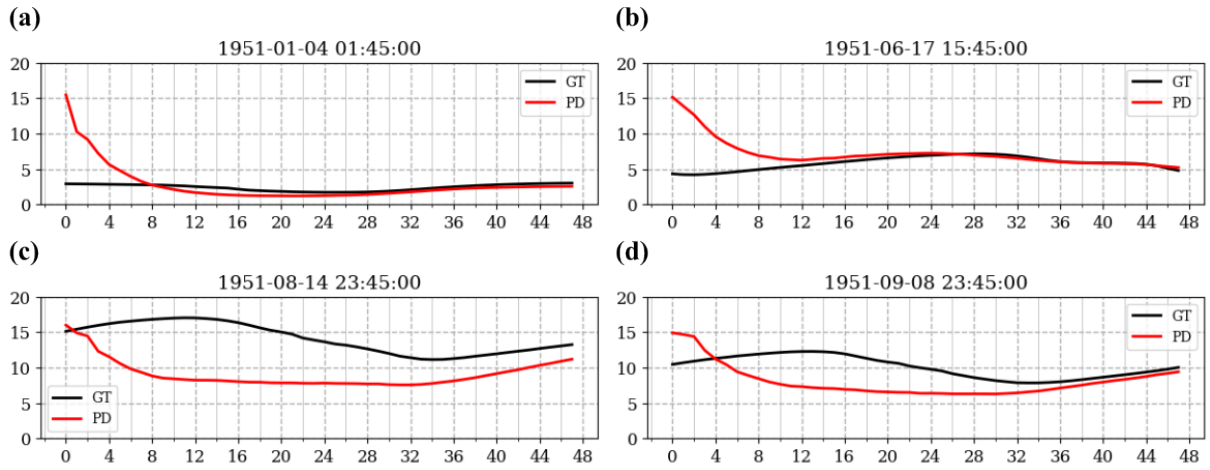


Figure S5. Evolution of the hidden state over 48 consecutive time steps. The initial phase represents a spin-up period during which the hidden state converges toward the ground-truth (GT) data. The duration of this period varies across trajectories, leading to different BCB behaviors: instance (a) converges in fewer than 10 time steps, (b) in about 15 time steps, while (c) and (d), which exhibit rising trajectories, require approximately 30 time steps to align with the GT direction.

