Accelerating GHG Emissions Inference: A Lagrangian Particle Dispersion Model Emulator Using Graph Neural Networks

Elena Fillola
Department of Engineering Mathematics
University of Bristol, UK
elena.fillolamayoral@bristol.ac.uk

Raul Santos-Rodriguez
Department of Computer Science
University of Bristol, UK
elena.fillolamayoral@bristol.ac.uk

Matt Rigby
Atmospheric Chemistry Research Group
University of Bristol, UK
elena.fillolamayoral@bristol.ac.uk

Abstract

Inverse modelling systems relying on Lagrangian Particle Dispersion Models (LPDMs) are a popular way to quantify greenhouse gas (GHG) emissions using atmospheric observations, providing independent validation to countries' self-reported emissions. However, the increased volume of satellite measurements cannot be fully leveraged due to computational bottlenecks. Here, we propose a data-driven architecture with Graph Neural Networks that emulates the outputs of LPDMs using only meteorological inputs, and demonstrate it in application with preliminary results for satellite measurements over Brazil.

1 Introduction

Monitoring the accuracy of GHG emissions reporting is key to evaluate the efficacy of climate agreements and support data-driven policymaking. Following IPCC guidelines, countries report their emissions through national greenhouse gas inventories (NGHGIs), aggregating activity data from sources and sectors together with country-specific emission factors. "Top-down" methods present an alternative approach, quantifying the movement of GHGs (GHG fluxes) using atmospheric observations. Incorporating real GHG measurements into the inference process enhances transparency and accountability in climate change mitigation efforts by independently verifying self-reported emissions, providing constraints on their uncertainties and identifying potential discrepancies.

Within the methods for GHG flux quantification, the IPCC recommends inverse modelling systems in particular for comparing inventories against atmospheric measurements [1]. These models produce estimates of the spatial distribution of GHG sources from observations, using an atmospheric transport model and a statistical inversion framework. They have been used, for example, to evaluate methane emissions in Europe using in situ sensors [2] and in India using satellite measurements [3], and they were key to identifing the origin of unexpected CFC-11 emissions from eastern China [4], as well as being used annually by the UK and Switzerland to evaluate their own emissions inventories [5].

These inference systems mentioned above were originally designed for relatively small datasets using in situ measurements (thousands of global observations per month), but new satellite-based instruments have increased the volume of GHG measurements by several orders of magnitude (millions of measurements per day). This growth is causing severe computational bottlenecks for GHG flux inference systems, particularly those relying on backward running Lagrangian Particle

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2023.

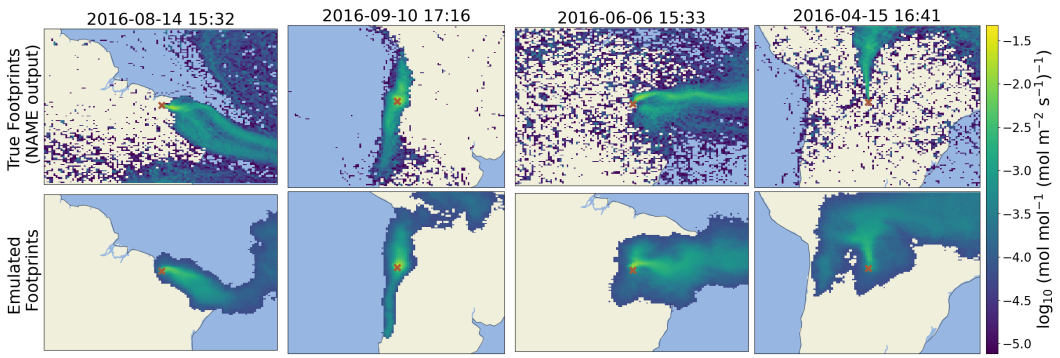


Figure 1: Samples of LPDM-generated footprints (top row) and the corresponding emulated footprints (bottom row), for an area of size $\approx 3300 \times 2500$ km over Brazil. The date in each column and the red cross in the centre of each image show when and where the satellite measurement was taken, and the footprint indicates the area to which that particular measurement is sensitive to. Note the $\log_{10}$ scale.

Dispersion Models (LPDMs) to solve for atmospheric transport. LPDMs simulate gas "particles" moving backwards in time from a GHG measurement location, producing a 2D histogram, known as a *footprint*, which indicates the upwind areas where emissions would contribute to the observed GHG measurement. To alleviate the computational requirements, existing approaches rely on approximating LPDM footprints mostly by interpolating from nearby footprints [6, 7] or smoothing of lower resolution simulations [8]. Currently, only [9] have developed a proof-of-concept emulator that can output footprints solely from meteorology, and only [6] address directly footprints for satellite measurements, which are more complex to produce.

