JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2025MS004969

PseudospectralNet: Toward Hybrid Atmospheric Models for Climate Simulations

Maximilian Gelbrecht[1,2], **Milan Klöwer**[3,4], **and Niklas Boers**[1,2,5]

[1]Earth System Modelling, School of Engineering & Design, Technical University Munich, Munich, Germany, [2]Potsdam Institute for Climate Impact Research, Potsdam, Germany, [3]Massachusetts Institute of Technology, Cambridge, MA, USA, [4]University of Oxford, Oxford, UK, [5]Department of Mathematics and Global Systems Institute, University of Exeter, Exeter, UK

Key Points:

- An architecture for hybrid atmospheric models is presented, combining physics-based and machine learning (ML) approaches
- Including physical model components yields better forecasts and numerical stability than pure ML approaches

Supporting Information:

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

Correspondence to:

M. Gelbrecht,
maximilian.gelbrecht@tum.de

Citation:

Gelbrecht, M., Klöwer, M., & Boers, N. (2025). PseudospectralNet: Toward hybrid atmospheric models for climate simulations. *Journal of Advances in Modeling Earth Systems*, *17*, e2025MS004969. https://doi.org/10.1029/2025MS004969

Received 21 JAN 2025
Accepted 15 SEP 2025

Author Contributions:

Conceptualization: Maximilian Gelbrecht, Niklas Boers
Data curation: Maximilian Gelbrecht, Milan Klöwer
Formal analysis: Maximilian Gelbrecht
Funding acquisition: Niklas Boers
Investigation: Maximilian Gelbrecht
Methodology: Maximilian Gelbrecht
Project administration: Maximilian Gelbrecht, Niklas Boers
Resources: Niklas Boers
Software: Maximilian Gelbrecht
Supervision: Niklas Boers
Validation: Maximilian Gelbrecht
Visualization: Maximilian Gelbrecht
Writing – original draft: Maximilian Gelbrecht

Abstract Recent machine learning (ML) models have shown great success for weather prediction tasks, suggesting that atmospheric dynamics can in principle be learned from data. However, such approaches suffer from instability or drift in long integrations and are hence usually not suited for climate simulations. Incorporating physical knowledge promises to alleviate such shortcomings. Here, we present PseudospectralNet (PSN), an architecture for a hybrid atmospheric model that combines a quasi-geostrophic physics-based dynamical core with a data-driven core based on an UNet. Our architecture transforms between grid and spectral space at every time step and therefore mimics the pseudospectral solution approach many intermediate-complexity atmospheric models follow. Neural networks are separately defined in the spectral and grid space and are combined with physics-based dynamical cores in each of these spaces. We train PSN separately on data from quasi-geostrophic models, primitive equation models, and reanalysis data. We use this model to study the effect of adding physics-based cores to ML models on forecast error and numerical stability. We find in both regards that adding the physics-based dynamical core to our model helps short-term predictability and long-term numerical stability of the hybrid model.

Plain Language Summary Many machine learning (ML) models that aim to learn atmospheric dynamics are not suitable for climate simulations because long simulations often suffer from numerical inaccuracies and instabilities. In this project, we investigate whether adding physics-based knowledge to ML models helps to improve those long simulations and also make forecasts of the models more accurate. For this purpose we divide a simple atmospheric model into its individual components and use their outputs at every time step of the simulation as an input to artificial neural networks that learn how to forecast example data accurately with the information from the data itself and the physics-based components. We find that such a hybrid model outperforms purely data-driven ML methods in our applications.

1. Introduction

Over the last years we have seen tremendous success of machine learning (ML) for weather prediction tasks with models such as Pangu (Bi et al., 2022), GraphCast (Lam et al., 2023), FourcastNet (Pathak et al., 2022), FengWu (K. Chen et al., 2023), or GenCast (Price et al., 2024). These purely data-driven deep learning models yield an accuracy comparable or better than traditional numerical weather prediction models (NWP) in benchmarks such as WeatherBench (Rasp et al., 2020, 2024). Not all NWP tasks are defined in these benchmarks—for example, aerosol forecasting is not present. While ML models require a costly training, they perform remarkably well at a fraction of the computational cost at inference of traditional NWP. However, the long-term numerical stability of such purely data-driven models remains a serious issue for many models, as is the generalization to changing climates (Rackow et al., 2024).

