JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2024MS004655

Atmospheric Transport Modeling of CO_2 With Neural Networks

Vitus Benson[1,2,3], Ana Bastos[2,4], Christian Reimers[1,2], Alexander J. Winkler[1,2], Fanny Yang[3], and Markus Reichstein[1,2]

[1]Max Planck Institute for Biogeochemistry, Jena, Germany, [2]ELLIS Unit Jena, Jena, Germany, [3]ETH Zürich, Zürich, Switzerland, [4]Leipzig University, Leipzig, Germany

Key Points:

- CarbonBench: a systematic benchmark for machine learning emulators of atmospheric tracer transport
- Adapted SwinTransformer deep neural network to achieve stable and mass-conserving transport of CO_2 by including physical constraints
- UNet, GraphCast, and Spherical Fourier Neural Operator baselines with the same customization are also strong models, for shorter lead times (up to 90 days)

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

V. Benson,
vbenson@bgc-jena.mpg.de

Citation:

Benson, V., Bastos, A., Reimers, C., Winkler, A. J., Yang, F., & Reichstein, M. (2025). Atmospheric transport modeling of CO_2 with neural networks. *Journal of Advances in Modeling Earth Systems*, *17*, e2024MS004655. https://doi.org/10.1029/2024MS004655

Received 20 AUG 2024
Accepted 27 JAN 2025

Abstract Accurately describing the distribution of CO_2 in the atmosphere with atmospheric tracer transport models is essential for greenhouse gas monitoring and verification support systems to aid implementation of international climate agreements. Large deep neural networks are poised to revolutionize weather prediction, which requires 3D modeling of the atmosphere. While similar in this regard, atmospheric transport modeling is subject to new challenges. Both, stable predictions for longer time horizons and mass conservation throughout need to be achieved, while IO plays a larger role compared to computational costs. In this study we explore four different deep neural networks (UNet, GraphCast, Spherical Fourier Neural Operator and SwinTransformer) which have proven as state-of-the-art in weather prediction to assess their usefulness for atmospheric tracer transport modeling. For this, we assemble the CarbonBench data set, a systematic benchmark tailored for machine learning emulators of Eulerian atmospheric transport. Through architectural adjustments, we decouple the performance of our emulators from the distribution shift caused by a steady rise in atmospheric CO_2. More specifically, we center CO_2 input fields to zero mean and then use an explicit flux scheme and a mass fixer to assure mass balance. This design enables stable and mass conserving transport for over 6 months with all four neural network architectures. In our study, the SwinTransformer displays particularly strong emulation skill: 90-day $R^2 > 0.99$ and physically plausible multi-year forward runs. This work paves the way toward high resolution forward and inverse modeling of inert trace gases with neural networks.

Plain Language Summary Changes in the CO_2 concentration can be measured in our atmosphere. To connect these to emissions, and activity from biosphere and ocean ecosystems, traditionally an atmospheric transport model is used that tracks the flow of CO_2 with the winds. Now, with progress in artificial intelligence (AI), it can be questioned, if these atmospheric transport models can be replaced with an AI model. In this work we introduce CarbonBench, a benchmark data set to train and compare different AI models. Moreover, we design a state-of-the-art AI model to predict how CO_2 distributes in the atmosphere. All our data and code are open-source, with the aim to enable further research toward leveraging AI for monitoring greenhouse gases and supporting climate agreements.

1. Introduction

Limiting greenhouse gas emissions in line with the Paris agreement to mitigate anthropogenic climate change requires monitoring, reporting and verification (MRV) efforts, especially of carbon dioxide (CO_2) (Friedlingstein et al., 2023). Atmospheric measurements of CO_2 from ground-based observatories, aircraft and satellite can provide independent, science-based estimates. However, these observations represent the concentration in the free air, not directly the emissions and other surface fluxes. Atmospheric transport models build the necessary bridge, allowing to understand CO_2 concentrations from the perspective of anthropogenic emissions, biosphere and ocean fluxes (Ciais et al., 2011; Gurney et al., 2002; Kaminski & Heimann, 2001). They solve the continuity equation of the mass of CO_2 in the atmosphere by computing horizontal advection and vertical movement of air parcels using driving meteorological reanalysis fields (Brasseur & Jacob, 2017).

Since its early ages in the late 1980s, solving 3D tracer transport with numerical schemes has been hampered by prohibitive computational costs when going to higher resolution (Williamson, 1992). Yet, low resolution transport models, suffer from a variety of modeling errors (Gaubert et al., 2019; Schuh et al., 2019). More specifically, representations of convective transport (Belikov et al., 2013; Munassar et al., 2023; Remaud et al., 2023; Schuh & Jacobson, 2023), turbulent vertical mixing (Kretschmer et al., 2012), summertime diabatic mixing (Jin et al., 2024), numerical advection scheme (Agusti-Panareda et al., 2017; Eastham & Jacob, 2017) and

reanalyzed meteorological fields (Yu et al., 2018; Zhang et al., 2021) in atmospheric transport models display significant uncertainties. Increasing resolution has been proposed as one potential remedy to the situation (Agustí-Panareda et al., 2019; Remaud et al., 2018).

However, a primary application of transport models is in inverse modeling of the surface fluxes to contribute regularly to MRV efforts such as the annual Global Carbon Budget updates (Friedlingstein et al., 2023). Starting from prior surface fluxes, the transport model is used to map them to atmospheric concentrations which can be compared against observations to subsequently optimize the fluxes through Bayesian calibration (Chandra et al., 2022; Chevallier et al., 2005, 2006; Peters et al., 2007; Remaud et al., 2018; Rödenbeck, 2005; Rödenbeck et al., 2003, 2018; van der Laan-Luijkx et al., 2017). This iterative process typically requires many expensive calls of the transport model and its adjoint, thereby rendering the usage of high fidelity solvers difficult (Chevallier et al., 2023).

Recently, AI-based emulation has revolutionized numerical weather prediction: deep neural networks trained on high resolution meteorological reanalysis can both, outpace and outperform, traditional medium-range weather forecasting systems (Bi et al., 2022; Bonev et al., 2023; Chen et al., 2023; Keisler, 2022; Kochkov et al., 2023; Lam et al., 2023; Pathak et al., 2022; Price et al., 2023). Crucially, these emulators require less vertical layers, allow for larger time steps and leverage computing infrastructure optimized for matrix multiplication like GPUs. Hence, the neural networks learn to solve the Navier Stokes equations, by implicitly representing both, the large-scale dynamics that could be explicitly solved, and subgrid-scale processes that have to be parameterized, some works even make this division explicit (Arcomano et al., 2022; Kochkov et al., 2023; Krasnopolsky & Fox-Rabinovitz, 2006). Furthermore, foundation models are being introduced which support other tasks beyond medium-range weather forecasting, such as climate modeling (Lessig et al., 2023; Nguyen et al., 2023) or short-term forecasts of atmospheric composition (Bodnar et al., 2024).

Modeling the atmospheric carbon cycle with neural networks has not yet gathered as much attention. Still, there are works on emulating the footprints obtained from Lagrangian particle dispersion models of CH_4, which are useful for regional inverse modeling: Over a few UK regions, the NAME model has been emulated with convolutional neural networks (CNNs) (Cartwright et al., 2023) and with Gradient Boosting Trees (Fillola et al., 2022) and over a few US regions, STILT has been emulated with FootNet (He et al., 2023), also a CNN. If more broadly considering approaches to modeling the CO_2 and CH_4 surface fluxes, machine learning has been used to upscale eddy covariance measurements as functions of climate and remote sensing to the globe, to obtain land fluxes of CH_4 (McNicol et al., 2023) and CO_2 (Jung et al., 2011, 2020; Nelson et al., 2024; Tramontana et al., 2016). For the latter, Upton et al. (2024) recently introduced additional atmospheric constraints, bridging between atmospheric inverse modeling and machine learning-based upscaling.

Here, we introduce atmospheric transport modeling of CO_2 with neural network emulators. Our main contributions are three-fold:

1. We create a new data set (CarbonBench), the first systematic benchmark for training and testing machine learning emulators of Eulerian atmospheric transport.
2. We develop a SwinTransformer-based emulator tailored for transport modeling through physics-based adjustments that allow for strong empirical performance: forward runs with global RMSE below 1 ppm are possible for multiple years.
3. We compare performance against three other large deep neural network architectures (UNet, GraphCast, and SFNO). While the SwinTransformer outperforms, with our generic architectural changes also the baselines achieve stable and mass-conserving transport for over 6 months.

Thus, we provide the first step toward a high resolution CO_2 inversion system leveraging AI to support the World Meteorological Organizations Global Greenhouse Gas Watch (G3W) and other efforts in line with the Paris agreement.

2. Methods

2.1. Task

In this work, we are tackling offline tracer transport with neural networks. That is, we solve the continuity equation for the inert trace gas CO_2 given prescribed meteorology. In other words, we predict the 3D field of CO_2

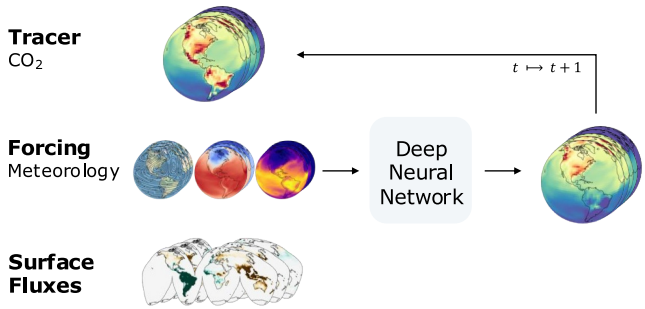

Figure 1. Offline atmospheric tracer transport modeling with deep neural networks.

concentration in the atmosphere at time $t + 1$ given the CO_2 concentration field from the previous time step t and meteorology and surface fluxes as additional inputs (Figure 1). Like conventional solvers, our learned neural networks are autoregressive: longer forward runs can be produced by feeding the predicted CO_2 concentrations back in as inputs, alongside prescribed fluxes and meteorology from the next time step. This allows in principle to generate arbitrarily long trajectories of CO_2 fields, if sufficient forcing data is available.

More specifically consider the CO_2 mass mixing ratio μ, a source/sink term Σ and the vector of wind fields V, then tracer transport follows from integrating

$$\frac{d\mu}{dt} + V \cdot \nabla\mu = \Sigma \tag{1}$$

over the spherical shell $\mathcal{D} = S^2 \times [r, r + h] \subset \mathbb{R}^3$, with S^2 the sphere, r the radius of Earth and h the height of the atmosphere. The integration is typically done by specifying von Neumann boundary conditions $\frac{d\mu}{dn} = 0$, with n being the outward-facing normal derivative on D, in other words: the flux out of the atmosphere is none. This would model surface fluxes with the source/sink term Σ, allowing for emissions inside the atmosphere. However, one may alternatively want to model surface fluxes as the lower boundary condition. In offline tracer transport models, the winds V are prescribed. An alternative approach would be online tracer transport, where in addition to the tracer transport, the full atmospheric dynamics are modeled (Patra et al., 2018).

