
Mass Conservation on Rails – Rethinking Physics-Informed Learning of Ice Flow Vector Fields

Kim Bente

School of Computer Science
The University of Sydney
Sydney, Australia
kim.bente@sydney.edu.au

Roman Marchant

Human Technology Institute
University of Technology Sydney
Sydney, Australia
roman.marchant@uts.edu.au

Fabio Ramos

NVIDIA
Seattle, USA
School of Computer Science
The University of Sydney
Sydney, Australia
fabio.ramos@sydney.edu.au

Abstract

To reliably project future sea level rise, ice sheet models require inputs that respect physics. Embedding physical principles like mass conservation into models that interpolate Antarctic ice flow vector fields from *sparse & noisy* measurements not only promotes physical adherence but can also improve accuracy and robustness. While physics-informed neural networks (PINNs) impose physics as soft penalties, offering flexibility but no physical guarantees, we instead propose divergence-free neural networks (dfNNs), which enforce local mass conservation exactly via a vector calculus trick. Our comparison of dfNNs, PINNs, and unconstrained NNs on ice flux interpolation over Byrd Glacier suggests that "mass conservation on rails" yields more reliable estimates, and that *directional guidance*, a learning strategy leveraging continent-wide satellite velocity data, boosts performance across models.

1 Introduction & Background

By integrating physical laws into data-driven learning, **physics-informed machine learning** (PIML) [1] has reshaped ML's impact across the physical sciences, including notable advances in climate science [2]–[4]. Many of the Essential Climate Variables (ECVs) [5], such as ocean currents, groundwater, and glacier flow, take the form of spatial vector fields, $\mathbf{v}(x, y)$, that describe Earth's fluid dynamics. These flows are governed by the continuity equation, which ensures that mass of the flowing fluid is conserved [6]. For steady, incompressible flow, this mass conservation constraint reduces to the *divergence-free condition*, $\nabla \cdot \mathbf{v} = 0$, enforcing that inflow equals outflow at any point in space. Diverse models of groundwater [7], ocean eddies and currents [8], [9], tidal flows [10], ice sheets [11], [12], glaciers [13], ocean surface winds [14], and large-scale atmospheric circulations [15], [16] have incorporated the **divergence-free condition** to promote physically consistent behaviour, thereby improving accuracy and robustness, which facilitates reliable downstream use.

Here, we focus on an application with especially grave implications — modelling how ice flows across the Antarctic Ice Sheet (AIS). Holding the ice equivalent of ~58 m global mean sea level rise [17], the AIS is the largest potential contributor to rising seas [18]. By 2100, sea level rise is projected to impose annual flood damage costs of ~2% of GDP [19] and to displace ~360 M people living on

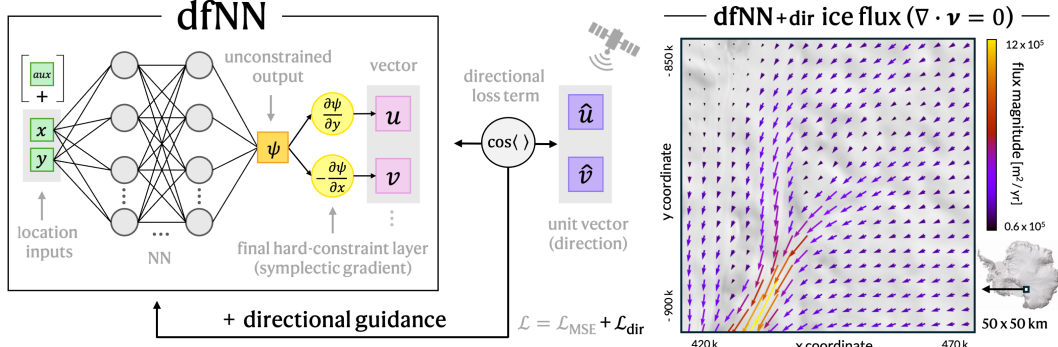


Figure 1: Diagram of the divergence-free NN (dfNN) model architecture and *directional guidance* learning strategy (left). The quiver plot (right) shows an on-grid ice flux reconstruction by the dfNN with *directional guidance* (best model) for a subset of the experimental region over Byrd Glacier.

