An Automatic Differentiation Method for Surface Carbon Flux Inversion

Yichen Yao[1], Guodong Chen[1], Zhibin Wang[1]and Hao Li[1]

[1]Damo Academy, Alibaba Group, Hangzhou 311121, China

Key Points:

- A fully-differentiable atmosphere transport model is developed, which encoded the PDE into a deep learning computational graph.
- Using the framework's automatic differentiation capabilities, the carbon inversion problem can be solved.
- The proposed framework has been tested in identical-twin OSSEs and has achieved favorable performance.

Corresponding author: Zhibin Wang, zhibin.waz@alibaba-inc.com

Abstract

We attempt, for the first time, to estimate the surface CO_2 flux by the method of automatic differentiation. The atmospheric transport model is developed using a deep learning framework and is validated against standard approaches. Depends on the built-in automatic differentiation feature of the deep learning framework, the system derivatives/gradients are readily available without any extra effort. We then formulate the surface carbon flux estimation as an inverse problem using the variational approach, driven by back-propagated objective gradients. The feasibility of the automatic differentiation method is demonstrated in identical-twin observing system simulation experiments (OSSEs). The proposed framework shows favorable accuracy and great efficiency in both fully and partially observable scenarios. The present study establishes a link between machine learning frameworks and general data assimilation or inverse modeling problems, and the promising results encourage more investigations in incorporating machine learning techniques in inverse carbon modeling.

Plain Language Summary

In this paper, we use the deep learning framework for the first time to solve the carbon inversion problem in atmospheric transport, and verify the feasibility of the framework on the inversion problem. In the scheme, The atmospheric transport model pyATM is implemented using a deep learning framework, and the performance of the model is quantitatively compared with the classical transport model GEOS-Chem. Relying on the built-in automatic differentiation function of the deep learning framework, the gradient of the error to the input variable can be calculated, that is, in the inversion problem, the gradient of the observed error to the surface carbon flux is observed, and the carbon flux value is updated. The proposed framework is demonstrated in an identical twin-observing system simulation experiment (OSSE), showing good accuracy and high efficiency in both fully and partially observable situations.

1 Introduction

Carbon dioxide (CO_2) is the most important long-lived greenhouse gas, which has caused climate issues such as global warming and ocean acidification (Solomon et al., 2009; Collins et al., 2013). Since the beginning of the Industrial Revolution (1750s), the atmospheric CO_2 concentration has increased dramatically due to anthropogenic emissions, mainly from fossil fuel burning. Approximately half of the emitted CO_2 was absorbed by the ocean and land through the natural carbon cycle, while the rest remains in the atmosphere and keeps pushing the atmospheric CO_2 level to new highs (Quéré et al., 2009; Friedlingstein et al., 2019; Blunden & Arndt, 2020). Moreover, the absorbing capacity of the oceanic and terrestrial ecosystems is not guaranteed to follow the fast growth of anthropogenic emissions or even remain at its present level as global warming tends to slow down the carbon uptake of the land and ocean (Fung et al., 2005; Quéré et al., 2007). On the other hand, the effectiveness of the measures that have been taken to reduce CO_2 emissions is hard to evaluate without a good understanding of the variations of the natural carbon sources/sinks. For more accurate projections of future CO_2 levels and the associated climate impacts, it is therefore essential to accurately quantify the temporal and spatial pattern of surface CO_2 fluxes and to better understand the natural carbon cycle. (Baker et al., 2006, 2010)

The surface carbon fluxes are often estimated by either the bottom-up or the top-down approach. The former uses a combination of human activity data and inventory-specific emission factors, process-based models, or remote-sensing approaches to estimate the fluxes (Poulter et al., 2022). The latter formulates it as an inverse problem, in which statistical methods or optimization approaches are used to invert the CO_2 transport process to find the optimal surface fluxes that best match the observed atmospheric CO_2 concentra-

tion (Thompson et al., 2022). The link between carbon dioxide concentration and surface carbon fluxes is established through atmospheric transport models (Community, 2021; Grell et al., 2005; Emmons et al., 2010).

The current study focuses on top-down techniques, with ensemble Kalman filters (EnKFs) (Peters et al., 2005, 2007; Feng et al., 2009; Kang et al., 2011) and variational methods (Baker et al., 2006, 2010; Chevallier, Engelen, et al., 2009; Chevallier, Maksyutov, et al., 2009) being the most common ones. Ensemble Kalman filters use a Bayesian framework to estimate the statistically optimal state given the prior or background predictions and the observations. EnKFs only rely on the model outputs and the observations, and thus are more flexible and developer-friendly. However, a reasonable ensemble size is typically required to capture accurate statistics, which requires considerable computational resources. On the other hand, the 4-dimensional variational (4D-Var) approach minimizes the differences between the model forecast trajectory and observations distributed over a period of time. These two approaches can be made in an equivalent Bayesian form if the background errors and the observational errors are both Gaussian, though the 4D-Var tends to give better results if the assimilation time is short while the EnKF needs some spinup time. However, an adjoint model of the forward transport process is required in 4D-Var to back-propagate the gradients, which is often a considerable development endeavor (Henze et al., 2007). This paper aims to reduce the development effort by developing the atmospheric transport model within deep learning frameworks, in which the adjoint model can be easily obtained by automatic differentiation.

In terms of observations, both in situ observations and satellite retrieved data have been used to study the carbon cycle and estimate the surface carbon fluxes. The in situ data such as Total Column Carbon Observing Network (TCCON) (Toon et al., 2009) and Observation Package (ObsPack) (Masarie et al., 2014) often provide pretty accurate observations and are more sensitive to surface emissions. However, these observations are extremely sparse spatially, which poses difficulties to the global flux estimates. On the other hand, satellite-based measurements, such as those of Greenhouse Gases Observing Satellite (GOSAT) (Yokota et al., 2009a) and the Orbiting Carbon Observatory-2 (OCO-2) (Crisp et al., 2004b), often have better spatial coverage over the globe, while they are less accurate and are sparse in both space and time. In the present study, we focus on algorithmic development and evaluating the new methodology. Therefore, we use the identical twin OSSEs, in which we know the "truth", to test the proposed framework, eliminating possible uncertainties associated with the model errors and the observational errors. However, we do mimic the temporal and spatial pattern of the satellite data to ensure its applicability in real-world systems.

Despite its extraordinary success achieved in computer vision and natural language processing, deep learning has not received much attention in computational physical modeling until recently (Zhu & Zabaras, 2018; Ravuri et al., 2021; Espeholt et al., 2021; Karniadakis et al., 2021; Chen & Fidkowski, 2021; Hu et al., 2022; Pathak et al., 2022). A few attempts have been devoted to data assimilation or inverse modeling. Laloyaux et al. (2022) introduced a neural network model into the 4d-Var to correct for the bias that accumulates along the model trajectory. Frerix et al. (2021) learned an inverse observation operator to map the observational data to physical states and then reformulated the objective function in the better-behaved physics space instead of the observation space. Fablet et al. (2021) adopted an end-to-end learning approach, which embedded the joint training of the forward dynamical process and the inverse assimilation in a physics-informed approach. Encoder-decoder type architectures have also been used in data assimilation to reduce the problem dimension and computational cost (Mack et al., 2020; Peyron et al., 2021; Amendola et al., 2021). Nevertheless, these studies treat the machine learning models mostly as black boxes and the generalization is largely limited. In this paper, we directly embedded the atmospheric transport model within a deep learning framework, *i.e.*, the physics is numerically "exact" instead of data-driven, which generalizes well to different initial and boundary conditions. The main contribution of this work can be summarized as follows:

- We developed a fully-differentiable atmosphere transport model called pyATM, which encoded the partial differential equations (PDEs) solver into a computational graph based on a deep learning framework.
- Using the framework's automatic differentiation capabilities, the derivatives of the loss function with respect to the learnable parameters are obtained. When the learnable parameter is the carbon flux and the loss the function is the observation error, the carbon flux can be corrected by observations.
- The proposed framework has been tested in identical-twin OSSEs and has achieved favorable performance in both fully-observed and partially-observed systems with simulated satellite observations.

