https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Python-Fortran Hybrid Programming for Deep Incorporation of AI and Physics Modeling and Data Assimilation (Hf2pMDA_1.0)

Xianrui Zhu[1+], Zikuan Lin[2+], Shaoqing Zhang[2*], Zebin Lu[2], Songhua Wu[3,4], Xiangyun Hou[2], Zhisheng Xiao[3], Zhicheng Ren[3], Jiangyu Li[5], Jing Xu[5], Yang Gao[6], Rixu Hao[7], Xiaolin Yu[2], Mingkui Li[2]

[1]Chongben Honors College, Ocean University of China, Qingdao, 266100, China
[2]Key Laboratory of Physical Oceanography, Ministry of Education, and Institute for Advanced Ocean Study, and Frontiers Science Center for Deep Ocean Multispheres and Earth System (FDOMES), College of Oceanic and Atmospheric Sciences, Ocean University of China, Qingdao, 266100, China
[3]Qingdao Leice Transient Technology Co., Ltd., Qingdao, 266100, China
[4]College of Marine Technology, Ocean Remote Sensing Institute, Ocean University of China, Qingdao, China
[5]Qingdao Marine and Meteorological Institute, Qingdao, 266100, China
[6]Key Laboratory of Marine Environmental Science and Ecology, Ministry of Education, Frontiers Science Center for Deep Ocean Multispheres and Earth System (FDOMES), Ocean University of China, Qingdao, 266100, China
[7]College of Intelligent Systems Science and Engineering, and Engineering Research Center of Navigation Instruments, Ministry of Education, Harbin Engineering University, Harbin, 150001, China.

[+]Co-first authors who contributed equally to this work.
*Correspondence to: Shaoqing Zhang (szhang@ouc.edu.cn)

Abstract. Artificial intelligence (AI) provides an unprecedented opportunity for advancing physics numerical modeling including data assimilation, which is a high-efficient and critically-important tool for advancing our understanding on Earth system and its applications. At the same time, deep incorporation of AI and physical modeling can make great driving to advance AI by injecting it rich physics from long time physics-based modeling development. However, since such physics models are conventionally coded in Fortran and AI algorithms usually are conveniently designed in Python, difficulties exist to directly incorporate AI algorithms into physics models, vice versa. Here, based on a f2py protocol, we have developed a procedure that implements an infrastructure which conveniently conducts Python and Fortran hybrid modeling and data assimilation (Hf2pMDA) to form a program entity so that AI algorithms and physical models can invoke mutually. As examples, within Hf2pMDA, a climate weakly coupled data assimilation (WCDA) system is naturally upgraded to a strongly CDA (SCDA) system, and a 1 km high-resolution weather DA system is conveniently implemented within a multi-layer downscaling model that has multiscale DA in different nesting layers. In the climate SCDA system, a coupled general circulation model (CGCM) and multiscale filtering algorithm is integrated by a Python main controller (PMC) that calls

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Fortran CGCM components and WCDA modules as well as a data-trained SCDA algorithm by latent space variational autoencoder (VAE) in Python. In the high-resolution weather DA system, the downscaled model consisting of traditional Fortran DA modules in all mother domains and Python VAE DA algorithm in the central child domain is integrated by a PMC that organizes these components. With convenient realization of deep incorporation of any AI algorithm and physics model, the Hf2pMDA has a great potential to make progresses on both AI and scientific modeling.

1 Introduction

Since the first Electronic Numerical Integrator and Computer (ENIAC) was born in 1940s, numerical weather prediction (NWP) models have been developed progressively (Benjamin et al., 2019) and advanced rapidly in recent years (Haarsma et al., 2016). An NWP model is a set of discretized momentum (dynamics) and energy (thermodynamics) budget equations of multi-sphere fluid motions, which can be dated back to the 17th century Newtonian mechanics (Newton, 1687) and 19th century Mayer's and Joule's energy conservation law (Mayer, 1842; Joule, 1843). Nowadays, an NWP model has advanced as an Earth system model (ESM) that consists of coupled atmosphere, ocean, sea-ice, land as well as biogeochemical processes etc. multi-sphere components (e.g., Kay et al., 2015; Danabasoglu et al., 2020). Now, the ESM is pursuing precise simulation and prediction for weather-climate variations through resolving multiscale interactions in the geofluid by developing high-resolution (HR) model (Zhang et al., 2023; Mouallem et al., 2025), which is a critically-important platform for Earth science studies (Haarsma et al., 2016).

However, as the spatial resolution continuously increases, physics-based modeling encounters many challenges (Chang et al., 2020; Marotzke, 2023). For example, as a consequence of water phase changing with strong nonlinearity, the genesis and melting processes of sea-ice have scales from millimeters (freezing coagulations, for instance) to hundreds of kilometers (glaciers, for instance) and rich scale processes such as melt ponds (e.g. Feng et al., 2022), ice leads (e.g. Qu et al., 2024) and polynyas (e.g. Diao et al., 2022) between them. Each scale band plays relatively independent and important role (Kenneth et al., 2020). Such a multiscale nature makes particular difficulties on HR modeling of sea-ice, since any HR model needs a specific parameterization for its sub-grid processes (Gou et al., 2025). The other outstanding example on HR ESM challenges is parameterization of planetary boundary layer (PBL) processes (Jia and Zhang, 2020). When the model horizontal resolution comes to a level of kilometers, classic PBL parameterization that deals with low resolution sub-grid boundary processes is no longer suitable for describing turbulences induced by differently-featured underneath surface structures, such as mechanical (roughness), thermal (urban heat) and shear-induced turbulences etc. (Kadivar et al., 2021).

Machine learning artificial intelligence (AI) technology provides unprecedented opportunities to resolve the challenging issues of physics-based HR modeling described above. As long as data that sufficiently represent an end-to-end process, no matter how complex the process is, it always can be resolved by a data-training procedure based on neural (deep learning) network (Hopfield, 1982; Rumelhart et al., 1986). Starting from the basic principle of least-square fitting to resolve optimal

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

weighting coefficients on a network, AI algorithms advance very rapidly and help greatly enhance accuracy of weather forecasts (Bi et al., 2023) and climate predictions (Ham et al., 2019). While data-driven AI technology rapidly spreads into almost every engineering and scientific discipline, the data-driven nature limits its further advancement and therefore physics-guided AI merges explosively (Faroughi et al., 2024; Yuan and Guo, 2024).

Deep incorporation of AI and physical models (let's call so for wording convenience) can make unbelievable opportunities to drive both even more rapidly advancing by injecting AI rich physics from long time physics-based modeling development and letting physical models being able to use any trained AI algorithm. Nevertheless, physical models are conventionally coded in Fortran (or Fortran-C hybrid) language and AI algorithms are designed and trained in Python language. Although great efforts are made for combining utilities of different languages together (Müller et al., 2024), deep incorporation of AI and physical models is still very difficult because of the lack of convenient infrastructure friendly to physical modelers between Fortran and Python. At present, on the one hand, most of the physics-guided AI employs some simplified physical expressions by Python (e.g. Häfner et al., 2018; Kochkov et al., 2024; Shu et al., 2025; Hao et al., 2025). On the other hand, physical models can only use some selected AI algorithms that can be translated into Fortran (Heuer et al., 2024) with FTorch library (Cambridge-ICCS, 2024) that does still not fully support all Torch application interfaces (Atkinson et al., 2025).

In this study, we document our developing process of Python-Fortran hybrid programming to establish a convenient infrastructure platform on which AI and physical modeling and data assimilation can deeply incorporate with respect to each other, called Hf2pMDA, based on the f2py protocol (Harris et al., 2020). Basically, through 3 steps of preparations in environment setting, recompiling of Fortran codes of physical model and plug interface, and simple programming of Python main controller, Hf2pMDA conveniently realizes the online deep incorporation of AI and physical modeling and data assimilation. Through exhibitions of two applications on climate strongly coupled data assimilation and high-precision small region weather Lidar assimilation, we show that Hf2pMDA is a readily using friendly platform to physical modelers and researchers.

This paper is organized as follows. **Section 2** describes the baseline of f2py protocol and outlines the general procedure to construct the framework of online Python-Fortran hybrid modeling and data assimilation (Hf2pMDA). **Sections 3 and 4** give examples of detailed implementation for applying Hf2pMDA to a climate model CDA system and a downscaling NWP system. Finally, summary and discussions are given in **Sect. 5**.

2 An infrastructure of Python-Fortran hybrid modeling and data assimilation (Hf2pMDA)

2.1 General f2py protocol and the outline of Hf2pMDA

The f2py protocol provides a basic idea for designing online Python-Fortran hybrid computation. A Python coded program with a kind of interpreted fashion is directly executed without pre-compiling requirement as compiled languages such as

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Fortran etc. A natural way to realize hybrid computation of Python and Fortran is that Python-callable Fortran (PCF) modules can be imported into a Python main controller (PMC) as callable objects and scientific Fortran computation can call Python-coded AI algorithms as subroutine or function. To that end, there are three aspects of concerns that need to be addressed in an infrastructure of Python-Fortran hybrid modeling and data assimilation (Hf2pMDA) designed in this study: (1) computational environment needs to be set to support both the Python and Fortran computations; (2) Fortran model codes need to be recompiled so that they become PCF modules; (3) application-oriented interfaces need to be designed to link PCF modules to PMC, and let scientific Fortran computation being able to call Python-coded AI algorithms. The logistics of Hf2pMDA can be illustrated a flow-chart in **Fig. 1**.

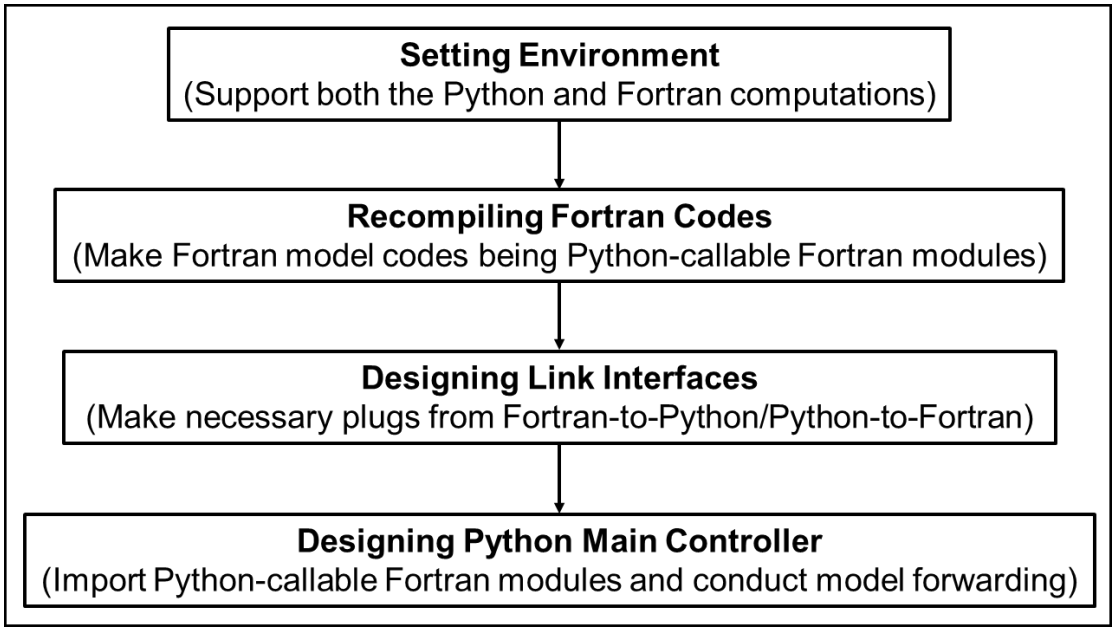


Figure 1: A logic flow-chart of the design idea for the infrastructure of Python-Fortran hybrid modeling and data assimilation (Hf2pMDA).

2.2 General procedure of Python-Fortran hybrid computation based on f2py

2.2.1 Environment setting

In theory, there is no specific environment variable for f2py. However, to minimize the interruption of other computation tasks to the Python-Fortran modeling development, we strongly recommend to use an environment setting tool such as *conda* etc. to set an isolated environment for the Python-Fortran hybrid application. We may start from a Python environment by typing the command "*conda create -n f2py python=3.11*" and continue to complete the setting by typing "*conda activate f2py*" and "*conda install numpy=1.26*" for follow-ups until necessary, where 3.11 and 1.26 are environment parameters of corresponding software. For some non-standardized environment variables only necessary for specific applications, the

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

"*source*" command is a convenient complementary. Detailed description will be given in specific applications by **Sects. 3 and 4**.

2.2.2 Creating of Python-callable Fortran (PCF) modules

In order to become Python-callable, the existed Fortran codes need to be recompiled with an additional new tag "-fPIC" that makes the Fortran codes being compiled become Dynamic Link Library (DLL) candidates with a "position independent" nature. Once relatively-independent Fortran codes are all recompiled as DLL candidates, they are ready to link with the Python main controller (PMC).