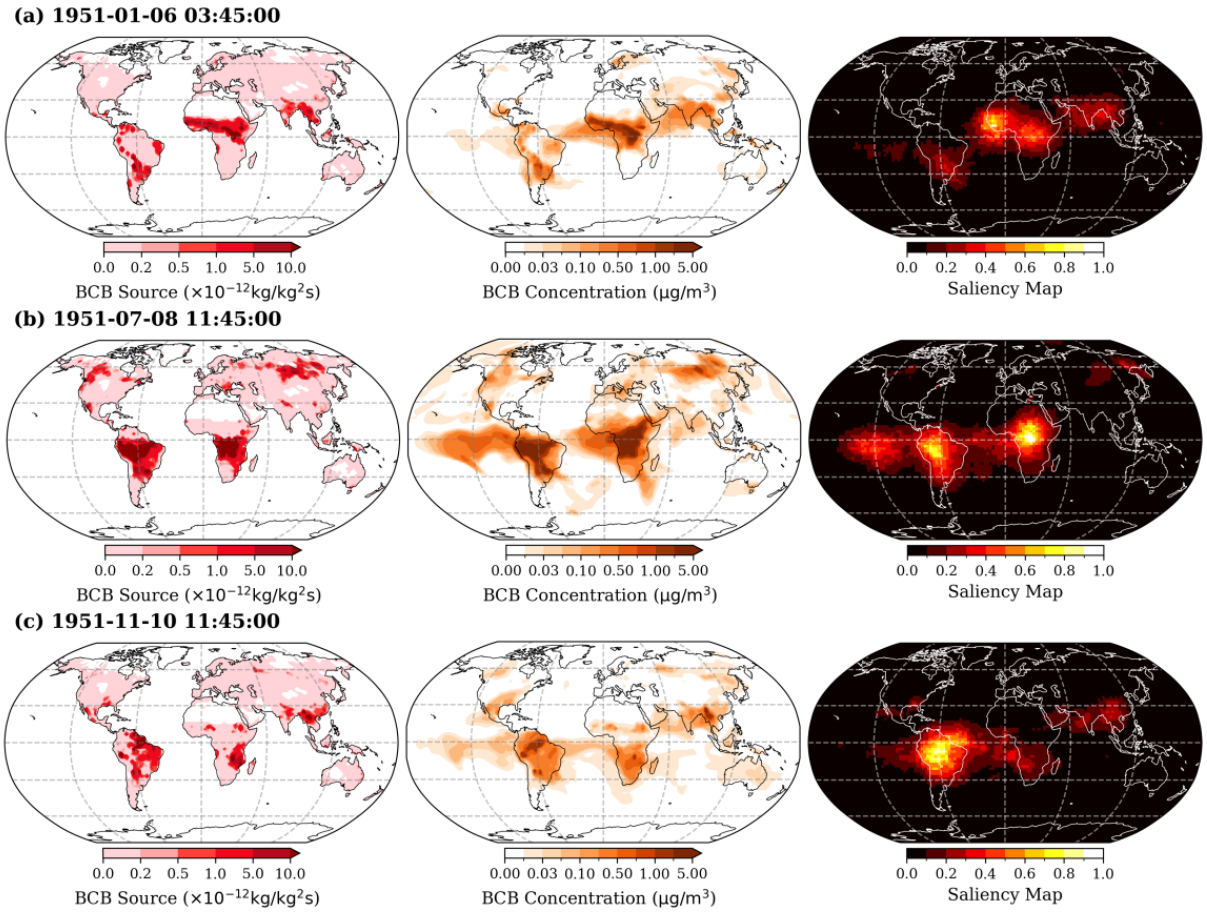


Figure S6. Visualization of BCB dynamics and model interpretability. The columns display the BCB Source emissions (left), the resulting BCB Concentration (middle), and the Saliency Maps generated by the ConvLSTM model (right), which highlight the spatial regions most influential to the prediction. The panels correspond to three distinct snapshots from the test period: (a) January 6, 1951 (03:45:00), (b) July 8, 1951 (11:45:00), and (c) November 10, 1951 (11:45:00).

