

We sincerely thank the Topic Editor for the careful assessment of our revised manuscript and for identifying the remaining issues that should be addressed before publication. We greatly appreciate the time and attention devoted to our manuscript. We have carefully considered each comment and revised the manuscript accordingly. In particular, we rechecked the numerical consistency of Figure 6, Figure 7, and Table 5, corrected the identified textual and numerical errors, defined the acronyms at their first occurrence, and carefully proofread the entire manuscript for additional grammatical, typographical, and formatting issues. Our point-by-point responses and the corresponding revisions in the manuscript are provided below.

1. Line 31: Be more specific about the data size number ("input data size of 10^8 values") and reword the surrounding phrase ("using the compiler optimization option ... baseline compiled option")

Response: Thank you for this helpful suggestion. We agree that the original description did not clearly specify the composition of the input data or the compilation options used in the comparison. We have clarified that the input consisted of a one-dimensional array containing 10^8 double-precision floating-point values and have explicitly identified the compilation options used for the HIP and Fortran implementations. The sentence at **lines 28-32** has been revised as follows:
Offline benchmark results demonstrated that, for a one-dimensional input array containing 10^8 double-precision floating-point values, GPU-HADVPPM4HIP V1.0 compiled with the -O3 optimization option achieved a maximum speedup of 556.5x on a GPU-like accelerator relative to the Fortran HADVPPM implementation compiled with the -O0 option as the baseline.

2. Line 88: "is Top1" -> "was the fastest"

Response: Thank you for this correction. We have revised the sentence at **lines 96-98** as suggested:

Remarkably, the computational efficiency of a single MI300A GPU exceeds 1.8 times the peak performance of the Earth-Simulator supercomputer (CPU-based architecture) in Japan, which was the fastest supercomputer in 2003.

3. Line 138: Delete ". "

Response: Thank you for identifying this typographical error. We have deleted the unnecessary period at the beginning of the paragraph at **line 160**. The revised paragraph now begins as follows:
The EPICC-Model V1.6.0 adopts a modular architecture developed using the Fortran programming language, a high-performance computing language specifically designed for scientific applications.

4. Line 330: Delete "which"

Response: Thank you for identifying this grammatical error. We have deleted “which” and revised the sentence at **lines 367-369** as follows:
To leverage this capability, we implemented the coordinated thread-block indexing method proposed by Cao et al. (2023), in which each grid cell in the two-dimensional horizontal plane is assigned to a dedicated thread.

5. Line 331: "whereby in which" -> either "whereby" or "in which"

Response: Thank you for this correction. We have retained “in which” and revised the sentence at **lines 367-369** to improve its grammatical clarity:
To leverage this capability, we implemented the coordinated thread-block indexing method proposed by Cao et al. (2023), in which each grid cell in the two-dimensional horizontal plane is assigned to a dedicated thread.

6. Line 334: "of across" -> either "of" or "across"

Response: Thank you for identifying this grammatical error. We have retained “across” and revised the sentence at **lines 373-375** as follows:
This hierarchical parallelization strategy enables parallel processing across the entire two-dimensional horizontal grid through coordinated thread-block resource allocation.

7. Line 345: "Top 100"-> "Top500"

Response: Thank you for this correction. We have replaced “Top 100” with “TOP500” at **line 389**.
The recent advancements in GPU technology have solidified the dominance of "CPU+GPU"

heterogeneous architectures in global high-performance computing, with 9 out of the top 10 supercomputers in 2024 utilizing heterogeneous configurations (TOP500, 2024).

Reference:

TOP500: Supercomputing Top500 list, TOP500 international organization, <https://www.top500.org/lists/top500/2024/11/> (last access: 26 May 2025), 2024.

8. Line 425: "(Intel Software, 2018)" inconsistent with bibliography

Response: Thank you for identifying this inconsistency. We verified the bibliographic information for the cited Quick Reference Guide to Optimization with Intel C++ and Fortran Compilers v19. The cited guide was published in 2018. We have therefore retained the in-text citation “Intel Software, 2018” at the original line 425 and corrected the publication year in the reference list in **lines 1064-1066** from 2020 to 2018.

Reference:

Intel Software: Quick Reference Guide to Optimization with Intel C++ and Fortran Compilers v19, Intel, <https://www.intel.cn/content/www/cn/zh/developer/articles/technical/intel-fortran-and-c-compiler-documentation.html> (last access: 13 September 2026), 2018.

9. Line 457: "Comparison of..."

Response: Thank you for this correction. We have replaced “Comparative of” with “Comparison of” in the caption of Table 4 at **lines 501-502**. The revised caption reads:

Table 4. Comparison of mean AE and RE across compilation configurations for the F-to-C, C-to-HIP, and F-to-HIP processes.