To minimize complexity of the infrastructure between PMC and PCF modules, a plug-featured Fortran subroutine that well organizes the PCF modules and makes a few plug-ins for PMC needs to be encapsulated with all PCF modules to form a signature file that shows a clear structure of such plugs. We use the following command line to initially create a signature file called *appname.pyf*:

```
$ f2py app_plugs.F90 -m appname -h --overwrite-signature appname.pyf
```

where *app_plugs.F90* consists of subroutines or functions that organize the PCF modules and serve as the plug-ins of PMC. Supposing that there are a few DLL candidates created in the compiling procedure described above called *libapp1.a*, *libapp2.a,...*etc., then, the encapsulating command looks like a continuous line of the above initial creation line of *appname.pyf* as:

```
$ FC="mpif90" CC="mpicc" CXX="mpicxx" LDSHARED="mpif90" LDFLAGS="-no-ipo \
-Wl, --export-dynamic" f2py -c appname.pyf app_plugs.F90 libapp1.a \
libapp2.a ... -lnetcdf -L/.../software/netcdf3/lib -L/.../openmpi/lib \
--backend distutils
```

where the option "-no-ipo" closes out Interprocedural Optimization to minimize potential uncertainties due to optimization cross modules and the option "-Wl" ensures used functions linked to DLL. Note that here the openmpi/lib also needs to be recompiled with the tag "-fPIC." Finally, as the above command is normally conducted without any error message, a complete DLL file called *appname.cpython-311-x86_64-linux-gnu.so* is formed (automatically named by the compiling environment parameters), which is a Python-callable DDL "shared object" where all subroutines and functions are callable by PMC. All information about the logic structure of *appname.cpython-311-x86_64-linux-gnu.so* is recorded in the signature file *appname.pyf*, which will be served as an import name in PMC, also serving as an efficient guideline for constructing PMC.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

2.2.3 Fortran-callable Python (FCP) algorithms

To let the Fortran codes be able to call Python-coded algorithms, we need an interface (called *callback_python.F90*, for instance) that translates the Python external functions into Fortran-callable public subroutines (called *python_algorithm{1,2}* etc., for instance). At the same time, the *app_plug.F90* shall include the definition of these external functions (called *python_foo{1,2}* etc., for instance) which are transported from PMC into Fortran application at plug-ins in *app_plug.F90* as arguments. This Fortran interface file *callback_python.F90* shall be compiled with *app_plug.F90* together by going through the procedure described above to join the signature file *appname.pyf*. Then, all subroutines representing python algorithms can be called in any Fortran application through a normal "*use*" statement.

2.3 The PMC's structure and execution of Python-Fortran hybrid computation

2.3.1 The PMC's structure

A PMC (called *appname_main.py*, for instance) shall consist of three important parts: import statements, model integration conductor and AI algorithm interface as described before. The import statements must include "from *appname* import *module_list*" (import selected Fortran modules from *appname* defined before) or simply "import *appname*" to import all modules in *appname*. The import statements may include any necessary modules that are used in PMC such as an MPI module which provides process-element identification (PE-id) etc. information, as well as defined AI algorithms and so on.

To minimize Python-recoding for the existed Fortran model, the PMC's statements for conducting model integration usually remain simple as possible as Python AI algorithms are conveniently called by the Fortran model. In the example of climate model application of H_{f2p}MDA, which will be described in **Sect. 3**, we implement an AI strongly-coupled data assimilation (SCDA) based on the existed Fortran-coded weakly-coupled data assimilation (WCDA). Due to the strongly-modular nature of existed Fortran codes, the AI algorithm trained by deep-learning is conveniently inserted at the atmosphere-ocean interface, we let PMC to take care of time integration loop (detailed in **Sect. 3.2**). However, in the example of H_{f2p}MDA's NWP model application described in **Sect. 4**, we implement AI DA in the 3rd domain downscaled from a recursive domain integration. In that circumstance, the PMC's model integration conductor only consists of a single line, which will be elucidated in **Sect. 4.2**.

2.3.2 The execution of Python-Fortran hybrid computation

We can still use *mpirun* command to reflect the MPI nature of Python-Fortran hybrid computation as:

```
$ mpirun -n xx python appname_main.py > appname.log
```

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

where "xx" is the PE number required by the program, *appname.log* is the log file recording the log information during the program running. We may insert this run command line into the existed *run_script* file of the Fortran model to let the H_{f2p}MDA run in a large scale of MPI usage by a *qsub* command to submit it to the back-end job queue.

Through the steps described above, the infrastructure of Python-Fortran hybrid modeling and data assimilation (H_{f2p}MDA) can be constructed. As illustrated in **Fig. 2**, within the H_{f2p}MDA infrastructure, the scientific modeling and machine learning AI algorithms can be deeply incorporated. Next, we will use examples of climate model CDA and weather model high-precision DA to give specific implementation of H_{f2p}MDA.

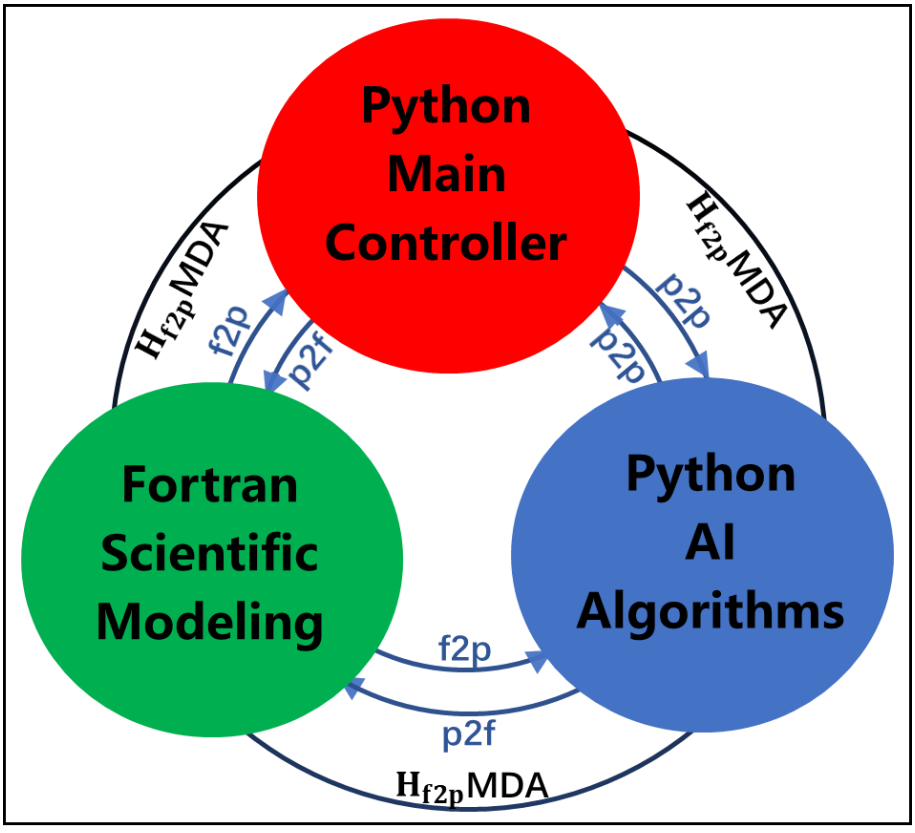


Figure 2: A schematic illustration of the designed infrastructure of Python-Fortran hybrid modeling and data assimilation (Hf2pMDA) for deep incorporation of scientific modeling and machine learning AI algorithms.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

3 H_{f2p}MDA applied to climate model

3.1 The CM2 model and its WCDA system

The CM2 is a fully-coupled Earth system model consisting of the atmosphere, ocean, land and sea-ice components (Lin, 2004; Gnanadesikan et al., 2006; Winton, 2000), having been an important member of the Coupled Model Intercomparison Project (CMIP) (Delworth et al., 2006a; Randall, 2007; Taylor et al., 2012; Eyring et al., 2016). The version we used in this study has the atmosphere model with 2° latitude × 2.5° longitude horizontal resolution and 24 vertical levels, and the ocean with 1°×1° horizontal resolution and 50 vertical levels, time stepping with 30 minutes and 2 hours for atmosphere and ocean as well as their coupling. As the first CGCM's CDA system, the first version of CM2-CDA was developed based on the ensemble Kalman filter (EnKF) (Zhang and Anderson, 2003; Zhang et al., 2007), implemented in a WCDA manner. However, an EnKF CDA system is very expensive and has a limitation on extracting low-frequency observational signals (Yu et al., 2019). To resolving this issue, based on scale disassembling of a long time series of single model solution and the EnKF's framework, a multiscale high-efficiency approximate EnKF (MSHea-EnKF) was designed (Yu et al., 2019) and implemented into the CM2 (Lu et al., 2020). Therefore, the current version of CM2-CDA employed in this study uses a sequential WCDA procedure which forwards on single CM2 coupled model integrated with atmosphere-ocean observations as they are available.

In a WCDA procedure, the atmosphere (ocean) observations only adjust the atmosphere (ocean) own itself state and observational information transfers cross the atmosphere and ocean by coupled model exchanged fluxes. Without direct observational adjustment between the atmosphere and ocean, imperfect coupling physics may still introduce errors into the coupled model and CDA remains in an incomplete fashion. The strongly-CDA (SCDA) has been pursued to fulfill sufficient balanced and coherent coupled state estimation in CDA (Zhang et al., 2020) but it keeps in difficulties (e.g. De Rosnay et al., 2022) imposed by the limitation of linear regression in traditional DA algorithms for such different characteristic scales cross the atmosphere and ocean. The natural nonlinear nature of machine deep learning AI algorithms makes SCDA feasible if it is inserted into a WCDA system as an additional term.

Next, we will describe how to apply the H_{f2p}MDA framework outlined in **Sect. 2** to the CM2-CDA to construct an H_{f2p}MDA-$CM2_{CDA}$ so that an AI SCDA scheme can be conveniently incorporated into CM2-CDA.

3.2 Design of modular H_{f2p}MDA-$CM2_{CDA}$

3.2.1 The environment requirement

Beside the general environment described in **Sect. 2.2.1**, we need to use *openmpi* for MPI application in this H_{f2p}MDA-$CM2_{CDA}$ case. We need to recompile *openmpi* package with added "-fPIC" tag and "*make install*". Then we *source* a file of command list containing the following 3 lines to add the *openmpi* application to the environment *myf2p*:

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026


```
export PATH="…/openmpi/bin:$PATH"
export LIBRARY_PATH="…/openmpi/lib:$LIBRARY_PATH"
export LD_LIBRARY_PATH="…/openmpi/lib:$LD_LIBRARY_PATH"
```

Again, the *openmpi* directory shall be in the specific application directory for H_{f2p}MDA-$CM2_{CDA}$ so that this *openmpi* setting does NOT influence any other applications.

3.2.2 Recompiling of CM2-CDA all static libraries

In general, the CM2 model has a very clear modular feature, in which the atmosphere and ocean components have relatively-independent structure, with land as atmospheric boundary processes and sea-ice being the interface of atmosphere and ocean. The main controller *coupler_main* organizes all interfaces between model components and loops time integration as every half hour for the atmosphere and every two hours for the ocean and atmosphere-ocean coupling. The weakly CDA conducts atmosphere (ocean) DA computation within the atmosphere (ocean) component and it therefore does not change the basic logic structure of *coupler_main.* The entire CM2-CDA package of codes consists of 4 static libraries: *libfms.a*, *libland.a*, *libecda.a* and *libcoupler.a*. Here *libfms.a* contains all the lowest shared information such as MPI and remapping functions etc. and *libland.a* is a relatively-independent application package for land model based on *libfms.a*, while *libecda.a* includes all atmosphere and ocean model components as well as CDA subroutines beside land model. Finally, *libcoupler.a* is the highest infrastructure to connect all parts above together by *coupler_main*. To make all Fortran modules in these static libraries becoming PCF modules, we first partition the main controller *coupler_main.F90* as two parts, *cm2_cda_mainsubs.F90* and *cm2_cda_plugs.F90.* The *cm2_cda_mainsubs.F90* is basically a modified version of *coupler_main.F90*, in which the main program is disassembled into a list of subroutines such as *cm2_cda_maininit()*, *atmos_step()* and *ocean_step()* etc. as module initialization and single step model integration. The *cm2_cda_plugs.F90* simply organizes various model component integration of single step as plug-ins (*tool_atmos_step()* and *tool_ocean_step()* etc., for instance) for PMC.

Following the procedure described in **Sect. 2.2.2**, we first recompile 4 static libraries *libfms.a*, *libland.a*, *libecda.a* and *libcoupler.a* by adding "-fPIC" option tag. Then we use *f2py* to combine *cm2_cda_plugs.F90* with these 4 static libraries together to create the signature file *cm2_cda.pyf* and DLL *cm2_cda.python-311-x86_64-linux-gnu.so*. Finally, the DLL *cm2_cda.python-311-x86_64-linux-gnu.so* to work directory where *cm2_cda_main.py*is stored.

3.2.3 H_{f2p}MDA-$CM2_{CDA}$'s PMC and its execution

Based on the guideline of PMC's structure described in **Sect. 2.3.1**, we construct the python main controller of H_{f2p}MDA-$CM2_{CDA}$, called *cm2_cda_main.py.* In this case, as arranging time integration of the coupled model, *cm2_cda_main.py* also takes in charge of connecting the atmosphere-ocean interface with the AI SCDA algorithm implemented by latent space

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

variational autoencoder, which will be described in the next section. Then following the procedure described in **Sect. 2.3.2**, we can run *cm2_cda_main.py* to complete the process of H_{f2p}MDA-$CM2_{CDA}$'s design and execution.

3.3 A latent space VAE SCDA algorithm within H_{f2p}MDA-$CM2_{CDA}$

Since the PMC *cm2_cda_main.py* conducts the coupled model time integration and it explicitly controls the evolution of coupling interface between the atmosphere and ocean, we can conveniently insert SCDA into the model integration. What follows describes a latent space variational autoencoder (VAE) SCDA algorithm, which consists of VAE latent space transformation and latent space minimization (Cheng et al., 2022).