When numerically integrating the continuity equation, one needs to discretize over a grid, which requires splitting the operator into resolved and unresolved scales. For atmospheric transport, one furthermore typically splits the operator into horizontal advection and vertical convection, whereby for the former any subgrid-scale closure is ignored, but for the latter it is parameterized (Heimann & Körner, 2003). Hence we end up with the equation

$$\frac{d\mu}{dt} + u\frac{d\mu}{dx} + v\frac{d\mu}{dy} + w(\omega, T, q, z)\frac{d\mu}{dz} = \Sigma \tag{2}$$

with the vertical velocity w being a function of updraft ω, temperature T, specific humidity q and geopotential height z. Throughout this work, we use neural networks to solve directly for the time derivative:

$$\frac{d\mu}{dt} = f(\mu, u, v, \omega, T, q, z, \ldots; \theta) \tag{3}$$

with $f(\cdot;\theta)$ being a neural networks with parameters θ. We then integrate using Euler steps $\mu_{t+1} = \mu_t + \frac{d\mu}{dt}$. During training, this means we approximate $\Delta\mu_t = \mu_{t+1} - \mu_t$ with the neural network $f(\cdot;\theta)$ by optimizing parameters through minimizing the squared loss:

$$\hat{\theta} = \underset{\theta}{arg\,min}\mathbb{E}\| \left(f(X_t;\theta) - \Delta\mu_t\right)\|_2^2 \tag{4}$$

2.2. CarbonBench Data Set

For training the neural network emulators, we collect two existing data sets and reprocess them into a deep learning-ready format. The first data set (CarbonTracker) is an inversion of CO_2, that is, it has been obtained by optimizing the surface fluxes by transporting them and then matching modeled atmospheric concentrations against observed ones. The second data set (ObsPack) contains atmospheric measurements of CO_2, allowing to compare our model predictions against an absolute baseline, independent of the training targets.

2.2.1. CarbonTracker

The CarbonTracker North America inversions (Peters et al., 2007) utilize the TM5 (Krol et al., 2005) transport model and the ensemble Kalman filter to perform inverse modeling of the surface fluxes. More specifically, they

start with a set of prior fluxes for the land and ocean (e.g., from Earth system models) and add these to prescribed fluxes for anthropogenic emissions and wildfires to obtain a first version of total CO_2 surface fluxes. In a next step, they leverage an atmospheric transport model and the ensemble Kalman filter to optimize the surface fluxes such that they match well to observed data of atmospheric CO_2 concentrations. Finally, the optimized fluxes are transported one more time to obtain a 3D field of atmospheric CO_2 concentrations. Here, we only use the final product from the inverse modeling process: the optimized surface fluxes and corresponding 3D fields. Moreover, we treat all surface fluxes as prescribed inputs, and not just the anthropogenic and wildfire components.

We collect 3D atmospheric CO_2 concentration fields, 2D CO_2 surface fluxes and 3D meteorological fields of q, T, u, v, ω, z from the CarbonTracker CT2022 version (Jacobson et al., 2023). These represent a closed system, that is, they fulfill a discretized version of the continuity Equation 1. Moreover, as they have been produced through inverse modeling, they are also closely resembling observations of atmospheric CO_2. For a complete list of the variables used, see Table S1 in Supporting Information S1.

We prepare three versions of the data set through aggregation that allow for quicker experimentation and testing of methods at multiple resolution. Each data set we split into training (years 2000–2016), validation (2017) and testing (2018–2020) sets, the three resolutions are:

- LowRes: 5.625° × 5.625° × 10 hybrid vertical levels ×6h.
- MidRes: 2.8125° × 2.8125° × 20 hybrid vertical levels ×6h.
- OrigRes: 2° × 3° × 34 hybrid vertical levels ×3h.

Note, while *OrigRes* is close to the original data resolution, it is not exact—we shift the time steps in comparison to CarbonTracker by 1.5 hr (except for fluxes) and we still regrid the surface fluxes, which had been optimized at 1° × 1° in CarbonTracker. In addition, in CarbonTracker North America, the full atmosphere is modeled at this higher resolution over a zoomed window in North America. We deliberately chose the horizontal resolution such that LowRes (MidRes) horizontal fields have 32 × 64 (64 × 128) pixels, which is ideal for most modern deep neural network architectures from computer vision (Rasp et al., 2024). Furthermore, the underlying TM5 atmospheric transport model of CarbonTracker North America solves the equations at higher temporal resolution with a dynamically varying time step of ≤90 minutes to ensure numerical stability. Still, external wind fields from ERA5 (Hersbach et al., 2020) are only provided at 3*h* resolution, in line with the OrigRes data in CarbonBench.

2.2.2. Data Preprocessing

In order to prepare the three deep learning-ready data set versions, we introduce a preprocessing chain. Through this chain, we aim to standardize data set format and ensure that the processed data is directly useable to implement offline tracer transport emulators in the spirit of Equation 3. Furthermore, the chain enables future work to leverage the presented neural networks on data sets from other transport models. We perform the following preprocessing steps:

1. Horizontal regridding: intensive meteorological variables with bilinear interpolation, extensive quantities (CO_2 mixing ratio and air mass) are divided by cell area, and then, alongside CO_2 surface fluxes regridded with conservative interpolation.
2. Conversion to standard units and variables: masses in $[Pg]$, Concentrations as ppm mass mixing ratio $\left[\frac{10^{-6} kgCO_2}{kg\text{DryAir}}\right]$, fluxes as $\left[\frac{kgCO_2}{m^2 s}\right]$, pressure in $[hPa]$. We aggregate surface fluxes into ocean, land and anthropogenic fluxes, where the former two would be optimized during an inversion and the latter one prescribed.
3. Vertical aggregation: interpolation in pressure coordinates through taking a pressure weighted mean for intensive quantities, and through summation for extensive quantities (masses).
4. Temporal resampling: linear resampling to target resolution.
5. Flux staggering: surface fluxes are staggered, such that they represent the mean flux between a time step and the next time step.
6. Flux mass correction: anthropogenic surface fluxes are corrected, such that any mass conservation errors introduced through preprocessing are removed and the mass difference between two time steps matches exactly the surface fluxes.
7. Temporally splitting into independent training, validation and testing data sets.

8. Deep learning-optimized storage: we store our data set in Zarr files, with chunking that optimizes loading of all data at a single time step: We store two arrays per time step, one with all 2D fields and one with all 3D fields.
9. Statistics: we compute mean and std. dev. statistics for all fields and for all per-level temporal deltas of all fields.

The preprocessing routines are implemented as part of the Neural Transport Python library (https://github.com/vitusbenson/neural_transport).

2.2.3. ObsPack Station Data

The NOAA ObsPack GLOBALVIEWplus product (Schuldt et al., 2023) collects measurements of atmospheric CO_2 from many different scientific laboratories around the globe with instruments at ground-based stations and towers and onboard ships, aircraft and weather balloons. In this study, we use all measurements flagged as representative from the *v9.1_2023-12-08* product. We compare these CO_2 measurements with our modeled data by extracting the grid cells closest to the horizontal (lat/lon) and vertical position (geopotential height) of the measurement and averaging over 6 hr time windows. We use the exact same method to extract station time series from the target CarbonTracker data, as we use for the AI models. This allows for an absolute comparison point: the target CarbonTracker data does not achieve perfect prediction of the ObsPack data, meaning we can compare the performance of AI models directly with TM5, the transport model used in CarbonTracker. In future work, the ObsPack station data does also allow for cross-dataset comparison. Note, however, if AI models trained on two different data sets are compared, differences in performance may also stem from the differences in the prescribed surface fluxes, meteorology and initial conditions, and not merely from the learned transport model.

2.2.4. Evaluation

We evaluate models by performing quarterly forward runs starting on 1 January, 1 April, etc. and running for 3 months each. We then average statistics over the full test period (2018–2020) and compute a range of performance metrics, such as RMSE, R^2, decorrelation time (#days with $R^2 > 0.9$), RMS mass error, relative mean and relative variability. We compute these metrics over individual spatial and temporal coordinate axes and also over sets of axes, to obtain a full picture.

2.3. Neural Networks

In this section we describe the neural networks studied in this work. We restrict ourselves to a rather conceptual description and refer the reader to the original papers for in-depth explanations of each architecture. In addition we report the adjustment to the original architectures which we introduce in this work to enable their applicability to atmospheric transport modeling.

2.3.1. Motivation

Atmospheric transport modeling requires processing high dimensional data: at the coarsest resolution, our model input has $32 \times 64 \times (10 \times 10 + 7) \approx 220k$ dimensions (assuming a setup of 10 3D and seven 2D fields, see Table S1 in Supporting Information S1—and accordingly $\sim 20k$ output dimensions). At such scales, training a standard 2-layer neural network, the multi-layer perceptron (MLP), becomes computationally intractible. In deep learning this challenge is typically approached by introducing inductive biases. These allow to significantly reduce the dimensionality of each matrix multiplication. In this study, from the vast variety of available architectures, we pick four. They are representative of generic architectural classes and previous work has found them successful at emulating weather and climate data.

Moreover, three out of the four networks coincide with general classes of conventional numerical methods (compare Figure 2): (a) UNet uses a regular mesh, like finite difference solvers on regular grids, (b) GraphCast uses an icosahedral mesh, again analogous to finite difference solvers, (c) SFNO is similar to a pseudo-spectral solver, only (d) SwinTransformer is unconventional in the way that it favors a brute-force split-process-combine approach, with little resemblance to conventional numerical methods, that is, it has the least inductive bias.

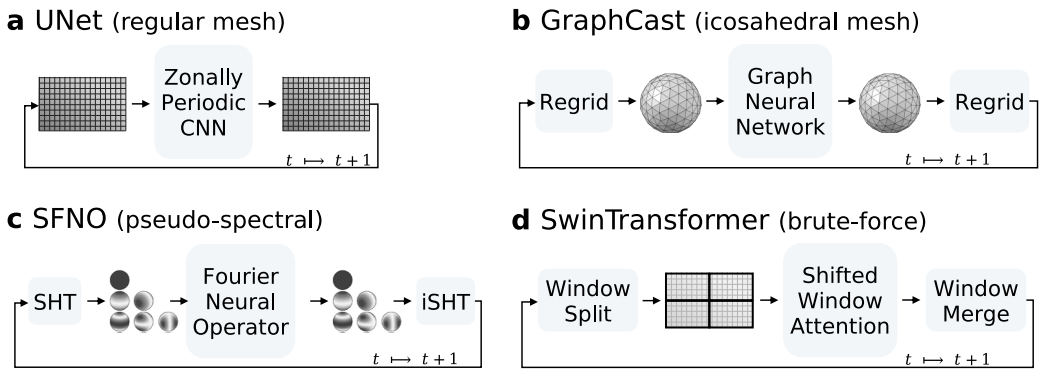

Figure 2. Conceptual depiction of the four deep neural networks included in this study.

2.3.2. Vertical Discretization

In all four approaches, we only consider inductive biases for the horizontal dimension. In the vertical direction we stack all data along the channel dimension and feed that as input. In other words, the models receive an array of values (for forcing, tracers and surface fluxes) per horizontal grid cell, and then process these in a latent space, allowing for vertical mixing and interactions across variables. This approach is independent of the particular vertical discretization pertinent in the data. While it is possible to include inductive biases that reflect vertical neighborhood, we refrain from doing so, as that is not standard for weather modeling—likely motivated by the possibility for strong vertical mixing due to convection.