The remainder of this paper proceeds as follows. We describe in Section 2 the detailed implementations of the proposed flux inversion framework, including the forward atmospheric transport model and its backward mode, as well as the variational inversion approach. Section 3 presents the observing system simulation experiment (OSSE) and provides a comprehensive summary of the problem setup. The primary results are shown in Section 4, and Section 5 concludes the present work and discusses potential future work.

2 Method

Deep learning has made extraordinary achievements in the field of artificial intelligence. However, its application in numerical computation has not been studied until recently. Here, we use the deep learning framework to encode the numerical solving process of partial differential equations as computational graphs. First, we use the generalized hyperbolic parabolic differential equations to formulate the inverse problem, including the forward dynamic modeling and the corresponding derivative calculations. Then, the pyATM atmospheric transport model built with this framework is introduced, followed by its structural design and the adopted inversion algorithms.

2.1 Implementation of PDE

Consider a PDE on $[0, T] \times D$ with the following form:

$$\partial_t u = \mu(u, \partial_1 u, \cdots, \partial_d u) + f, \tag{1}$$

$$u(0, x) = u_0(x), \tag{2}$$

where $t \in [0, T]$, $x \in D \subset \mathbb{R}^d$. $u(x, t)$ is the state variable of the PDE, $\mu(u, \partial_1 u, \cdots, \partial_d u)$ denotes the state-dependent spatial derivative term, f is the external forcing term, and u_0 represents the initial condition. The external force term f can be independent of state variables. For this set of equations, we can obtain the transition relation of the corresponding state variables between two adjacent moments t_i and t_{i+1} by the integration in time,

$$u_{i+1} = u_i + \int_{t_i}^{t_{t+1}} [\mu(u_i, \partial_1 u_i, \cdots, \partial_d u_i) + f_i] dt. \tag{3}$$

The dynamical process of the partial differential equation is implemented using a deep learning framework within a computational graph. The proposed architecture consists of three parts: the spatial derivative calculations, the construction of the right-hand side (RHS), and the time integration. All orders of partial derivatives are calculated first, which are then added up with the source term f to complete the RHS. In time integration, the RHS is integrated forward in time to update the system state.

The spatial discretization of partial differential equations adopts the finite difference method, mainly due to its simple matrix operation form that can be favorably implemented by tensor operations in the deep learning framework. In the present study, we adopted second-order differencing for the transport problem, in which either upwind schemes or central schemes with filtering are used to stabilize the system. The spatial derivative degenerates to first-order forms at the boundaries corresponding to the north and south poles. With regards to time integration, explicit Runge-Kuta schemes are used.

2.2 Inverse Calculation

In traditional numerical computing, the forward PDE model is usually programmed by languages such as Fortran or C for speed and efficiency. However, the reverse adjoint model often requires dedicated development efforts. On the other hand, if the forward PDE is encoded within a deep learning framework such as torch(Paszke et al., 2019), tensorflow(Paszke et al., 2019), and JAX(Bradbury et al., 2018), it automatically stores operational dependencies and intermediate results, which then enables flexible and convenient gradient computation between variables. Moreover, those deep learning frameworks provide user-friendly interfaces for interpreted languages while they rely on backends that are written in compiled languages for intensive computations to compensate for efficiency loss. Therefore, deep learning frameworks have great potential in physical modeling and are especially suitable for inverse problems.

For inverse problems, the loss function, as shown in Eqn. 4, is usually defined as the difference between the projected state variables in the observation space $\mathcal{H}(u_i)$ and observation values o_i. Therefore, the solution process of the inverse problem is to iteratively update the learnable variables such that the loss function decreases along the gradient descending direction. If the learnable variable is the external force term f in Eqn. 1, then the update of f depends on the gradients of the loss with respect to the external force terms as in Eqn. 5. In the expression, the external force f_i applied at time i will have an impact on the states of all subsequent moments. As a result, the gradient calculation also includes the gradient of all the subsequent times.

$$\mathcal{L} = \sum_{i=1}^{T} \|\mathcal{H}(u_i) - o_i\| \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial f_i} = \sum_{j=i+1}^{T} \frac{\partial \mathcal{L}_j}{\partial f_i} = \sum_{j=i+1}^{T} \frac{\partial \|\mathcal{H}(u_j) - o_j\|}{\partial f_i} \tag{5}$$

In the forward time integration, the state change at a certain moment will affect all the states along the entire time trajectory. The perturbation of the state at each moment will gradually expand its influence range as time progresses. Conversely, the error function at some future moment will contribute to the state derivatives at all previous moments. According to the chain rule, the derivative of the error (loss) function at moment j with respect to the flux at moment i can be written as:

$$\frac{\partial \mathcal{L}_j}{\partial f_i} = \frac{\partial \|\mathcal{H}(u_j) - o_j\|}{\partial f_i} = \frac{\partial \|\mathcal{H}(u_j) - o_j\|}{\partial u_j} \frac{\partial u_j}{\partial u_{j-1}} \cdots \frac{\partial u_{i+2}}{\partial u_{i+1}} \frac{\partial u_{i+1}}{\partial f_i} \tag{6}$$

The above chain expression contains the product of several terms, including the derivative of the loss function with respect to the last state $\partial \|\mathcal{H}(u_j) - o_j\| / \partial u_j$, a series of derivatives of the next state with respect to the previous state $(\partial u_j / \partial u_{j-1}) \cdots (\partial u_{i+2} / \partial u_{i+1})$, and finally the derivative of the state with respect to the flux $\partial u_{i+1} / \partial f_i$. The derivative of the state variable at the next moment with respect to the state variable at the previous

moment $\partial u_{i+1}/\partial u_i$, can be obtained from Eqn. 3 by taking the derivatives on both sides,

$$\frac{\partial u_{i+1}}{\partial u_i} = 1 + \frac{\partial}{\partial u_i}\int_{t_i}^{t_{t+1}} \mu(u)dt. \tag{7}$$

Similarly, the derivative of the current states with respect to the external force term at the previous moment can also be derived from Eqn. 3,

$$\frac{\partial u_{i+1}}{\partial f_i} = \int_{t_i}^{t_{i+1}} 1dt = t_{i+1} - t_i. \tag{8}$$

Eqn. 7 and Eqn. 8 assume that the external force term is independent of the state variables, which is true for the carbon flux inversion problems that are of particular interest in this paper.

The gradient of the loss function in Eqn. 5 can be easily computed through automatic differentiation in our proposed method. By using a deep learning framework, chain rules can be automatically applied by backpropagation of the computation graph, without the need to explicitly compute all derivatives individually.

2.3 Flux Inversion Framework

2.3.1 Atmosphere Transport Model

The atmospheric transport model used in the inversion problem takes a special form of the hyperbolic parabolic differential equation in Eqn. 1, where only the first order derivatives (transport) are present in the system,

$$\frac{\partial \rho c}{\partial t} = -\nabla(\rho c \cdot V) + s. \tag{9}$$

The equation mainly describes the evolution of the carbon concentration field c under the transport effect of meteorological velocity field V, and the source/sink effects of the surface flux s. ρ is the density of the air which is constant by the incompressible flow assumption. In this problem, the meteorological velocity field V is considered a known quantity, which can be obtained through historical reanalysis data. In Eqn. 9, we do not consider the atmospheric chemistry-related terms. This set of transport equations applies to relatively long-lived gases in the atmospheric circulation such as CO2, since their chemical changes are small and thus negligible relative to transport effects.

Here, we use the pytorch framework to implement the atmospheric transport model, denoted as pyATM. The numerical implementation of pyATM mainly refers to GEOS-Chem (Community, 2021), which is a numerical model commonly adopted by researchers in a wide range of atmospheric chemistry problems. In our ATM implementation, the influence of the atmospheric boundary layer is taken into account. In the atmospheric boundary layer, the gas concentration field is usually mixed in a relatively short period of time. The TURBDAY mixing scheme is the default atmospheric boundary layer(ABL) mixing scheme in GEOS-Chem. In the TURBDAY scheme, it assumes instantaneous vertical mixing from the surface through the mixing depth (Bey et al., 2001). In our pyATM, the same implementation is also adopted, which assumes that the gas concentrations within the boundary layer are instantaneously mixed. The boundary layer thickness data are obtained from meteorological reanalysis data. We also adopt some numerical tricks employed in GEOS-Chem to help stabilize the system within the polar regions. Specifically, the concentration fields are regionally averaged in the two layers that are closest to the polar boundaries.