flood-prone land under RCP 4.5 [20]. However, extreme polar conditions and remoteness severely limit data collection, making AIS modelling challenging, yet scientifically paramount. Studies have found that unconstrained interpolations of sparse and noisy ice thickness measurements from airborne radar surveys produce unrealistic behaviour in numerical ice sheet models due to flux divergences [11], [21] (*flux*: transport of ice volume per unit width and time). In response to this issue, numerical inversions [11], [17] and, more recently, ML approaches [12], [13], [22] have been developed to **reduce flux artefacts by constraining ice flux interpolations with mass conservation**.

Both proposed ML approaches [12], [13] employ physics-informed neural networks (PINNs) [23], a widely used PIML framework. PINNs incorporate an additional loss term that quantifies violations of physical principles, to learn not only to fit the training data, but to simultaneously minimise physical inconsistencies, such as flux artefacts. Although PINNs have the flexibility to incorporate multiple physical principles at once, as demonstrated in [12] and [13], and are hence broadly adopted, the penalty-based ‘*soft*’ constraining does not guarantee physical consistency, which has been found to hinder generalisation performance in the face of data sparsity and noise [24], [25]. Furthermore, unstable trade-offs between data fit and physics can also affect convergence [26]. In this work, we show how lesser known hard-constrained **divergence-free NNs** (dfNNs, see Fig. 1), rooted in [24], [26], [27], can be used to **model exactly mass-conserving ice flux vector fields**. To assess competing PIML paradigms for modelling divergence-free vector fields, we compare hard-, soft-, and unconstrained NN models—dfNNs, PINNs, and regular NNs—on ice flux reconstruction for Byrd Glacier, Antarctica, shown in Fig. 1, evaluating both predictive accuracy and physics adherence. Lastly, informed by the application context [28], we test two extensions across all three models: (i) *directional guidance*, a learning strategy that leverages continent-wide ice velocity observations from satellites via a directional loss term $\mathcal{L}_{\text{dir}}$ to align predicted flow with this directional information beyond sparse flux observation locations, and (ii) incorporating auxiliary predictors (e.g., surface elevation). Reproducible experiments and implementations in PyTorch [29] are available at https://github.com/kimbente/mass_conservation_on_rails.

2 Exact mass conservation with dfNNs

Introducing dfNNs. In this work, we address the problem of learning 2D vector fields with NNs under the constraint of local mass conservation. Let $\mathbf{v} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ denote a 2D vector field, expressed as $\mathbf{v}(x, y) = (u(x, y), v(x, y))$, where u and v are the vector components in x - and y -direction, respectively. The divergence of a vector field is defined as the sum of the partial derivatives of its components, $\nabla \cdot \mathbf{v} = \partial u / \partial x + \partial v / \partial y$, which quantifies the local rate of expansion or compression. In balanced flows, this quantity must equal zero. The first NN model to integrate the divergence-free constraint into the model architecture was introduced in early work of Kuroe et al. [27]. The proposed approach, *model inclusive learning*, leverages the property that the symplectic gradient of a scalar stream function is by design divergence-free (see [6] for vector calculus background). By training the NN to predict the stream function rather than the vector components, akin to the change of variables trick, and attaining the vector components by taking the symplectic gradient in a deterministic

differential step, the network outputs continuous, exactly divergence-free vector fields, i.e. with $\nabla \cdot \mathbf{v} = 0$. Closely related model architectures have reappeared more recently: Hamiltonian Neural Networks [24], which were introduced with energy rather than mass conservation in mind, and Neural Conservation Laws [26], a theoretically comprehensive framework that extends to solving general continuity equations, both impose the divergence-free constraint through a construction equivalent to [27]. Since the original model [27], like [9], addresses vector fields beyond the purely divergence-free case by decomposing them into curl-free and divergence-free components, we refer to the divergence-free component as a divergence-free neural network (dfNN) throughout this paper. The left panel of Fig. 1 illustrates the dfNN architecture. Spatial coordinates (x, y) , together with optional auxiliary inputs, are passed through a feed-forward neural network whose width and depth can be adjusted to the task complexity. For each input location, the network outputs a scalar *stream function* ψ (refer to [6]). As proposed by [27] this unconstrained scalar field is then mapped to a strictly divergence-free vector field by applying the symplectic operator. Its components, given by the symplectic gradient $(\partial\psi/\partial y, -\partial\psi/\partial x)$, define the velocity field (u, v) and can be computed efficiently via automatic differentiation in ML frameworks such as PyTorch [29], all in a fully mesh-free manner.