In this work, we use CarbonTracker data, which comes at hybrid model levels. Hybrid levels interpolate smoothly between a terrain-following component in the lower troposphere (close to the surface) and constant pressure levels in the upper stratosphere. More specifically, the pressure of each vertical layer is an affine transformation of the surface pressure (which varies with orography).

2.3.3. UNet

UNets (Ronneberger et al., 2015) are fully CNNs consisting of an encoder and a decoder arranged in a *U-shape*—referring to gradual spatial downsampling and subsequent upsampling. We employ UNets that treat the globe as a cylinder, having periodic convolutions in zonal (longitude) direction and zero-padded convolutions at the poles (Rasp et al., 2020; Scher, 2018). Vertical layers and different variables are simply stacked along the channel dimension.

Our UNet has 4 stages within the encoder and decoder, each consisting of two 3×3 2d conv layers, that are followed by LeakyReLU and BatchNorm layers and a residual connection. Spatial downsampling is achieved through 2×2 MaxPooling and upsampling through 2×2 nearest interpolation. In the first encoder stage, we use a single 7×7 conv layer instead. We add skip connections between the encoder and decoder stages. The network operates on input sizes that are divisible by 16, but through bilinear upsampling in the first and nearest downsampling in the last layer, we allow for other input shapes as well.

2.3.4. SwinTransformer

SwinTransformers (Liu et al., 2021) are transformer neural networks processing 2D inputs by attention between embeddings of windows, which are shifted in each layer. We allow for periodic shifts in zonal (longitude) direction and retain processing at the highest resolution (no hierarchical layers), two architectural design choices which have been proven useful for weather forecasting (Willard et al., 2024). However, in contrast to Willard et al. (2024), we adopt relative positional encoding, as in Liu et al. (2022).

Our SwinTransformer has 12 layers each consisting of a Multi-head Self-Attention block followed by LayerNorm and a pixelwise MLP (with GELU activation and LayerNorm) and residual connections between blocks. The self-attention is masked in such a way, that only attention within windows of nearby pixels is computed, we use 4×8 pixel windows. Windows are shifted by half their size at every second layer, with zonally periodic shifts and masked shifts at the poles—to cover information transfer in all spatial directions. In contrast to previous work we

found using patch embedding to introduce artifacts at longer rollouts, which is why our model directly operates at pixel level (i.e., in 1 × 1 patches). Input shapes need to be divisible by the window shape, we allow for other input shapes through nearest interpolation.

2.3.5. GraphCast

GraphCast is a graph neural network tailored for weather forecasting. It follows an encode-process-decode layout (Battaglia et al., 2018), with the encoder and decoder mapping between the regular grid (lat-lon) and an icosahedral mesh (Keisler, 2022). Thus, they are responsible for two tasks: first, they perform regridding, akin to conventional regridding tools, but here learned, and second, they map the input data into an high-dimensional latent space, as typical for deep neural networks. On the icosahedral mesh in latent space, the processor component processes the data to obtain a powerful embedding from which the time delta of the target variables can be extracted. More specifically, the processor uses message passing layers in local neighborhoods of each grid cell with additional long-range connections (Lam et al., 2023). This can be understood as local stencils on the sphere that process information just like in a conventional finite difference solver, with the addition of some non-local interactions between supernodes, that can further enhance predictions.

Our GraphCast has a processor with 8 layers, each performing message passing between neighboring nodes on an icosahedral multi-mesh that has been refined 3 times (levels 0–3). The encoder uses a bipartite graph to map between the regular grid representation and multi-mesh nodes by assigning all grid cells to a multi-mesh node whose center is less than 0.75 times the maximum inter-node distance in the level 3 mesh away from that node. The encoder and decoder map between data space and a latent space with 256 channels. Like the original GraphCast we use Swish activations and layer norm. Our message passing layer use a mean operation to aggregate incoming information from neighboring nodes.

2.3.6. Spherical Fourier Neural Operator

Spherical Fourier Neural Operators (SFNO) (Bonev et al., 2023) are an extension of the Fourier Neural Operator (FNO) (Li et al., 2021) to the sphere, by replacing Fourier transforms with spherical harmonics transforms (SHT). An FNO Block performs channel-wise spatial processing in the spectral domain and combines this with channel-mixing in the grid domain. The SFNO consists of many blocks, each using the SHT and inverse SHT to map between grid and spectral space. We use linear transformations in spectral space and local MLPs in grid space.

2.4. Details

We train our deep neural networks using the Neural Transport Python library (https://github.com/vitusbenson/neural_transport). Our experiment scripts are published in the CarbonBench Python repo (https://github.com/vitusbenson/carbonbench).

2.4.1. Optimization

We train our models with ADAM in a two-stage fashion. First, with a cosine learning rate schedule and linear warm up on next-step prediction. Afterward with a constant learning rate and a n-steps-ahead schedule, where we iteratively increase the lead time during training every 2 epochs until 31-steps-ahead. For hyperparameter tuning (we tune the learning rate with a coarse grid search per model architecture) and ablation studies, we train for 100k steps, and for the final models (i.e., the best performing ablations) we train for 300k steps during the first optimization stage, that is, next-step prediction. In both cases we use the same n-steps-ahead training during the second optimization stage after the next-step training, as this had a big impact on performance over longer rollouts. In this work, we optimize always against the full 3D CO_2 field from CarbonTracker, future work may consider additionally including a part of the ObsPack measurements (which are only used for evaluation in this work) or weighting targets differently.

2.4.2. CentFlux

We scale and shift the model output with the std. dev. and mean of the temporal deltas of each target variable vertical layer. Afterward, we add the previous time step 3D field to obtain a raw prediction for the next time step.

In addition, we add the surface fluxes to the lowest vertical layer. Due to steadily rising anthropogenic emissions, the input CO_2 mean is increasing over time, which would represent a covariate shift, to which neural networks are rarely robust. To account for this, we center the input CO_2 field at each time step to have zero mean. This fix should allow stable transport for arbitrary levels of atmospheric CO_2. Throughout this manuscript we call the addition of surface fluxes at the lowest vertical level and the centering of CO_2 input fields jointly *CentFlux*.

2.4.3. SpecLoss

Previous work identified divergence in the power spectra to be symptomatic for models becoming unstable for longer rollouts (Chattopadhyay & Hassanzadeh, 2023). To improve in this regard we introduce an additional loss term that regularizes predictions. *SpecLoss* measures the difference in spectral power densities between observed and predicted 2D fields (i.e., at each vertical level). We leverage the spherical harmonics transform to obtain spectral coefficients, from which we compute the spectral power density. Our approach is similar to a regularization term used in NeuralGCM (Kochkov et al., 2023).

2.4.4. Massfixer

Tracer transport fulfills the continuity equation, which stems from mass conservation, in other words, the total mass of simulated CO_2 in the atmosphere at $t + 1$ should match the mass at t plus the total mass input through the surface fluxes. While some conventional numerical approaches like finite volume methods fulfill tracer mass conservation by design, others, such as semi-Lagrangian or pseudo-spectral schemes do not. Also deep neural networks are only softly constrained to fulfill mass conservation (if zero emulation error is achieved, mass is necessarily conserved). Similarly to previous attempts to correct conventional approaches (Diamantakis & Flemming, 2014), we adopt a simple mass fixer, that scales the predicted mass at each time step by the desired mass calculated from the surface fluxes. This fixer leads to proportionally larger adjustments in grid cells with more tracer mass.

3. Results

3.1. Model Intercomparison

We evaluate global and local test set performance of the four neural network architectures, each with tuned hyperparameters, and report the results in Figure 3. UNet, GraphCast, SFNO and SwinTransformer all achieve stable transport for at least 6 months with local performance almost equal to TM5, that is, to the ground truth that models had been trained on. The best model is SwinTransformer, which achieves a global R^2 of 0.99 over quarterly forecasts, that is, almost perfect emulation. Performance degrades when looking at the other three models, with UNet > SFNO > GraphCast. Here, GraphCast has more than double the global RMSE compared to SwinTransformer, but still stays below 1 ppm over 90 day forward runs. Furthermore, GraphCast runs become unstable after 178 days, while the other three models display decorrelation times above 3 years, indicating long-term stability (Figure 3a). At station level, the difference are of lower magnitude, but still significant (Figures 3c and 3d). In the following we assess the performance of the SwinTransformer, the best performing model, in more detail, with the equivalent plots for the other models provided in the supplementary material.

3.2. Best Performing Model

The SwinTransformer produces stable forecasts in terms of RMSE, R^2, relative mean and relative std. dev. over 90 days. Figure 4 compares the performance for different levels. Mostly, the performance varies little for different layers, with the exception that the surface layer has a significantly larger RMSE compared to all other layers (over 1.5×). Moreover, while in the lower troposphere after a brief annealing phase during the first few forecast steps the predictions are of approximately constant quality, there is a drift with increased performance degradation in the upper stratosphere (the top three layers).

Qualitatively, SwinTransformer captures the large-scale motion of CO_2 in the atmosphere, as depicted by maps of total column CO_2 (Figure 5). The largest errors appear in eastern Asia, a region known for large anthropogenic

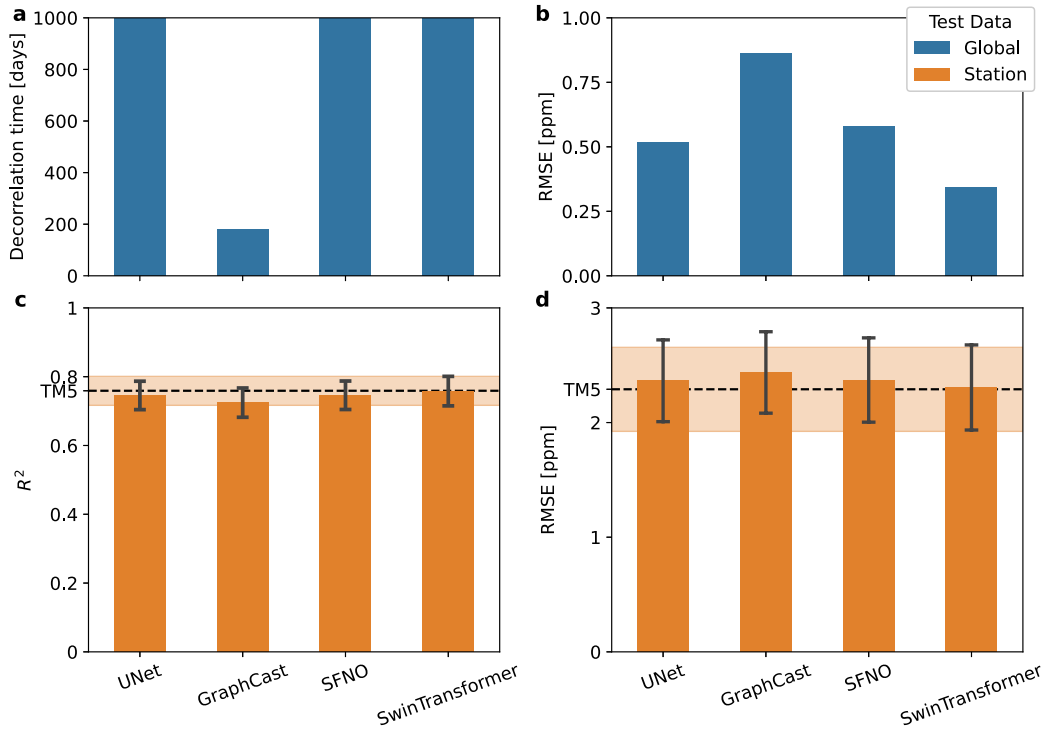

Figure 3. Intercomparison between the best models per architecture. In blue (a and b), the performance is evaluated by scoring the global predicted 3D field against the ground truth CO_2 field from the test period of the LowRes data set—this allows for comparisons between the AI models. In orange (c and d), the performance is evaluated at ObsPack stations. This allows, in addition, to compare against TM5 (dashed black lines), the transport model used to produce the ground truth data set. At ObsPack stations, in addition to the mean scores (mean over the stations), we also display uncertainty estimates: the std. dev. over stations scaled by the square root of the number of stations. Local R^2 (c) and global (b) and local RMSE (d) are computed for quarterly 90-day forward runs, the decorrelation time (a) is estimated from a single 3 year forward run.

emission. Otherwise, error patterns appear to follow fronts in the atmospheric field, indicating mildly decreased performance over sharper gradients (Figure 5).