10. Figs. 6, 7: Please clarify why the results changed since the last version

Response: Thank you for requesting this clarification. We sincerely apologize that the reasons for the changes in Figs. 6 and 7 were not explained in our previous response. In the original manuscript, Figs. 6 and 7 showed the concentration fields at the end of a 24 h integration. When revising the manuscript in response to Referee 2, we replaced the original NH₃ comparison with the requested O₃ comparison. To keep the numerical comparison consistent with the 28 h integration used in the performance evaluation presented in Fig. 12, we regenerated Figs. 6 and 7

using the concentration fields at the end of the 28 h integration. However, we overlooked updating the corresponding experimental description in Sect. 4.2 and the captions of Figs. 6 and 7, which still referred to a 24 h integration.

Because the simulated concentrations evolved between the 24th and 28th hours, changing the output time resulted in corresponding changes in the spatial distributions of the gas and aerosol species. We also adjusted the color-bar ranges of the concentration fields in the left and middle columns to reflect the ranges of the 28 h results. This adjustment further changed the visual appearance of the figures but did not alter the underlying numerical data.

As part of our continuing research on the heterogeneous acceleration of air quality models, we subsequently identified and corrected a minor implementation error in the HIP-Opt2 advection routine. Specifically, the updated routine now accounts for the **map-scale factors** associated with the Lambert projection in the advection computation. This correction further reduced the already small numerical differences between the HIP-Opt2 and Fortran results. We repeated the performance tests using the corrected implementation and confirmed that the computational times and speedups remained unchanged at the reported precision. Therefore, the correction did not affect the reported performance results or the conclusions of the study. The corrected code has been uploaded as a new version of the Zenodo record (Cao et al., 2026), ensuring that the publicly archived code corresponds exactly to the implementation used to generate the results in the revised manuscript.

Accordingly, we have revised the experimental description and the corresponding discussion in Sect. 4.2, corrected the captions of Figs. 6 and 7, and updated the Code availability section in **lines 375-377, lines 503-511, lines 521-522, lines 572-576, lines 584-588, and lines 928-932**, as follows:

Lines 375-377 :

*To ensure consistency with the original Fortran implementation on the projected grid, the HIP-Opt2 advection routine incorporates **the map-scale factors associated with the Lambert projection** when calculating horizontal transport.*

Lines 503-511:

By coupling GPU-HADVPPM4HIP V1.0 with the EPICC-Model and implementing optimizations such as reducing CPU-GPU communication frequency, adopting coordinated

thread-block indexing, and employing MPI+HIP hybrid parallelization, we developed the HIP-Opt2 version. Using identical input data and model configurations, the differences between the simulation results produced by HIP-Opt2 and the original Fortran version were evaluated in a real-world case study. The model input data and configurations were set as described in Sect. 4.1. Both versions were integrated for 28 h beginning at 00:00 UTC on 1 July 2021, consistent with the simulation duration used in the performance evaluation in Sect. 4.3.2. The resulting output fields were used for the numerical comparison presented in this section.

Lines 521-522:

Figures 6 and 7 compare the gas and aerosol concentrations calculated by the original Fortran and HIP-Opt2 versions at the end of the 28 h integration.

Lines 572-576:

Figure 6. Spatial distributions of HONO, SO₂, NO₂, and O₃ concentrations calculated by the original Fortran and HIP-Opt2 versions at the end of the 28 h integration. Panels (a), (d), (g), and (j) show the Fortran results; panels (b), (e), (h), and (k) show the HIP-Opt2 results; and panels (c), (f), (i), and (l) show the differences between the two versions. The units of the concentrations and differences are ppbV.

Lines 584-588:

Figure 7. Spatial distributions of BC, PM_{2.5}, ASO₄, and ANH₄ concentrations calculated by the original Fortran and HIP-Opt2 versions at the end of the 28 h integration. Panels (a), (d), (g), and (j) show the Fortran results; panels (b), (e), (h), and (k) show the HIP-Opt2 results; and panels (c), (f), (i), and (l) show the differences between the two versions. The units of the concentrations and differences are $\mu\text{g} \cdot \text{m}^{-3}$.

Lines 928-932:

The source codes of EPICC-Model V1.6.0 are available online via ZENODO (<https://zenodo.org/records/20303367>, EPICC-Model Working Group, 2026). The GPU acceleration code of HADVPPM scheme in the EPICC-Model, datasets, and the offline test code related to this paper are available online via ZENODO (<https://zenodo.org/records/22743837>, Cao et al., 2026).

Reference:

Cao, K., Wu, Q., and Tang, X.: The dataset of the manuscript "Enhancing the advection module

performance in the EPICC-Model V1.6.0 via GPU-HADVPPM4HIP V1.0 coupling and GPU-optimized strategies", Zenodo [data set], <https://doi.org/10.5281/zenodo.22743837>, 2026.

11. Figs. 6, 7: The right column indicates that deltas are small, but in the present form provide no further information. Please adjust the scale to the data range to actually show the deltas and their regional distribution.