Modelling ice flux as a divergence-free flow. *Ice flux* denotes the horizontal transport of ice volume per unit width and time, with volume equivalent to mass under incompressibility. Ice flux vectors $\mathbf{v}$ have units $[\text{m}^3 \text{m}^{-1} \text{yr}^{-1}]$, or simply $[\text{m}^2 \text{yr}^{-1}]$, and are defined as the product $\mathbf{v} = h \cdot \mathbf{s}$ where ice thickness $h \in \mathbb{R}$ is in $[\text{m}]$ and horizontal velocity $\mathbf{s} \in \mathbb{R}^2$ is in $[\text{m} \text{yr}^{-1}]$. Following [11] and [12], we take the measured surface-level velocity as a proxy for depth-averaged velocity, to model the depth-integrated transport of ice. Given the AIS’s quasi-steady flow, minimal thickness changes, and a net surface/basal mass balance that is negligible relative to fluxes along ice streams [11], [17], we model the problem as a spatial (time-independent) process under divergence-free flow assumptions. As motivated above, the resulting spatial ice flux fields are a critical input for numerical ice sheet models [11], [21], and, conversely, ice thickness maps can be extracted with $h = \mathbf{v}/\mathbf{s}$ [11]–[13].

Directional guidance. While observations of environmental processes are typically sparse, InSAR satellites provide continent-wide ice surface velocity observations $\mathbf{s}$ on a dense grid over Antarctica [30]. Given $\mathbf{v} = h \cdot \mathbf{s}$, these observations only determine the direction of $\mathbf{v}$, but not its magnitude. Still, this partial information can help improve and further constrain reconstructions in unsurveyed regions. In a learning strategy that we refer to as *directional guidance*, conceptually similar to the physical guidance term in PINNs [31], we integrate a loss term $\mathcal{L}_{\text{dir}}$ that quantifies the directional misalignment between domain-wide unit velocity observations $\hat{\mathbf{s}}$, and corresponding normalised model-predicted vectors $\hat{\mathbf{v}}$, via the cosine similarity, as defined in Eq. (1),

$$\mathcal{L} = (1 - w_{\text{dir}}) \cdot \mathcal{L}_{\text{MSE}} + w_{\text{dir}} \cdot \mathcal{L}_{\text{dir}}, \quad \text{with } \mathcal{L}_{\text{dir}} = 1 - \cos(\hat{\mathbf{s}}, \hat{\mathbf{v}}) \in [0, 2]. \quad (1)$$

Weighted by w_{dir} , $\mathcal{L}_{\text{dir}}$ is integrated with the regular training loss $\mathcal{L}_{\text{MSE}}$, so that gradient updates simultaneously fit flux observations over training regions and align predicted directions with satellite-derived flow directions elsewhere.

3 Experiments & Results

Data. To assess how best to enforce local mass conservation constraints in learning vector fields, we compare hard-constrained dfNNs [24], [26], [27], soft-constrained PINNs [23], and unconstrained NNs using real Antarctic observations. Similar to [12], we focus on a 200×200 km region over Byrd Glacier (see Fig. 1), a fast-flowing outlet glacier with sufficiently dense observations to withhold data for testing. Ice flux observations $\mathbf{v}$ are obtained by combining ice thickness measurements h from the comprehensive *Bedmap* data collection [32] with ice surface velocities $\mathbf{s}$ from NASA’s *MEaSURES Phase-Based Antarctica Ice Velocity Map* [30]. The data are split into train and test regions using a chequerboard pattern with 15 km squares (Fig. 4), yielding 27,172 training and 22,045 testing points. Further details on data sources and preprocessing are provided in Appendix D.