3.3.1 VAE latent space transformation

The VAE refers to a class of deep generative models that combines artificial neural network and variational Bayesian methods (Cheng et al., 2024; Fan et al., 2025). We can use **Fig. 3** to illustrate the transformation process of between the physical and latent spaces by VAE used in this study. Unlike a basic autoencoder architecture, we here use a versionincluding a Gaussian distribution constraint of latent space so that the minimized total loss includes the Kullback-Leibler (KL) loss which ensures the distribution of induced tensor in latent space close to a Gaussian (Solera-Rico et al., 2024; Lin et al., 2025), called β-VAE. The principle of the used algorithm is to induce a reduced-order tensor in the latent space (denoted as z in the upper panel of **Fig. 3**) by minimizing the total loss $L(x)$ consisting of reconstruction and KL losses (shown in the lower panel of **Fig. 3**). We set the value of parameter β as 10^{-6} to seek a relatively compact representation of the latent space while maintaining the reconstruction accuracy. While the detailed description of the algorithm can be referred to Lin et al. (2025), we will outline the data-training process of transformation between the physical and latent spaces below.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

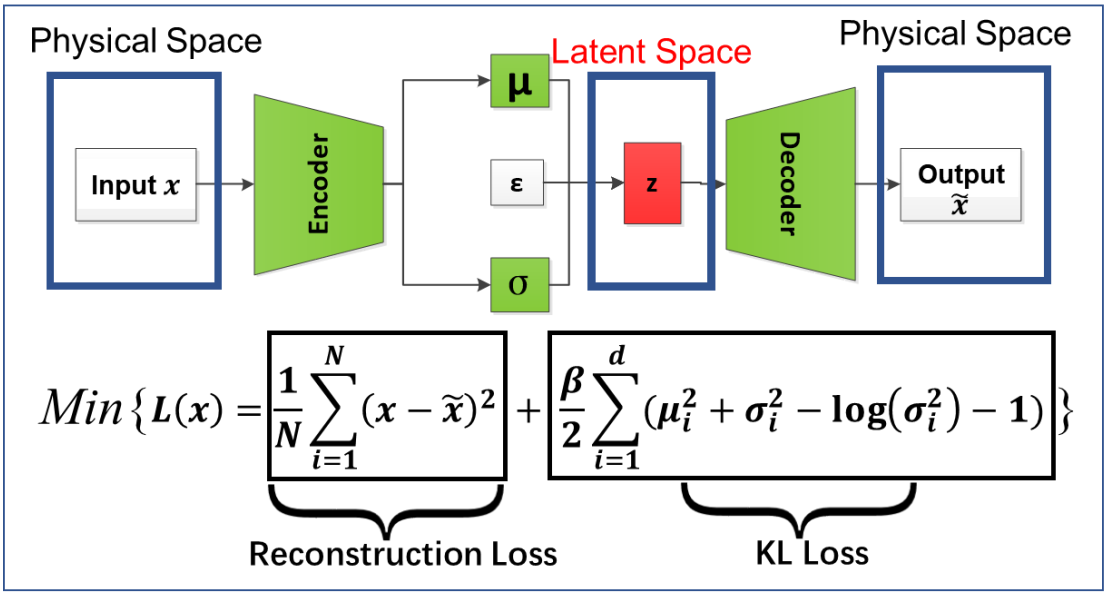


Figure 3: A schematic illustration of the principle of transformation between physical and latent spaces in which the loss function includes reconstruction loss and Kullback-Leibler (KL) loss weighted by a β value (called β-VAE). The reconstruction loss measures the accuracy of VAE reconstruction while the KL loss measures the distance between the generated probability distribution and Gaussian distribution. The algorithm is implemented by data-trained machine learning.

3.3.2 A SCDA algorithm in latent space minimization

Based on the technology of VAE latent space transformation described above, as shown in **Fig. 4**, Latent Data Assimilation (LDA) further introduces a minimization process in the latent space, which includes an observational term (see lower panel of **Fig. 4**) in the loss function being minimized, to implement data assimilation (Cheng et al., 2024). Note that the utilization error convariance matrices B and R (also serving as scaling factors for the background and observational terms respectively) makes the loss function non-dimensional in minimization for which we choose 3-dimensional variational (3D-Var) for simplicity in this study (although 4D-Var could also be used).

The explicit minimization process (will be outlined in **Sect. 3.3.3**) to implement LDA is very fast since the size of states in the latent space is quite small due to its reduced-order nature. It is convenient to include atmosphere and ocean model variables in the physical space vector x (input in **Fig. 4**) and atmosphere and ocean observational observations (see **Fig. 4**) at the air-sea interface. Then, the final decoded physical space vector $\tilde{\boldsymbol{x}}$ is an updated model state that incorporates the atmosphere and ocean observations. The LDA illustrated by **Fig. 4** therefore implements SCDA in which the atmosphere (ocean) observations realize instantaneous and direct adjustment for the ocean (atmosphere) state. Since the encoding and decoding process is done by machine deep learning that consisting of nonlinear activation functions, such SCDA is nonlinear although a minimization process is used to incorporate observational data into the latent space (Lin et al., 2025).

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

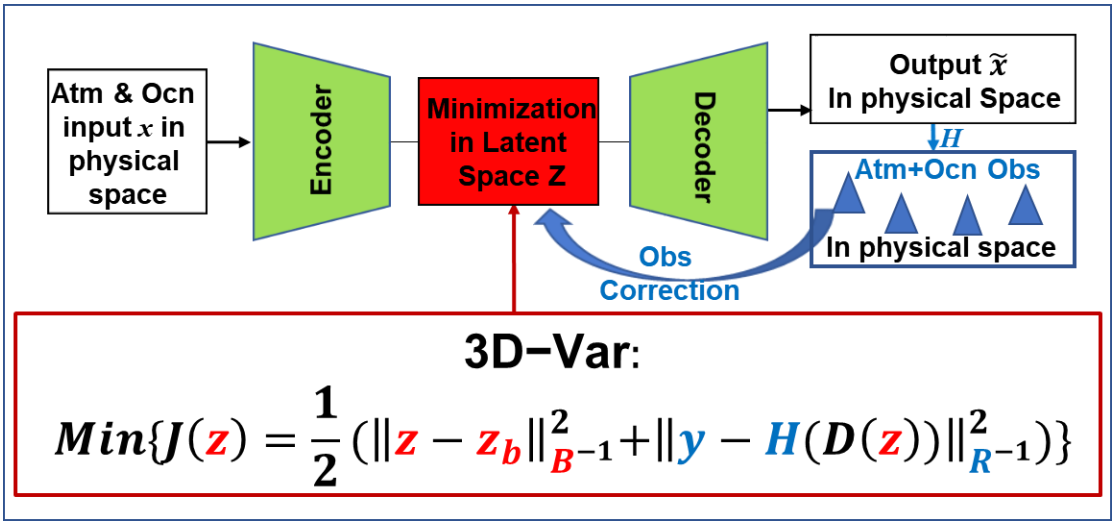


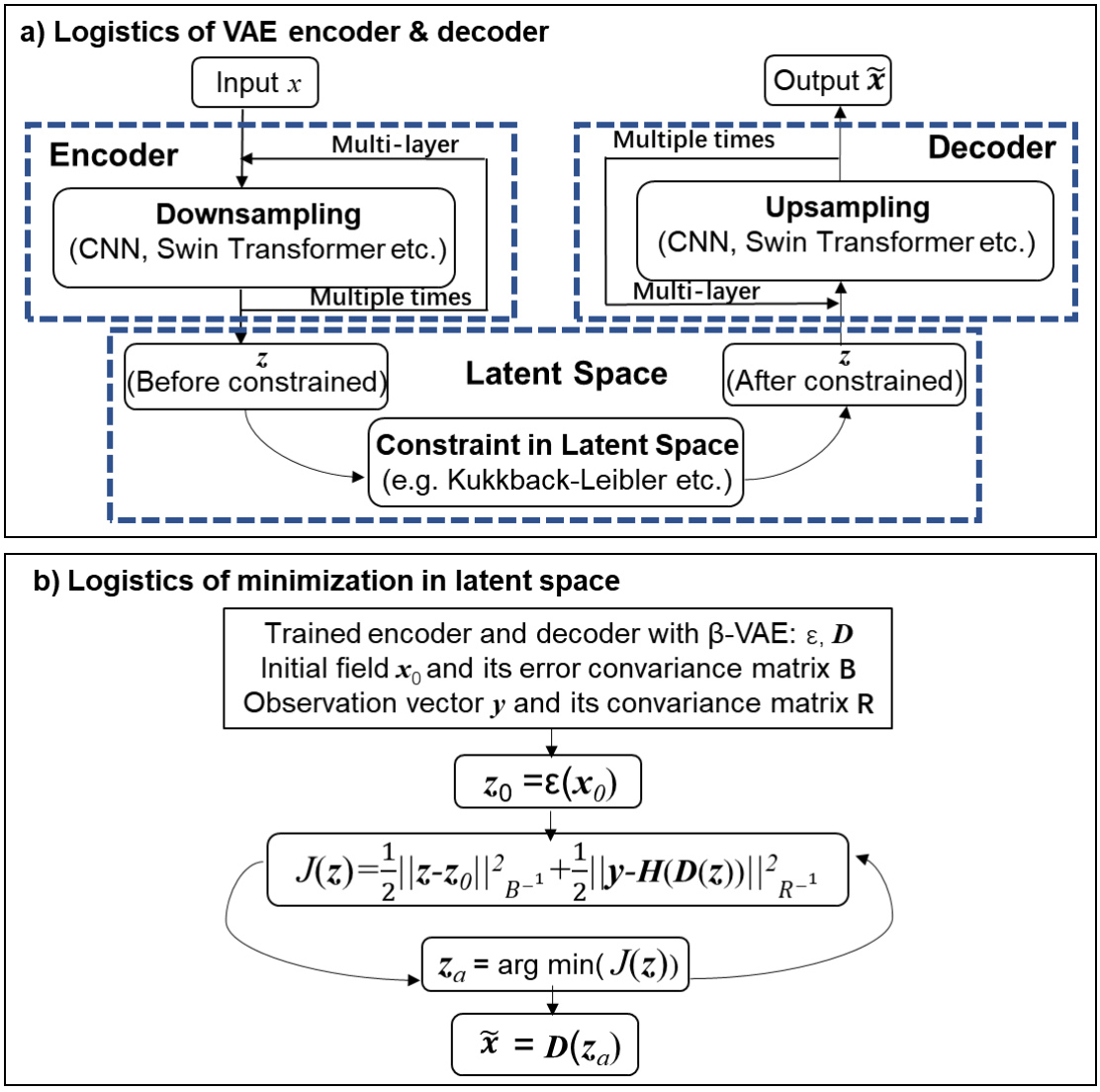
Figure 4: A schematic illustration of the strongly-coupled data assimilation (SCDA) principle of observational correction of atmosphere-ocean fields by latent space minimization i.e. 3-dimensional variational (3D-Var) working on the atmospheric and oceanic observations. B and R are error covariance matrices for the latent space state background and observations, also serving as the normalization (scaling) factor of the loss function being minimized. Such an SCDA algorithm is implemented by a high-efficient machine learning process described in Sect. 3.2.3.

3.3.3 The VAE's data-training and latent space minimization

This section gives programming comments on VAE training and latent space minimization of SCDA in H_{f2p}MDA-$CM2_{CDA}$, serving as a guideline on concrete coding. For VAE training, on the one hand, the basic procedure of transformation between physical and latent spaces by VAE encoder and decoder is very simple and standardized as illustrated by **Fig. 5*a***. In general, it only consists of downsampling, upsampling and latent space constraint (if necessary), three or even two parts. On the other hand, the training process for a specific application could be very unique in selections of parameters and loop levels as well as neural networks etc. To that end, we will provide the specific procedure descriptions when we present the experiment results for CM2-SCDA in **Sect. 3.4** and WRF-DA within the H_{f2p}MDA in **Sect. 4.3**.

In contrast to VAE training, in all applications, latent space minimization is conducted in a quite uniform way as shown in **Fig. 5*b***. Once the training process of VAE encoder Ɛ and decoder D is done, a joint atmosphere-ocean state vector x and a latent space state vector z can be transformed back and forth very high-effiviently by the trained Ɛ and D. Due to the order of state vector z in the non-dimensional latent space is reduced to a necessarily-small size, the minimization can be conducted in a very fast way. As the error criterion in prior is satisfied, the assimilation solution $\tilde{\boldsymbol{x}}$ is ready.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026



295 **Figure 5: Logistics of VAE latent space data assimilation in Hf2pMDA. *a*) A general implementation flow-chart of encoder and decoder for transformation between physical and latent spaces as principally illustrated in Fig. 3 for a β-VAE. *b*) A general implementation flow-chart of minimization in latent space.**

3.4 Preliminary results of SCDA by H_{f2p}MDA-CM2$_{CDA}$

3.4.1 The VAE reconstruction of physical variables at air-sea interface

300 In the H_{f2p}MDA-CM2$_{CDA}$ system, to produce SCDA results, we first train the encoder and decoder to reconstruct the reduced-order state in latent space using the atmosphere and ocean state variables at the air-sea interface. High-efficiently

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

reconstructing the latent space with necessary accuracy is the core of training scheme which consists of various selections on parameters and loop levels as well as neural networks etc. For the CM2-SCDA, we set the input vector x_0 consisting of the atmosphere surface wind, temperature and pressure (U_s, V_s, T_s, P_s), and ocean surface currents, temperature and height (SSU, SSV, SST, SSH). The concrete training process is shown in details by **Fig. 6**. As summarized by **Fig. 5*a***, the most part of encoder (decoder) conducts a training of downsampling (upsampling). However, it's worth to mention that different from a general case in which the physical state usually only includes information within a component of multi-sphere geofluid, in this SCDA case, the physical state being transformed consists of atmospheric and oceanic information with different characteristic scales. Under this circumstance, the VAE training process therefore includes a Cross Attention step which consists of mixing and its inverse (separation). Apart from this point, the VAE still follows a pretty general training procedure for downsampling and upsampling.