Response: Thank you for this helpful suggestion. We agree that the color-bar ranges used in the previous revision were too broad to reveal the regional distribution of the small differences between the HIP-Opt2 and Fortran results. We have therefore redrawn the right-hand panels of Figs. 6 and 7 using specific color bar ranges appropriate for their respective magnitudes.

Specifically, for the gas species, the HONO range was reduced from $\pm 1.0 \times 10^{-3}$ to $\pm 8.0 \times 10^{-6}$ ppbV, while the ranges for SO₂ and NO₂ were reduced from $\pm 1.0 \times 10^{-2}$ to $\pm 1.0 \times 10^{-3}$ ppbV. The O₃ range was retained at $\pm 1.0 \times 10^{-2}$ ppbV. For the aerosol species, the BC range was reduced from $\pm 1.0 \times 10^{-2}$ to $\pm 8.0 \times 10^{-5}$ $\mu\text{g} \cdot \text{m}^{-3}$, while the ranges for PM_{2.5}, ASO₄, and ANH₄ were reduced from $\pm 1.0 \times 10^{-2}$ to $\pm 1.0 \times 10^{-3}$ $\mu\text{g} \cdot \text{m}^{-3}$.

With these revised scales, the localized positive and negative differences are now clearly visible. The differences are spatially sparse and primarily confined to a limited number of grid cells, without a large-scale systematic positive or negative pattern. These revisions affect only the graphical presentation and do not alter the underlying simulation results or the statistical metrics reported in Table 5. We have revised this part at **lines 521-540**, which are as follows:

Figures 6 and 7 compare the gas and aerosol concentrations calculated by the original Fortran and HIP-Opt2 versions at the end of the 28 h integration. Visually, the two versions produce nearly identical spatial distributions and concentration magnitudes. To reveal the regional distributions of the small numerical differences, species-specific color-bar ranges appropriate for the corresponding difference magnitudes were used in the right-hand panels. The displayed differences are of the order of 1.0×10^{-6} ppbV for HONO, 1.0×10^{-3} ppbV for SO₂ and NO₂, and 1.0×10^{-2} ppbV for O₃. For the aerosol species, the displayed differences are of the order of 1.0×10^{-5} $\mu\text{g} \cdot \text{m}^{-3}$ for black carbon (BC) and 1.0×10^{-3} $\mu\text{g} \cdot \text{m}^{-3}$ for PM_{2.5}, ASO₄, and ANH₄. Cao et al. (2024) reported that the differences between the HIP and Fortran

versions of CAMx were within $\pm 4.0 \times 10^{-2}$ ppbV for O_3 and NO_2 and within $\pm 2.0 \times 10^{-2} \mu g \cdot m^{-3}$ for ASO_4 . By comparison, in the present study, the differences in O_3 and NO_2 are within $\pm 1.0 \times 10^{-2}$ ppbV and $\pm 1.0 \times 10^{-3}$ ppbV, respectively, while the difference in ASO_4 is within $\pm 1.0 \times 10^{-3} \mu g \cdot m^{-3}$, all of which are smaller than the corresponding differences reported by Cao et al. (2024). For each species, the grid-cell differences were further averaged. The resulting mean differences are -5.1×10^{-9} , 3.6×10^{-7} , 1.7×10^{-6} , and 3.2×10^{-4} ppbV for HONO, SO_2 , NO_2 , and O_3 , respectively, and 1.1×10^{-6} , 1.8×10^{-5} , 4.0×10^{-6} , and $1.0 \times 10^{-6} \mu g \cdot m^{-3}$ for BC, $PM_{2.5}$, ASO_4 , and ANH_4 , respectively. Taken together, the smaller difference ranges and near zero domain mean differences confirm the close numerical agreement between the Fortran and HIP-Opt2 versions.