Experiments. In total, we train nine models and evaluate their reconstruction performance on the test set. These include the three base models, their *directionally guided* counterparts, and variants incorporating surface elevation from *Bedmap* as an auxiliary input. All models share a consistent NN backbone and are trained with PyTorch’s AdamW optimiser with weight decay, which we found to improve convergence. We evaluate predictive accuracy using RMSE and MAE, and quantify physics violations with the Mean Absolute Divergence (MAD, defined in Appendix C). We also monitor the carbon emissions of our experiments with *CodeCarbon*, and report them in Appendix E. Further training details are provided in Appendix B.

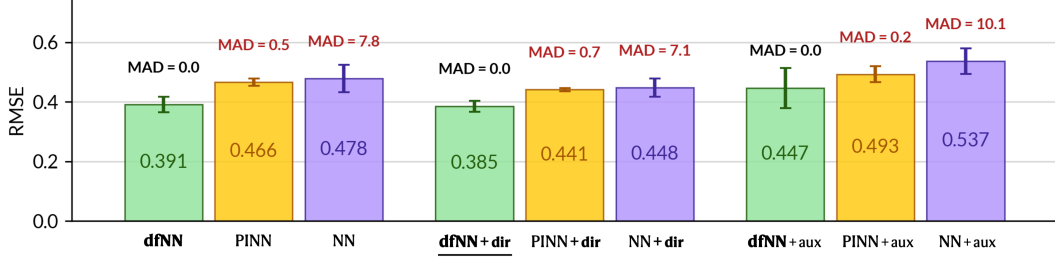


Figure 2: Test RMSE (↓) comparison across all model variants, averaged over five runs. Boxed values indicate mean RMSE, with error bars showing $\pm$ std. MAD (top) denotes the Mean Absolute Divergence. **dfNNs** (proposed, in bold) outperform PINNs & NNs, while **directional guidance** (proposed, in bold) improves all models and yields the best-performing variant, dfNN + dir (underlined).

Results. Test results from all nine models, averaged over five independent training runs, are visualised in Fig. 2, with full metrics (mean $\pm$ std) reported in Table 1 (Appendix). Across all metrics, dfNNs (and their variants) consistently yield more reliable reconstructions of unseen ice flux vectors than both PINNs and NNs, underscoring the benefits of enforcing exact mass conservation for both accuracy and physical compliance. PINNs substantially reduce MAD compared to NNs, but their improvement in interpolation accuracy is modest. Furthermore, they do not converge to fully divergence-free solutions (MAD > 0). *Directional guidance* improves performance across all models, with larger relative gains for PINNs and NNs, which have weaker inductive biases than dfNNs. Predictions by all models + dir, the best variant of each model family, are shown in Fig. 3. By contrast, incorporating surface elevation as an auxiliary predictor degrades performance, with exploratory tests using surface gradients showing an even stronger decline. Overall, the dfNN with *directional guidance* is the best-performing model, producing guaranteed divergence-free interpolations with MAD = 0.

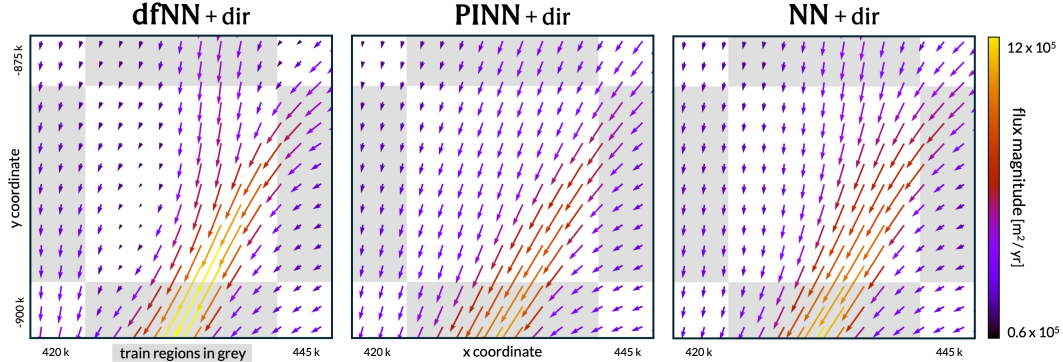


Figure 3: Gridded predictions of models + dir (best variant per model) for a small test region (white).