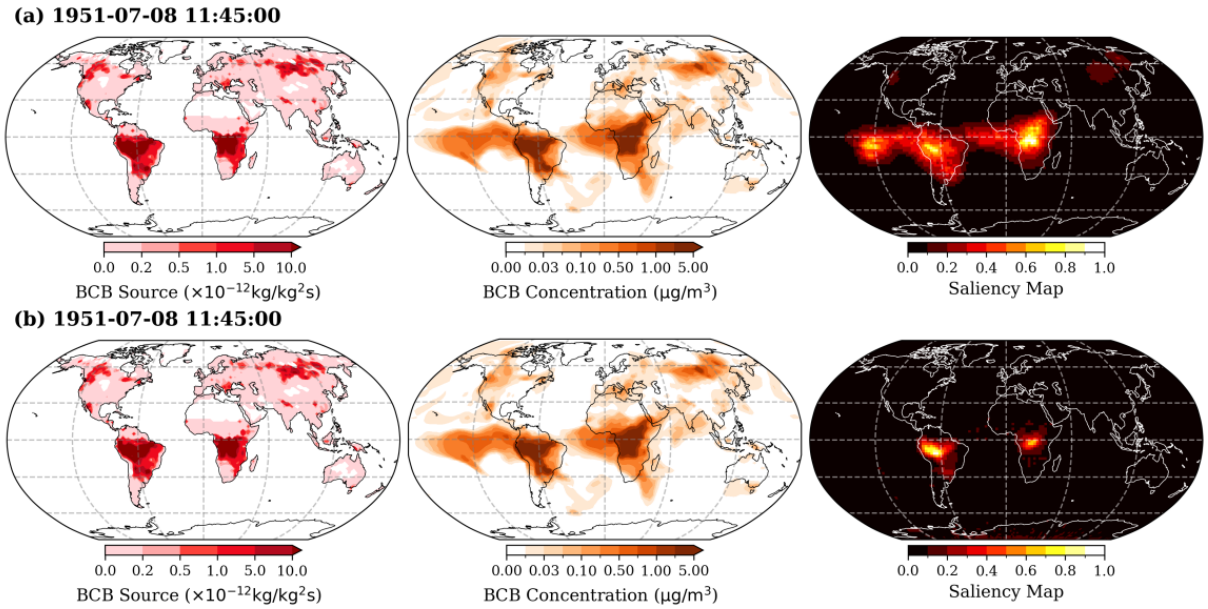


Figure S7. Comparison of interpretability across different model architectures at the same timestep (July 8, 1951, 11:45:00). Columns display the BCB Source (left), BCB Concentration (middle), and Saliency Map (right). Row (a) shows the results for the Gated Conv3D model, and row (b) shows the SimVP model. Both architectures exhibit spatial attention patterns similar to the ConvLSTM baseline shown in Figure S6b.

Table S1. Model specifications and inference time

Model	**ConvLSTM**	**Conv3D**	**SimVP**
Trainable Params	6,375,249	561,787	57,788,882
Total Mult-Adds (GB)	887.66	263.38	396.77
Input Size (MB)	14.78	14.78	24.58
Forward/Backward Pass Size (MB)	2649.42	2992.40	6406.35
Params Size (MB)	4.80	1.43	231.13
Estimated Total Size (MB)	2669.00	3008.62	6662.05
Average Inference Time (CPU)	0.573011 s	0.699617 s	1.420980 s
Average Inference Time (GPU)	0.057585 s	0.022775 s	0.034831 s

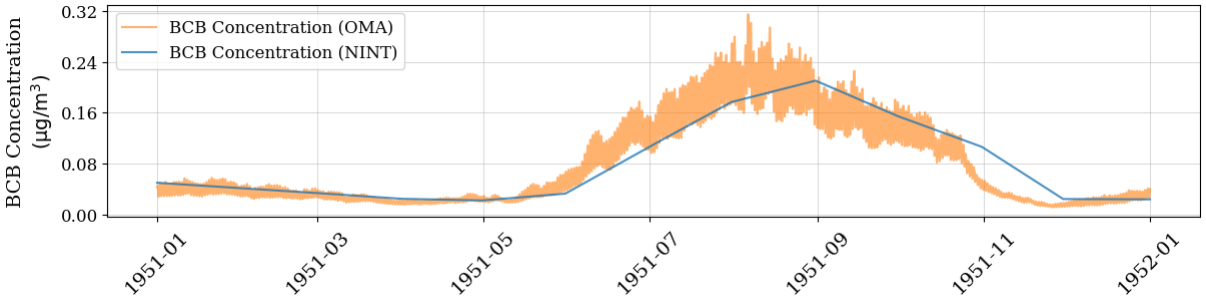


Figure S8. Comparison of global average values of OMA Interactive vs. NINT scheme over the test period.