Using the atmospheric transport model implemented within pyATM, the computational graph has the following two characteristics. First, the depth of the computational graph is proportional to the length of the time integral. The whole process consists of many

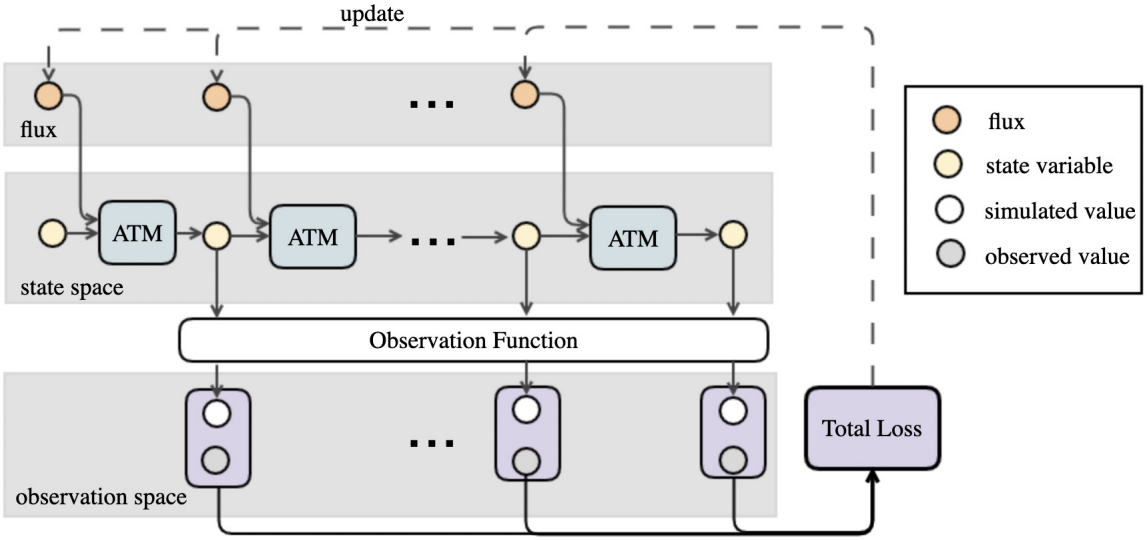

Figure 1. Schematic diagram of solving the carbon inversion problem using the automatic gradient inversion framework pyATM.

computational blocks, with each block representing the numerical evolution within a specific time step. Second, the structure of the computation graph is exactly the same among all blocks, only the inputs are different. In each computation block, the meteorological data and flux data corresponding to the time step is fed into the graph, while the former is fixed and the latter is learnable.

2.3.2 Flux Inversion

The atmospheric transport equation can be denoted as: $c_p = \mathcal{M}(c_0, f_p)$, where the model $\mathcal{M}$ represents the full dynamics of the concentration evolution. Conditioned on the initial concentration field c_0, as well as the surface flux f_p emitted within these periods, the predicted concentration vector c_p can then be obtained. The concentration in the state space can be mapped to the observation space by the observation operator $\mathcal{H}(c_p)$. In the flux inversion problem, the decision variables are the gas emissions f_p specified to the model. Estimation of the surface flux can be obtained by minimizing the following object function $\mathcal{L}(f_p)$ with respect to f_p:

$$\mathcal{L}(f_p) = \mathbb{E}[\mathcal{H}(c_p) - c_o)^T R^{-1}(\mathcal{H}(c_p) - c_o)] + \mathbb{E}[(f_p - f_b)^T B^{-1}(f_p - f_b)] \tag{10}$$

The error function contains two parts. The first part is the observation error term, and the second part is the background error term. The observation error term describes the error between the observation c_o and the simulated concentration values in the observation space $\mathcal{H}(c_p)$, weighted by the measurement uncertainty matrix R. The second part is called the background error, which describes the difference between the predicted flux f_p and background flux f_b, weighted by the background error covariance matrix B.

Based on the above description of the atmospheric transport model pyATM, the derivative calculation for the flux update, and the definition of the error function, we summarize the inversion process in the pyATM framework. Figure 1 depicts the schematic diagram of solving the carbon inversion problem using automatic gradient in the pyATM framework. At each time step, the state variable and flux input are fed into the single-step ATM block, and the state at the next moment can be obtained. These time steps are concatenated to

complete the entire numerical simulation. The states obtained by pyATM are then mapped to the observation space in which the errors compared to the actual observations are computed. Finally, all errors are aggregated and contribute to the derivative of the total error with respect to fluxes at each time step. After a complete forward state prediction and backward gradient propagation, the fluxes are updated by gradient descent methods to minimize the loss function. The problem of particular interest in this study is carbon flux inversion, while the proposed framework is formulated without loss of generality such that it is also applicable to general PDE systems.

The proposed framework has several advantages compared to other inversion methods. Using the automatic gradient scheme, the flux updating is similar to the variational method. Both methods update the flux driven by the derivative of the error with respect to the flux input. However, compared with traditional variational methods that rely on the adjoint model, the proposed framework avoids complicated adjoint development, which is often a dedicated endeavor. Moreover, the flexibility of the variational method is usually limited. Once the code implementation of the forward atmospheric transport has changed, the corresponding adjoint model also needs to be updated accordingly. On the other hand, deep learning frameworks provide automatic differentiation of all tensor operations. Once the forward computational graph is built, it can automatically obtain the gradient relationship between any two connected tensors, and the dependencies get automatically updated if the forward computational graph changes. Using automatic differentiation, the computational cost of the inversion algorithm is independent of the input parameter dimensions, making it efficient in high-dimensional inversion problems.

3 Experimental Setup

In the experimental part, this paper mainly discusses two aspects: one is to verify the effectiveness of the pyATM atmospheric transport model, and the other is to demonstrate the automatic differentiation method for solving the inversion problem through twin experiments.

3.1 Numerical Setting

The implementation of the pyATM forward process mainly refers to the numerical schemes in GEOS-Chem, such that numerical settings and the input meteorological conditions are the same as those in GEOS-Chem. The atmospheric transport models operate on a global scale, which is discretized evenly in both the longitude and latitude directions with rectangular meshes. The spatial resolution is 5° (longitude) $\times 4^\circ$ (latitude), corresponding to a horizontal mesh grid of 72×46 cells. In the vertical direction, 47 pressure layers are used as those in GEOS-Chem. The time step Δt for the numerical simulation is 30 minutes. The MERRA2(ref) reanalysis data are used for the meteorological fields, providing the velocity vector data as well as the boundary layer thickness information.