4 Conclusion

Our experiments on real Antarctic ice flux data highlight the advantages of enforcing known physical inductive biases with hard constraints (dfNNs) rather than loss terms (PINNs). Nonetheless, additional loss term guidance can help hard-, soft-, and unconstrained models alike: Our proposed *directional guidance* learning strategy shows an effective way of leveraging dense satellite data to constrain the direction of environmental flows, boosting performance across all models. Furthermore, the ablations substantiate the premise that parsimonious models generalise best: Adding auxiliary surface predictors like elevation introduces more noise than signal, indicating that relevant topographic flow controls are not expressed at the surface over Byrd Glacier. Overall, our work suggests that when available, hard-constrained "models on rails" should be the method of choice for physics-informed learning, as they offer both physical fidelity and maintain model design flexibility to incorporate additional guidance, while remaining robust under the *noisy & sparse* data typical of climate and environmental applications in the real world.

Acknowledgments and Disclosure of Funding

This research was supported by an Australian Government Research Training Program (RTP) Scholarship, as well as through the Australian Research Council’s Industrial Transformation Training Centre in Data Analytics for Resources and Environments (DARE) (project ID IC190100031).

References

- [1] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, Jun. 2021.
- [2] M. Reichstein, G. Camps-Valls, B. Stevens, *et al.*, “Deep learning and process understanding for data-driven earth system science,” *Nature*, vol. 566, no. 7743, pp. 195–204, 2019.
- [3] D. Rolnick, P. L. Donti, L. H. Kaack, *et al.*, “Tackling climate change with machine learning,” *ACM Computing Surveys*, vol. 55, no. 2, 42:1–42:96, 2022.
- [4] C. Monteleoni, G. A. Schmidt, F. Alexander, *et al.*, “Climate informatics,” in *Computational Intelligent Data Analysis for Sustainable Development*, Chapman and Hall/CRC, 2013, p. 440.
- [5] W. M. Organization. “Essential climate variables,” World Meteorological Organization. (Nov. 13, 2024), [Online]. Available: <https://gcos.wmo.int/site/global-climate-observing-system-gcos/essential-climate-variables> (visited on 08/19/2025).
- [6] P. K. Kundu, I. M. Cohen, and D. R. Dowling, “Chapter 4 - conservation laws,” in *Fluid Mechanics (Sixth Edition)*, P. K. Kundu, I. M. Cohen, and D. R. Dowling, Eds., Boston: Academic Press, Jan. 1, 2016, pp. 109–193.
- [7] S. Zhang, X.-S. Wang, P.-F. Han, and X.-W. Jiang, “New particle tracking method for 2d steady-state groundwater flow around wells: A modified algorithm using the stream function,” *Journal of Hydrology*, vol. 632, p. 130 928, Mar. 1, 2024.
- [8] R. Berlinghieri, B. L. Trippe, D. R. Burt, *et al.*, *Gaussian processes at the helm(holtz): A more fluid model for ocean currents*, Jun. 20, 2023.
- [9] A. Sosanya and S. Greydanus, *Dissipative hamiltonian neural networks: Learning dissipative and conservative dynamics separately*, Jan. 26, 2022.
- [10] R. Vennell and R. Beatson, “A divergence-free spatial interpolator for large sparse velocity data sets,” *Journal of Geophysical Research: Oceans*, vol. 114, C10 2009.
- [11] M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia, and D. Aubry, “A mass conservation approach for mapping glacier ice thickness,” *Geophysical Research Letters*, vol. 38, no. 19, 2011.
- [12] T. O. Teisberg, D. M. Schroeder, and E. J. MacKie, “A machine learning approach to mass-conserving ice thickness interpolation,” in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, Jul. 2021, pp. 8664–8667.
- [13] V. Steidl, J. L. Bamber, and X. X. Zhu, *Physics-aware machine learning for glacier ice thickness estimation: A case study for svalbard*, Jun. 14, 2024.
- [14] M. Fan, D. Paul, T. C. M. Lee, and T. Matsuo, *Modeling tangential vector fields on a sphere*, Dec. 23, 2016.
- [15] F. J. Narcowich, J. D. Ward, and G. B. Wright, “Divergence-free RBFs on surfaces,” *Journal of Fourier Analysis and Applications*, vol. 13, no. 6, pp. 643–663, Dec. 2007.
- [16] B. Löptien, L. Gizon, A. C. Birch, *et al.*, “Global-scale equatorial rossby waves as an essential component of solar internal dynamics,” *Nature Astronomy*, vol. 2, no. 7, pp. 568–573, Jul. 2018.
- [17] M. Morlighem, E. Rignot, T. Binder, *et al.*, “Deep glacial troughs and stabilizing ridges unveiled beneath the margins of the antarctic ice sheet,” *Nature Geoscience*, vol. 13, no. 2, pp. 132–137, 2020, [<https://nsidc.org/data/nsidc-0756/versions/3>].
- [18] IPCC, “Summary for policymakers,” in *IPCC Special Report on the Ocean and Cryosphere in a Changing Climate [H.-O. Pörtner, D.C. Roberts, V. Masson-Delmotte, P. Zhai, M. Tignor, E. Poloczanska, K. Mintenbeck, A. Alegría, M. Nicolai, A. Okem, J. Petzold, B. Rama, N.M. Weyer (eds.)]*. Cambridge, UK and New York, NY, USA: Cambridge University Press, 2022, pp. 3–35.