Zooming in on a few stations from the ObsPack Globalview product, SwinTransformer generally performs similar to the training target TM5 (Figure 6). Interestingly, for the Svalbard station, SwinTransformer captures the

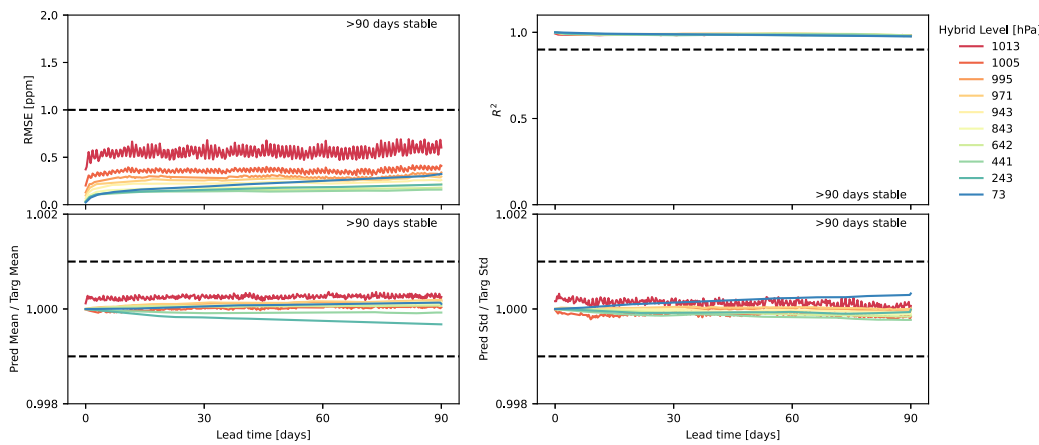

Figure 4. Key metrics per vertical layer for quarterly forecasts over the test set for SwinTransformer. We report metrics per time step and vertical level, that is, they represent properties of the 2D maps of atmospheric CO_2 mass mixing ratios at different vertical levels. The metrics are averaged over quarterly reset 90-day forward runs. Dashed lines indicate arbitrarily set thresholds which subjectively signify stable simulation (e.g., RMSE < 1 ppm is a goal for many CO_2 MRV systems).

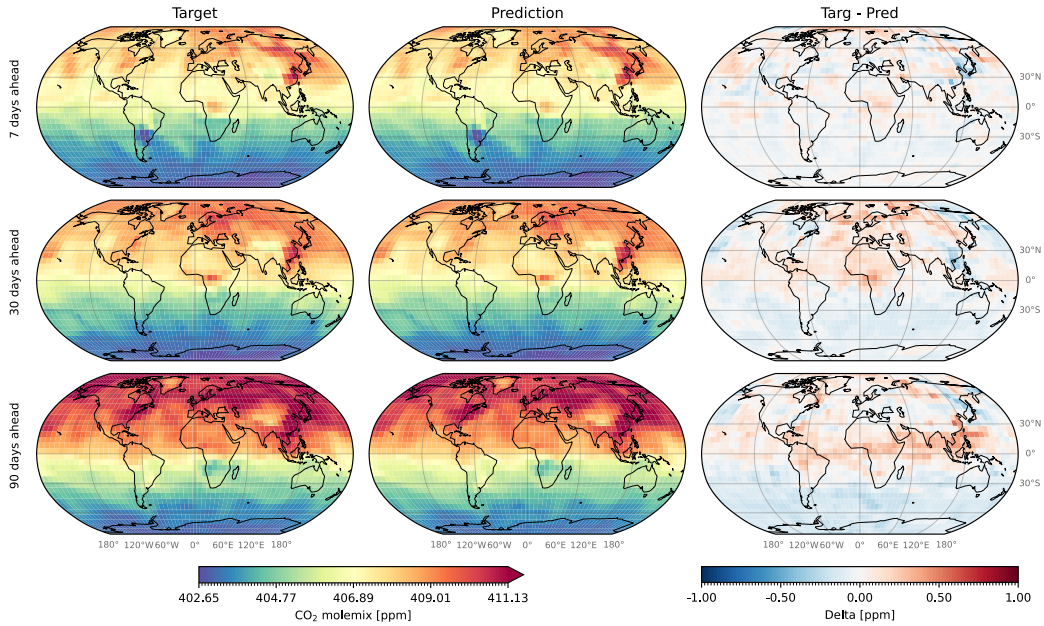

Figure 5. Maps of Total Column CO_2 Target, Prediction by SwinTransformer and Error for different lead times. Shown is a single forward run starting from 1 January 2018.

seasonal cycle in the observations well, whereas TM5 oversmoothes it. There are barely any jumps visible at the quarterly intervals (gray dotted lines), where the SwinTransformer initial state is reset. This is in line with the previous result, that SwinTransformer displays little performance degradation over 90 day horizons. While it is unclear exactly why SwinTransformer outperforms TM5 in Svalbard, this may be related to the stations vicinity to the poles and differences in the boundary layer vertical transport of the two models.

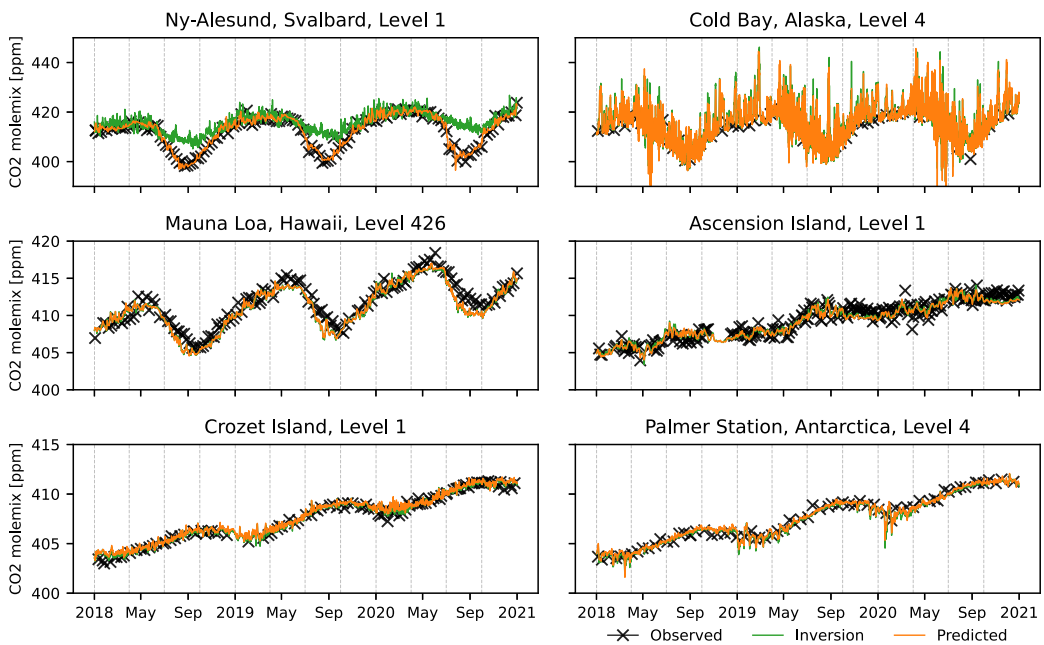

Figure 6. Performance of SwinTransformer (orange line) compared to TM5 (the training target, here: *Inversion*, green line) at six measurement stations from the ObsPack Globalview product. Shown are 90-day forward runs, the light gray lines indicate the dates on which the runs are reset.

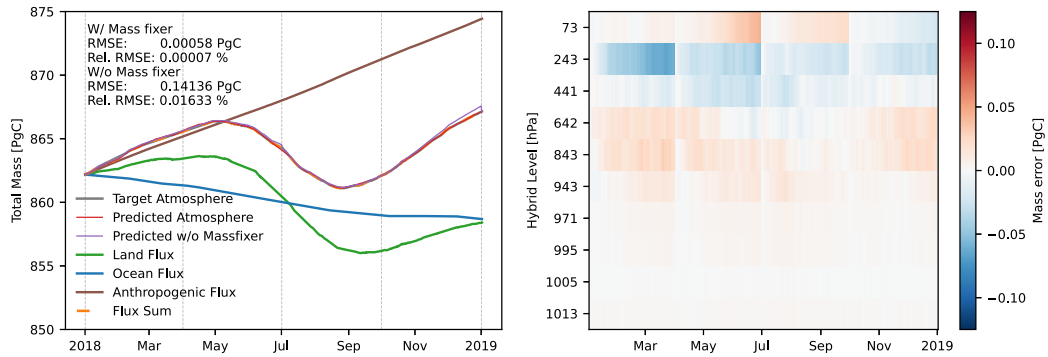

Figure 7. Mass Conservation of SwinTransformer globally (left) and per level (right). In the left panel, the total mass in the target atmosphere (gray line) and in the predicted atmosphere (red line) match exactly with a cumulative sum of the surface fluxes (orange dotted line), that is, they are plotted on top of each other indicating mass conservation. The flux sum is the sum of the Anthropogenic (brown), Land (green) and Ocean (blue) fluxes. In addition, we show performance without the massfixer (purple line). The right panel shows the difference of the total mass per level and time step between the SwinTransformer prediction (after applying the massfixer) and the target.

3.3. Mass Conservation

Figure 7 presents global and per-level mass conservation results with SwinTransformer. Globally SwinTransformer with the mass fixer achieves an RMSE of 0.00058 PgC, which may be considered negligible in comparison to the total atmospheric mass of ∼865 PgC in 2018. This remaining mass error likely stems from numerical problems: our deep neural networks operate with 32-bit floating points, which can give performance issues especially when dealing with division of relatively large numbers. Notably, the mass fixer greatly enhances the conservative properties of SwinTransformer in comparison to the free-running neural network (purple line, Figure 7 left side): it has over 0.01% relative mass RMSE, which particularly manifests in an overprediction of mass in november and december.

Analyzing the mass error per vertical layer gives insight into the vertical transport learned by SwinTransformer. Figure 7, right side, indicates that the upward vertical transport is too weak in northern hemisphere winter (too little mass in upper stratosphere) and too strong in summer. Notably, vertical transport in the lower layers close to the surface displays little mass error, albeit those layers being more heavily influenced by diurnal variability and surface fluxes.