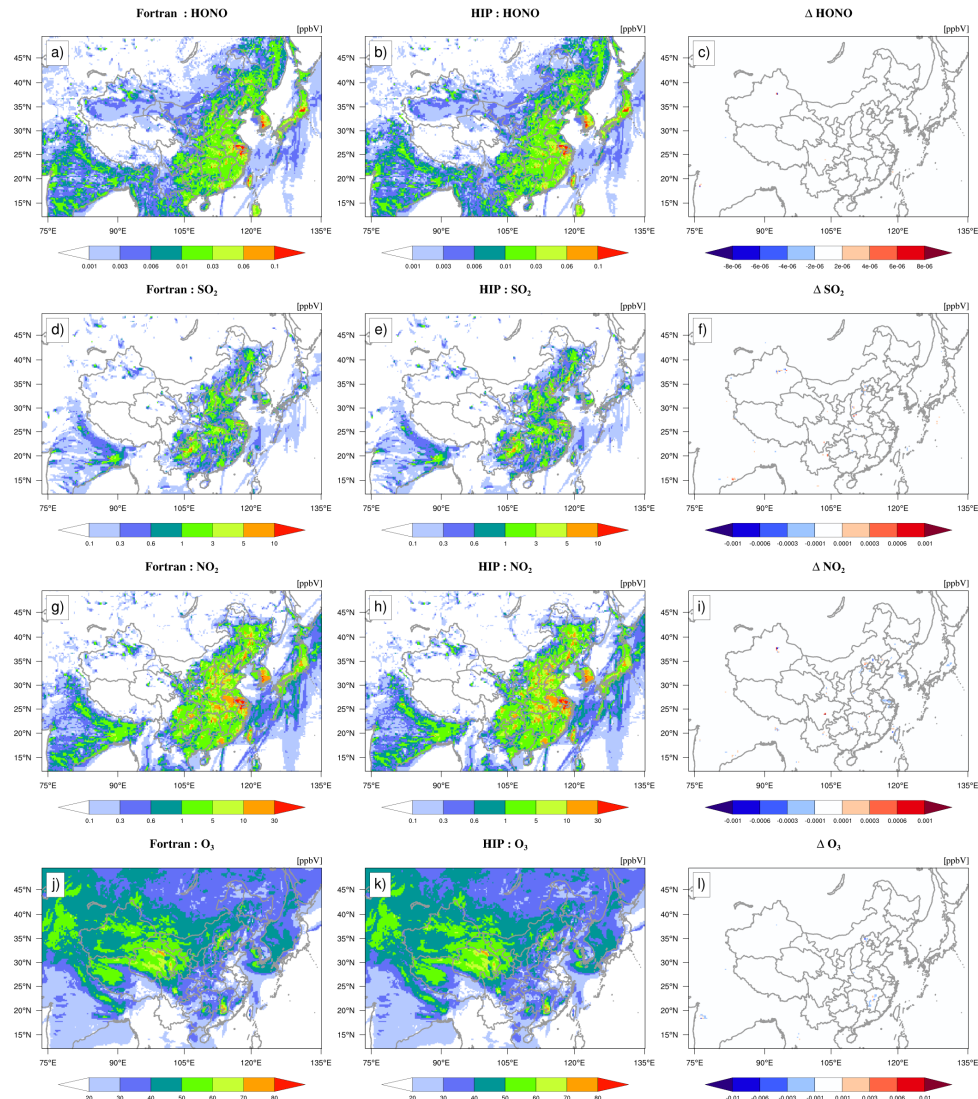


Figure 6. Spatial distributions of HONO, SO_2 , NO_2 , and O_3 concentrations calculated by the original Fortran and HIP-Opt2 versions at the end of the 28 h integration. Panels (a), (d), (g),

and (j) show the Fortran results; panels (b), (e), (h), and (k) show the HIP-Opt2 results; and panels (c), (f), (i), and (l) show the differences between the two versions. The units of the concentrations and differences are ppbV.