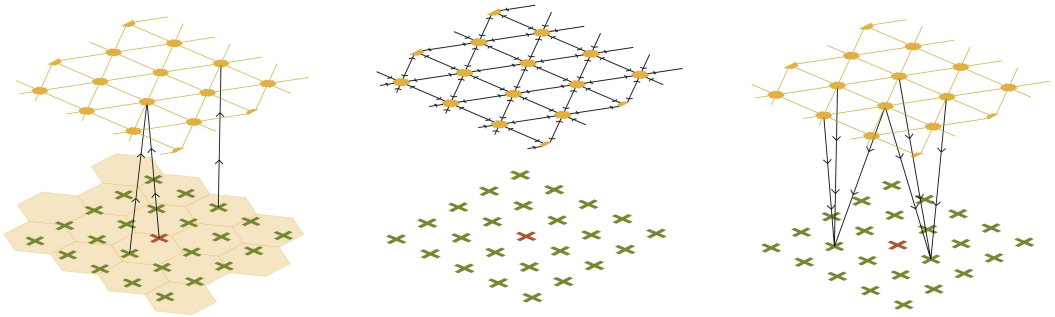
In this paper, we propose a Graph Neural Network to output satellite footprints for any location in a domain using only meteorological and topographical inputs, without needing nearby footprints or further simulations for new measurements. Such a data-driven emulator could leverage the potential of satellite data, potentially enabling near real-time, global emissions inference, improving transparency and supporting countries without the necessary measurement infrastructure or computational resources to evaluate their emissions reports. We demonstrate and evaluate preliminary results for satellite measurements over Brazil, and describe further steps to impact.

2 Method

Our data-driven approach to LPDM emulation is based on previous Machine Learning (ML) approaches to meteorological forecasting, in particular [10] and [11], using message-passing Graph Neural Networks (GNN) in an Encode-Process-Decode architecture. Meteorological forecasting and particle dispersion simulations are closely related problems, sharing limitations and input data, but there are some key differences. One of the main changes in implementation with respect to traditional ML forecasting is that rather than using a fixed domain (i.e. the area covered by inputs and outputs is always the same), we choose a moving domain setup, where the latitude-longitude grid for the inputs and outputs is a square of side S centered around the measurement location.

Model The model is formed of three components, shown in figure 2: an Encoder, a Processor and a Decoder. Meteorological data and other variables like topography are arranged in a latitude-longitude grid, that represents the native data space. This data gets Encoded into an abstract mesh composed of equilateral triangles. The data is Processed in this abstract layer, and then gets Decoded back to the original data space, emulating the footprint value at each grid node. Representing the data as a graph encourages flexibility in the setup: the input and output lat-lon grids don't need to have the same resolution, and the training can be done on a domain size S^{train} different than the testing size S^{test}.

In the Encoder (figure 2a), the input data for the grid nodes closest to each mesh node is aggregated using some function and mapped to the mesh node using a multilayer perceptron (MLP). This function should be permutation invariant and accept a variable number of inputs (e.g. mean or maximum, here we use a distance-weighted mean). Another MLP encodes distance information in the mesh edges.



(a) Encoder (b) Processor (c) Decoder

Figure 2: Local graph connectivity in the Encoder, Processor, and Decoder. Green crosses represent nodes in the native data space (latitude-longitude grid) and yellow hexagons represent nodes in the abstract mesh layer. Black arrows show information flow. The lat-lon grid is a square of side S (in the image, $S = 5$) centred around the measurement point, shown as a red cross. a) Local information is encoded into the nodes in the abstract layer using data from the closest grid nodes. Light yellow projection shows the area of influence of each mesh node. b) Data encoded in the mesh nodes is updated iteratively, processing information from neighbouring edges and nodes. c) The information in the mesh nodes is decoded into the output value at each of the lat-lon coordinates, by aggregating information from the N closest mesh nodes (here, $N = 3$).

In the Processor (figure 2b), we implement a message-passing GNN only in the abstract layer. Message-passing GNNs, or graph networks, were defined in [12] and act as a graph-to-graph module. Each message-passing block is built of two MLPs: one that updates edge features using the neighbouring nodes and another MLP that updates node features based on a node's neighbouring edges and nodes (including itself). The Processor is built as a sequence of independent message-passing blocks, which spread the information across neighbours.