3.4. Long-Term Stability

While this paper mostly focuses on prediction horizons up to 90 days, we also performed a 3-year rollout of the SwinTransformer over the full test period. SwinTransformer remains stable even after over 3 years rollout, but starts to display errors above 1 ppm in many regions (Figure 9).

More specifically, the surface layer RMSE first crosses 1 ppm after 217 days (Figure 8) and the RMSE near the surface generally displays cyclical behavior, with highest errors in northern hemisphere summer. The highest layer, representing the upper stratosphere, is unstable over rollout time: it is being oversmoothed and accumulates too little mass over time. For most inverse modeling purposes, this is of lesser concern, as the upper stratosphere contains less carbon and there are typically no direct measurements of CO_2 taken at such altitude.

Overall the results are particularly promising as previous work has repeatedly noted challenges in the stability of long-term rollouts of neural network-based PDE emulators (Bonev et al., 2023; Brandstetter et al., 2022; Lippe et al., 2023). Moreover, CO_2 transport may be considered particularly challenging as atmospheric CO_2 concentrations keep rising, naturally pushing the distribution of the atmospheric tracer field away from the training distribution and constituting an out-of-domain (OOD) problem. Still, future work needs to assess the robustness of our models to distribution shifts beyond the rise in CO_2 during the test set. For example, considering generalization to significantly different surface fluxes could be relevant. While preliminary experiments with transporting zeroed-out surface fluxes indicated no non-physical behavior, caution needs to preside and thus extrapolation far from training data may be a limitation of the transport emulator.

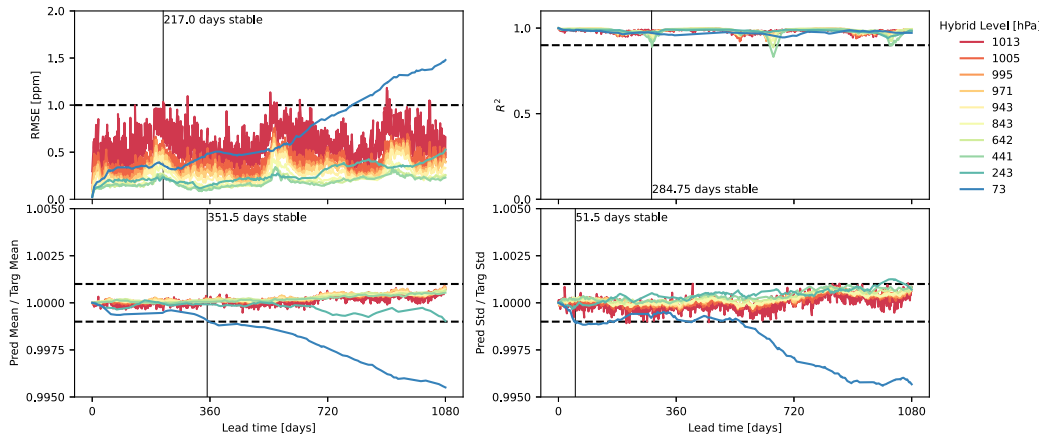

Figure 8. Key metrics per vertical layer for a single 3-year rollout with SwinTransformer starting from 1 January 2018. As in Figure 4, we report metrics per time step and vertical level. Dashed lines indicate arbitrarily set thresholds.

3.5. Differences Between AI Model Architectures

The four AI models included in this study build on different underlying principles (mesh-based vs. pseudo-spectral vs. brute-force). Hence it is less surprising that there are differences in the patterns of model residuals between models. Figure 10 presents RMSE patterns. For all models, RMSE seemingly scales with CO_2 variability: regions with large biosphere dynamics such as the tropics or boreal forests, and areas with large anthropogenic emissions such as eastern Asia in the near-surface layers have consistently larger errors. UNet, SFNO and GraphCast all have higher errors at the poles. For SFNO this is very limited to the pole grid cell itself, likely because the spherical harmonics there do not allow for zonal variability. For UNet, the impact is a bit larger, mirroring the smoothing effect of convolutions with zero padding at the poles. GraphCast has the most severe problems with the poles. This might be related to the encoder and decoder of GraphCast, which map between grid cells on the regular grid and nodes on the icosahedral mesh. Near the poles, many grid cells are mapped to a single node, which could potentially result in stability problems.

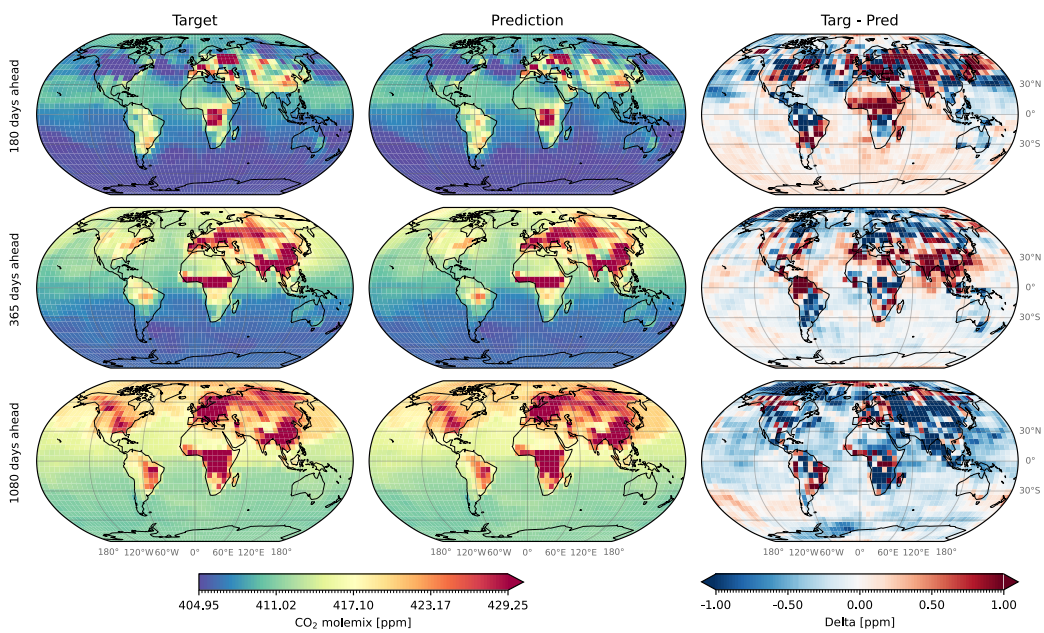

Figure 9. Maps of surface layer (1,013 hPa in a standard atmosphere) CO_2 Target, Prediction by SwinTransformer and Error for different lead times of a 3-year rollout starting from 1 January 2018.

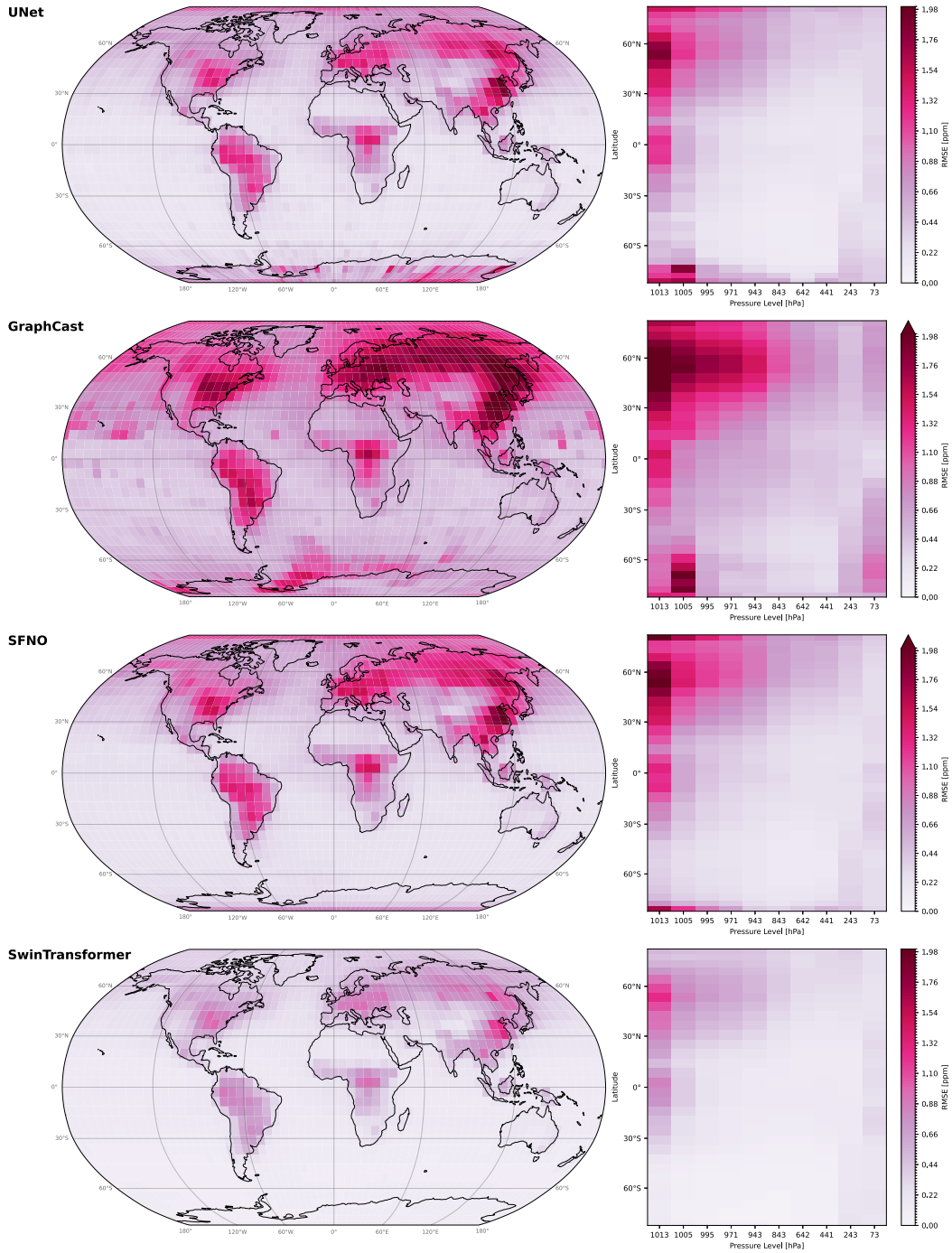

Figure 10. RMSE patterns of the four AI models. For each model, shown is the RMSE per horizontal grid cell averaged over time and vertical level (left side) and per latitude and vertical level averaged over time and longitude (right side). Scores are for quarterly 90-day forward runs.