- [19] S Jevrejeva, L. P. Jackson, A Grinsted, D Lincke, and B Marzeion, “Flood damage costs under the sea level rise with warming of 1.5 °c and 2 °c,” *Environmental Research Letters*, vol. 13, no. 7, p. 074 014, Jul. 2018.
- [20] S. A. Kulp and B. H. Strauss, “New elevation data triple estimates of global vulnerability to sea-level rise and coastal flooding,” *Nature Communications*, vol. 10, no. 1, p. 4844, Oct. 29, 2019.
- [21] H. Seroussi, M. Morlighem, E. Rignot, *et al.*, “Ice flux divergence anomalies on 79north glacier, greenland,” *Geophysical Research Letters*, vol. 38, no. 9, May 16, 2011.
- [22] Z. Liu, Y. Koo, and M. Rahnemounfar, *Physics-informed machine learning on polar ice: A survey*, Apr. 30, 2024.
- [23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations*, Nov. 28, 2017.
- [24] S. Greydanus, M. Dzamba, and J. Yosinski, *Hamiltonian neural networks*, Sep. 5, 2019.
- [25] I. Macêdo and R. Castro, *Learning divergence-free and curl-free vector fields with matrix-valued kernels*. IMPA, 2010.
- [26] J. Richter-Powell, Y. Lipman, and R. T. Q. Chen, *Neural conservation laws: A divergence-free perspective*, Dec. 11, 2022.
- [27] Y. Kuroe, M. Mitsui, H. Kawakami, and T. Mori, “A learning method for vector field approximation by neural networks,” in *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, vol. 3, May 1998, 2300–2305 vol.3.
- [28] D. Rolnick, A. Aspuru-Guzik, S. Beery, *et al.*, “Position: Application-driven innovation in machine learning,” in *Proceedings of the 41st International Conference on Machine Learning*, PMLR, Jul. 8, 2024, pp. 42 707–42 718.
- [29] A. Paszke, S. Gross, F. Massa, *et al.*, *PyTorch: An imperative style, high-performance deep learning library*, Dec. 3, 2019.
- [30] J. Mouginot, E. Rignot, and B. Scheuchl, *MEaSUREs phase-based antarctica ice velocity map, version 1*, [<https://nsidc.org/data/nsidc-0754/versions/1>], 2019.
- [31] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.
- [32] A. C. Frémand, P. Fretwell, J. A. Bodart, *et al.*, “Antarctic bedmap data: Findable, accessible, interoperable, and reusable (FAIR) sharing of 60 years of ice bed, surface, and thickness data,” *Earth System Science Data*, vol. 15, no. 7, pp. 2695–2710, Jul. 17, 2023, [<https://www.bas.ac.uk/project/bedmap/#data>].