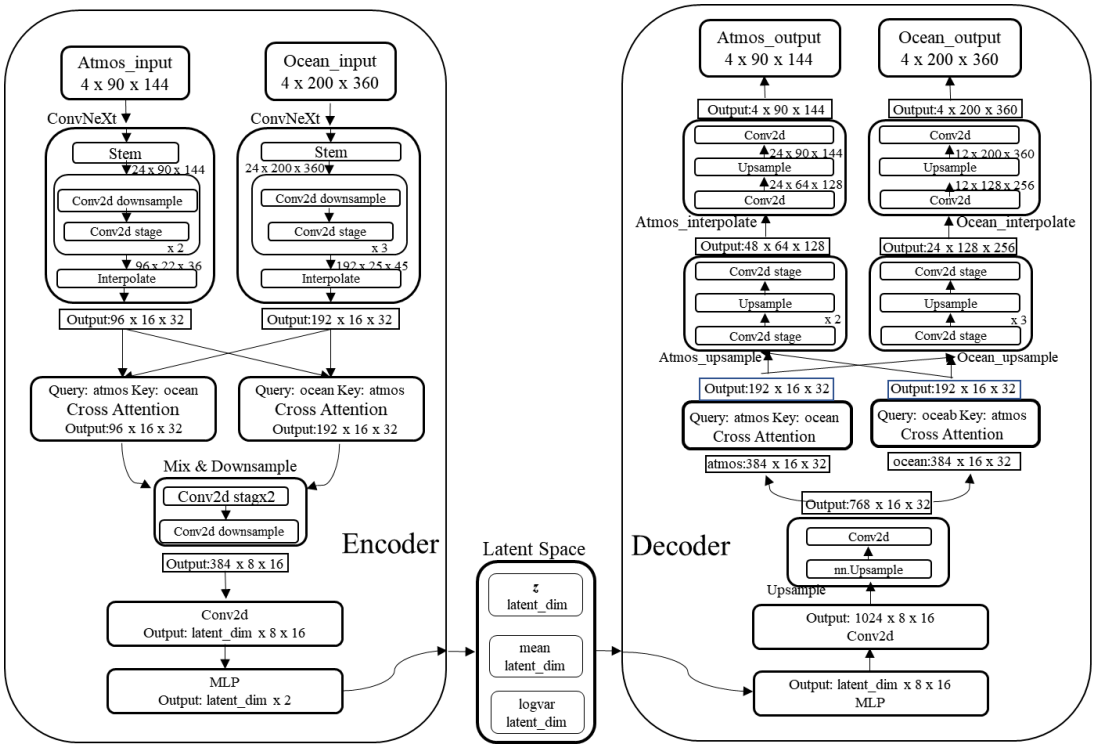


Figure 6: The training process of VAE encoder (left) and decoder (right) in physical-latent space transformation of SCDA in Hf2pMDA-CM2$_{CDA}$ as a specific implementation of general VAE logistics shown in Fig. 5*a*. In this case, we set the latent_dim as 20480. The denotation "×N" at the lower right corner of some box means repeating this operation by N times.

After simply conducting normalization on each variable of U_s, V_s, T_s, P_s, SSU, SSV, SST and SSH using the difference of its maximum and minimum values, we form the input vector x and go through the training procedure shown by **Fig. 6** with a dataset of 50-year time series with 3-hour intervals (totally 144540 samples in time) to obtain the latent space vector z. The

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

reconstruction accuracy for each variable is presented in **Figs. 7 and 8**. From there, we can see that the reconstructed fields (**Figs. 7*e-h***) are visually nearly-identical to the original physical fields (**Figs. 7*a-d***). The absolute errors of the reconstruction always remain in an order of 10^{-2} (**Figs. 7*i-l***). Although the reconstructed fields still show a little smoother than the original physical fields, the accuracy of reconstruction shall be sufficient for the first time practicing on climate SCDA issues.

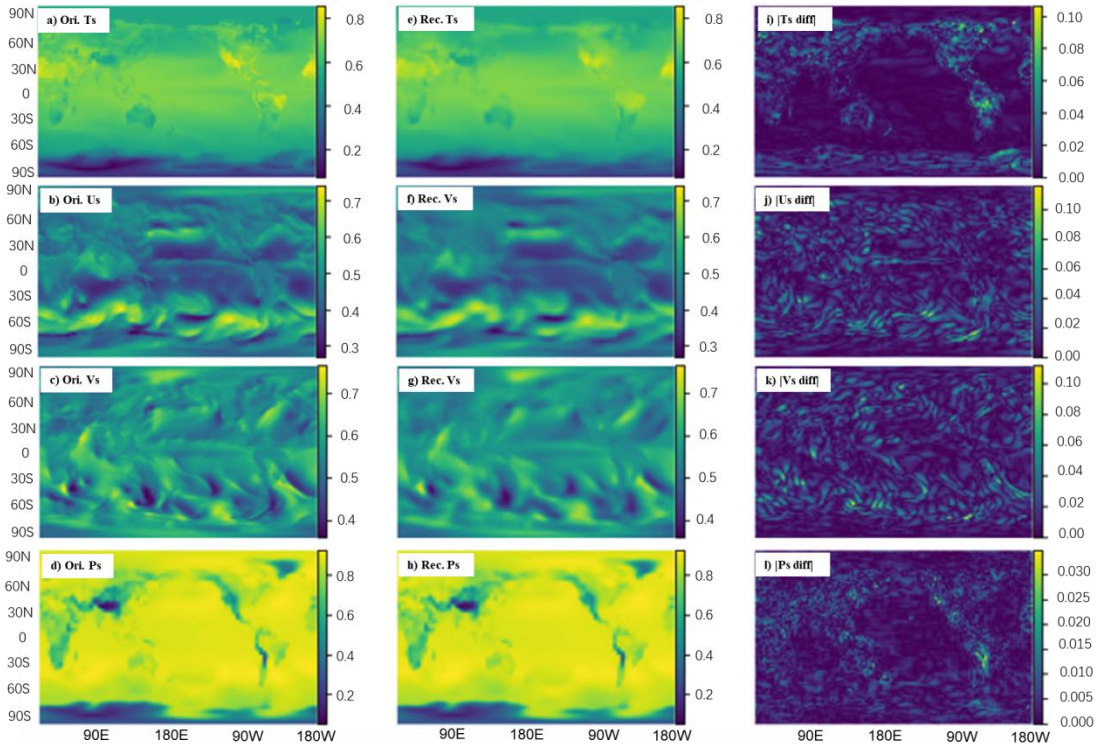
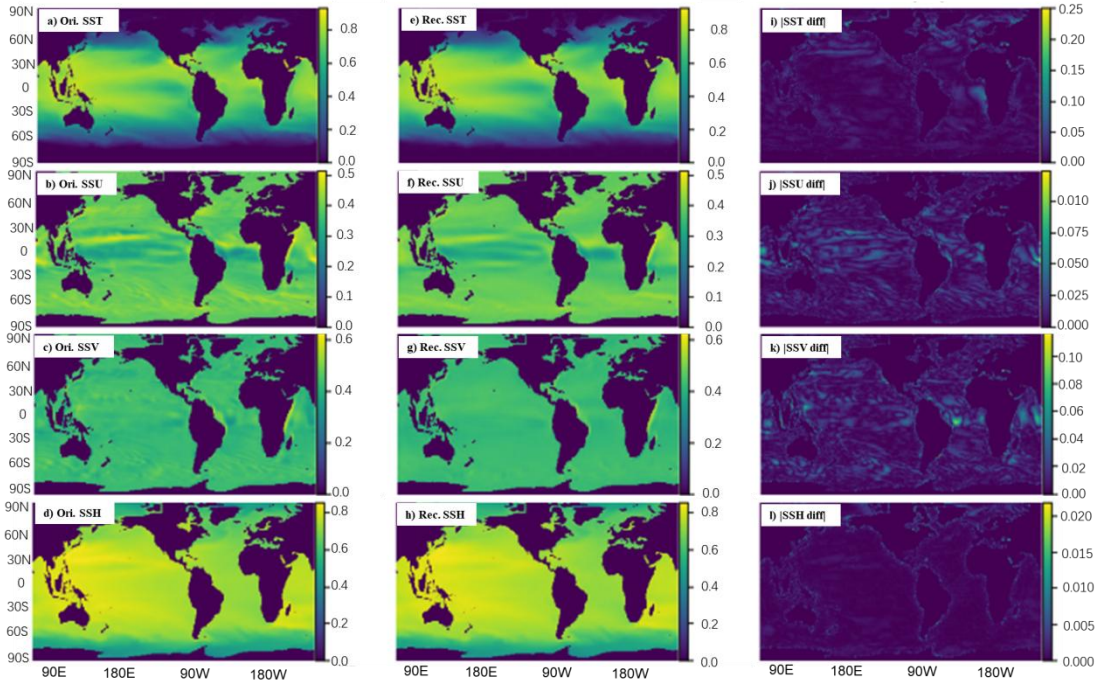


Figure 7: The typical VAE reconstruction accuracy for atmosphere surface *aei*) temperature Ts, *bfj*) u-component Us, *cgk*) v-component Vs and *dhl*) pressure Ps. The time slide is randomly taken from a 5-year validation set. The leftmost (panels *a-d*) is the input *x*, the middle (panels *e-h*) is the reconstructed $\tilde{x}$ through decoding from *z*, the rightmost (panels *i-l*) is the absolute values of their differences.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026



330 **Figure 8: Same as Fig. 7 but for sea surface *aei*) temperature SST, *bfj*) u-component SSUs *cgk*) v-component SSV and *dhl*) height SSH.**

3.4.2 The SCDA results

In the case of H_{f2p}MDA-CM2$_{CDA}$, through the VAE training, we establish a latent space which is transformed from the air-sea interface space so that we can efficiently conduct minimization to blending atmospheric and oceanic observations into
335 the model space. Here while the size of coupled state vector in physical space is 343,296, through space transformation, the state vector size in the latent space becomes 20480 with a reduction rate of 94%, which makes fast and efficient minimization feasible. Next, we will show that although only 6% of physical space vector size is used in latent space minimization, the reconstruction fields are reasonably to represent the background physical information for extracting observational information.

340 In this test case, we use the ERA5 (5th generation atmospheric reanalysis of European Centre for Medium-Range Weather Forecasts) surface pressure Ps (with a 25 km horizontal resolution) and OISST (NOAA 1/4°Daily Optimum Interpolation Sea Surface Temperature) as atmospheric and oceanic observations respectively. We run the H_{f2p}MDA-CM2$_{CDA}$ system on the whole 1982 and present a typical minimization process in **Fig. 9*a*** and the whole year verification results in **Figs. 9*b-c***. The latent space can be comprehended as a reduced-order holographic space that represents all nonlinear complex
345 relationships of physical variables. In this space, as minimization proceeds, when the observation loss decreases (brown curve in **Fig. 9*a***, the latent space state vector z gets adjusted as the background loss increases (green curve in **Fig. 9*a***) so that

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

the observations are coherently incorporated into the background at the convergent point that the background and observation reach equilibrium.

From **Figs. 9*b-c***, we can see that due to the low-frequency and quasi-linear nature of ocean motions, both WCDA and SCDA make the model SST quickly converged to observations and keep in nearly identical root mean squared errors (RMSEs) in most of the time. In the contrary, because of strong internal variability of the atmosphere, the curves of Ps RMSEs in WCDA and SCDA quickly separate after a few days, but SCDA does not show his merit within a few months. After about 4~5 months, SCDA starts to show a little smaller RMSEs. In terms of a 6-month mean in July to December, SCDA reduces the WCDA's RMSE by roughly 4%, in which SCDA's RMSE is 4.62 out of 4.81 of WCDA's.