3.6. Ablations

In our experiments we found the four AI models to not work very well for CO_2 prediction out-of-the-box. Especially the mesh-based methods UNet and GraphCast displayed issues with low stability over longer rollouts. In contrast, the final models presented in this paper are stable and mass-conserving for over 90 days. Table 1 presents insights into the design choices that lead to the improved performance on the LowRes data set. It reports

Table 1
Ablation Study Highlighting the Best Configuration Per Model Architecture (Underline) and the Best Overall Model (Bold)

Model	CentFlux	SpecLoss	No. of parameters (M)	Decorr time	R^2	RMSE
UNet S	✗	✗	9.6	1.5	0.07	>100
UNet S	✓	✗	9.6	>90	0.98	0.57
UNet S	✓	✓	9.6	>90	0.98	0.52
UNet XS	✓	✓	2.7	>90	0.98	0.62
UNet M	✓	✓	35.7	>90	0.98	0.52
GraphCast XS	✗	✗	5.2	41.25	0.87	1.63
GraphCast XS	✓	✗	5.2	>90	0.95	0.96
GraphCast XS	✓	✓	5.2	>90	0.96	0.86
GraphCast XXS	✓	✓	1.3	>90	0.95	0.92
GraphCast S	✓	✓	8.8	>90	0.96	0.87
GraphCast XS mesh = 0–2	✓	✓	5.2	>90	0.94	0.99
SFNO M	✗	✗	35.7	>90	0.97	0.67
SFNO M	✓	✗	35.7	>90	0.98	0.59
SFNO M	✓	✓	35.7	>90	0.98	0.58
SFNO S	✓	✓	8.9	>90	0.98	0.59
SFNO L	✓	✓	53.5	>90	0.98	0.59
SwinTransformer M	✗	✗	37.9	>90	0.97	0.79
SwinTransformer M	✓	✗	37.9	>90	0.99	0.37
SwinTransformer M	✓	✓	37.9	>90	0.99	0.34
SwinTransformer S	✓	✓	6.4	>90	0.99	0.36
SwinTransformer L	✓	✓	85.2	>90	0.99	0.34
SwinTransformer M $ps = 4$	✓	✓	38.8	>90	0.97	0.70

Note. For each model, we compare three different sizes, whether to center the input CO_2 field to account for covariate shift and to add surface fluxes directly to the lowest vertical layer (Centering and Flux Addition, i.e., *CentFlux*), and, whether to leverage an additional loss term which measures divergence in the spectral power densities (*SpecLoss*). For GraphCast, we additionally ablate the resolution of the icosahedral multi-mesh (*mesh*, default is 0–3), and for SwinTransformer, we ablate the patch size (*ps*, default is 1). We report three metrics: decorrelation time, R^2 and RMSE—all over 90-day forward runs.

metrics over 90 day forward runs. In our experiments the ranking of models remained consistent if only studying 7 day RMSE and R^2, except for models that develop instabilities (e.g., UNet or GraphCast).

For each of the four models, we ablate the model size and two training tricks that particularly increased the stability. The first one, *CentFlux*, is a combination of centering the 3D CO_2 fields and adding the prescribed surface fluxes to the lowest vertical layer. The second one, *SpecLoss*, is an additional loss term that penalizes deviations in the spectral power spectrum (computed with the spherical harmonic transform) between the model output and the target.

For all models, *CentFlux* is essential to achieve stable rollouts and improves the performance significantly. *SpecLoss* additionally enhances scores, but the gains are smaller. In fact, *SpecLoss* only marginally improves power spectral densities: Figure S1 in Supporting Information S1, we compare these for GraphCast, which displays the largest gain from *SpecLoss*, with only small improvements visible. As the additional loss term still improves error metrics, we keep it nonetheless.

The four models have different optimal model sizes. While the best GraphCast in our experiment (size XS), has 5.2 M parameters, the best UNet (size S) has 9.6 M, and the best SFNO and SwinTransformer (both size M) have 35.7 and 37.9 M parameters respectively. Note, models with more parameters do not necessarily have better performance: UNet outperforms SFNO slightly on RMSE. Still, that in our experiments it was significantly more challenging to scale GraphCast to larger size compared to SwinTransformer is probably one of the reasons why

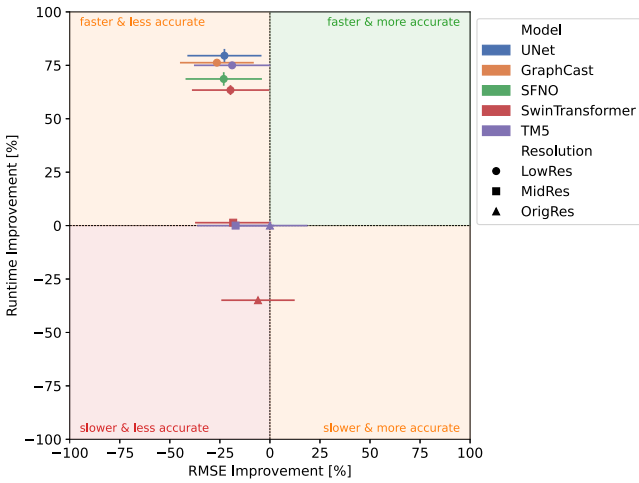

Figure 11. Pareto frontier of inference runtime versus model performance (RMSE at ObsPack Globalviewplus stations), plotted relative to the TM5 OrigRes target data. Runtime improvements are excluding IO and based on estimated TM5 runtimes.

GraphCast is the worst model architecture in our intercomparison. Finally, note that due to computational constraints we did not re-tune all hyperparameters for all ablations, but relied on one set of hyperparameters per model architecture. Doing this tuning could, in principle, change results.

3.7. Computational Costs

One reason why AI-based emulators of ERA5 have garnered interest is because they offer significant speed-ups over conventional NWP models at inference time. Conceptually these speed ups arise as most ERA5 emulators use 10× less vertical layers (13 instead of 137), 30× higher time step (6 hr instead of 12 min), purely explicit solvers (no iterative scheme for implicit steps necessary) and compute accelerators (GPUs/TPUs instead of CPUs). These speed ups at inference time come with a trade-off: first significant compute resources need to be allocated in order to train the models. For weather forecasting such an investment is often quickly justified, as many model runs are necessary every day, so after training a single AI-based emulator can be used many times.

Naturally, one may wonder if a parallel to neural network emulation of atmospheric transport can be drawn. However, the SwinTransformer is not significantly faster compared to TM5 (Figure 11). Performance here is highly hardware dependent, but as a rough estimate SwinTransformer takes ∼1.5sec for a 30 day forward run on a single Nvidia A40 GPU. Figure 11 compares the speed of the four AI models and TM5 at different resolutions. For this, we measured the model time in an idealized scenario, removing all pre- and postprocessing of model inputs and outputs, and instead directly reading and writing the raw tensors from memory. We then measure the speed of 30 day forecasts with 10 repetitions on a Nvidia A40 GPU. Generally, we notice only small differences between the AI models.

Running the TM5-MP model, which improves upon TM5 through OpenMPI parallelization (Williams et al., 2017), takes ∼8 minutes on a machine with 24 CPUs for a 1 month forward run on a 3° × 2° grid and ∼2 minutes on 6° × 4° (Segers et al., 2020). We assume 50% time is spent in IO and plot estimated runtimes for TM5 without IO in Figure 11, with OrigRes and MidRes runs to take 4 min and LowRes to take 1 min on a single modern machine with 24 CPUs.

The lack of speed-up can possibly be explained with a number of factors. First, TM5 is run on a 2° × 3° grid, which does not require an extremely small time step. Second, TM5 uses about the same number of vertical layers as SwinTransformer. Third, tracer transport in TM5 is entirely linear (in the surface fluxes), and the mass fluxes for each grid cell are pre-computed. After this is done, transport becomes cheap. Fourth, while TM5 still does not run on GPUs, it reaps a number of benefits from its maturity, such as leveraging fast FORTRAN code and parallelization through OpenMPI.

4. Discussion

In this work, we trained deep neural networks to emulate the atmospheric transport of CO_2. We test four models and find SwinTransformer to perform best, with almost perfect emulation for 90 days, and stable and mass-conserving emulation for multiple years ahead. For this we adjust the model architecture, decoupling the drift in CO_2 from its dynamics by leveraging centered CO_2 fields as inputs and using a post-hoc flux scheme to correct the mass balance. Yet, the presented model is not giving large computational advantages compared to conventional approaches, at least not at low resolution.

Storm-resolving models allow for explicit treatment of convection, with large impact on vertical transport of air masses and CO_2. Some modeling centers are already experimenting with storm-resolving transport model runs (Agustí-Panareda et al., 2014, 2022, 2023; Gelaro et al., 2015), which typically require to run an online transport model. Here, AI models could leverage model output and offer an alternative route ahead.

Considering higher resolution might offer room for speed ups: doubling the horizontal resolution of conventional solvers increases the computation costs by roughly 10× (Hoefler et al., 2023), partly due to a need for smaller time stepping. Yet, some of the errors of transport representation in current inverse modeling schemes are attributed to

low resolution (Agustí-Panareda et al., 2019; Remaud et al., 2018). Hence, developing multi-resolution training schemes, for example, by utilizing the cross attention mechanism (Alkin et al., 2024; Jaegle et al., 2021, 2022; Serrano et al., 2024), which is straight-forward with the data in CarbonBench, may enable more accurate low-resolution models that are still computationally feasible for inverse modeling by emulating the high resolution solvers. In other words, by leveraging high resolution training data, AI-based solvers could exhibit higher accuracy even if run at low resolution. Moreover, modeling the atmosphere in a highly compressed space may yield further improvements (Han et al., 2024), for instance, such a transport model could render the usage of full resolution wind fields from ERA5 feasible.

Furthermore, there is still a lot of room for common techniques used to speed up AI models. Model distillation is a technique to significantly reduce the parameter count of neural networks without loosing much in terms of skill. Quantization leverages lower numerical precision to decrease memory footprint and increase speed. On a programming language level, just-in-time compilation, for example, through torchscript, can speed up certain operations. And more generally, data loading can be optimized through asynchronous techniques, clever caching and parallelization.

Future work may also explore the applicability of the neural network solvers for inverse modeling, that is inferring surface fluxes from observed atmospheric measurements. The implementation of the neural networks is fully differentiable, which opens new avenues for obtaining the sensitivities required for the inversions. Furthermore, SwinTransformer already displays high stability in surface layers over at least 3 year forward runs and matches TM5 accuracy at measurement stations, which underlines its suitability as a forward model for inverse modeling, to be explored still is the robustness of its tangent-linear or adjoint, which would be required for variational inversions. Additionally, some inverse modeling approaches rely on the creation of large ensembles. Since neural networks natively support batched processing, there is potential for speed ups (generating a full ensemble can be as cheap as a single forward run).

Data Availability Statement

We construct the CarbonBench data set from existing open data from CarbonTracker North America version CT2022 (Jacobson et al., 2023) (http://doi.org/10.25925/z1gj-3254) and from ObsPack GLOBALVIEWplus CO_2 v9.1 (Schuldt et al., 2023) (http://doi.org/10.25925/20231201). We provide code that downloads the data efficiently from the original data providers and processes it into the formats used in this study, yet in line with the original data licenses we do not re-distribute the data. The software to run all our experiments and reproduce the results in this paper is archived in Benson (2024). The code consists in the CarbonBench Python repo (https://github.com/vitusbenson/carbonbench), with the deep neural networks implemented in the Neural Transport Python library (https://github.com/vitusbenson/neural_transport), a versatile software package containing data set creation, data loading, training and evaluation routines intended to be easily usable in other research projects.

Acknowledgments
VB is grateful for stimulating discussions to Fabian Gans, Maximilian Gelbrecht, Martin Heimann, Martin Jung, Nick McGreivy, Albrecht Schall, Sam Upton and many others at MPI Jena and ETH Zürich. AW, CR, and MR acknowledge funding by the European Research Council (ERC) Synergy Grant Understanding and modeling the Earth System with Machine Learning (USMILE) under the Horizon 2020 research and innovation programme (Grant agreement No. 855187). Open Access funding enabled and organized by Projekt DEAL.