In the Decoder (figure 2c), the final encodings of the mesh nodes are mapped back to the native data space. The N closest mesh nodes to each grid node are aggregated and passed to a decoding MLP, which outputs the footprint value at that lat-lon coordinate, performing node-level regression.

Metrics and evaluation The emulated footprints can be evaluated at four levels: on a pixel by pixel level (e.g. MSE), on a footprint level (e.g. image similarity and segmentation metrics), on a concentration level by convolving with the expected emissions in that area (see Appendix A), and at an implementation level, using them for GHG quantification and comparing the estimated emissions to those estimated with LPDM footprints. On a more qualitative side, the emulator can also be evaluated on the balance achieved between performance and computational expense. For example, [10] and [11] present results that are comparable or outperform the physical model, but train on 30+ years of data. Our goal is devising an emulator that can produce footprints that infer emissions within uncertainty of the physics-based model, but minimise the computing and data requirements.

3 Application: emissions over Brazil

Data We use a subset of the footprints presented in [13], produced by NAME, the UK's Met Office LPDM, [14] for GOSAT measurements over Brazil for 2014, 2015 and 2016 (GOSAT stands for Greenhouse gases Observing SATellite, launched in 2009 [15]). See [3] for more information on how footprints for satellite are generated. At each node in the lat-lon grid, the model takes as inputs global meteorology fields at different pressure levels and topography (extracted from the UK Met Office's Unified Model [16, 17]) as well as location variables, including the coordinates of each node and the distance from the measurement point. The input and output lat-lon grids have the same resolution of $0.352° \times 0.234°$ ($\approx 33 \times 25$km). We sample the meteorology at the time of the measurement as well as six and twelve hours before this, and use this as inputs to the network.

Training and calibration We train the model with a square lat-lon grid of side $S^{train} = 50$ gridcells centered around the measurement point (covering an area of $\approx 1650 \times 1250$ km) to reduce

memory usage and computational expense, and we test on a grid of size $S^{test} = 100$ gridcells at the same resolution, therefore covering an area of $\approx$ 3300$\times$2500 km. We reduce the data volume by sampling regularly in the time axis, using one in every three footprints. For these preliminary results, we train on data for 2014-2015 ($\approx$11000 data points after sampling), validate on the first three months of 2016 ($\approx$900 data points) and test on the rest of 2016 ($\approx$4000 data points), using MSE as a loss function. We manually optimise the parameters on the validation set, but comprehensive parameter tuning should be conducted in further stages. All MLPs have two layers, ReLU activation, and layer normalisation. All input variables are normalised to zero mean and unit variance, for each pressure level if required. As the footprint values decay quickly with distance from the measurement point, we apply a node-wise Box-Cox transformation followed by standardisation to ensure all output values are closer to a normal distribution and are within the same range of values.

The outputs can be bias-corrected if needed using, for example, approaches from precipitation modelling [18, 19]. We use the validation set (transformed back to the original data range) to define a "footprint threshold" such that the threshold exceedance matches the above-zero frequency in the LPDM-generated footprints [19]. In the test dataset, all values under this threshold are set to zero.

Preliminary results We use the setup described here to emulate footprints for GOSAT measurements in Brazil in April-December 2016, and show some examples of the NAME-generated and emulator-generated footprints in figure 1. Table 1 shows the evaluation metrics for the test dataset at the pixel, footprint and concentration level. The accuracy obtained and the Normalised Mean Absolute Error (NMAE) at a concentration level are comparable to those found by [9], which is promising, given that emulating satellite footprints is a more difficult task than in situ measurement footprints.

Although often with good spatial agreement, the emulator has room for improvement: the emulated footprints are smoother than the LPDM ouputs, some meteorological conditions are better predicted than others (due to an imbalanced distribution of wind directions etc.), and some geographical locations prove challenging (e.g. the footprint over the Andes mountains in the second panel of Figure 1). Moreover, the positive Mean Bias Error (MBE) indicates that the model has a tendency to under-predict, also seen by [6] and [9], likely due to the imbalanced distribution of values and the sparsity of the footprints.