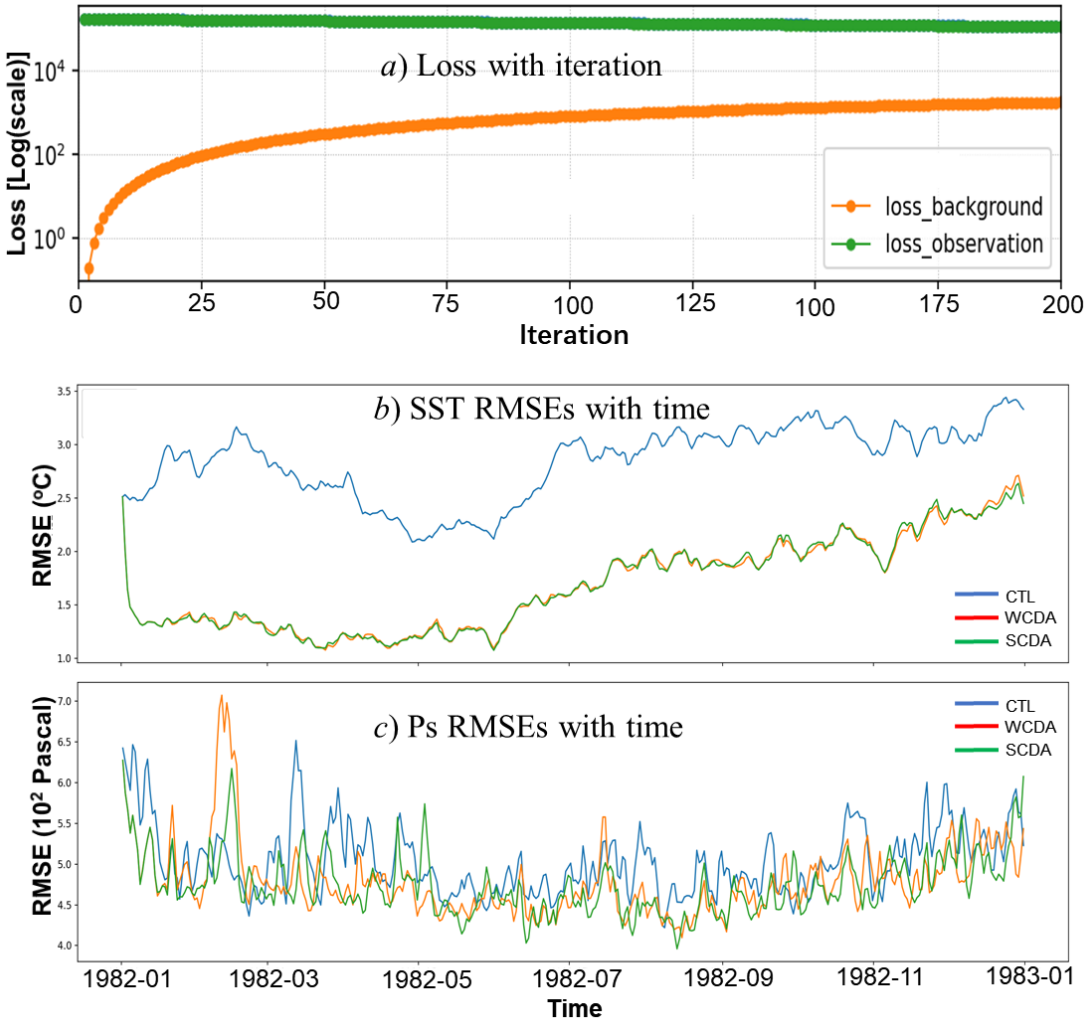


Figure 9: *a*) Variations of the background loss (brown) and observation loss (green) by the iteration in the latent space minimization as shown in Fig. 5b for carrying out SCDA, and *bc*) the time series of root mean squared errors (RMSEs) of SST and

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Ps in free model control simulation (CTL) (blue), and WCDA (brown) as well as SCDA (green) produced by Hf2pMDA-CM2$_{CDA}$ at the 200th iteration of minimization. The WCDA is a multiscale high-efficient approximate filter (MSHea-EnKF).

It's worth to mention that in this H$_{f2p}$MDA-CM2$_{CDA}$ case, the model resolution is pretty low (~200 km for the atmosphere and ~100 km for the ocean). Under the circumstance, the coupled model dynamics in general is quasi-linear (Perlwitz et al., 2017), and it is possible that WCDA can extract the most of observational information into the coupled model. However, our preliminary results using H$_{f2p}$MDA-CM2$_{CDA}$ exhibits the promise that after a sufficient time scale, SCDA can create additional values to the CDA system. This may become critically-important as the model attempts to resolve fine-scale and frequent air-sea coupling processes as a tropical cyclone or mesoscale eddy passes by.

4 H$_{f2p}$MDA applied to an NWP model

4.1 Brief description of WRF and its DA system in multi-layer nesting

4.1.1 A multi-layer nesting WRF model

Here we use a version of Weather Research and Forecasting (WRF, v3.7.1) model configured as 3-layer nesting at 9, 3 and 1 km horizontal resolutions for the West of China (D01), Central Sichuan Province (D02) and Chengdu City (D03) respectively as shown in **Fig. 10*a***. The WRF model is divided into 50 vertical levels from the surface to the model top at 50 hPa. The model includes most of the primary physics such as the Kain-Fritsch convection parameterization (Kain, 2004), Rapid Radiative Transfer Model for GCMs (RRTMG) longwave and shortwave radiations (Clough et al., 2005) and Yonsei University (YSU) boundary layer (Hu et al., 2013) as well as WRF single-moment 3-class (WSM3) microphysics (Hong et al., 2004) etc. The initial and boundary conditions are drawn from the ERA5 reanalysis dataset in this study. The original purpose of this WRF configuration is to study the impact of lidar observations on the Chengdu City. Here we employ it to exhibit the capability to conveniently carry out parallel experiments using H$_{f2p}$MDA conducting traditional DA and AI LDA.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

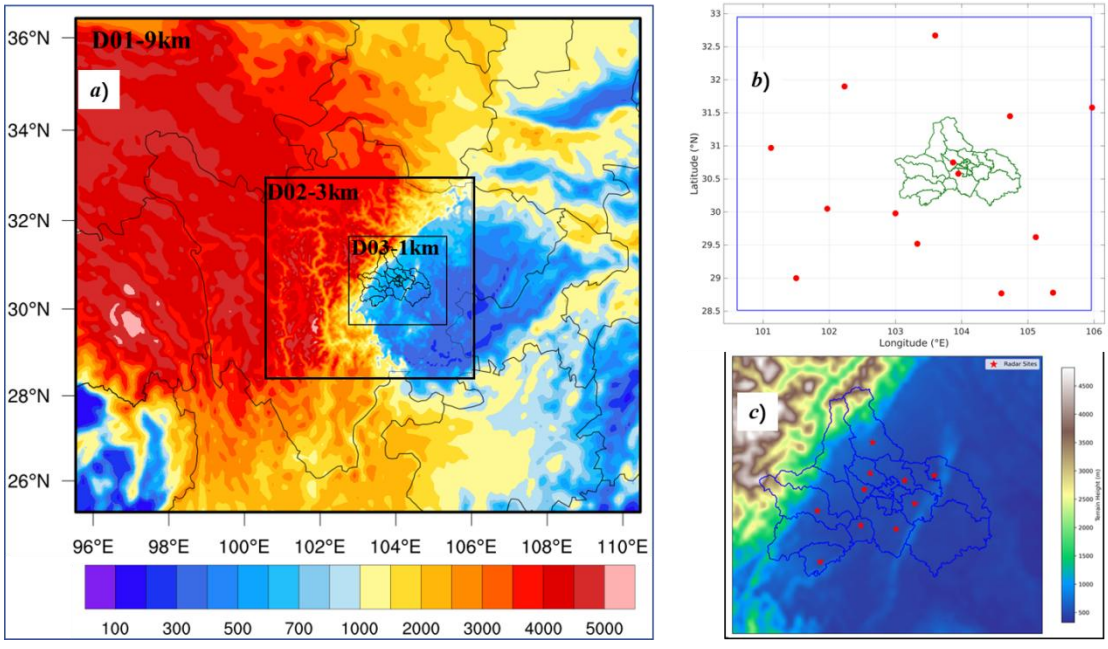


Figure 10: The configurations of *a*) 3-layer nesting model domain at 9, 3 and 1 km horizontal resolutions for the West China (D01), Central Sichuan Province (D02) and Chengdu City (D03), *b*) conventional observation stations in D02, and *c*) Lidar stations deployed in Chengdu City D03.

4.1.2 A multi-scale WRF DA system

Same as the CM2-CDA described in **Sect. 3.1**, a multiscale high-efficiency approximate EnKF (MSHea-EnKF) (Yu et al., 2019) data assimilation algorithm is used in this multi-layer nesting WRF to carry out multiscale observational information extraction (Wang et al., 2024). To do that, we conduct the large scale "observational" constraint in D01 using ERA5's wind and temperature, local data constraint in D02 and D03 using the conventional observations shown in **Fig. 10*b*** (in D02) as well as additional Lidar observations shown in **Fig. 10*c*** (in D03). In this study, once the H_{f2p}MDA-WRF$_{DA}$ system (will be described in the next section) is set, we use it to conduct parallel assimilation experiments in the 1 km resolution Chengdu City core domain by the classic MSHea-EnKF and VAE LDA to test validation of the H_{f2p}MDA-WRF$_{DA}$.

4.2 Design of H_{f2p}MDA-WRF$_{DA}$

4.2.1 Code modification and recompiling of WRF and its multiscale DA system

Unlike the CM2 model which is highly modularized, the WRF model uses a recursive function called *integrate* to perform downscaling simulations by inputting differently-defined domain as its argument. Under this circumstance, we adopt a strategy in which the WRF main program is changed to a module containing subroutine *wrf* and the whole model and its DA

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

codes are packed as an entity that is used by *wrfda_plug.F90*. In this case, the *wrfda_plug.F90* only defines a subroutine called *call_wrf_main* which calls the subroutine *wrf* and defines an *external python_foo* before the subroutine call. The *python_foo* is passed in by the *call_wrf_main* as an argument from the PMC that will be described in **Sect. 4.2.2**. The external *python_foo* is a trained LDA algorithm which is used by *callback_python.F90* where a *python_da* interface is defined as a Fortran subroutine which is public to any Fortran module if applicable.

Once the preparation described above is done, we follow the procedures described in **Sects. 2.2.1 and 2.2.2** to complete environment setting and Fortran codes compiling. At this point, the signature file *wrfda.pyf* and the "shared object" *wrfda.cpython-311-x86_64-linux-gnu.so* are ready for being imported to PMC.

4.2.2 The PMC of H_{f2p}MDA-WRF$_{DA}$ and its execution

In this case, the PMC *wrfda_main.py* is very concise. After importing necessary Fortran and Python modules including some interfaces and utilities for usage of LDA, the *wrfda_main.py* directly makes a Python call to *call_wrf_main* as *wrfda.wrfda_plug.call_wrf_main*(*transpond*). Here the *transpond* is the function that will become external *python_foo* in the subroutine *call_wrf_main* by argument passing. Following the procedure described in **Sect. 2.3.2**, we can execute the H_{f2p}MDA-WRF$_{DA}$ system.

4.3 Preliminary results of H_{f2p}MDA-WRF$_{DA}$

4.3.1 The VAE reconstruction of WRF D03

We use ERA5 dataset to create boundary and initial conditions and conduct 3-layer nesting downscaling simulations as described in **Sect. 4.1.1**. Each case is run for 36 hours with hourly output in D03. We discard the first 12 hours as downscaling spinup and connect all data in last 24 hours of these cases in 2022-2024 to form a 3-year hourly dataset (totally 8760 samples in time) for the VAE reconstruction training as shown in **Fig. 11**. It's worth to mention that in this case, pursuing 1 km resolution high-precision expression, we particularly concern local small scale information, and therefore we set a relatively large latent_dim (196608) with only a reduction rate of roughly 25%.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

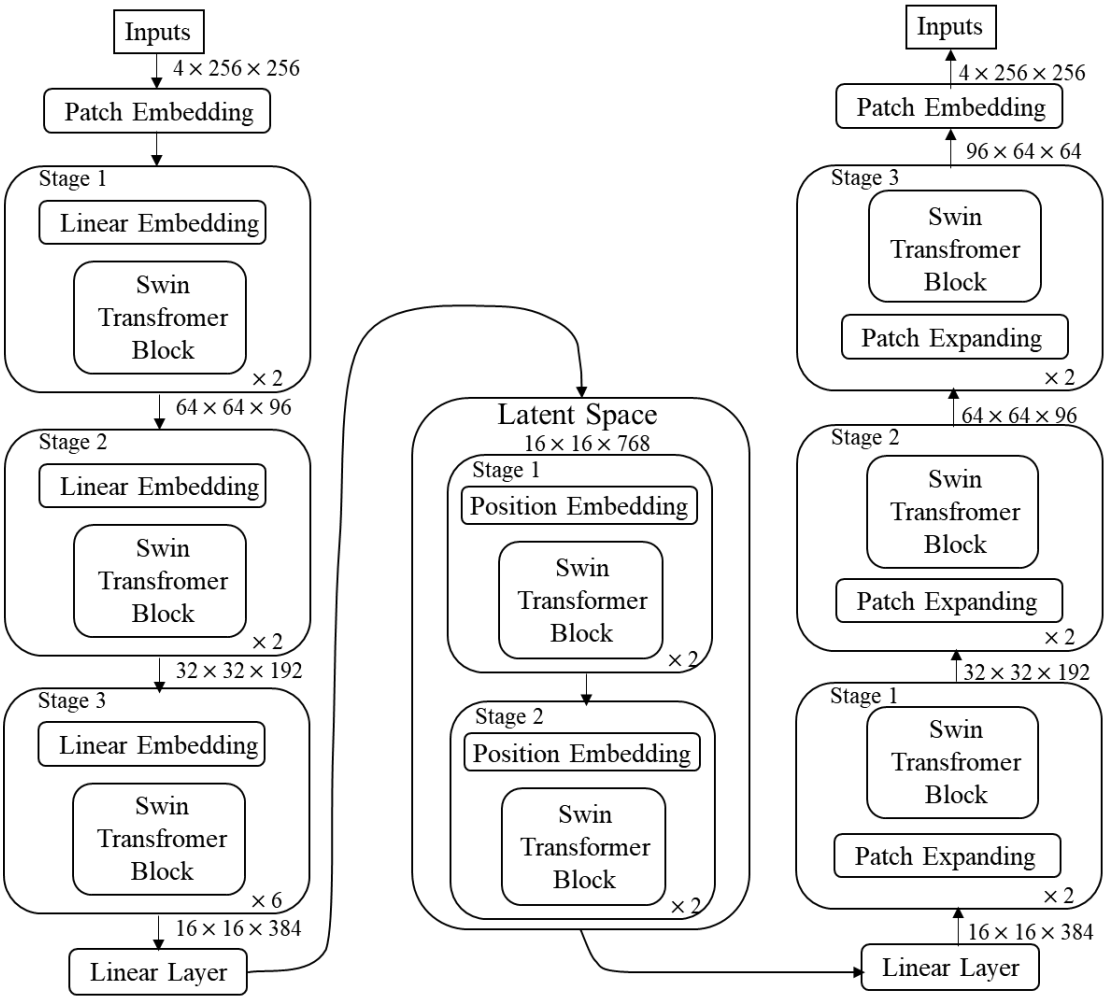


Figure 11: Same as Fig. 6 but for WRF DA of Hf2pMDA-WRF$_{DA}$ in D03. In this case, we set the latent_dim as 196608.