3.2 Identical-twin Observing System Simulation Experiments

In the experiments, we adopt the identical-twin observing system simulation experiments to examine the effectiveness of the inversion algorithm. The concept of twin experiments is very useful for testing the effectiveness of inversion algorithms as the “true" solution is known. The schematic diagram of the identical-twin inversion problem using pyATM is shown in Figure 2. In the identical-twin experimental setup, "true" observations are obtained by running a forward numerical process with known initial conditions and "true" fluxes. In the inversion stage, the initial concentration field is fixed, and only the surface flux parameters are zero initialized and then iteratively updated. Through such an experiment, the influence of other interference factors can be excluded, and the backward relationship from the observation error to the flux can be reflected more clearly. The

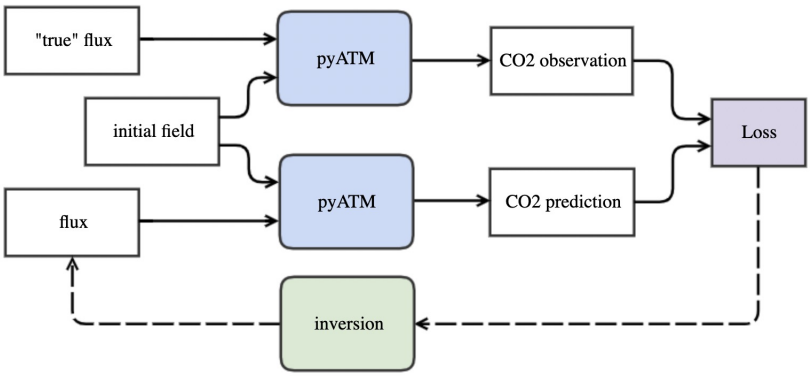

Figure 2. Schematic diagram of the identical-twin observing system simulation experiments using pyATM. In the twin experimental setup, CO2 observations are obtained by running a forward numerical procedure with known initial conditions and "true" fluxes.

identical-twin problem has a "true" flux that can provide a quantitative evaluation of the correctness and accuracy of the inversion algorithm. At the same time, the identical-twin experiment makes it more convenient to perform some operations on the observation data, such as adding perturbations, using sparse sampling observations, etc.

The observable variable is XCO_2 , the column-weighted average CO_2 concentration, which is typically available in the satellite retrieved data. Two sets of validation under different observation coverage are used, namely the full observation inversion and the partial observation one. With regards to the partial observation case, the time stamps and spatial coordinates of the observation points are generated based on the sampling pattern of carbon satellites. In the partial observation scenarios, the inversion problem is usually ill-posed due to the sparsity of valid observations, as well as the dissipative nature of the atmospheric transport. The problem of inversion underdetermination caused by sparse observations can be alleviated by making more observations or by reducing the degrees of freedom of the flux parameters.

To verify the validity of the proposed inversion framework, we do not consider the background flux error term that is commonly adopted, and thus only the first term on the right-hand side of Eqn. 10 is included in the loss function. By ignoring the prior information of the background flux, the inverse problem is made more challenging as less information is used. However, excluding prior information also helps to examine more clearly the carbon flux updates that directly correspond to observational errors, while eliminating interferences from background terms. Therefore, carbon fluxes are updated from zero initialization, and no background term is added to the loss function. Here we assume that all observations are uncorrelated since all sensors observe independently. In such a case, a unit diagonal matrix is used as the error covariance matrix R. During the iterative process, we use the Adam optimizer with a learning rate of 1e-5. The inversion process is stopped after 300 iterative steps.

4 Results

4.1 ATM validation

The atmospheric transport equation encoded in pyATM is implemented with reference to GEOS-Chem. To verify its effectiveness, we quantitatively verify the pyATM results against the GEOS-Chem simulations. In the experiment, the time spans over 7 days, from July 1^{st} 2019 T 00:00 UTC to July 8^{st} 2019 T 00:00 UTC. We use the same initial concen-

Table 1. MAE error under different pyATM numerical schemes, compared with the corresponding GEOS-Chem numerical simulation in the range of 7 days.

version	Spatial scheme	Vertical Transport	PBL Mixiing	XCO2 MAE(ppm)	bottom CO2 MAE(ppm)
V01	central	off	off	0.304	1.044
V02	upwind	off	off	0.229	0.949
V03	upwind	on	off	0.181	0.949
V04	upwind	on	on	0.111	0.192

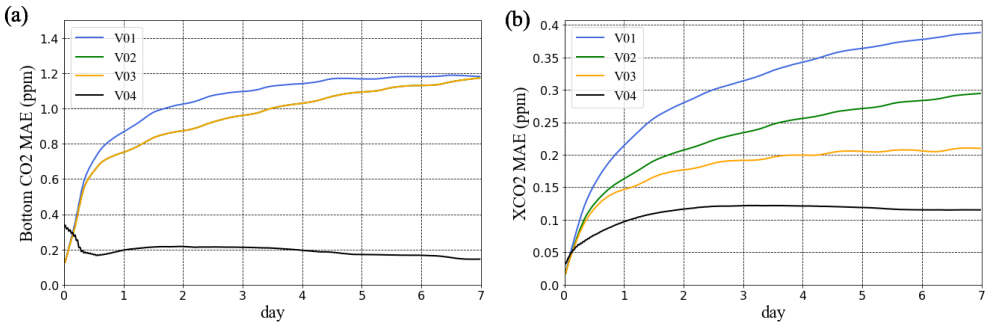

Figure 3. MAE error under different pyATM numerical schemes, compared with the corresponding GEOS-Chem numerical simulation in the range of 7 days. (a) bottom CO2 MAE(ppm). (b) XCO2 MAE(ppm).

tration field and meteorological data to quantitatively compare the results of the forward integrations between pyATM and GEOS-Chem. In order to examine the implementation of the atmospheric transport model, the carbon flux is not included in the simulation process. To obtain numerical results that are closer to GEOS-Chem, we conducted a series of explorations of numerical schemes. On the spatial discretization, we tried both the second-order central scheme and the upwind scheme. The effect of adding vertical convection, as well as the effect of boundary layer mixing, are also within the scope of the numerical investigation. Table 1 lists different versions of the pyATM and the corresponding mean absolute error (MAE) within a 7-day numerical simulation. The V01 version adopts the central difference scheme with filtering while ignoring both vertical convection and boundary layer mixing. The V02 version uses the second-order upwind scheme to increase the numerical stability. Based on version V02, V03 adds the convection effect in the vertical direction. Furthermore, version V04 adds the implementation of boundary layer mixing compared with version V03. The daily error curves for the four versions within the 7-day range are illustrated in Figure 3. In the quantitative results, we present the bottom CO_2 and XCO_2 errors, which are most relevant to the actual observations. For observation stations located close to the ground, only CO_2 concentration at the bottom of the atmosphere can be obtained. On the other hand, only the column weighted concentration XCO_2 can be retrieved from satellite observations.

Considering the spatial discretization scheme, the upwind scheme outperforms the central one, reducing the XCO_2 error from 0.304 parts per million (ppm) to 0.229 ppm. The upwind scheme achieves better numerical stability compared to the central one since it respects the convective nature of the system. The V03 version takes into account convection effects in the vertical direction, making the momentum transfer three-dimensional. Since the bottom layer is not affected by the 3D effect, the error of the bottom layer CO_2 will not change. However, for a full vertical level, the error of XCO_2 is significantly reduced to

0.181 ppm. If the influence of the boundary layer mixing is further considered, the error is greatly reduced as shown in Table 1 for the version V04. After considering the boundary layer mixing, the bottom CO_2 error can be reduced to 0.192 ppm, and the error of XCO2 is further reduced to 0.111 ppm, achieving similar results compared to GEOS-Chem.

The errors during the 7-day range are plotted in Figure 3 for different numerical settings. V01, V02, and V03 versions all have a monotonic increasing trend, in which the error grows fast at the beginning and then gradually flattens. Among them, the V01 version exhibits the highest XCO_2 error around 0.4 ppm after 7 days while the V03 achieves the lowest error level around 0.2 ppm. For the V04 version, the error peaks after about 3 days, and finally tends to decrease slowly. The error of the predicted XCO_2 after 7 days is about 0.12 ppm. According to the carbon satellite observations, the overall XCO_2 retrieval error is about 0.3ppm. Therefore, the model error of pyATM (assumes the GEOS-Chem as the "true" model) can be made much smaller than the observation error, indicating its capability of serving as a forward model in carbon flux inversion. For the error curves of the bottom CO_2 concentration, it can be seen that error of V04 flattens right after the CO_2 has been well mixed in the boundary layer while the error accumulates but gradually flattens in the other schemes.

Figure 4 presents the predicted results of the pyATM model on days 1, 4, and 7, compared with the results from GEOS-Chem. The XCO_2 distribution is smoothed out under the atmospheric transport and boundary layer mixing, in which both the maximum and minimum values are attenuated. In terms of spatial distribution, the characteristics of high concentration in the northern hemisphere and low concentration in the southern hemisphere are retained. Visually, the results of pyATM are very close to those of GEOS-Chem, indicating a reliable capability of pyATM to simulate carbon atmospheric transport.