Table 1: Performance metrics of footprint emulator with current setup. See A for metric definitions

Pixel-level metrics	Footprint-level metrics		Concentration-level metrics		
MAE	Dice similarity	Accuracy	R2	NMAE	Mean Bias Error
$1.1{\cdot}10^{-4}$	$57.2\% \pm 18\%$	$65.8\% \pm 9\%$	0.448	0.3829	6.43

4 Next steps

Here, we describe a data-driven LPDM emulator that can infer footprints based only on meteorological and topography data, and demonstrate its application over Brazil. This approach presents an opportunity to leverage the growing volume of satellite data through well-known GHG emissions inference methods, by accelerating one of the most computationally expensive steps of the inference pipeline. Implementing this model in operational systems could make GHG emissions inference faster and more scalable, paving the way for near-time, global emissions estimates. This integration would also have far-reaching effects in data-driven policy making, as it would provide countries and organisations with an accessible method to understand their emissions, independently verify self-reported inventories and identify discrepancies in modelling. To achieve this effectively, the emulator needs to be developed further and tested in diverse geographic regions beyond Brazil but most importantly, implemented and evaluated with the full emissions inference pipeline.

References

[1] S. Dhakal, J.C. Minx, F.L. Toth, A. Abdel-Aziz, M.J. Figueroa Meza, K. Hubacek, I.G.C. Jonckheere, Y. Kim, G.F. Nemet, S. Pachauri, X.C. Tan, and T. Wiedmann. Chapter 2: Emissions Trends and Drivers. In *IPCC: Climate Change 2022: Mitigation of Climate Change.*

Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022.

[2] P. Bergamaschi, U. Karstens, A. J. Manning, M. Saunois, A. Tsuruta, A. Berchet, A. T. Vermeulen, T. Arnold, G. Janssens-Maenhout, S. Hammer, et al. Inverse modelling of European CH4 emissions during 2006–2012 using different inverse models and reassessed atmospheric observations. *Atmospheric Chemistry and Physics*, 18(2):901–920, 2018.

[3] A. L. Ganesan, M. Rigby, M. F. Lunt, R. J. Parker, H. Boesch, N. Goulding, T. Umezawa, A. Zahn, A. Chatterjee, R. G. Prinn, and others. Atmospheric observations show accurate reporting and little growth in India's methane emissions. *Nature Communications*, 8, 2017. See section "Generation of NAME sensitivity maps".

[4] M. Rigby, S. Park, T. Saito, L. M. Western, A. L. Redington, X. Fang, S. Henne, A. J. Manning, R. G. Prinn, G. S. Dutton, et al. Increase in CFC-11 emissions from eastern China based on atmospheric observations. *Nature*, 569:546–550, 2019.

[5] A. Manning, S. O'Doherty, D. Young, A. Redington, D. Say, J. Pitt, T. Arnold, C. Rennick, M. Rigby, A. Wisher, A. Wenger, and P. Simmonds. Long-Term Atmospheric Measurement and Interpretation of Radiatively Active Trace Gases - Detailed Report (October 2020 to September 2021). Technical report, Department for Business, Energy & Industrial Strategy, 2022.

[6] D. Roten, D. Wu, B. Fasoli, T. Oda, and J. C. Lin. An interpolation method to reduce the computational time in the stochastic Lagrangian particle dispersion modeling of spatially dense XCO2 retrievals. *Earth and Space Science*, 8(4), 2021.

[7] L. Cartwright, A. Zammit-Mangion, and N. M. Deutscher. Emulation of greenhouse-gas sensitivities using variational autoencoders. *Environmetrics*, 34(2), 2023.

[8] B. Fasoli, J. C. Lin, D. R. Bowling, L. Mitchell, and D. Mendoza. Simulating atmospheric tracer concentrations for spatially distributed receptors: Updates to the Stochastic Time-Inverted Lagrangian Transport model's R interface (STILT-R version 2). *Geoscientific Model Development*, 11(7):2813–2824, 2018.

[9] E. Fillola, R. Santos-Rodriguez, A. Manning, S. O'Doherty, and M. Rigby. A machine learning emulator for Lagrangian particle dispersion model footprints: A case study using NAME. *Geoscientific Model Development*, 16(7):1997–2009, 2023.

[10] R Keisler. Forecasting Global Weather with Graph Neural Networks. *arXiv preprint arXiv:2202.07575*, 2022.

[11] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, others, and P. Battaglia. GraphCast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022.

[12] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[13] R. L. Tunnicliffe, A. L. Ganesan, R. J. Parker, H. Boesch, N. Gedney, B. Poulter, Z. Zhang, J. V. Lavrič, D. Walter, M. Rigby, S. Henne, D. Young, and S. O'Doherty. Quantifying sources of Brazil's CH4 emissions between 2010 and 2018 from Satellite Data. *Atmospheric Chemistry and Physics*, 20(21):13041–13067, 2020.