References

Agustí-Panareda, A., Barré, J., Massart, S., Inness, A., Aben, I., Ades, M., et al. (2023). Technical note: The CAMS greenhouse gas reanalysis from 2003 to 2020. *Atmospheric Chemistry and Physics*, *23*(6), 3829–3859. https://doi.org/10.5194/acp-23-3829-2023

Agusti-Panareda, A., Diamantakis, M., Bayona, V., Klappenbach, F., & Butz, A. (2017). Improving the inter-hemispheric gradient of total column atmospheric CO_2 and CH_4 in simulations with the ECMWF semi-Lagrangian atmospheric global model. *Geoscientific Model Development*, *10*(1), 1–18. https://doi.org/10.5194/gmd-10-1-2017

Agustí-Panareda, A., Diamantakis, M., Massart, S., Chevallier, F., Muñoz-Sabater, J., Barré, J., et al. (2019). Modelling CO_2 weather – Why horizontal resolution matters. *Atmospheric Chemistry and Physics*, *19*(11), 7347–7376. https://doi.org/10.5194/acp-19-7347-2019

Agustí-Panareda, A., Massart, S., Chevallier, F., Boussetta, S., Balsamo, G., Beljaars, A., et al. (2014). Forecasting global atmospheric CO_2. *Atmospheric Chemistry and Physics*, *14*(21), 11959–11983. https://doi.org/10.5194/acp-14-11959-2014

Agustí-Panareda, A., McNorton, J., Balsamo, G., Baier, B. C., Bousserez, N., Boussetta, S., et al. (2022). Global nature run data with realistic high-resolution carbon weather for the year of the Paris Agreement. *Scientific Data*, *9*(1), 160. https://doi.org/10.1038/s41597-022-01228-2

Alkin, B., Fürst, A., Schmid, S., Gruber, L., Holzleitner, M., & Brandstetter, J. (2024). *Universal physics transformers: A framework for efficiently scaling neural operators* (No. arXiv:2402.12365). arXiv. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2024/hash/2cd36d327f33d47b372d4711edd08de0-Abstract-Conference.html

Arcomano, T., Szunyogh, I., Wikner, A., Pathak, J., Hunt, B. R., & Ott, E. (2022). A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model. *Journal of Advances in Modeling Earth Systems*, *14*(3), e2021MS002712. https://doi.org/10.1029/2021MS002712

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., et al. (2018). *Relational inductive biases, deep learning, and graph networks* (No. arXiv:1806.01261). arXiv. https://doi.org/10.48550/arXiv.1806.01261

Belikov, D. A., Maksyutov, S., Krol, M., Fraser, A., Rigby, M., Bian, H., et al. (2013). Off-line algorithm for calculation of vertical tracer transport in the troposphere due to deep convection. *Atmospheric Chemistry and Physics*, *13*(3), 1093–1114. https://doi.org/10.5194/acp-13-1093-2013

Benson, V. (2024). Code for Benson et al. (2024) - Atmospheric transport modeling of CO_2 with neural networks [Dataset]. *Zenodo*. https://doi.org/10.5281/zenodo.14502316

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2022). *Pangu-weather: A 3D high-resolution model for fast and accurate global weather forecast* (No. arXiv:2211.02556). arXiv. Retrieved from https://www.nature.com/articles/s41586-023-06185-3

Bodnar, C., Bruinsma, W. P., Lucic, A., Stanley, M., Brandstetter, J., Garvan, P., et al. (2024). *Aurora: A foundation model of the atmosphere* (No. arXiv:2405.13063). arXiv. https://doi.org/10.48550/arXiv.2405.13063

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., & Anandkumar, A. (2023). *Spherical Fourier neural operators: Learning stable dynamics on the sphere* (No. arXiv:2306.03838). arXiv. Retrieved from https://proceedings.mlr.press/v202/bonev23a.html

Brandstetter, J., Worrall, D. E., & Welling, M. (2022). Message passing neural PDE solvers. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=vSix3HPYKSU

Brasseur, G. P., & Jacob, D. J. (2017). *Modeling of atmospheric chemistry* (1st ed.). Cambridge University Press. https://doi.org/10.1017/9781316544754

Cartwright, L., Zammit-Mangion, A., & Deutscher, N. M. (2023). Emulation of greenhouse-gas sensitivities using variational autoencoders. *Environmetrics*, *34*(2), e2754. https://doi.org/10.1002/env.2754

Chandra, N., Patra, P. K., Niwa, Y., Ito, A., Iida, Y., Goto, D., et al. (2022). Estimated regional CO_2 flux and uncertainty based on an ensemble of atmospheric CO_2 inversions. *Atmospheric Chemistry and Physics*, *22*(14), 9215–9243. https://doi.org/10.5194/acp-22-9215-2022

Chattopadhyay, A., & Hassanzadeh, P. (2023). *Long-term instabilities of deep learning-based digital twins of the climate system: The cause and a solution* (No. arXiv:2304.07029). arXiv. https://doi.org/10.48550/arXiv.2304.07029

Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.-J., et al. (2023). *FengWu: Pushing the skillful global medium-range weather forecast beyond 10 days lead* (No. arXiv:2304.02948). arXiv. https://doi.org/10.48550/arXiv.2304.02948

Chevallier, F., Fisher, M., Peylin, P., Serrar, S., Bousquet, P., Bréon, F.-M., et al. (2005). Inferring CO_2 sources and sinks from satellite observations: Method and application to TOVS data. *Journal of Geophysical Research*, *110*(D24), D24309. https://doi.org/10.1029/2005JD006390

Chevallier, F., Lloret, Z., Cozic, A., Takache, S., & Remaud, M. (2023). Toward high-resolution global atmospheric inverse modeling using graphics accelerators. *Geophysical Research Letters*, *50*(5), e2022GL102135. https://doi.org/10.1029/2022GL102135

Chevallier, F., Viovy, N., Reichstein, M., & Ciais, P. (2006). On the assignment of prior errors in Bayesian inversions of CO_2 surface fluxes. *Geophysical Research Letters*, *33*(13), L13802. https://doi.org/10.1029/2006GL026496

Ciais, P., Rayner, P., Chevallier, F., Bousquet, P., Logan, M., Peylin, P., & Ramonet, M. (2011). Atmospheric inversions for estimating CO_2 fluxes: Methods and perspectives. In M. Jonas, Z. Nahorski, S. Nilsson, & T. Whiter (Eds.), *Greenhouse gas inventories: Dealing with uncertainty* (pp. 69–92). Springer Netherlands. https://doi.org/10.1007/978-94-007-1670-4_6

Diamantakis, M., & Flemming, J. (2014). Global mass fixer algorithms for conservative tracer transport in the ECMWF model. *Geoscientific Model Development*, *7*(3), 965–979. https://doi.org/10.5194/gmd-7-965-2014

Eastham, S. D., & Jacob, D. J. (2017). Limits on the ability of global Eulerian models to resolve intercontinental transport of chemical plumes. *Atmospheric Chemistry and Physics*, *17*(4), 2543–2553. https://doi.org/10.5194/acp-17-2543-2017

Fillola, E., Santos-Rodriguez, R., Manning, A., O'Doherty, S., & Rigby, M. (2022). A machine learning emulator for Lagrangian particle dispersion model footprints: A case study using NAME. *EGUsphere*, 1–19. Retrieved from https://gmd.copernicus.org/articles/16/1997/2023/

Friedlingstein, P., O'Sullivan, M., Jones, M. W., Andrew, R. M., Bakker, D. C. E., Hauck, J., et al. (2023). Global carbon budget 2023. *Earth System Science Data*, *15*(12), 5301–5369. https://doi.org/10.5194/essd-15-5301-2023

Gaubert, B., Stephens, B. B., Basu, S., Chevallier, F., Deng, F., Kort, E. A., et al. (2019). Global atmospheric CO_2 inverse models converging on neutral tropical land exchange, but disagreeing on fossil fuel and atmospheric growth rate. *Biogeosciences*, *16*(1), 117–134. https://doi.org/10.5194/bg-16-117-2019

Gelaro, R., Putman, W. M., Pawson, S., Draper, C., Molod, A., Norris, P. M., et al. (2015). *Evaluation of the 7-km GEOS-5 Nature Run* (Technical Report No. 36). NTRS - NASA Technical Reports Server. Retrieved from https://ntrs.nasa.gov/citations/20150011486

Gurney, K. R., Law, R. M., Denning, A. S., Rayner, P. J., Baker, D., Bousquet, P., et al. (2002). Towards robust regional estimates of CO_2 sources and sinks using atmospheric transport models. *Nature*, *415*(6872), 626–630. https://doi.org/10.1038/415626a

Han, T., Chen, Z., Guo, S., Xu, W., & Bai, L. (2024). *CRA5: Extreme compression of ERA5 for portable global climate and weather research via an efficient variational transformer* (No. arXiv:2405.03376). arXiv. https://doi.org/10.48550/arXiv.2405.03376

He, T.-L., Dadheech, N., Thompson, T. M., & Turner, A. J. (2023). FootNet: Development of a machine learning emulator of atmospheric transport. Retrieved from https://eartharxiv.org/repository/view/6392/

Heimann, H., & Körner, S. (2003). *The global atmospheric tracer model TM3* (Technical Report 5, p. 131). Max-Planck-Institut für Biogeochemie. Retrieved from https://www.bgc-jena.mpg.de/archived/bgc-systems/bgc-systems/uploads/Publications/5.pdf

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., et al. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049. https://doi.org/10.1002/qj.3803

Hoefler, T., Stevens, B., Prein, A. F., Baehr, J., Schulthess, T., Stocker, T. F., et al. (2023). Earth virtualization engines: A technical perspective. *Computing in Science & Engineering*, *25*(3), 50–59. https://doi.org/10.1109/MCSE.2023.3311148

Jacobson, A. R., Schuldt, K. N., Tans, P., Andrews, A., Miller, J. B., Oda, T., et al. (2023). *CarbonTracker CT2022*. NOAA Global Monitoring Laboratory. https://doi.org/10.25925/Z1GJ-3254

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., et al. (2022). Perceiver IO: A general architecture for structured inputs & outputs. arXiv:2107.14795 [cs, eess]. Retrieved from https://openreview.net/forum?id=fILj7WpI-g

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). Perceiver: General perception with iterative attention. arXiv:2103.03206 [cs, eess]. Retrieved from https://proceedings.mlr.press/v139/jaegle21a.html

Jin, Y., Keeling, R. F., Stephens, B. B., Long, M. C., Patra, P. K., Rödenbeck, C., et al. (2024). Improved atmospheric constraints on Southern Ocean CO_2 exchange. *Proceedings of the National Academy of Sciences*, *121*(6), e2309333121. https://doi.org/10.1073/pnas.2309333121

Jung, M., Reichstein, M., Margolis, H. A., Cescatti, A., Richardson, A. D., Arain, M. A., et al. (2011). Global patterns of land-atmosphere fluxes of carbon dioxide, latent heat, and sensible heat derived from eddy covariance, satellite, and meteorological observations. *Journal of Geophysical Research*, *116*(G3), G00J07. https://doi.org/10.1029/2010JG001566