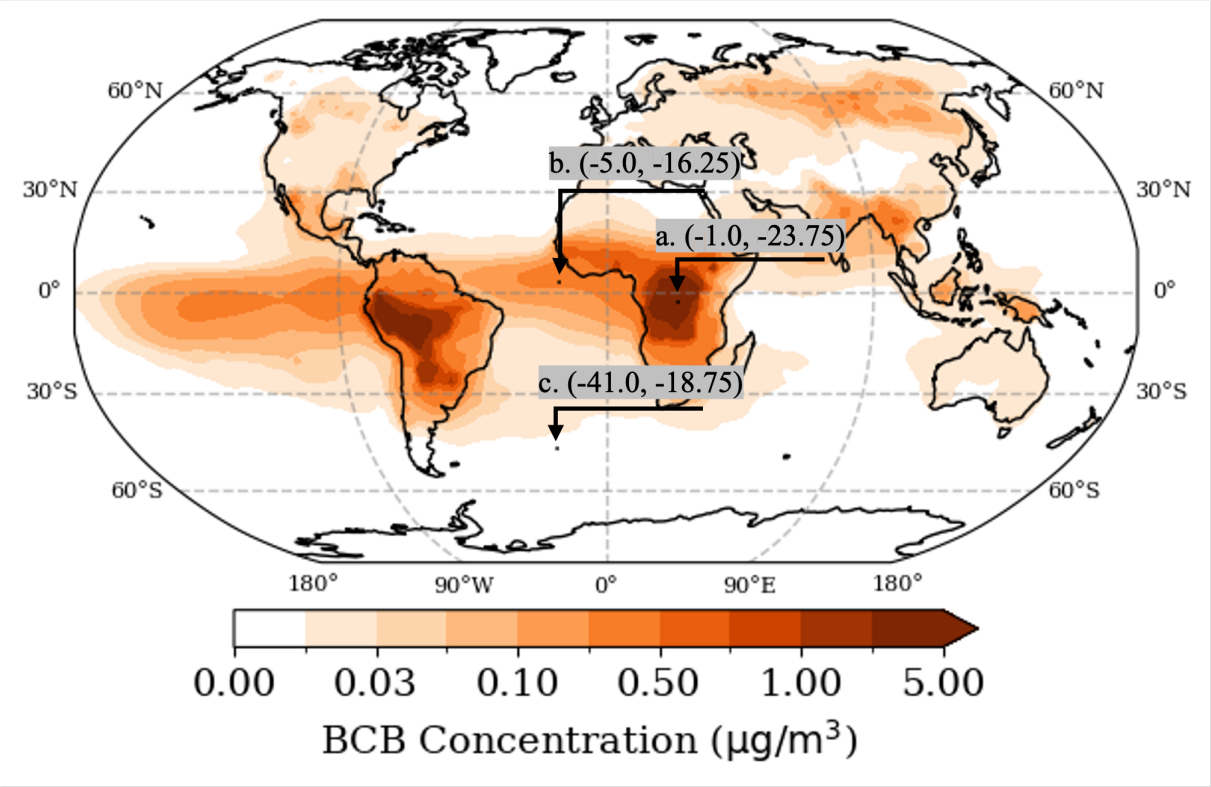


Figure S9. Geographical locations of the three selected grid cells representing regions with high, moderate, and low BCB concentrations. The corresponding (latitude and longitude) coordinates for each grid are also indicated.