In this test case, we only assimilate the surface observations as shown in **Figs. 10*b-c***. We first show the reconstruction accuracy. As examples, the reconstructed surface wind (u_1, v_1), temperature (T_1) and moisture (q_1) fields at 00UTC on 23 June 2025 are shown in **Figs. 12 and 13**, which are randomly drawn from a 1-month validation dataset of June 20 - July 20, 2025, for which we will conduct parallel MSHea-EnKF and LDA experiments using H_{f2p}MDA-WRF$_{DA}$ in D03. From **Figs. 12 and 13**, we can see that the VAE encoder and decoder reconstructed fields (**Figs. 12*c-d,* 13*c-d***) represent these physical variables (**Figs. 12*a-b,* 13*a-b***) very accurately, with a nearly-perfect SSIM (Structure Similarity Index Measure ≈ 0) (**Figs. 12*e-f,* 13*e-f***). Note that the shown quantity is normalized values by the difference of maximum and minimum of corresponding physical variable.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

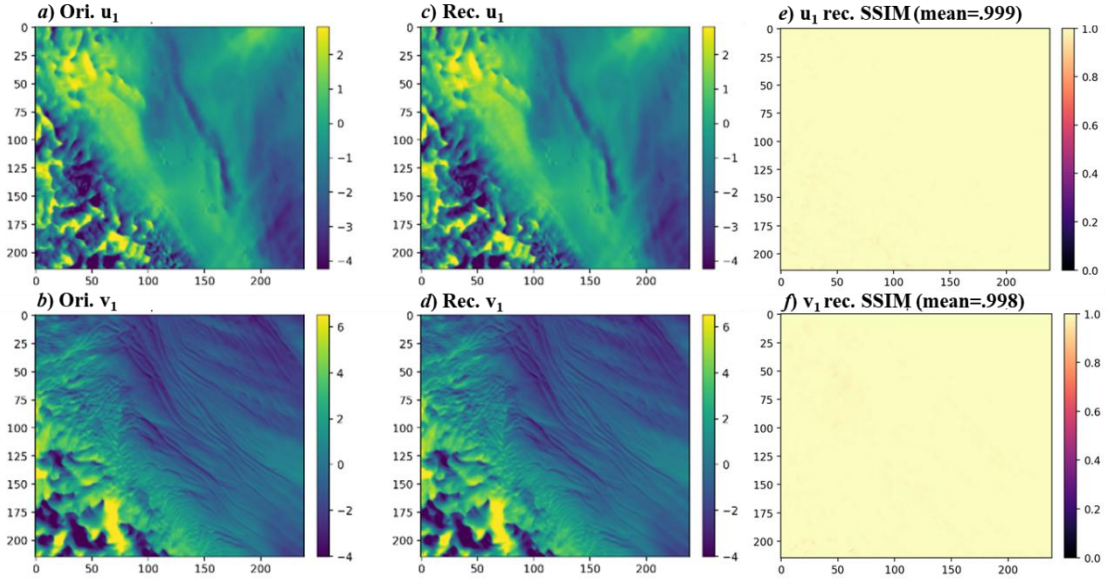


Figure 12: The typical VAE reconstruction accuracy for surface wind *ace*) u-component u_1, *bdf*) v-component v_1. The time slide is at 00UTC on 23 June 2025, randomly taken from a 1-month validation dataset for 20 June - 20 July, 2025. The leftmost panels *a-b* are the input *x*, the middle panels *c-d* are the reconstructed $\tilde{x}$ through decoding from *z*, the rightmost panels e-*f* are the Structure Similarity Index Measure (SSIM) for the corresponding reconstruction.

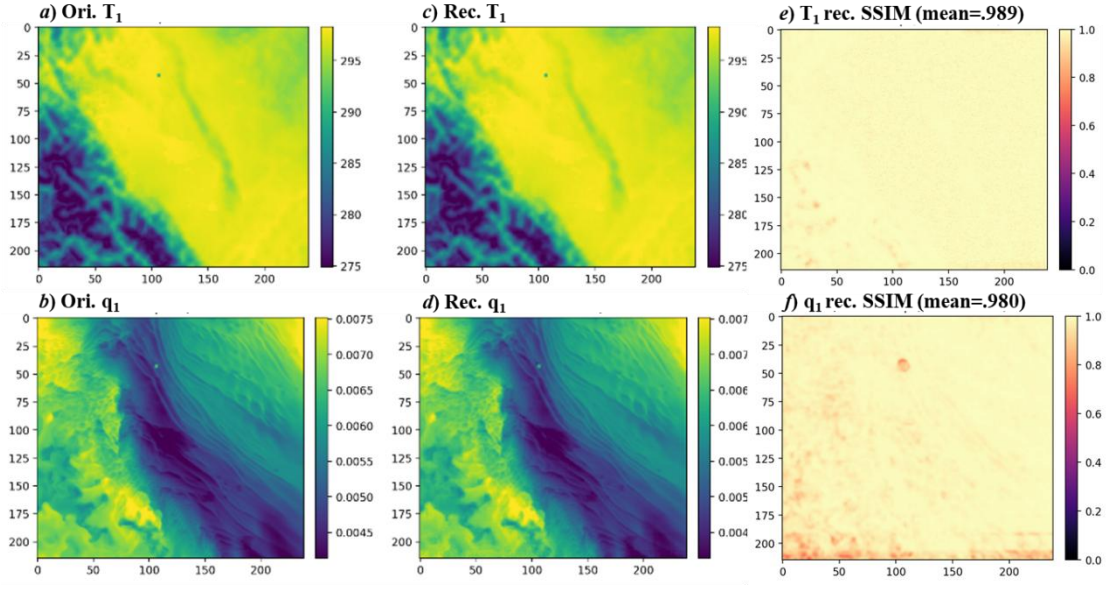


Figure 13: Same as Fig. 12 but for surface *ace*) temperature T_1, and *bdf*) moisture q_1.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

4.3.2 The LDA results in D03

We insert the VAE encoder and decoder trained in **Sect. 4.3.1** into the H_{f2p}MDA-WRF_{DA} system designed in **Sect. 4.2** and employ the latent space minimization described in **Sect. 3.3.2** to replace the D03 DA procedure. In this way, we can perform DA experiments with classic multiscale MSHea-EnKF and new VAE-LDA within the H_{f2p}MDA-WRF_{DA} framework in a parallel fashion. We run the system from 00UTC 21 June 2025 to 00UTC July 21 2025 with initial and boundary conditions remapped from ERA5 reanalysis, in which the WRF model with 9/3 (1) km resolutions in D01/D02 (D03) assimilate conventional surface station (Lidar) observations shown in **Figs. 10*b-c***. We first show the distribution of analysis increments at the first step assimilation in **Fig. 14**. We see very local u_1 and v_1 adjustments that distribute around Lidar locations (**Figs. 14*a-b***), and although the Lidar only provides wind measurements in this case, the temperature and moisture also get consistent adjustments (**Figs. 14*c-d***) under this VAE-LDA multivariate nonlinear assimilation framework.

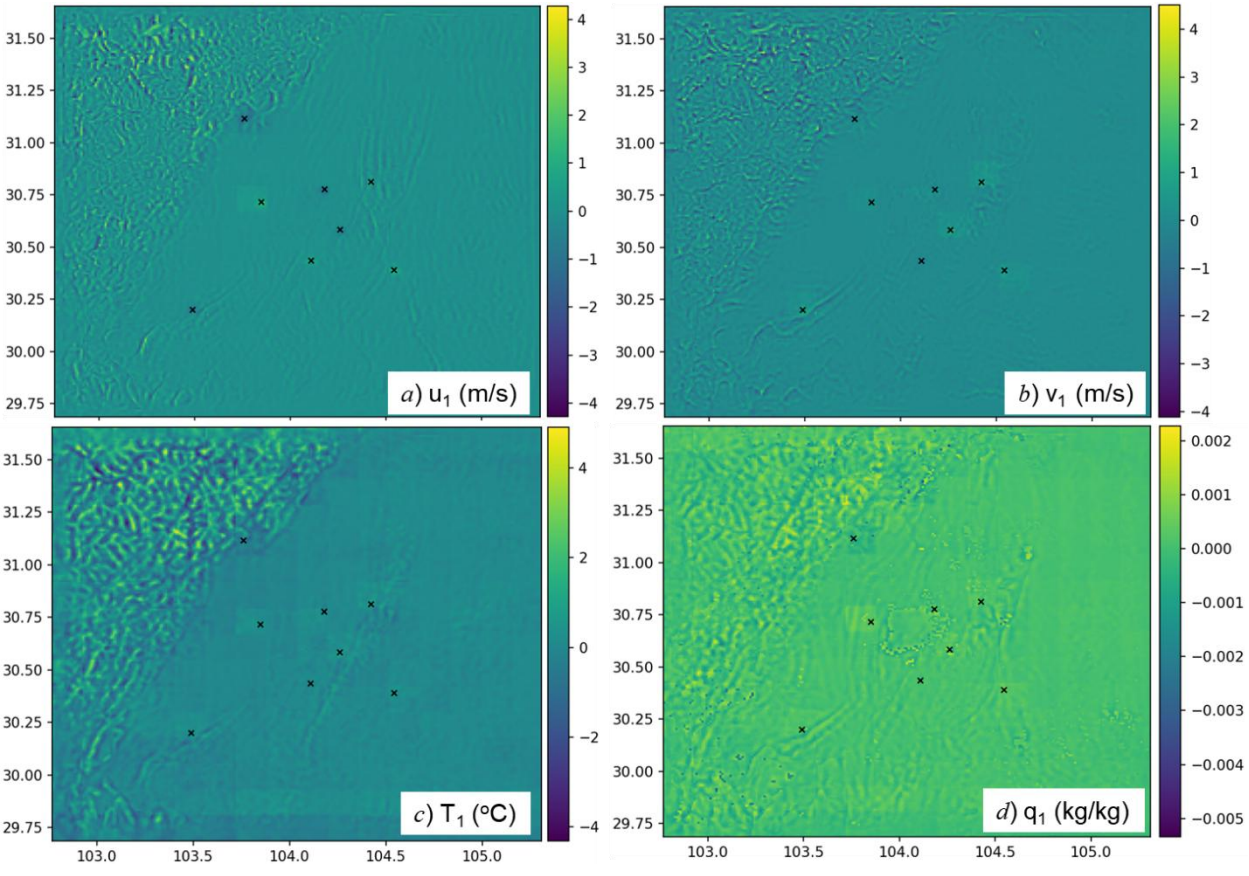


Figure 14: The distributions of analysis increments (the analysis minus the first guess from background) of the first model level *a*) u-component (u_1) and *b*) v-component v_1 and *c*) temperature T_1 as well as *d*) specific humidity q_1 at the minimization convergent point of the first assimilation step in D03 LDA of H_{f2p}MDA-WRF_{DA}. The H_{f2p}MDA-WRF_{DA} system is run from 00UTC 21 June

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

2025 to 00UTC July 21 2025 with initial and boundary conditions remapped from ERA5 reanalysis. The black asterisks denote the locations of 8 Lidars.

Then, we present the time series of RMSEs of the first model level (near surface) u-component (called u_1) and v-component (called v_1) produced by the classic multiscale DA (denoted by MDA) and new VAE-LDA (denoted by LDA) in **Fig. 15**. Within the H_{f2p}MDA-WRF$_{DA}$ it is very convenient to perform traditional DA and newly-developed AI DA in a parallel fashion. We see that except for the period of 06UTC-10UTC 21 June, in most of the time, both the MDA and LDA reduce the RMSEs from the CTL. However, we also see that the VAE-LDA gives equivalent results of classic MDA. Since the model version used in this study does still not resolve turbulence-featured extreme events (1 km horizontal resolution and 100 m vertical resolution in planetary boundary layer in this case), when one of 8 Lidars measures out a local gust event at 06UTC-10UTC 21 June, either MDA or LDA does not capture this event. This leads to large RMSEs during this period for all of CTL, MDA and LDA. Follow studies will first detect the sensitivities of these DA schemes on model horizontal and vertical resolutions, and then perform deep research on VAE-LDA to get optimal results, serving for low-altitude economy environment safety insurance.