Appendix A – Results Table

Table 1 contains all experimental results. A visual comparison of RMSE (Root Mean Square Error) and MAD (Mean Absolute Divergence) values can be found above in Fig. 2.

Table 1: Test metrics (mean $\pm$ std) over 5 runs. MAD denotes the Mean Absolute Divergence. Lower values $\downarrow$ indicate better performance. The overall best model is shown in bold, and physics violations (flux divergences) are highlighted in red. The model order follows Fig. 2.

Model (+ extension)	RMSE $\pm$ std	MAE $\pm$ std	MAD $\pm$ std
dfNN	0.391 $\pm$ 0.03	0.199 $\pm$ 0.01	0.000 $\pm$ 0.00
dfNN + dir	0.385 $\pm$ 0.02	0.193 $\pm$ 0.01	0.000 $\pm$ 0.00
dfNN + aux	0.447 $\pm$ 0.07	0.209 $\pm$ 0.01	0.000 $\pm$ 0.00
PINN	0.466 $\pm$ 0.01	0.236 $\pm$ 0.01	0.471 $\pm$ 0.06
PINN + dir	0.441 $\pm$ 0.01	0.221 $\pm$ 0.01	0.685 $\pm$ 0.11
PINN + aux	0.493 $\pm$ 0.03	0.237 $\pm$ 0.01	0.235 $\pm$ 0.03
NN	0.478 $\pm$ 0.05	0.215 $\pm$ 0.01	7.792 $\pm$ 0.43
NN + dir	0.448 $\pm$ 0.03	0.205 $\pm$ 0.01	7.137 $\pm$ 0.31
NN + aux	0.537 $\pm$ 0.04	0.237 $\pm$ 0.02	10.13 $\pm$ 0.75

Appendix B – Training details

Table 2 contains all hyperparameters used for training, where the reported values correspond to the best-performing settings identified in our investigation. PINNs are trained with an additional divergence reduction step at every epoch, to ensure that the model also learns to reduce divergences over test regions.

Table 2: Training hyperparameters used in experiments. Please also refer to the repository’s configs.py file on https://github.com/kimbente/mass_conservation_on_rails directly.

Hyperparameter	Value
dfNN learning rate (all model variants)	0.0001
PINN learning rate (all model variants)	0.0001
NN learning rate (all model variants)	0.0005
Number of runs	5
Maximum number of epochs	3000
Patience for early stopping	100
Batch size	1024
Number of hidden layers (for all NNs)	4
Width of each hidden layer (for all NNs)	64
Optimiser	AdamW
Weight decay	0.001
Weight w_{dir} for $\mathcal{L}_{\text{dir}}$ (dfNN + dir, NN + dir)	0.4
PINN number of points used per epoch for additional domain divergence reduction step	10,240
PINN weight w_{div} for $\mathcal{L}_{\text{div}}$ (PINN, PINN + aux)	0.2
for PINN + dir both w_{dir} and w_{div} are halved to avoid underweighing the training loss $\mathcal{L}_{\text{MSE}}$	
Adjusted ($\times 0.5$) weight w_{dir} for $\mathcal{L}_{\text{dir}}$ (PINN + dir)	0.2
Adjusted ($\times 0.5$) PINN weight w_{div} for $\mathcal{L}_{\text{div}}$ (PINN + dir)	0.1

Appendix C – Mean Absolute Divergence (MAD)

We introduce the metric Mean Absolute Divergence (MAD) to quantify flux divergences (i.e. violations of local mass conservation). Taking the absolute divergence ensures that local positive and negative divergences, which both violate the divergence-free constraint, do not cancel out. MAD is calculated as defined in Eq. (2).

$$\text{MAD} = \frac{1}{N} \sum_{i=1}^N |\nabla \cdot \mathbf{q}(\mathbf{x}_i)| = \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial u}{\partial x}(\mathbf{x}_i) + \frac{\partial v}{\partial y}(\mathbf{x}_i) \right| \quad (2)$$

Appendix D – Training and testing data

Byrd Glacier is a major Antarctic outlet glacier draining a basin that holds 6 m sea level equivalent (SLE) [17]. The 200×200 km region, about the size of Denmark, is characterised by fast, topographically steered flow, and is relatively densely surveyed, allowing us to withhold a subset of data for testing.