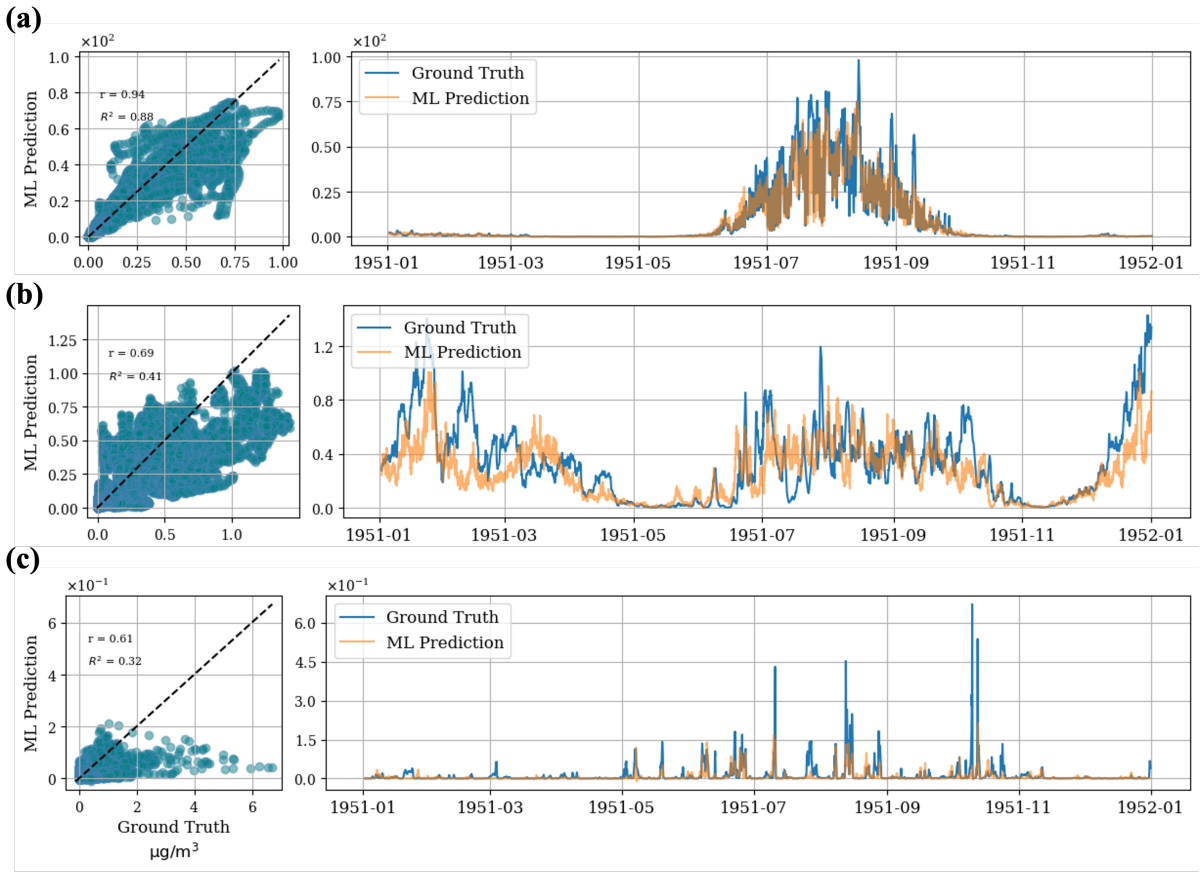


Figure S10. Time series of ground truth BCB concentrations and ConvLSTM model predictions over the test period for the three selected grid cells. (a), (b), and (c) correspond to a different concentration regime (high, moderate, and low, respectively), illustrating the model's ability to capture temporal variability across varying emission intensities.

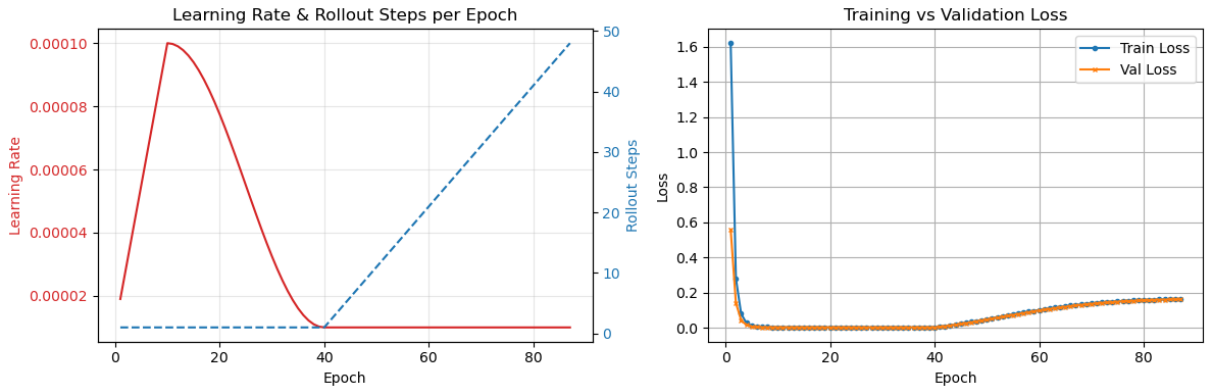


Figure S11. Training Dynamics and Curriculum Schedule. *Left:* The evolution of the learning rate (red solid line) and the autoregressive rollout horizon (blue dashed line). The vertical alignment at epoch 40 marks the transition from the Teacher Forcing phase (single-step prediction with variable LR) to the Curriculum phase (increasing multi-step rollout with fixed LR). *Right:* Training and validation loss curves. The step-change increase in loss at epoch 40 corresponds to the onset of the autoregressive rollout, where the model must contend with compounding errors over increasingly long horizons.