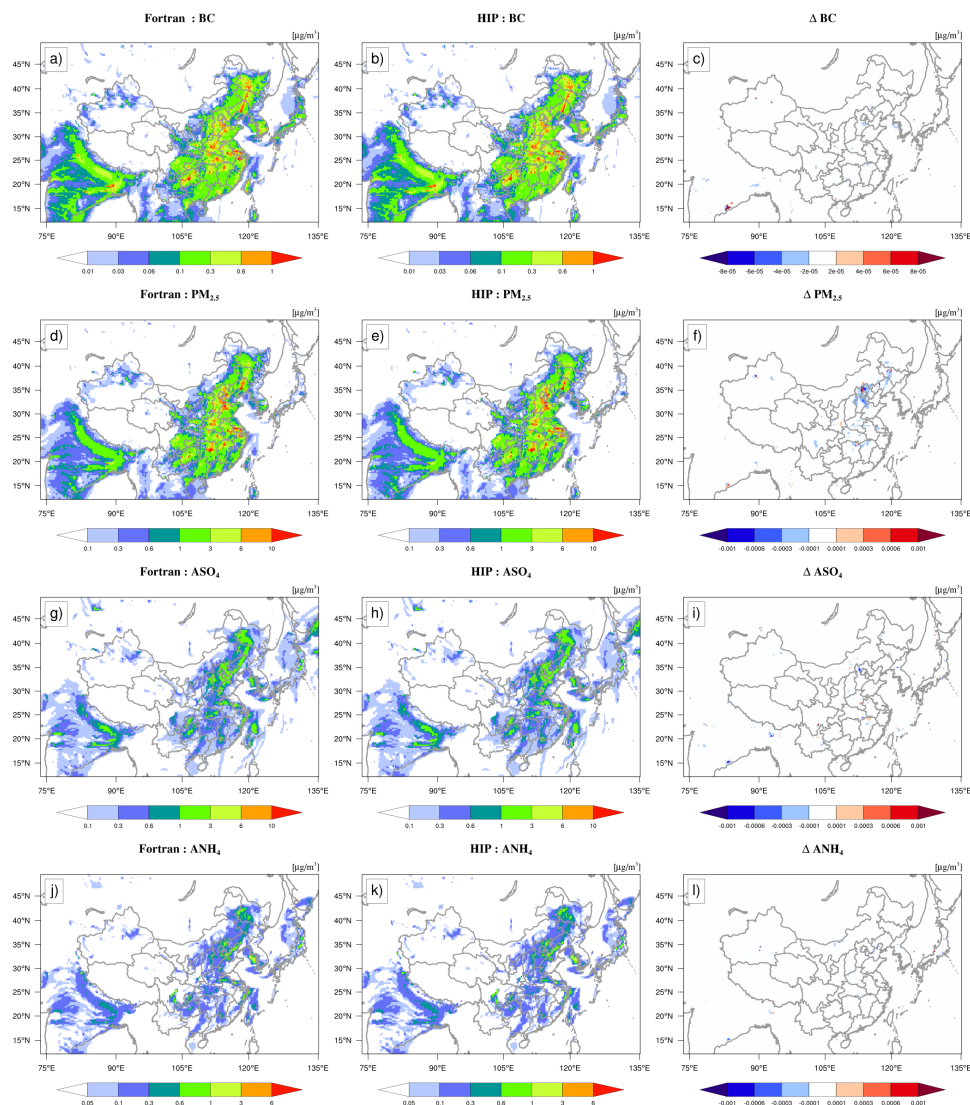


Figure 7. Spatial distributions of BC, PM_{2.5}, ASO₄, and ANH₄ concentrations calculated by the original Fortran and HIP-Opt2 versions at the end of the 28 h integration. Panels (a), (d), (g), and (j) show the Fortran results; panels (b), (e), (h), and (k) show the HIP-Opt2 results; and panels (c), (f), (i), and (l) show the differences between the two versions. The units of the concentrations and differences are $\mu\text{g} \cdot \text{m}^{-3}$.

12. Table 5: Please clarify why the results changed since the last version

Response: Thank you for pointing out this unexplained change. We sincerely apologize that the reason for updating the values in Table 5 was not stated in our previous response. The statistics in

the original manuscript were calculated from the concentration fields at the end of the 24 h integration. During the previous revision, when replacing the original NH₃ comparison with the O₃ comparison requested by Referee 2 and adding NMB, we recalculated all the statistics using the fields at the end of the 28 h integration, consistent with the updated Figs. 6 and 7 and the simulation duration used in Fig. 12. Because the concentration distributions evolved between the 24th and 28th hours, changing the output time affected the standard deviations and the associated comparison metrics.

As part of our continuing research on the heterogeneous acceleration of air quality models, we subsequently identified and corrected a minor implementation error in the HIP-Opt2 advection routine. Specifically, the updated routine now accounts for the **map-scale factor** associated with the Lambert projection in the advection computation. The correction further reduced the numerical differences between the HIP-Opt2 and Fortran results and consequently affected the resulting RMSE, RMSE/std, and NMB values. The adjustment of the color-bar ranges affected only the graphical presentation of Figs. 6 and 7 and had no effect on the statistics in Table 5.

All values in the revised Table 5 have now been recalculated from the same pair of 28 h Fortran and HIP-Opt2 simulations used to generate Figs. 6 and 7. We also repeated the performance tests after correcting the HIP-Opt2 implementation and confirmed that the reported computational performance and the conclusions of the study remain unchanged. The corrected code has been uploaded as a new version of the Zenodo record (Cao et al., 2026), ensuring that the archived code corresponds exactly to the implementation used to generate the results in the revised manuscript. Accordingly, we have clarified the simulation outputs used to calculate Table 5 in Sect. 4.2 and updated the Code availability section in **lines 375-377**, **lines 553-557** and **lines 928-932**, respectively, as follows:

Lines 375-377 :

*To ensure consistency with the original Fortran implementation on the projected grid, the HIP-Opt2 advection routine incorporates **the map-scale factors associated with the Lambert projection** when calculating horizontal transport*

Lines 553-557:

All the statistical metrics presented in Table 5 were calculated from the concentration fields at the end of the 28 h Fortran and HIP-Opt2 simulations used to generate Figures 6 and 7. To

further evaluate the suitability of HIP-Opt2 for scientific applications, we followed Wang et al. (2021) and Cao et al. (2024) and calculated the root mean square error (RMSE), standard deviation (std), RMSE/std ratio, and normalized mean bias (NMB).

Lines 928-932:

The source codes of EPICC-Model V1.6.0 are available online via ZENODO (<https://zenodo.org/records/20303367>, EPICC-Model Working Group, 2026). The GPU acceleration code of HADVPPM scheme in the EPICC-Model, datasets, and the offline test code related to this paper are available online via ZENODO (<https://zenodo.org/records/22743837>, Cao et al., 2026).