[14] A. Jones, D. Thomson, M. Hort, and B. Devenish. The U.K. Met Office's next-generation atmospheric dispersion model, NAME III. *Air Pollution Modeling and Its Application XVII*, page 580–589, 2007.

[15] R. J. Parker, A. Webb, H. Boesch, P. Somkuti, R. Barrio Guillo, A. Di Noia, N. Kalaitzi, J. S. Anand, P. Bergamaschi, F. Chevallier, and others. A decade of GOSAT proxy satellite CH4 observations. *Earth System Science Data*, 12(4):3383–3412, 2020.

[16] Met Office (2013). Operational Numerical Weather Prediction (NWP) Output from the Global Atmospheric Part of the Met Office Unified Model (UM), NCAS British Atmospheric Data Centre. Last Accessed: March 2022.

[17] Met Office (2016). NWP-UKV: Met Office UK Atmospheric High Resolution Model data, Centre for Environmental Data Analysis. Last Accessed: March 2022.

[18] C. Teutschbein and J. Seibert. Bias correction of regional climate model simulations for hydrological climate-change impact studies: Review and evaluation of different methods. *Journal of Hydrology*, 456-457:12–29, 2012.

[19] J. Schmidli, C. Frei, and P. L. Vidale. Downscaling from GCM precipitation: A benchmark for dynamical and statistical downscaling methods. *International Journal of Climatology*, 26(5):679–689, 2006.

A Metrics

A.1 Segmentation Metrics

- Accuracy: Measures the spatial agreement of two footprints, giving true positives and true negatives the same weight. It is calculated by creating a binary representation of the LPDM footprints f above a certain threshold b (here $b = 0$):

$$F^{bin}_{t,i,j} = \begin{cases} 1, & \text{if } f_{t,i,j} > b \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

  and similarly calculating $\hat{F}^{bin}_t$ for the emulated footprints $\hat{f}$, and evaluating their spatial agreement:

$$\text{AC}_t = 100\% \times \frac{\left|F^{bin}_t \cap \hat{F}^{bin}_t\right| + \left|\overline{F^{bin}_t} \cap \overline{\hat{F}^{bin}_t}\right|}{\left|F^{bin}_t\right| + \left|\overline{F^{bin}_t}\right|}$$

- Dice Similarity: Measures the spatial overlap of the true and predicted footprints. Applied on F^{bin}_t and $\hat{F}^{bin}_t$ as above,

$$\text{DICE}_t = 100\% \times \frac{2\left|F^{bin}_t \cap \hat{F}^{bin}_t\right|}{\left|F^{bin}_t\right| + \left|\hat{F}^{bin}_t\right|}$$

A.2 Concentration Metrics

The LPDM footprint can be convolved with a map of gridded emissions to provide the expected above-baseline column-averaged mole fraction for that measurement location and time, by doing element-wise multiplication of the two grids (Hadamard product) and summing over the area. However, here we consider only the summed values of each footprint $F^{tot}_t = \sum_{ij} f(x_i, y_j, t)$ and similarly $\hat{F}^{tot}_t$ for emulated footprints $\hat{f}$, to remove any potential influences from the emissions prior. This is equivalent to convolving the footprints with a uniform emissions map of value 1. The two concentration timeseries, where $\mathbf{C}$ is the concentration timeseries calculated with the LDPM-generated footprints and arbitrary maps of gridded emissions, and $\hat{\mathbf{C}}$ is the timeseries generated with the emulated footprints and the same set of emissions, can be compared with the following metrics:

- R-squared score (R^2): represents the proportion of variance in the true time series that is explained by the emulated time series. It is calculated with

$$R^2(\mathbf{C}, \hat{\mathbf{C}}) = 1 - \frac{\sum_{t=1}^{m}(C_t - \hat{C}_t)^2}{\sum_{t=1}^{m}(C_t - \bar{C})^2}$$

- Normalised Mean Absolute Error (NMAE): MAE normalised by the mean of the true data, making it a scale-invariant metric. It can be calculated as

$$NMAE(\mathbf{C}, \hat{\mathbf{C}}) = \frac{1}{m\bar{C}} \sum_{t=1}^{m} \left| C_t - \hat{C}_t \right|$$

- Mean Bias Error (MBE): measures any systematic errors in the predictions, and is defined as

$$MBE(\mathbf{C}, \hat{\mathbf{C}}) = \frac{1}{m} \sum_{t=1}^{m} (C_t - \hat{C}_t)$$