4.2 Inversion Result

For the inversion tests, we designed experiments for both the full observation and partial observation scenarios respectively, to examine the influence of observation sparsity on the inversion results. Full observation means that the concentration information on any latitude and longitude coordinates can be obtained at any time. However, in the vertical direction, only the column weighted concentration can be obtained (XCO_2). The generation of sparse observation data mainly refers to the samples from satellite observation data. This verifies the effectiveness of solving the inversion problem at the practical spatial and temporal sampling frequency of on-orbit satellites. Here, we use the combined dataset of the two most commonly used carbon satellites, the Japanese Greenhouse gases Observing Satellite (GOSAT) (Yokota et al., 2009b) and the NASA Orbiting Carbon Observatory (OCO-2)(Crisp et al., 2004a). Figure 6 shows the spatial sampling points for the first two days of July 2019. The sampling trajectory of the satellite is different every day, and only a small area can be scanned in a short period of time, and the overall sampling frequency is about 1%. In addition, these two satellites cannot achieve full coverage over the globe. For the month of July, we are concerned about the lack of observation data coverage in the high-latitude areas in the southern hemisphere, which will be further discussed later.

We mainly investigate the long-period inversion capability of the proposed framework. Therefore, a temporally fixed while spatially varied carbon flux field is adopted at all timestamps as the time-averaged flux field. Utilizing the automatic differentiation capability of deep learning frameworks to compute the derivatives, the update of carbon fluxes is then performed iteratively. Figure 5 shows the statistics during the iterative process, in which (a) is the MAE at observation points, and (b) presents the MAE of the inversed flux field. For the MAE error of the concentration at the observation points, it decreases rapidly with the progress of the iterative step, and it presents a log-linear law in the later stage. For the full observation case, the XCO_2 error decreases to around 1.6×10^{-3} ppm after 300 iterations. As for the partial observation one, the observation error is slightly higher, which is about

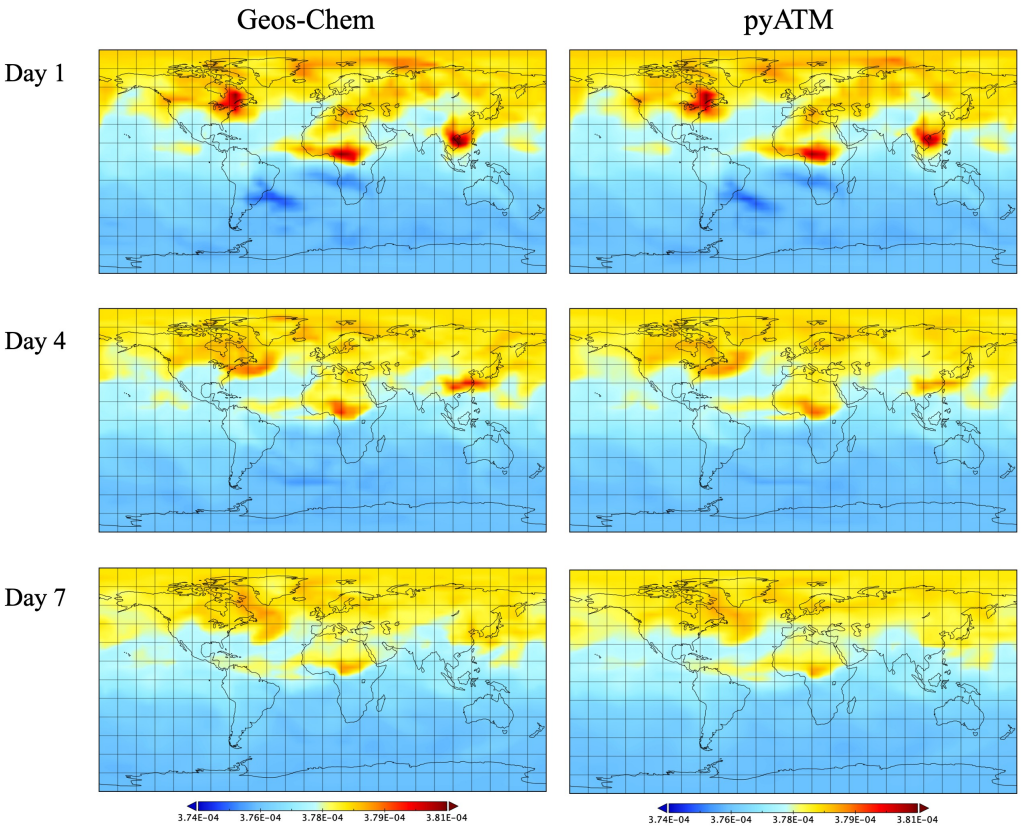

Figure 4. Simulation results of XCO2 for GEOS-Chem and pyATM at days 1, 4, and 7 respectively.

2.2×10^{-3} ppm after 300 iterations. For the inversed flux error as shown in Figure 5 (b), the error decreases gradually as the iteration proceeds for both the full and the partial observation cases. However, the error is significantly smaller under the full observations, indicating high sensitivities of the XCO_2 with respect to the fluxes. After 300 iterations, the flux errors of the full observation and the partial observation cases are around $2.59 \times 10^{-8} mol \cdot m^{-2} s^{-1}$ and $6.10 \times 10^{-8} mol \cdot m^{-2} s^{-1}$, respectively. In the case of sparse observations, due to the data sparseness in both space and time, only the flux within a small influence range of the observation point can be effectively updated. As discussed earlier, the observation in the high-latitude areas of the southern hemisphere is obviously insufficient, so the nearby flux cannot be effectively updated.

The update history of the inversed flux during the optimization is shown in Figure 7 and Figure 8. These can further help to investigate the capability as well as the effect of the inversion algorithm. Figure 7 shows the inversed fluxes after 1, 5, 25, 100, and 300 iterations, along with the ground truth under the full observation scenario. After one iteration step, the flux source and sink patterns can be seen. In the northern part of Eurasia and North America, the flux sinks due to vegetation absorption are manifested. While in East Asia, South Asia, Central America, North America, and South America, there exhibit carbon sources. In the fifth iteration step, the intensities of carbon sources and sinks are further enhanced, showing a large carbon sink in Siberia, as well carbon sources corresponding to several urban areas in South Asia. After 25 iteration steps, the overall flux distribution is well captured, and all major carbon sources and sinks in the ground truth are reflected in the inversion results, although small-scale differences still exist. Iterating from the 25^{th} to the 100^{th} iteration, the major carbon sources and sinks are further refined. The peak flux sources corresponding to East and South Asian urban regions are more clearly and accurately depicted. From the 100^{th} iteration to the 300^{th} iteration, the update slowly saturates, and the flux field at the 300^{th} iteration is very close to the ground truth.

For partially observed scenarios, the flux update is consistent with the full observation scenario, the flux gradually converges as the optimization proceeds. However, the flux convergence in the partial observation scenario is less effective compared to that of the full observation one. After 300 iterations, there are still noticeable discrepancies between the inversed flux and the ground truth, which is particularly prominent in the southern hemisphere with little observation coverage. The inversion results failed to recover oceanic carbon sinks in the southern hemisphere. Meanwhile, the inversed flux shows abnormal sources in the Antarctic region. These deficiencies are sort of expected due to inadequate observations in the high-latitude areas of the southern hemisphere. As for the northern hemisphere, the overall flux distribution is well captured, although the carbon sources in East and South Asia are underestimated. Moreover, more spikes in the flux distributions can also be observed due to sparse observations.

In summary, the spatial-temporal distribution pattern and the magnitudes are better retrieved in fully-observed scenarios. However, considering the difficulties of practical data acquisition, the system is only partially observable most of the time. Despite some flaws in data-poor regions, the inversion results employing the synthesized satellite data show good agreement with the ground truth, indicating the applicability of the proposed pyATM framework in real-world applications. Furthermore, we anticipate that prior knowledge of the flux distributions can also help regularize the flux inversion in practice.