Reference:

Cao, K., Wu, Q., and Tang, X.: The dataset of the manuscript "Enhancing the advection module performance in the EPICC-Model V1.6.0 via GPU-HADVPPM4HIP V1.0 coupling and GPU-optimized strategies", Zenodo [data set], <https://doi.org/10.5281/zenodo.22743837>, 2026.

13. Line 551: "MAD8" -> "MDA8"

Response: Thank you for identifying this typographical error. We have corrected “MAD8” to “MDA8” in the caption of Fig. 9 at **lines 641-643** and checked the manuscript to ensure that this abbreviation is used consistently throughout.

Figure 9. Time series of observed and simulated MDA8 O₃ concentrations in major cities of China from 8 to 31 July 2021. Panels (a)-(h) represent Beijing, Taiyuan, Hangzhou, Hefei, Fuzhou, Qingdao, Guangzhou, and Kunming, respectively.

14. Line 609: "was analyzed" -> "were analyzed"

Response: Thank you for this correction. Because the subject contains both time consumption and proportional contributions, we have replaced “was analyzed” with “were analyzed” at **lines 700-702**. The revised sentence reads:

To quantify the overheads of these processes, the time consumption and proportional contributions were analyzed under the -O0 compilation option for different data sizes.

15. Line 684: The time values appear to be missing a factor of 10

Response: Thank you for carefully identifying this numerical error. We rechecked the total elapsed times against the recorded average hourly elapsed times and confirmed that the two total elapsed-time values were indeed missing a factor of 10. We have corrected the values from 2,287.4 to 22,874.0 s for the Fortran version and from 2,152.0 to 21,520.0 s for the HIP-Opt2 version at **lines 779-781**. These corrections are consistent with the corresponding average hourly elapsed times of 817.0 and 768.6 s, respectively. The correction does not affect the reported performance comparison or the conclusions of the study.

The total elapsed time for the 28 h simulation decreased from 22,874.0 s for the Fortran version to 21,520.0 s for the HIP-Opt2 version, while the average hourly elapsed time decreased from 817.0 to 768.6 s.

16. Lines 834, 835: "TX" -> "XT"

Response: Thank you for identifying this inconsistency in the author initials. We have corrected both occurrences of “TX” to “XT” at **lines 942-943**, which are as follows:

KC, QW, XT, XC, WW, JieL, JZ, and ZW carried out the formal analysis of the model results. KC, QW, XT, JinL, XC, HC, WW, LK, and ZW took part in the discussion.

17. Line 835: "JinxL" -> "JinL" (consistent with line 832)

Response: Thank you for identifying this inconsistency. We have corrected “JinxL” to “JinL” at **line 943** and checked all author initials for consistency throughout the manuscript, which are as follows:

KC, QW, XT, JinL, XC, HC, WW, LK, and ZW took part in the discussion.

18. Acronyms: several are used before or without definition, e.g. CUDA, CMAQ, CAMx, CBM-Z, HTAP, TUV, YSU, MDA8, FNL/GFS/WPS in Fig. 1. Please define all acronyms at first use. In particular, "BC" denotes black carbon in Sect. 4.2 and the tables, but "boundary conditions" at l. 152 - please disambiguate.

Response: Thank you for this careful and important comment. We agree that all acronyms should be defined at their first occurrence and that the use of “BC” for both black carbon and boundary

conditions could cause confusion. We have systematically checked the Abstract, main text, figures, tables, and captions and defined all acronyms at their first occurrence. In particular, CUDA, CMAQ, CAMx, CBM-Z, HTAP, TUV, YSU, and MDA8 are now defined in the main text, while FNL, GFS, and WPS are defined in the caption of Fig. 1.

To distinguish boundary conditions from black carbon, we now use the plural abbreviation “BCs” exclusively for boundary conditions and retain the singular abbreviation “BC” exclusively for black carbon. Accordingly, “IC/BC” in Fig. 1 has been replaced with “ICs/BCs”, which are defined as initial conditions and boundary conditions, respectively, in both the main text and the figure caption. “Black carbon (BC)” is explicitly defined at its first occurrence in Sect. 4.2. The principal revisions are provided below.

Lines 70-75:

Wang et al. (2017) ported the Global Nested Air Quality Prediction Modeling System (GNAQPMS) to the second-generation Intel Xeon Phi processor, Knights Landing (KNL), achieving a 3.5x computational speedup through hybrid Message Passing Interface (MPI) and Open Multi-Processing (OpenMP) parallelization, vectorization, memory-access optimization, reduction of thread-local storage, and global communication optimization.

Lines 82-83:

Wang et al. (2019) developed the multi-point Carbon Bond Mechanism Z (MP CBM-Z) by implementing vectorized computation techniques within the CBM-Z framework.

Lines 102-106:

Alvanos and Christoudias (2017) developed a software package for the ECHAM/MESSy Atmospheric Chemistry (EMAC) model, enabling the automated generation of Compute Unified Device Architecture (CUDA) kernels for the numerical integration of atmospheric chemical kinetics using the Kinetic PreProcessor (KPP; Damian et al., 2002).