During preprocessing, we apply a uniform firn correction to ice thickness measurements, downsample along densely spaced airborne survey flight lines, and interpolate the velocity grid to the ice thickness point locations to obtain $\mathbf{v}$.

Fig. 4 shows the chequerboard pattern which we use to divide the Byrd Glacier domain into training and testing regions.

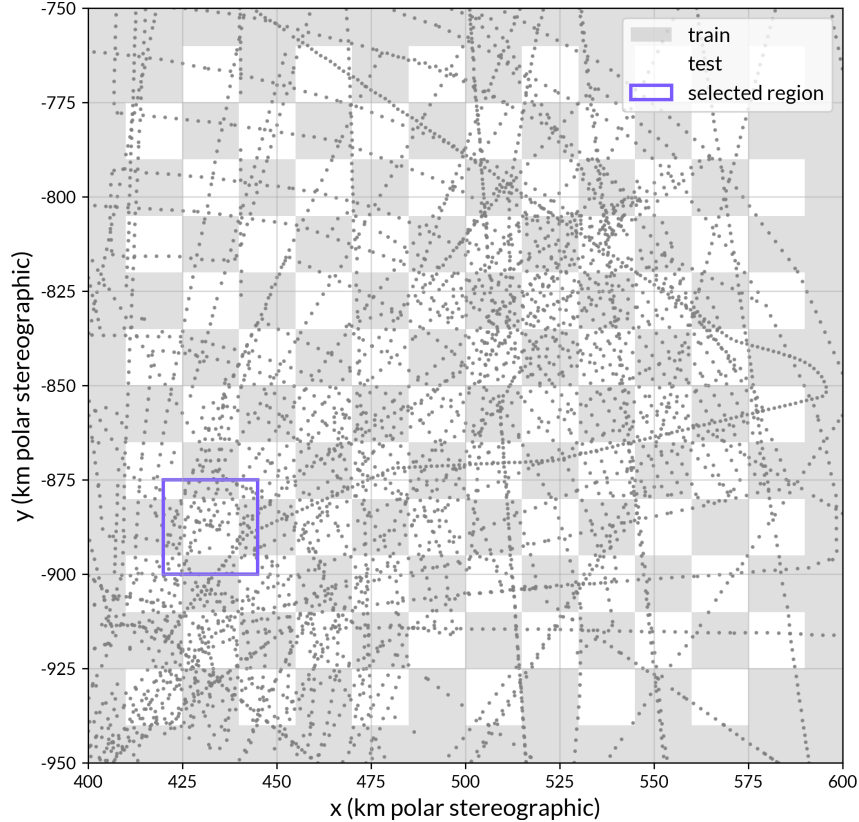


Figure 4: Train-Test chequerboard over Byrd Glacier, Antarctica. Training regions are shown in grey and test regions are shown in white. Points indicate locations with ice flux observations, collected mainly through airborne geophysical surveys, highlighting the anisotropic nature of the ‘flight line’ data. The Antarctic Polar stereographic coordinate system is used (ESPG:3031, <https://epsg.io/3031>). The selected region (purple frame) corresponds to the region shown in Fig. 3.

Appendix E – Emissions

We use *CodeCarbon* to track our experiments’ emissions and report them in Table 3.

Table 3: Total computational demands of 5 experiment runs (model training and inference), as tracked by *CodeCarbon*, see <https://codecarbon.io/>. (GPU: 1 × NVIDIA RTX 4090, RAM: 63 GB).

Model (+ extension)	Emissions [in kgCO ₂ eq]	Energy consumed [in kWh]	Wall clock time [in minutes]
dfNN	0.219	0.399	139.19
dfNN + dir	0.171	0.312	171.53
dfNN + aux	0.236	0.430	131.29
PINN	0.078	0.143	62.49
PINN + dir	0.072	0.132	75.27
PINN + aux	0.098	0.178	102.85
NN	0.076	0.138	81.89
NN + dir	0.083	0.151	90.16
NN + aux	0.073	0.134	79.34