5 Conclusions

In this paper, we developed a fully-differentiable atmospheric transport model called pyATM, which encodes the PDE solver into computational graphs based on a deep learning framework. Using the framework's automatic differentiation capabilities, the derivatives of the loss function with respect to the learnable parameters are obtained. When the learnable parameter is the carbon flux and the loss function is the observation error, the flux can be

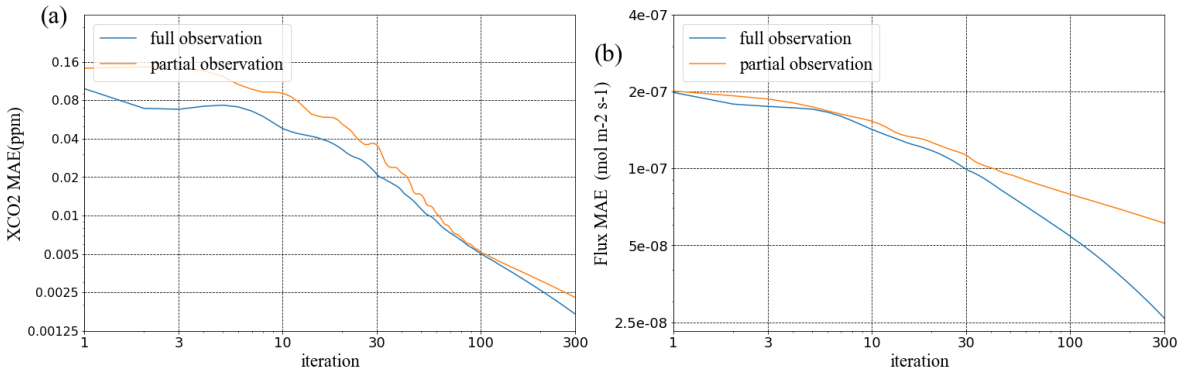

Figure 5. Error convergence history in the inversion tests. (a) XCO2 MAE in the observation points. (b) Inversed flux MAE.

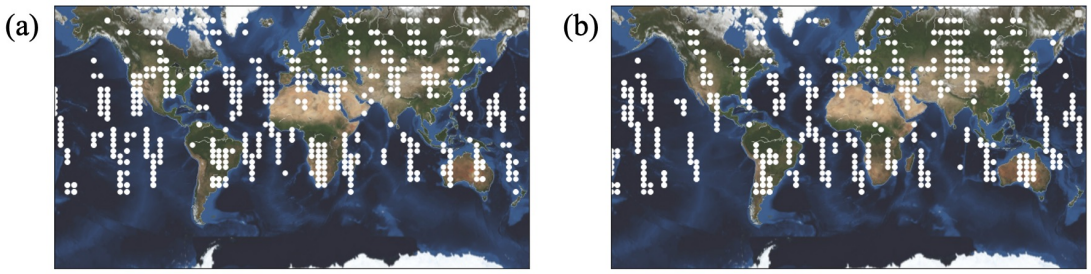

Figure 6. Satellite observation sampling points of 2 typical days. (a) July 1^{st} 2019 (b) July 2^{nd} 2019.

corrected by observation. The proposed framework has been demonstrated in identical-twin OSSEs and has achieved favorable performance in both fully-observed and partially-observed systems.

Based on the proposed framework, the connection between the machine learning framework and inverse problem solving is established. Since the PDE solver is embedded in the machine learning framework, popular machine learning models can be easily added to the PDE solving procedure, making it possible to seamlessly integrate machine learning models within PDE-constrained inverse problems in future research. Furthermore, the application of automatic differentiation is not limited to carbon flux inversion problems but also applies to inverse/optimization problems constrained on more general PDEs, such as data assimilation in numerical weather predictions and shape optimization in engineering design.

Acknowledgments

This work was supported in part by the Zhejiang Science and Technology Program under Grant 2021C01017.

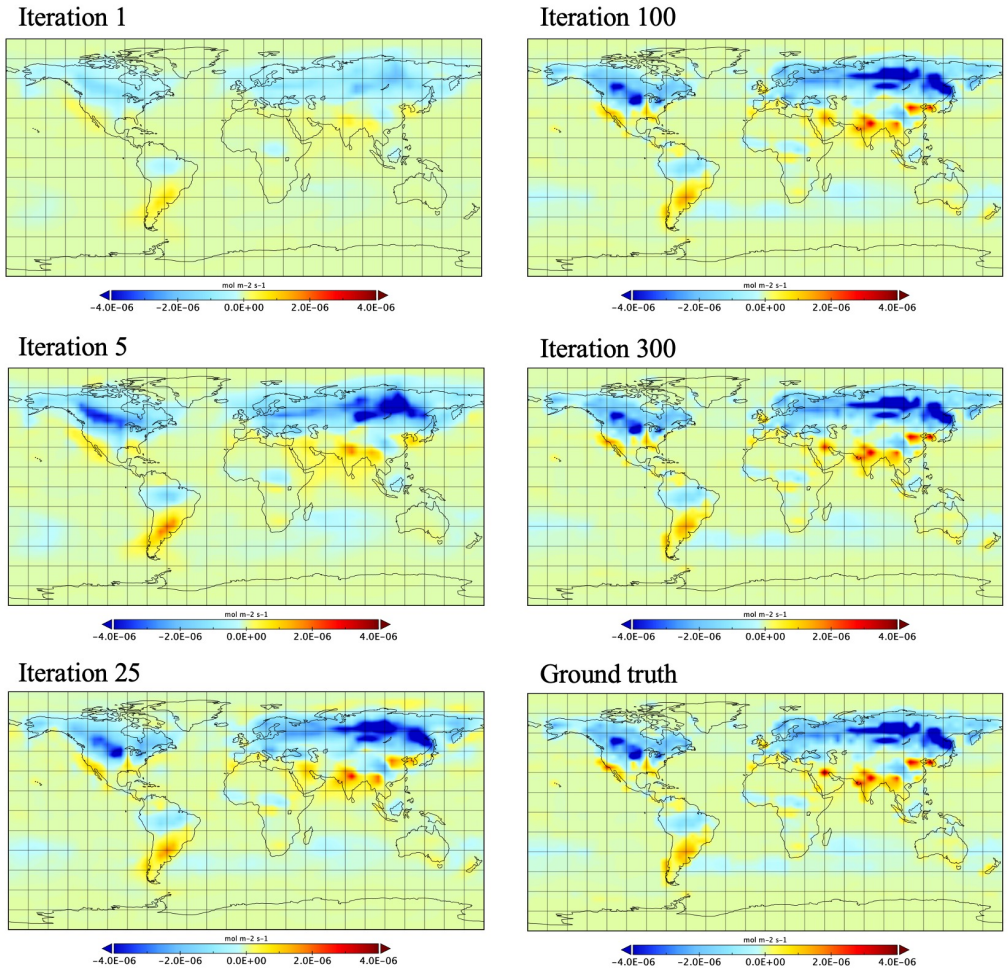

Figure 7. The flux update history in the full observation case. The 6 images present the inversed flux fields at the 1^{st}, 5^{th}, 25^{th}, 100^{th}, 300^{th} iteration steps, and the ground truth.