Lines 116-117:

Quevedo et al. (2025) adapted the third-order Rosenbrock solver in the Community Multiscale Air Quality (CMAQ) model by converting the Fortran code to CUDA Fortran.

Lines 122-124:

Through code refactoring from Fortran to standard C and the use of the heterogeneous-compute interface for portability (HIP) programming framework, Cao et al. (2025) successfully

parallelized the fourth-order Rosenbrock solver on a GPU-like architecture.

Lines 128-130:

Cao et al. (2023, 2024) implemented GPU-accelerated adaptations of the advection module in the Comprehensive Air Quality Model with Extensions (CAMx) using CUDA and HIP, respectively.

Lines 168-171:

Emission inventories such as the Multi-resolution Emission Inventory for China (MEIC; Li et al., 2017), Emissions Database for Global Atmospheric Research (EDGAR; Crippa et al., 2024), Hemispheric Transport of Air Pollution version 3 (HTAP_v3; Crippa et al., 2023), and Inversed Emission Inventory for Chinese Air Quality (CAQIEI; Kong et al., 2024) can be used.

Lines 176-178:

Other input datasets include configuration files, terrain data, photolysis data generated by the Tropospheric Ultraviolet and Visible (TUV) radiation model, initial conditions (ICs), and boundary conditions (BCs).

Lines 182-185:

For vertical diffusion, either the scheme of Byun and Dennis (1995) or the Yonsei University (YSU) scheme (Hong et al., 2006) can be selected to calculate the turbulent vertical diffusion coefficient.

Lines 200-207:

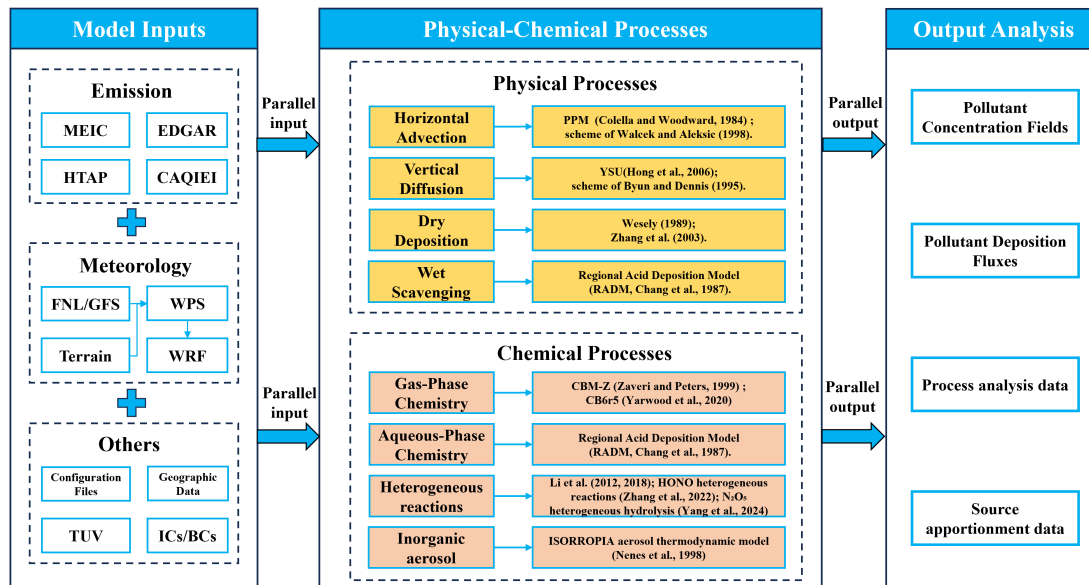


Figure 1. Computational framework and workflow of EPICC-Model V1.6.0. In the physical and chemical processes component, yellow denotes physical modules and orange denotes chemical modules. ICs and BCs denote initial conditions and boundary conditions, respectively. FNL

denotes the Final Operational Global Analysis produced by the National Centers for Environmental Prediction, GFS denotes the Global Forecast System, and WPS denotes the WRF Preprocessing System, where WRF represents the Weather Research and Forecasting model.

Lines 260-263:

As summarized in Table 1, each GPU-like computing node contains two of China' s domestic CPU processors and two GPU-like accelerators connected through Peripheral Component Interconnect Express (PCIe) 4.0 buses, and both the CPUs and GPU-like accelerators are first-generation devices.

Lines 281-284:

Analogous to AMD's Radeon Open Compute (ROCm) software stack (AMD, 2023), the dtk-23.04.1 toolkit (Cao et al., 2024) includes programming models, tools, compilers, libraries, and runtimes for artificial intelligence and high-performance computing applications on GPU-like accelerators.