Jung, M., Schwalm, C., Migliavacca, M., Walther, S., Camps-Valls, G., Koirala, S., et al. (2020). Scaling carbon fluxes from eddy covariance sites to globe: Synthesis and evaluation of the FLUXCOM approach. *Biogeosciences*, *17*(5), 1343–1365. https://doi.org/10.5194/bg-17-1343-2020

Kaminski, T., & Heimann, M. (2001). Inverse modeling of atmospheric carbon dioxide fluxes. *Science*, *294*(5541), 259. https://doi.org/10.1126/science.294.5541.259a

Keisler, R. (2022). Forecasting global weather with graph neural networks. arXiv:2202.07575 [physics]. Retrieved from http://arxiv.org/abs/2202.07575

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., et al. (2023). *Neural general circulation models* (No. arXiv: 2311.07222). arXiv. Retrieved from https://www.nature.com/articles/s41586-024-07744-y

Krasnopolsky, V. M., & Fox-Rabinovitz, M. S. (2006). Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, *19*(2), 122–134. https://doi.org/10.1016/j.neunet.2006.01.002

Kretschmer, R., Gerbig, C., Karstens, U., & Koch, F.-T. (2012). Error characterization of CO_2 vertical mixing in the atmospheric transport model WRF-VPRM. *Atmospheric Chemistry and Physics*, *12*(5), 2441–2458. https://doi.org/10.5194/acp-12-2441-2012

Krol, M., Houweling, S., Bregman, B., van den Broek, M., Segers, A., van Velthoven, P., et al. (2005). The two-way nested global chemistry-transport zoom model TM5: Algorithm and applications. *Atmospheric Chemistry and Physics*, *5*(2), 417–432. https://doi.org/10.5194/acp-5-417-2005

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., et al. (2023). Learning skillful medium-range global weather forecasting. *Science*, *382*(6677), 1416–1421. https://doi.org/10.1126/science.adi2336

Lessig, C., Luise, I., Gong, B., Langguth, M., Stadtler, S., & Schultz, M. (2023). *AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning* (No. arXiv:2308.13280). arXiv. https://doi.org/10.48550/arXiv.2308.13280

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=c8P9NQVtmnO

Lippe, P., Veeling, B., Perdikaris, P., Turner, R., & Brandstetter, J. (2023). PDE-refiner: Achieving accurate long rollouts with neural PDE solvers. *Advances in Neural Information Processing Systems*, *36*, 67398–67433. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., et al. (2022). Swin transformer V2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12009–12019). Retrieved from https://openaccess.thecvf.com/content/CVPR2022/html/Liu_Swin_Transformer_V2_Scaling_Up_Capacity_and_Resolution_CVPR_2022_paper.html

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 10012–10022). Retrieved from https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper

McNicol, G., Fluet-Chouinard, E., Ouyang, Z., Knox, S., Zhang, Z., Aalto, T., et al. (2023). Upscaling wetland methane emissions from the FLUXNET-CH4 eddy covariance network (UpCH4 v1.0): Model development, network assessment, and budget comparison. *AGU Advances*, *4*(5), e2023AV000956. https://doi.org/10.1029/2023AV000956

Munassar, S., Monteil, G., Scholze, M., Karstens, U., Rödenbeck, C., Koch, F.-T., et al. (2023). Why do inverse models disagree? A case study with two European CO_2 inversions. *Atmospheric Chemistry and Physics*, *23*(4), 2813–2828. https://doi.org/10.5194/acp-23-2813-2023

Nelson, J. A., Walther, S., Gans, F., Kraft, B., Weber, U., Novick, K., et al. (2024). X-BASE: The first terrestrial carbon and water flux products from an extended data-driven scaling framework, FLUXCOM-X. *EGUsphere*, 1–51. Retrieved from https://bg.copernicus.org/articles/21/5079/2024/

Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., & Grover, A. (2023). *ClimaX: A foundation model for weather and climate* (No. arXiv: 2301.10343). arXiv. https://doi.org/10.48550/arXiv.2301.10343

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., et al. (2022). *FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators* (No. arXiv:2202.11214). arXiv. https://dl.acm.org/doi/10.1145/3592979.3593412

Patra, P. K., Takigawa, M., Watanabe, S., Chandra, N., Ishijima, K., & Yamashita, Y. (2018). Improved chemical tracer simulation by MIROC4.0-based atmospheric chemistry-transport model (MIROC4-ACTM). *Sola*, *14*(0), 91–96. https://doi.org/10.2151/sola.2018-016

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

Price, I., Sanchez-Gonzalez, A., Alet, F., Ewalds, T., El-Kadi, A., Stott, J., et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather* (No. arXiv:2312.15796). arXiv. Retrieved from https://www.nature.com/articles/s41586-024-08252-9

Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., & Thuerey, N. (2020). WeatherBench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, *12*(11), e2020MS002203. https://doi.org/10.1029/2020MS002203

Rasp, S., Hoyer, S., Merose, A., Langmore, I., Battaglia, P., Russell, T., et al. (2024). WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, *16*(6), e2023MS004019. https://doi.org/10.1029/2023MS004019

Remaud, M., Chevallier, F., Cozic, A., Lin, X., & Bousquet, P. (2018). On the impact of recent developments of the LMDz atmospheric general circulation model on the simulation of CO_2 transport. *Geoscientific Model Development*, *11*(11), 4489–4513. https://doi.org/10.5194/gmd-11-4489-2018

Remaud, M., Ma, J., Krol, M., Abadie, C., Cartwright, M. P., Patra, P., et al. (2023). Intercomparison of atmospheric carbonyl sulfide (TransCom-COS; Part one): Evaluating the impact of transport and emissions on tropospheric variability using ground-based and aircraft data. *Journal of Geophysical Research: Atmospheres*, *128*(6), e2022JD037817. https://doi.org/10.1029/2022JD037817

Rödenbeck, C. (2005). *Estimating CO_2 sources and sinks from atmospheric concentration measurements using a global inversion of atmospheric transport* (Technical Report 6, p. 53). Max-Planck-Institut für Biogeochemie. Retrieved from https://www.bgc-jena.mpg.de/archived/bgc-systems/bgc-systems/uploads/Publications/6.pdf

Rödenbeck, C., Houweling, S., Gloor, M., & Heimann, M. (2003). CO_2 flux history 1982–2001 inferred from atmospheric data using a global inversion of atmospheric transport. *Atmospheric Chemistry and Physics*, *3*(6), 1919–1964. https://doi.org/10.5194/acp-3-1919-2003

Rödenbeck, C., Zaehle, S., Keeling, R., & Heimann, M. (2018). History of El Niño impacts on the global carbon cycle 1957–2017: A quantification from atmospheric CO_2 data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *373*(1760), 20170303. https://doi.org/10.1098/rstb.2017.0303

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, *45*(22), 12616–12622. https://doi.org/10.1029/2018GL080704

Schuh, A. E., & Jacobson, A. R. (2023). Uncertainty in parameterized convection remains a key obstacle for estimating surface fluxes of carbon dioxide. *Atmospheric Chemistry and Physics*, *23*(11), 6285–6297. https://doi.org/10.5194/acp-23-6285-2023

Schuh, A. E., Jacobson, A. R., Basu, S., Weir, B., Baker, D., Bowman, K., et al. (2019). Quantifying the impact of atmospheric transport uncertainty on CO_2 surface flux estimates. *Global Biogeochemical Cycles*, *33*(4), 484–500. https://doi.org/10.1029/2018GB006086

Schuldt, K. N., Mund, J., Aalto, T., Abshire, J. B., Aikin, K., Allen, G., et al. (2023). *Multi-laboratory compilation of atmospheric carbon dioxide data for the period 1957–2022; obspack_co2_1_GLOBALVIEWplus_v9.1_2023-12-08*. NOAA Global Monitoring Laboratory. https://doi.org/10.25925/20231201

Segers, A., Tokaya, J., Houweling, S., van Peet, J., & Huijnen, V. (2020). TM5-MP-4DVAR, AND SOMETHING ON CH4 SINKS. Retrieved from https://www2.projects.science.uu.nl/tm5/TM5_PW/TM5_presentations_Oct2020/segers.pdf

Serrano, L., Wang, T. X., Naour, E. L., Vittaut, J.-N., & Gallinari, P. (2024). *AROMA: Preserving spatial structure for latent PDE modeling with local neural fields* (No. arXiv:2406.02176). arXiv. Retrieved from https://papers.nips.cc/paper_files/paper/2024/hash/185a120a3f709187e68bd092e6098851-Abstract-Conference.html

Tramontana, G., Jung, M., Schwalm, C. R., Ichii, K., Camps-Valls, G., Ráduly, B., et al. (2016). Predicting carbon dioxide and energy fluxes across global FLUXNET sites with regression algorithms. *Biogeosciences*, *13*(14), 4291–4313. https://doi.org/10.5194/bg-13-4291-2016

Upton, S., Reichstein, M., Gans, F., Peters, W., Kraft, B., & Bastos, A. (2024). Constraining biospheric carbon dioxide fluxes by combined top-down and bottom-up approaches. *Atmospheric Chemistry and Physics*, *24*(4), 2555–2582. https://doi.org/10.5194/acp-24-2555-2024

van der Laan-Luijkx, I. T., van der Velde, I. R., van der Veen, E., Tsuruta, A., Stanislawska, K., Babenhauserheide, A., et al. (2017). The CarbonTracker Data Assimilation Shell (CTDAS) v1.0: Implementation and global carbon balance 2001–2015. *Geoscientific Model Development*, *10*(7), 2785–2800. https://doi.org/10.5194/gmd-10-2785-2017

Willard, J. D., Harrington, P., Subramanian, S., Mahesh, A., O'Brien, T. A., & Collins, W. D. (2024). *Analyzing and exploring training recipes for large-scale transformer-based weather prediction* (No. arXiv:2404.19630). arXiv. Retrieved from http://arxiv.org/abs/2404.19630

Williams, J. E., Boersma, K. F., Le Sager, P., & Verstraeten, W. W. (2017). The high-resolution version of TM5-MP for optimized satellite retrievals: Description and validation. *Geoscientific Model Development*, *10*(2), 721–750. https://doi.org/10.5194/gmd-10-721-2017

Williamson, D. L. (1992). Review of numerical approaches for modeling global transport. In H. Van Dop & G. Kallos (Eds.), *Air Pollution Modeling and Its Application IX* (pp. 377–394). Springer US. https://doi.org/10.1007/978-1-4615-3052-7_38

Yu, K., Keller, C. A., Jacob, D. J., Molod, A. M., Eastham, S. D., & Long, M. S. (2018). Errors and improvements in the use of archived meteorological data for chemical transport modeling: An analysis using GEOS-Chem v11-01 driven by GEOS-5 meteorology. *Geoscientific Model Development*, *11*(1), 305–319. https://doi.org/10.5194/gmd-11-305-2018

Zhang, B., Liu, H., Crawford, J. H., Chen, G., Fairlie, T. D., Chambers, S., et al. (2021). Simulation of radon-222 with the GEOS-Chem global model: Emissions, seasonality, and convective transport. *Atmospheric Chemistry and Physics*, *21*(3), 1861–1887. https://doi.org/10.5194/acp-21-1861-2021