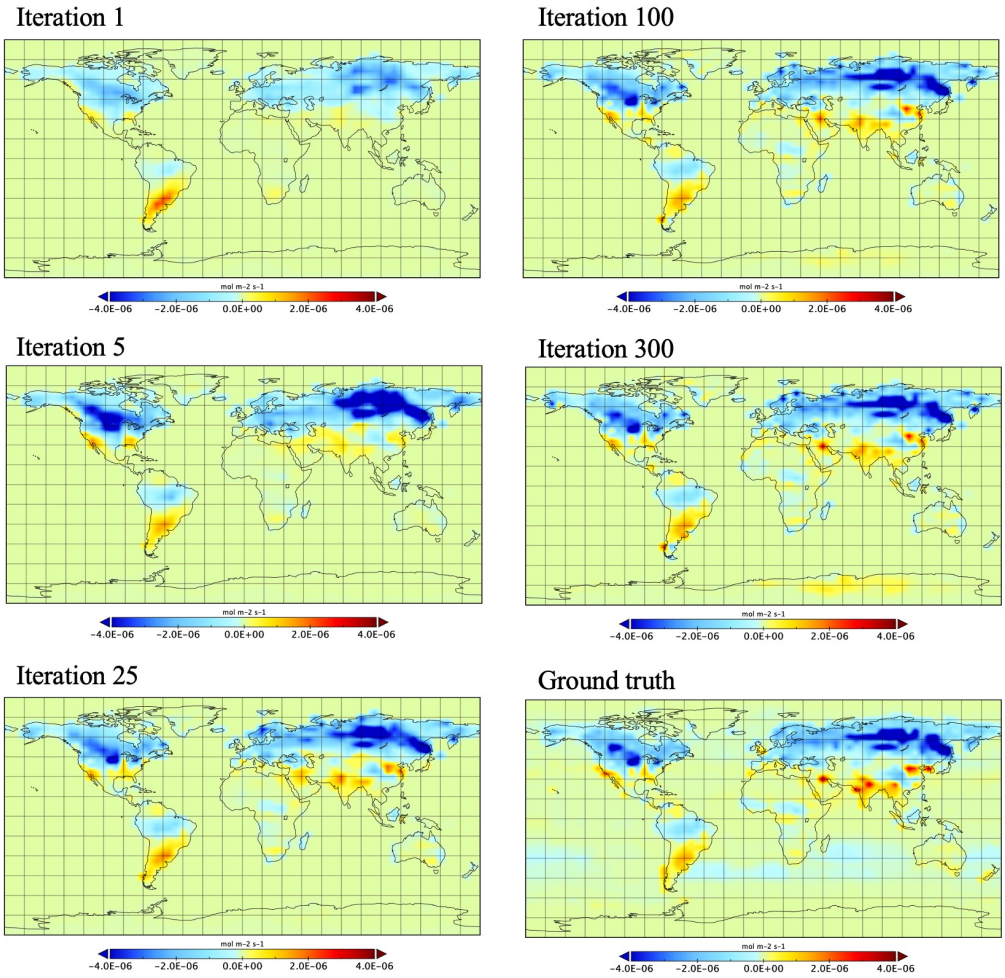

Figure 8. The flux update history in the partial observation case. The 6 images present the inversed flux fields at the 1^{st}, 5^{th}, 25^{th}, 100^{th}, 300^{th} iteration steps, and the ground truth.

References

Amendola, M., Arcucci, R., Mottet, L., Casas, C. Q., Fan, S., Pain, C., ... Guo, Y.-K. (2021). Data assimilation in the latent space of a convolutional autoencoder. In *Computational science – ICCS 2021* (pp. 373–386). Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-030-77977-1_30 doi: 10.1007/978-3-030-77977-1_30

Baker, D. F., Bösch, H., Doney, S. C., O'Brien, D., & Schimel, D. S. (2010, May). Carbon source/sink information provided by column CO_2 measurements from the orbiting carbon observatory. *Atmospheric Chemistry and Physics*, *10*(9), 4145–4165. Retrieved from https://doi.org/10.5194/acp-10-4145-2010 doi: 10.5194/acp-10-4145-2010

Baker, D. F., Doney, S. C., & Schimel, D. S. (2006, January). Variational data assimilation for atmospheric CO_2. *Tellus B: Chemical and Physical Meteorology*, *58*(5), 359–365. Retrieved from https://doi.org/10.1111/j.1600-0889.2006.00218.x doi: 10.1111/j.1600-0889.2006.00218.x

Bey, I., Jacob, D. J., Yantosca, R. M., Logan, J. A., Field, B. D., Fiore, A. M., ... Schultz, M. G. (2001). Global modeling of tropospheric chemistry with assimilated meteorology: Model description and evaluation. *Journal of Geophysical Research: Atmospheres*, *106*(D19), 23073–23095.

Blunden, J., & Arndt, D. S. (Eds.). (2020, August). State of the climate in 2019. *Bulletin of the American Meteorological Society*, *101*(8), S1–S429. Retrieved from https://doi.org/10.1175/2020bamsstateoftheclimate.1 doi: 10.1175/2020bamsstateoftheclimate.1

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., ... Zhang, Q. (2018). *JAX: composable transformations of Python+NumPy programs*. Retrieved from http://github.com/google/jax

Chen, G., & Fidkowski, K. J. (2021, June). Output-based adaptive aerodynamic simulations using convolutional neural networks. *Computers & Fluids*, *223*, 104947. Retrieved from https://doi.org/10.1016/j.compfluid.2021.104947 doi: 10.1016/j.compfluid.2021.104947

Chevallier, F., Engelen, R. J., Carouge, C., Conway, T. J., Peylin, P., Pickett-Heaps, C., ... Xueref-Remy, I. (2009, October). AIRS-based versus flask-based estimation of carbon surface fluxes. *Journal of Geophysical Research*, *114*(D20). Retrieved from https://doi.org/10.1029/2009jd012311 doi: 10.1029/2009jd012311

Chevallier, F., Maksyutov, S., Bousquet, P., Bréon, F.-M., Saito, R., Yoshida, Y., & Yokota, T. (2009, October). On the accuracy of the CO_2 surface fluxes to be estimated from the GOSAT observations. *Geophysical Research Letters*, *36*(19). Retrieved from https://doi.org/10.1029/2009gl040108 doi: 10.1029/2009gl040108

Collins, M., Knutti, R., Arblaster, J., Dufresne, J.-L., Fichefet, T., Friedlingstein, P., ... others (2013). Long-term climate change: projections, commitments and irreversibility. In *Climate change 2013-the physical science basis: Contribution of working group i to the fifth assessment report of the intergovernmental panel on climate change* (pp. 1029–1136). Cambridge University Press.

Community, T. I. G.-C. U. (2021). *geoschem/gcclassic: Geos-chem 13.3.3.* Zenodo. Retrieved from https://zenodo.org/record/5748260 doi: 10.5281/ZENODO.5748260

Crisp, D., Atlas, R., Breon, F.-M., Brown, L., Burrows, J., Ciais, P., ... others (2004a). The orbiting carbon observatory (oco) mission. *Advances in Space Research*, *34*(4), 700–709.

Crisp, D., Atlas, R., Breon, F.-M., Brown, L., Burrows, J., Ciais, P., ... Schroll, S. (2004b, January). The orbiting carbon observatory (OCO) mission. *Advances in Space Research*, *34*(4), 700–709. Retrieved from https://doi.org/10.1016/j.asr.2003.08.062 doi: 10.1016/j.asr.2003.08.062

Emmons, L. K., Walters, S., Hess, P. G., Lamarque, J.-F., Pfister, G. G., Fillmore, D., ... Kloster, S. (2010, January). Description and evaluation of the model for ozone and

related chemical tracers, version 4 (MOZART-4). *Geoscientific Model Development*, *3*(1), 43–67. Retrieved from https://doi.org/10.5194/gmd-3-43-2010 doi: 10.5194/gmd-3-43-2010

Espeholt, L., Agrawal, S., Sønderby, C., Kumar, M., Heek, J., Bromberg, C., ... Kalchbrenner, N. (2021). *Skillful twelve hour precipitation forecasts using large context neural networks.*

Fablet, R., Chapron, B., Drumetz, L., Mémin, E., Pannekoucke, O., & Rousseau, F. (2021, October). Learning variational data assimilation models and solvers. *Journal of Advances in Modeling Earth Systems*, *13*(10). Retrieved from https://doi.org/10.1029/2021ms002572 doi: 10.1029/2021ms002572

Feng, L., Palmer, P. I., Bösch, H., & Dance, S. (2009, April). Estimating surface CO_2 fluxes from space-borne CO_2 dry air mole fraction observations using an ensemble kalman filter. *Atmospheric Chemistry and Physics*, *9*(8), 2619–2633. Retrieved from https://doi.org/10.5194/acp-9-2619-2009 doi: 10.5194/acp-9-2619-2009