Lines 528-529:

For the aerosol species, the displayed differences are of the order of $1.0 \times 10^{-5} \mu\text{g} \cdot \text{m}^{-3}$ for black carbon (BC).

Lines 616-618:

Figures 8 and 9, together with Table 6, present time-series comparisons and statistical evaluations of the simulated and observed daily $\text{PM}_{2.5}$ and maximum daily 8 h average (MDA8) O_3 concentrations.

19. Please perform a general check for further typos (missing periods etc)

Response: Thank you for this helpful comment. We apologize for the remaining typographical and grammatical errors. We have carefully proofread the entire manuscript and corrected the identified errors, as well as other issues in grammar, spelling, punctuation, terminology, and formatting. The principal corrections and their locations in the revised manuscript are summarized below:

Lines in the revised manuscript	Original text	Revised text
Line 68	“Thompson and Selin 2012”	“Thompson and Selin, 2012”

Line 162	<i>“EPICC-Model Working Group, 2026”</i>	<i>“EPICC-Model Working Group, 2026”</i>
Line 217	<i>“Appel, et al. (2021); Emery, et al. (2024)”</i>	<i>“Appel et al., 2021; Emery et al., 2024”</i>
Line 230	<i>“Figure 2. illustrates...”</i>	<i>“Figure 2 illustrates...”</i>
Line 267	<i>The software environment includes the Intel OneAPI 2021.3.0 toolkit for compiling the CPU code and the dtk-23.04.1 toolkit for compiling the GPU-like code</i>	<i>The software environment includes the Intel OneAPI 2021.3.0 toolkit for compiling the CPU code and the dtk-23.04.1 toolkit for compiling the GPU-like code.</i>
Lines 361-363	<i>Figure 4. The code-level implementation of CPU-GPU communication optimization. Panel (a) is the baseline Fortran code of the HIP-ori Panel (b) is the optimized Fortran code of the HIP-Opt1.</i>	<i>Figure 4. Code-level implementation of the CPU-GPU communication optimization. Panel (a) shows the baseline Fortran code for HIP-Ori, and panel (b) shows the optimized Fortran code for HIP-Opt1.</i>
Line 440	<i>“the model domain configurations maintains...”</i>	<i>“the model domain configuration maintains...”</i>
Line 465	<i>“toolkit”</i>	<i>“toolkit”</i>
Lines 635-637	<i>Figure 8. Time series of daily observed and simulated PM_{2.5} concentrations in major cities of China on 8-31 July, 2021. Panels (a) - (h) represent Beijing, Taiyuan, Hangzhou, Hefei, Fuzhou, Qingdao, Guangzhou, and Kunming.</i>	<i>Figure 8. Time series of observed and simulated daily PM_{2.5} concentrations in major Chinese cities from 8 to 31 July 2021. Panels (a)–(h) represent Beijing, Taiyuan, Hangzhou, Hefei, Fuzhou, Qingdao, Guangzhou, and Kunming, respectively.</i>
Lines 704 and 708	<i>“Figure 11.”</i>	<i>“Figure 11”</i>
Line 704	<i>“the label of ... represent”</i>	<i>“the labels ... represent”</i>
Lines 729-731	<i>Figure 11. Time consumption (a) and proportional contributions (b) for memory allocation, memory copy, kernel, and synchronize process under the -O0 compilation flag and varying data scales.</i>	<i>Figure 11. Time consumption (a) and proportional contributions (b) of GPU memory allocation, CPU-GPU memory copy, kernel execution, and synchronization under the -O0 compilation option for different input-data sizes.</i>

Line 741	<i>and the collaboratively indexed version HIP-Opt2</i>	<i>and the collaboratively indexed version HIP-Opt2.</i>
Lines 785-786	<i>Figure 12. (a) The average hourly elapsed time required for a 28-hour simulation across HIP-Ori, HIP-Opt1, Opt2 versions;</i>	<i>Figure 12. (a) The average hourly elapsed time required for a 28-hour simulation across the HIP-Ori, HIP-Opt1, and HIP-Opt2 versions;</i>
Lines 831-833	<i>This study systematically conducted comparative efficiency analyses by the validation of computational consistency in GPU-HADVPPM4HIP V1.0 through offline testing methodologies, as well as the verification of EPICC-Model's pollutant simulation performance on the EarthLab.</i>	<i>This study systematically evaluated the numerical consistency and computational efficiency of GPU-HADVPPM4HIP V1.0 through offline tests and further assessed the pollutant simulation performance of the EPICC-Model on EarthLab.</i>
Lines 844-847	<i>Memory copy operations (i.e., CPU-GPU data transfers) exhibited elapsed time increases rapidly with data size increasing, accounting for approximately 50% of total computation time at 10^8 data size with a continuing upward trend.</i>	<i>The elapsed time of memory-copy operations, namely CPU-GPU data transfers, increased rapidly with input-data size and accounted for approximately 50% of the total computational time for an input array containing 10^8 values.</i>