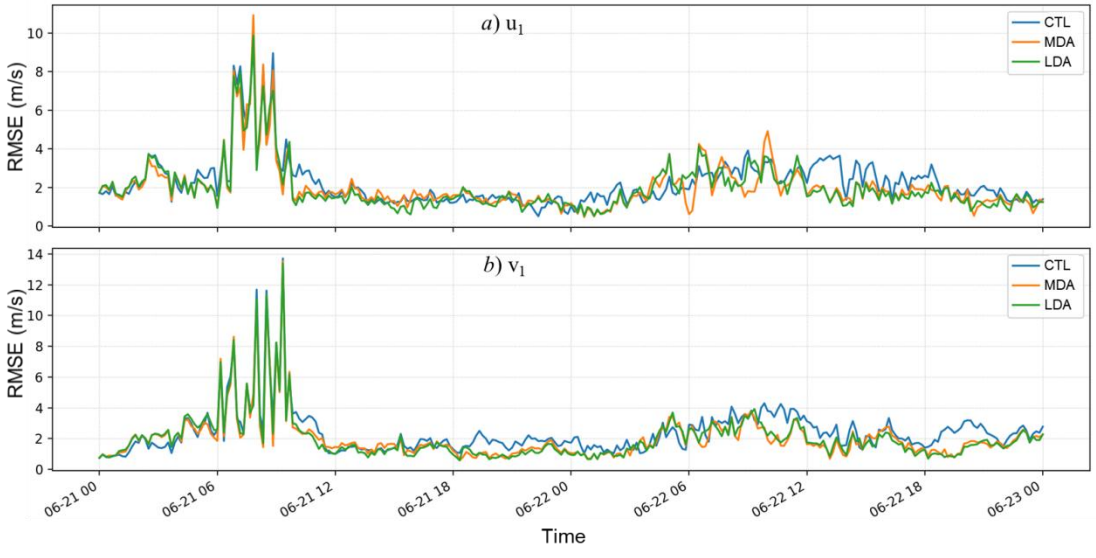


Figure 15: The time series of root mean squared errors (RMSEs) of the first model level u-component (u_1) and v-component v_1 in free model control simulation (CTL) (blue), and multiscale high-efficient approximate filter (MSHea-EnKF) data assimilation (MDA, brown) as well as VAE latent data assimilation (LDA, green) produced by Hf2pMDA-WRF$_{DA}$ in the D03 domain. Both MDA and LDA experiments are conducted under the configuration of multi-layer nested downscaling shown in Fig. 10*a* in which the WRF model with 9/3 (1) km resolutions in D01/D02 (D03) assimilate conventional surface station (Lidar) observations shown in Figs. 10*b-c*. What are shown here are the results of the first two days in the one month test period of 00UTC 21 June 2025 to 00UTC July 21 2025.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

5 Summary and discussions

Based on the f2py protocol, a Fortran-Python hybrid programming infrastructure platform called H_{f2p}MDA has been developed for deep incorporation of AI and scientific modeling and data assimilation. In this H_{f2p}MDA framework, any Python-coded (Fortran-coded) AI (scientific) algorithm (scheme) can be used by a scientific modeling and data assimilation module (machine learning procedure). As two typical application examples of H_{f2p}MDA, the implementations with a coupled global climate model CM2 with modularized Fortran structures and a multi-layer nesting downscaling weather model WRF featured as recursive Fortran integration, have been presented in details. The CM2's (WRF's) H_{f2p}MDA implementation establishes a nonlinear strongly-coupled (high-precision) climate (weather) data assimilation system called H_{f2p}MDA-$CM2_{CDA}$ (H_{f2p}MDA-WRF_{DA}). Test results of H_{f2p}MDA-$CM2_{CDA}$ (H_{f2p}MDA-WRF_{DA}) show that while the H_{f2p}MDA conveniently inserts an AI latent DA algorithm into the $CM2_{CDA}$ (WRF_{DA}) system and fulfills global strongly-CDA (regional high-precision DA), it readily improves the assimilation quality from a traditional DA method. Except for deeply evaluating scientific values in H_{f2p}MDA-$CM2_{CDA}$ and H_{f2p}MDA-WRF_{DA}, follow-up studies also include incorporating more advanced AI DA algorithms into H_{f2p}MDA, for example, the generative assimilation and prediction (GAP) (Yang et al., 2025) based on image recognition technology.

More generally, the H_{f2p}MDA designed in this study is a high-efficient infrastructure platform for deep-incorporation of AI and science since it only requires a minimum change on legacy Fortran codes of sciences that have a long persistently-developing history. Following the procedure of H_{f2p}MDA, on the one hand, it is feasible to incorporate data-driven machine deep learning algorithms to improve scientific modeling. For example, there are urgent demands of new parameterization schemes for high-resolution (HR) Earth system model (e,g. Chang et al., 2020) or even resolving the challenging problems of HR Earth system modeling (e.g. Gou et al., 2025). On the other hand, the rich achievements on scientific modeling from long time development over more than a half century since 1950s can directly advance development of science-guided AI algorithms (e.g. Hao et al., 2025). In that sense, while the H_{f2p}MDA has a very wide scope of applications, it requires more efforts to optimize its structure and deepen its infrastructure development so that it can be more conveniently plugged in any application scenario.

Code and data availability

The ERA5 dataset (Hersbach et al., 2020) can be obtained from https://doi.org/10.24381/cds.adbb2d47.

The OISST v2.1 sea surface temperature dataset (Huang et al., 2021) can be obtained from https://www.ncei.noaa.gov/products/optimum-interpolation-sst.

The CM2.1 model (Delworth et al., 2006a) can be obtained from https://github.com/mom-ocean/MOM5 and the CM2.1 model version with DA modules is also archived on Zenodo (https://doi.org/10.5281/zenodo.18883209; Delworth et al., 2006b).

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

The Weather Research and Forecasting model version 3.7.1 (WRF v3.7.1; Skamarock et al., 2008) can be obtained from https://www2.mmm.ucar.edu/wrf/users/download/get_source.html.

The model code for Hf2pMDA-CM2CDA and Hf2pMDA-WRFDA developed in this study is archived on Zenodo at https://doi.org/10.5281/zenodo.18800167 (Zhu et al., 2026b). The observational datasets used in the experiments are available at https://doi.org/10.5281/zenodo.18799861 (Zhu et al., 2026a).

Author contributions

XZ and ZLin are co-first authors who contributed equally to this work. Both of them conduct all test work on Python-Fortran hybrid programming as well as VAE and latent minimization and join the paper edits. SZ is the corresponding author who proposes the idea, designs and organizes the research project, writes and edits the paper. ZLu participates the test experiments on WRF multiscale data assimilation and joins the discussions. SW and XH manage the Leice Transient team and OUC team respectively, and join the discussions and make comments on the research work. ZX participates the test experiments on WRF VAE training experiments and joins the discussions. ZR, JL, JX, YG, RH, XY and ML contribute to this research by joining the discussions and making comments on the research and manuscript.

Competing interests

The authors declare that they have no conflict of interest.

Disclaimer

Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements

This work is supported by the Science and Technology Innovation Project of Laoshan Laboratory under Grants (Nos. LSKJ202300400, LSKJ202300401-03, LSKJ202202200, LSKJ202202201-04), the National Natural Science Foundation of China (42361164616), the Shandong Province's "Taishan" Scientist Program (ts201712017).

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Review statement

The review statement will be added by Copernicus Publications listing the handling editor as well as all contributing referees according to their status anonymous or identified.

References

Anderson, J. L.: An Ensemble Adjustment Kalman Filter for Data Assimilation, Mon. Wea. Rev., 129, 2884–2903, https://doi.org/10.1175/1520-0493(2001)129%253C2884:AEAKFF%253E2.0.CO;2, 2001.

Atkinson, J., Elafrou, A., Kasoar, E., Wallwork, J. G., Meltzer, T., Clifford, S., Orchard, D., and Edsall, C.: FTorch: a library for coupling PyTorch models to Fortran, JOSS, 10, 7602, https://doi.org/10.21105/joss.07602, 2025.

Benjamin, S. G., Brown, J. M., Brunet, G., Lynch, P., Saito, K., and Schlatter, T. W.: 100 Years of Progress in Forecasting and NWP Applications, Meteorological Monographs, 59, 13.1-13.67, https://doi.org/10.1175/AMSMONOGRAPHS-D-18-0020.1, 2019.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., and Tian, Q.: Accurate medium-range global weather forecasting with 3D neural networks, Nature, 619, 533–538, https://doi.org/10.1038/s41586-023-06185-3, 2023.

Ftorch - A library for coupling (py)torch machine learning models to Fortran.: https://github.com/Cambridge-ICCS/FTorch.

Cheng, S., Prentice, I. C., Huang, Y., Jin, Y., Guo, Y.-K., and Arcucci, R.: Data-driven surrogate model with latent data assimilation: Application to wildfire forecasting, Journal of Computational Physics, 464, 111302, https://doi.org/10.1016/j.jcp.2022.111302, 2022.

Cheng, S., Zhuang, Y., Kahouadji, L., Liu, C., Chen, J., Matar, O. K., and Arcucci, R.: Multi-domain encoder–decoder neural networks for latent data assimilation in dynamical systems, Computer Methods in Applied Mechanics and Engineering, 430, 117201, https://doi.org/10.1016/j.cma.2024.117201, 2024.

Clough, S. A., Shephard, M. W., Mlawer, E. J., Delamere, J. S., Iacono, M. J., Cady-Pereira, K., Boukabara, S., and Brown, P. D.: Atmospheric radiative transfer modeling: a summary of the AER codes, Journal of Quantitative Spectroscopy and Radiative Transfer, 91, 233–244, https://doi.org/10.1016/j.jqsrt.2004.05.058, 2005.

Danabasoglu, G., Lamarque, J. -F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., Emmons, L. K., Fasullo, J., Garcia, R., Gettelman, A., Hannay, C., Holland, M. M., Large, W. G., Lauritzen, P. H., Lawrence, D. M., Lenaerts, J. T. M., Lindsay, K., Lipscomb, W. H., Mills, M. J., Neale, R., Oleson, K. W., Otto-Bliesner, B., Phillips, A. S., Sacks, W., Tilmes, S., Van Kampenhout, L., Vertenstein, M., Bertini, A., Dennis, J., Deser, C., Fischer, C., Fox-Kemper, B., Kay, J. E., Kinnison, D., Kushner, P. J., Larson, V. E., Long, M. C., Mickelson, S., Moore, J. K., Nienhouse, E., Polvani, L., Rasch, P. J., and Strand, W. G.: The Community Earth System Model Version 2 (CESM2), J Adv Model Earth Syst, 12, e2019MS001916, https://doi.org/10.1029/2019MS001916, 2020.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

De Rosnay, P., Browne, P., De Boisséson, E., Fairbairn, D., Hirahara, Y., Ochi, K., Schepers, D., Weston, P., Zuo, H., Alonso-Balmaseda, M., Balsamo, G., Bonavita, M., Borman, N., Brown, A., Chrust, M., Dahoui, M., Chiara, G., English, S., Geer, A., Healy, S., Hersbach, H., Laloyaux, P., Magnusson, L., Massart, S., McNally, A., Pappenberger, F., and Rabier, F.: Coupled data assimilation at ECMWF: current status, challenges and future developments, Quart J Royal Meteoro Soc, 148, 2672–2702, https://doi.org/10.1002/qj.4330, 2022.

Delworth, T. L., Broccoli, A. J., Rosati, A., Stouffer, R. J., Balaji, V., Beesley, J. A., Cooke, W. F., Dixon, K. W., Dunne, J., Dunne, K. A., Durachta, J. W., Findell, K. L., Ginoux, P., Gnanadesikan, A., Gordon, C. T., Griffies, S. M., Gudgel, R., Harrison, M. J., Held, I. M., Hemler, R. S., Horowitz, L. W., Klein, S. A., Knutson, T. R., Kushner, P. J., Langenhorst, A. R., Lee, H.-C., Lin, S.-J., Lu, J., Malyshev, S. L., Milly, P. C. D., Ramaswamy, V., Russell, J., Schwarzkopf, M. D., Shevliakova, E., Sirutis, J. J., Spelman, M. J., Stern, W. F., Winton, M., Wittenberg, A. T., Wyman, B., Zeng, F., and Zhang, R.: GFDL's CM2 Global Coupled Climate Models. Part I: Formulation and Simulation Characteristics, Journal of Climate, 19, 643–674, https://doi.org/10.1175/JCLI3629.1, 2006a.

Delworth, T. L., Broccoli, A. J., Rosati, A., Stouffer, R. J., Balaji, V., Cooke, W. F., Dixon, K. W., Dunne, J., Dunne, K. A., Durachta, J. W., Findell, K. L., Ginoux, P., Gnanadesikan, A., Gordon, C. T., Griffies, S. M., Gudgel, R., Harrison, M. J., Held, I. M., Hemler, R. S., Langenhorst, A. R., Lee, H.-C., Lin, S.-J., Lu, J., Malyshev, S. L., Milly, P. C. D., Ramaswamy, V., Russell, J., Schwarzkopf, M. D., Shevliakova, E., Sirutis, J. J., Spelman, M. J., Stern, W. F., Winton, M., Wittenberg, A. T., Wyman, B., Zeng, F., and Zhang, R.: CM2.1 Model, , https://doi.org/10.5281/ZENODO.18883209, 2006b.

Diao, X., Stössel, A., Chang, P., Danabasoglu, G., Yeager, S. G., Gopal, A., Wang, H., and Zhang, S.: On the Intermittent Occurrence of Open-Ocean Polynyas in a Multi-Century High-Resolution Preindustrial Earth System Model Simulation, JGR Oceans, 127, e2021JC017672, https://doi.org/10.1029/2021JC017672, 2022.

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization, Geosci. Model Dev., 9, 1937–1958, https://doi.org/10.5194/gmd-9-1937-2016, 2016.

Fan, H., Liu, Y., Huo, Z., Liu, Y., Shi, Y., and Li, Y.: A Novel Latent Space Data Assimilation Framework with Autoencoder-Observation to Latent Space (AE-O2L) Network. Part I: The Observation-Only Analysis Method, Monthly Weather Review, 153, 1335–1348, https://doi.org/10.1175/MWR-D-24-0057.1, 2025.

Faroughi, S. A., Pawar, N. M., Fernandes, C., Raissi, M., Das, S., Kalantari, N. K., and Kourosh Mahjour, S.: Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks and Operators in Scientific Computing: Fluid and Solid Mechanics, Journal of Computing and Information Science in Engineering, 24, 040802, https://doi.org/10.1115/1.4064449, 2024.

Feng, J., Zhang, Y., Cheng, Q., and Tsou, J. Y.: Pan-Arctic melt pond fraction trend, variability, and contribution to sea ice changes, Global and Planetary Change, 217, 103932, https://doi.org/10.1016/j.gloplacha.2022.103932, 2022.