Frerix, T., Kochkov, D., Smith, J. A., Cremers, D., Brenner, M. P., & Hoyer, S. (2021). *Variational data assimilation with a learned inverse observation operator.*

Friedlingstein, P., Jones, M. W., O'Sullivan, M., Andrew, R. M., Hauck, J., Peters, G. P., ... Zaehle, S. (2019, December). Global carbon budget 2019. *Earth System Science Data*, *11*(4), 1783–1838. Retrieved from https://doi.org/10.5194/essd-11-1783-2019 doi: 10.5194/essd-11-1783-2019

Fung, I. Y., Doney, S. C., Lindsay, K., & John, J. (2005, August). Evolution of carbon sinks in a changing climate. *Proceedings of the National Academy of Sciences*, *102*(32), 11201–11206. Retrieved from https://doi.org/10.1073/pnas.0504949102 doi: 10.1073/pnas.0504949102

Grell, G. A., Peckham, S. E., Schmitz, R., McKeen, S. A., Frost, G., Skamarock, W. C., & Eder, B. (2005, December). Fully coupled "online" chemistry within the WRF model. *Atmospheric Environment*, *39*(37), 6957–6975. Retrieved from https://doi.org/10.1016/j.atmosenv.2005.04.027 doi: 10.1016/j.atmosenv.2005.04.027

Henze, D. K., Hakami, A., & Seinfeld, J. H. (2007, May). Development of the adjoint of GEOS-chem. *Atmospheric Chemistry and Physics*, *7*(9), 2413–2433. Retrieved from https://doi.org/10.5194/acp-7-2413-2007 doi: 10.5194/acp-7-2413-2007

Hu, Y., Chen, L., Wang, Z., & Li, H. (2022). *Swinvrnn: A data-driven ensemble forecasting model via learned distribution perturbation.*

Kang, J.-S., Kalnay, E., Liu, J., Fung, I., Miyoshi, T., & Ide, K. (2011, May). "variable localization" in an ensemble kalman filter: Application to the carbon cycle data assimilation. *Journal of Geophysical Research*, *116*(D9). Retrieved from https://doi.org/10.1029/2010jd014673 doi: 10.1029/2010jd014673

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021, May). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. Retrieved from https://doi.org/10.1038/s42254-021-00314-5 doi: 10.1038/s42254-021-00314-5

Laloyaux, P., Kurth, T., Dueben, P. D., & Hall, D. (2022, May). Deep learning to estimate model biases in an operational NWP assimilation system. *Journal of Advances in Modeling Earth Systems*. Retrieved from https://doi.org/10.1029/2022ms003016 doi: 10.1029/2022ms003016

Mack, J., Arcucci, R., Molina-Solana, M., & Guo, Y.-K. (2020, December). Attention-based convolutional autoencoders for 3d-variational data assimilation. *Computer Methods in Applied Mechanics and Engineering*, *372*, 113291. Retrieved from https://doi.org/10.1016/j.cma.2020.113291 doi: 10.1016/j.cma.2020.113291

Masarie, K. A., Peters, W., Jacobson, A. R., & Tans, P. P. (2014, December). ObsPack: a framework for the preparation, delivery, and attribution of atmospheric greenhouse gas measurements. *Earth System Science Data*, *6*(2), 375–384. Retrieved from https://doi.org/10.5194/essd-6-375-2014 doi: 10.5194/essd-6-375-2014

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In

H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., ... Anandkumar, A. (2022). *Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators.*

Peters, W., Jacobson, A. R., Sweeney, C., Andrews, A. E., Conway, T. J., Masarie, K., ... Tans, P. P. (2007, November). An atmospheric perspective on north american carbon dioxide exchange: CarbonTracker. *Proceedings of the National Academy of Sciences*, *104*(48), 18925–18930. Retrieved from https://doi.org/10.1073/pnas.0708986104 doi: 10.1073/pnas.0708986104

Peters, W., Miller, J. B., Whitaker, J., Denning, A. S., Hirsch, A., Krol, M. C., ... Tans, P. P. (2005). An ensemble data assimilation system to estimate CO_2 surface fluxes from atmospheric trace gas observations. *Journal of Geophysical Research*, *110*(D24). Retrieved from https://doi.org/10.1029/2005jd006157 doi: 10.1029/2005jd006157

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P., & Goret, G. (2021, September). Latent space data assimilation by using deep learning. *Quarterly Journal of the Royal Meteorological Society*, *147*(740), 3759–3777. Retrieved from https://doi.org/10.1002/qj.4153 doi: 10.1002/qj.4153

Poulter, B., Bastos, A., Canadell, J. G., Ciais, P., Huntzinger, D., Houghton, R. A., ... Luyssaert, S. (2022). Bottom-up approaches for estimating terrestrial GHG budgets: Bookkeeping, process-based modeling, and data-driven methods. In *Balancing greenhouse gas budgets* (pp. 59–85). Elsevier. Retrieved from https://doi.org/10.1016/b978-0-12-814952-2.00010-1 doi: 10.1016/b978-0-12-814952-2.00010-1

Quéré, C. L., Raupach, M. R., Canadell, J. G., Marland, G., Bopp, L., Ciais, P., ... Woodward, F. I. (2009, November). Trends in the sources and sinks of carbon dioxide. *Nature Geoscience*, *2*(12), 831–836. Retrieved from https://doi.org/10.1038/ngeo689 doi: 10.1038/ngeo689

Quéré, C. L., Rödenbeck, C., Buitenhuis, E. T., Conway, T. J., Langenfelds, R., Gomez, A., ... Heimann, M. (2007, June). Saturation of the southern ocean sink due to recent climate change. *Science*, *316*(5832), 1735–1738. Retrieved from https://doi.org/10.1126/science.1136188 doi: 10.1126/science.1136188

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., ... Mohamed, S. (2021, September). Skilful precipitation nowcasting using deep generative models of radar. *Nature*, *597*(7878), 672–677. Retrieved from https://doi.org/10.1038/s41586-021-03854-z doi: 10.1038/s41586-021-03854-z

Solomon, S., Plattner, G.-K., Knutti, R., & Friedlingstein, P. (2009, February). Irreversible climate change due to carbon dioxide emissions. *Proceedings of the National Academy of Sciences*, *106*(6), 1704–1709. Retrieved from https://doi.org/10.1073/pnas.0812721106 doi: 10.1073/pnas.0812721106

Thompson, R. L., Chevallier, F., Maksyutov, S., Patra, P. K., & Bowman, K. (2022). Top-down approaches. In *Balancing greenhouse gas budgets* (pp. 87–155). Elsevier. Retrieved from https://doi.org/10.1016/b978-0-12-814952-2.00008-3 doi: 10.1016/b978-0-12-814952-2.00008-3

Toon, G., Blavier, J.-F., Washenfelder, R., Wunch, D., Keppel-Aleks, G., Wennberg, P., ... Notholt, J. (2009). Total column carbon observing network (TCCON). In *Advances in imaging*. OSA. Retrieved from https://doi.org/10.1364/fts.2009.jma3 doi: 10.1364/fts.2009.jma3

Yokota, T., Yoshida, Y., Eguchi, N., Ota, Y., Tanaka, T., Watanabe, H., & Maksyutov, S. (2009a). Global concentrations of CO2 and CH4 retrieved from GOSAT: First preliminary results. *SOLA*, *5*, 160–163. Retrieved from https://doi.org/10.2151/sola.2009-041 doi: 10.2151/sola.2009-041

Yokota, T., Yoshida, Y., Eguchi, N., Ota, Y., Tanaka, T., Watanabe, H., & Maksyutov, S. (2009b). Global concentrations of co2 and ch4 retrieved from gosat: First preliminary

results. *Sola*, *5*, 160–163.

Zhu, Y., & Zabaras, N. (2018, August). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, *366*, 415–447. Retrieved from https://doi.org/10.1016/j.jcp.2018.04.018 doi: 10.1016/j.jcp.2018.04.018