Gnanadesikan, A., Dixon, K. W., Griffies, S. M., Balaji, V., Barreiro, M., Beesley, J. A., Cooke, W. F., Delworth, T. L., Gerdes, R., Harrison, M. J., Held, I. M., Hurlin, W. J., Lee, H.-C., Liang, Z., Nong, G., Pacanowski, R. C., Rosati, A.,

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Russell, J., Samuels, B. L., Song, Q., Spelman, M. J., Stouffer, R. J., Sweeney, C. O., Vecchi, G., Winton, M., Wittenberg, A. T., Zeng, F., Zhang, R., and Dunne, J. P.: GFDL's CM2 Global Coupled Climate Models. Part II: The Baseline Ocean Simulation, Journal of Climate, 19, 675–697, https://doi.org/10.1175/JCLI3630.1, 2006.

Haarsma, R. J., Roberts, M. J., Vidale, P. L., Senior, C. A., Bellucci, A., Bao, Q., Chang, P., Corti, S., Fučkar, N. S., Guemas, V., Von Hardenberg, J., Hazeleger, W., Kodama, C., Koenigk, T., Leung, L. R., Lu, J., Luo, J.-J., Mao, J., Mizielinski, M. S., Mizuta, R., Nobre, P., Satoh, M., Scoccimarro, E., Semmler, T., Small, J., and Von Storch, J.-S.: High Resolution Model Intercomparison Project (HighResMIP v1.0) for CMIP6, Geosci. Model Dev., 9, 4185–4208, https://doi.org/10.5194/gmd-9-4185-2016, 2016.

Häfner, D., Jacobsen, R. L., Eden, C., Kristensen, M. R. B., Jochum, M., Nuterman, R., and Vinter, B.: Veros v0.1 – a fast and versatile ocean simulator in pure Python, Geosci. Model Dev., 11, 3299–3312, https://doi.org/10.5194/gmd-11-3299-2018, 2018.

Ham, Y.-G., Kim, J.-H., and Luo, J.-J.: Deep learning for multi-year ENSO forecasts, Nature, 573, 568–572, https://doi.org/10.1038/s41586-019-1559-7, 2019.

Hao, R., Zhao, Y., Zhang, S., Wang, G., and Deng, X.: Interpretable Cross-Sphere Multiscale Deep Learning Predicts ENSO Skilfully Beyond 2 Years, https://doi.org/10.48550/ARXIV.2503.21211, 2025.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E.: Array programming with NumPy, Nature, 585, 357–362, https://doi.org/10.1038/s41586-020-2649-2, 2020.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., De Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.: The ERA5 global reanalysis, Quart J Royal Meteoro Soc, 146, 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Heuer, H., Schwabe, M., Gentine, P., Giorgetta, M. A., and Eyring, V.: Interpretable Multiscale Machine Learning-Based Parameterizations of Convection for ICON, J Adv Model Earth Syst, 16, e2024MS004398, https://doi.org/10.1029/2024MS004398, 2024.

Hong, S.-Y., Dudhia, J., and Chen, S.-H.: A Revised Approach to Ice Microphysical Processes for the Bulk Parameterization of Clouds and Precipitation, Mon. Wea. Rev., 132, 103–120, https://doi.org/10.1175/1520-0493(2004)132%253C0103:ARATIM%253E2.0.CO;2, 2004.

Hopfield, J. J.: Neural networks and physical systems with emergent collective computational abilities., Proc. Natl. Acad. Sci. U.S.A., 79, 2554–2558, https://doi.org/10.1073/pnas.79.8.2554, 1982.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Hu, X., Klein, P. M., and Xue, M.: Evaluation of the updated YSU planetary boundary layer scheme within WRF for wind resource and air quality assessments, JGR Atmospheres, 118, https://doi.org/10.1002/jgrd.50823, 2013.

Huang, B., Liu, C., Banzon, V., Freeman, E., Graham, G., Hankins, B., Smith, T., and Zhang, H.-M.: Improvements of the Daily Optimum Interpolation Sea Surface Temperature (DOISST) Version 2.1, Journal of Climate, 34, 2923–2939, https://doi.org/10.1175/JCLI-D-20-0166.1, 2021.

Jia, W. and Zhang, X.: The role of the planetary boundary layer parameterization schemes on the meteorological and aerosol pollution simulations: A review, Atmospheric Research, 239, 104890, https://doi.org/10.1016/j.atmosres.2020.104890, 2020.

Joule, J. P.: On the calorific effects of magneto-electricity, and on the mechanical value of heat, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 23, 263–276, https://doi.org/10.1080/14786444308644730, 1843.

Kadivar, M., Tormey, D., and McGranaghan, G.: A review on turbulent flow over rough surfaces: Fundamentals and theories, International Journal of Thermofluids, 10, 100077, https://doi.org/10.1016/j.ijft.2021.100077, 2021.

Kain, J. S.: The Kain–Fritsch Convective Parameterization: An Update, J. Appl. Meteor., 43, 170–181, https://doi.org/10.1175/1520-0450(2004)043%253C0170:TKCPAU%253E2.0.CO;2, 2004.

Kay, J. E., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G., Arblaster, J. M., Bates, S. C., Danabasoglu, G., Edwards, J., Holland, M., Kushner, P., Lamarque, J.-F., Lawrence, D., Lindsay, K., Middleton, A., Munoz, E., Neale, R., Oleson, K., Polvani, L., and Vertenstein, M.: The Community Earth System Model (CESM) Large Ensemble Project: A Community Resource for Studying Climate Change in the Presence of Internal Climate Variability, Bulletin of the American Meteorological Society, 96, 1333–1349, https://doi.org/10.1175/BAMS-D-13-00255.1, 2015.

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., Klöwer, M., Lottes, J., Rasp, S., Düben, P., Hatfield, S., Battaglia, P., Sanchez-Gonzalez, A., Willson, M., Brenner, M. P., and Hoyer, S.: Neural general circulation models for weather and climate, Nature, 632, 1060–1066, https://doi.org/10.1038/s41586-024-07744-y, 2024.

Lin, Z.: A High-Efficiency Paleoclimate Reconstruction Approach by Online Strongly-Coupled Data Assimilation in Latent Space, Climate Dynamics, under review, 2025.

Lu, L., Zhang, S., Jiang, Y., Yu, X., Li, M., Chen, Y., Chang, P., Danabasoglu, G., Liu, Z., Zhu, C., Lin, X., and Wu, L.: An Improved Coupled Data Assimilation System with a CGCM Using Multi-Time-Scale High-Efficiency EnOI-Like Filtering, Journal of Climate, 36, 6045–6067, https://doi.org/10.1175/JCLI-D-22-0558.1, 2023.

Marotzke, J.: From theory to RAPID AMOC observations: a personal voyage of discovery, Phil. Trans. R. Soc. A., 381, 20220192, https://doi.org/10.1098/rsta.2022.0192, 2023.

Mayer, J. R.: Bemerkungen über die Kräfte der unbelebten Natur, Justus Liebigs Ann. Chem., 42, 233–240, https://doi.org/10.1002/jlac.18420420212, 1842.

Mouallem, J., Gao, K., Reichl, B. G., Chilutti, L., Harris, L., Benson, R., Zadeh, N., Chen, J., Chen, J.-H., and Zhang, C.: Development of a high-resolution coupled SHiELD-MOM6 model – Part 1: Model overview, coupling technique, and validation in a regional setup, Geosci. Model Dev., 18, 6461–6478, https://doi.org/10.5194/gmd-18-6461-2025, 2025.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Müller, S., Lange, M., Fischer, T., König, S., Kelbling, M., Leal Rojas, J. J., and Thober, S.: FINAM is not a model (v1.0): a new Python-based model coupling framework, Geosci. Model Dev., 18, 4483–4498, https://doi.org/10.5194/gmd-18-4483-2025, 2025.

Newton, I.: Philosophiae naturalis principia mathematica, Jussu Societatis Regiae ac Typis Josephi Streater. Prostat apud plures bibliopolas, https://doi.org/10.5479/sil.52126.39088015628399, 1687.

Perlwitz, J., Knutson, T., Kossin, J. P., and LeGrande, A. N.: Ch. 5: Large-Scale Circulation and Climate Variability. Climate Science Special Report: Fourth National Climate Assessment, Volume I, U.S. Global Change Research Program, https://doi.org/10.7930/J0RV0KVQ, 2017.

Qu, M., Lei, R., Liu, Y., and Li, N.: Arctic Sea ice leads detected using sentinel-1B SAR image and their responses to atmosphere circulation and sea ice dynamics, Remote Sensing of Environment, 308, 114193, https://doi.org/10.1016/j.rse.2024.114193, 2024.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning representations by back-propagating errors, Nature, 323, 533–536, https://doi.org/10.1038/323533a0, 1986.

Shu, R., Gao, Y., Wu, H., Gou, R., Wang, K., Xiang, Y., Xu, F., Wen, Q., and Huang, X.: Ocean-E2E: Hybrid Physics-Based and Data-Driven Global Forecasting of Extreme Marine Heatwaves with End-to-End Neural Assimilation, https://doi.org/10.48550/arXiv.2505.22071, 15 August 2025.

Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Wang, W., Huang, X.-Y., and Duda, M.: A Description of the Advanced Research WRF Version 3, UCAR/NCAR, https://doi.org/10.5065/D68S4MVH, 2008.

Solera-Rico, A., Sanmiguel Vila, C., Gómez-López, M., Wang, Y., Almashjary, A., Dawson, S. T. M., and Vinuesa, R.: β-Variational autoencoders and transformers for reduced-order modelling of fluid flows, Nat Commun, 15, 1361, https://doi.org/10.1038/s41467-024-45578-4, 2024.

Taylor, K. E., Stouffer, R. J., and Meehl, G. A.: An Overview of CMIP5 and the Experiment Design, Bulletin of the American Meteorological Society, 93, 485–498, https://doi.org/10.1175/BAMS-D-11-00094.1, 2012.

Wang, J., Zhang, S., Yu, X., Jin, Y., Liu, X., Qi, Y., Yang, G., and Li, M.: A Numerical Study of the Impact of Topography on the July 2021 Extreme Rainfall Event in Zhengzhou, China, JGR Atmospheres, 129, e2024JD041332, https://doi.org/10.1029/2024JD041332, 2024.

Winton, M.: A Reformulated Three-Layer Sea Ice Model, J. Atmos. Oceanic Technol., 17, 525–531, https://doi.org/10.1175/1520-0426(2000)017%253C0525:ARTLSI%253E2.0.CO;2, 2000.

Yang, S., Nai, C., Liu, X., Li, W., Chao, J., Wang, J., Wang, L., Li, X., Chen, X., Lu, B., Xiao, Z., Boers, N., Yuan, H., and Pan, B.: Generative assimilation and prediction for weather and climate, https://doi.org/10.48550/ARXIV.2503.03038, 2025.

Yu, X., Zhang, S., Li, J., Lu, L., Liu, Z., Li, M., Yu, H., Han, G., Lin, X., Wu, L., and Chang, P.: A Multi-Timescale EnOI-Like High-Efficiency Approximate Filter for Coupled Model Data Assimilation, J Adv Model Earth Syst, 11, 45–63, https://doi.org/10.1029/2018MS001504, 2019.

https://doi.org/10.5194/egusphere-2025-6479
Preprint. Discussion started: 9 March 2026

Yuan, W.-X. and Guo, R.: Physics-guided multistage neural network: A physically guided network for step initial values and dispersive shock wave phenomena, Phys. Rev. E, 110, 065307, https://doi.org/10.1103/PhysRevE.110.065307, 2024.

Zhang, S. and Anderson, J. L.: Impact of spatially and temporally varying estimates of error covariance on assimilation in a simple atmospheric model, Tellus A: Dynamic Meteorology and Oceanography, 55, 126, https://doi.org/10.3402/tellusa.v55i2.12087, 2003.

Zhang, S., Harrison, M. J., Rosati, A., and Wittenberg, A.: System Design and Evaluation of Coupled Ensemble Data Assimilation for Global Oceanic Climate Studies, Monthly Weather Review, 135, 3541–3564, https://doi.org/10.1175/MWR3466.1, 2007.

Zhang, S., Xu, S., Fu, H., Wu, L., Liu, Z., Gao, Y., Zhao, C., Wan, W., Wan, L., Lu, H., Li, C., Liu, Y., Lv, X., Xie, J., Yu, Y., Gu, J., Wang, X., Zhang, Y., Ning, C., Fei, Y., Guo, X., Wang, Z., Wang, X., Wang, Z., Qu, B., Li, M., Zhao, H., Jiang, Y., Yang, G., Lu, L., Wang, H., An, H., Zhang, X., Zhang, Y., Ma, W., Yu, F., Xu, J., Lin, X., and Shen, X.: Toward Earth system modeling with resolved clouds and ocean submesoscales on heterogeneous many-core HPCs, National Science Review, 10, nwad069, https://doi.org/10.1093/nsr/nwad069, 2023.

Zhu, X., Lin, Z., Lu, Z., Zhang, S., and Wu, S.: Hf2p CM2.1_SCDA & WRF_LDA Dataset, https://doi.org/10.5281/ZENODO.18799861, 2026a.

Zhu, X., Lin, Z., Zhang, S., Lu, Z., Wu, S., Hou, X., Xiao, Z., Ren, Z., Li, J., Jing, X., Gao, Y., Hao, R., Yu, X., and Li, M.: Hf2p CM2.1_SCDA and WRF_LDA, Zenodo [software], https://doi.org/10.5281/ZENODO.18800167, 2026b.