With hybrid models that combine physics-based models with ML we hope to close the gap between classical ESMs that have to rely manual tuning but don't suffer from out-of-sample and stability problems in the same way as ML models do. In the present work we focus on atmospheric models. A hybrid atmospheric model, NeuralGCM, has recently been introduced (Kochkov et al., 2024). Based on a primitive equation physics-based dynamical core wrapped inside an encoder-decoder structure, it applied an artificial neural network (ANN) as a column model to represent physical processes in the vertical column of the atmosphere. While it already performs well on many decade-long simulations, only 22 out of 37 initial conditions yielded stable simulations for a time span of 40 simulated years. Therefore, although NeuralGCM is promising, stability remains an issue as

Writing – review & editing: Maximilian Gelbrecht, Milan Klöwer, Niklas Boers

well. In this study, we aim to address the stability issue with a new architecture for hybrid atmospheric models. We suggest both an alternative architecture for hybrid atmospheric modeling based on Neural Differential Equations (R. T. Q. Chen et al., 2018; Gelbrecht, Boers, & Kurths, 2021) and verify it by analyzing forecast-error and numerical stability, using various kinds of training and test data. We use this model as an opportunity for a case study on integrating physics into data-driven models and investigating its impact when learning from data of varying complexity.

Similar to studies in fluid dynamics such as (Kochkov et al., 2021; Um et al., 2020) and the aforementioned NeuralGCM, we set up a fully differentiable hybrid model. This means that we can compute gradients with respect to all model parameters via automatic differentiation (AD) (e.g., Innes, 2019). Earlier studies suggest more stable solutions for such AD-enabled models (Kochkov et al., 2021; Um et al., 2020), in addition to many other substantial benefits, such as easier, systematic and transparent calibration and more comprehensive uncertainty and sensitivity studies (Gelbrecht et al., 2023). Differentiability also enables us to combine ML- and physics-based components much more naturally and freely, and allows us to train all model components jointly on the atmospheric target data. The big advantage of this approach is that the parameters of the physical model parts do not need to be manually re-tuned, but are optimized online, together with the parameters of the ML model parts. This addresses substantial, current challenges when including ML-based parameterizations in existing, non-differentiable General Circulation Models (GCMs) or ESMs, where the manual re-tuning of the physical model parameters is extremely time- and resource-intensive. The caveat of such an approach, however, is that most existing atmospheric models are not differentiable by AD. Only a few models (mainly the operational weather forecast models using data assimilation) are differentiable by manually maintaining an additional adjoint model, which requires additional human resources (e.g., Rabier et al., 2000) and slows down development. In this work, we therefore have to begin our investigation with a comparably simple atmospheric model: three-layer quasi-geostrophic atmospheric model (QG3) in the formulation of Marshall and Molteni (1993).

We introduce a framework for hybrid atmospheric modeling, with a physics-based dynamical core that we implement to be differentiable and GPU-compatible. The QG3 model is used to study for example, the predictability of large-scale atmospheric motions, northern hemisphere teleconnection patterns, and weather regimes such as atmospheric blocking events (e.g., Corti et al., 1997; Lucarini & Gritsun, 2020; Kondrashov et al., 2004; Vannitsem, 2017). In the Supporting Information S1, we give a complete accord of its governing equations. The QG3 model is considerably less complex than any operational GCM. The QG3 model gives a simplistic but effective representation of the synoptic-to-planetary scale atmospheric dynamics of the mid-latitudes (Lucarini & Gritsun, 2020). We will therefore begin with training our hybrid model on data from models of similar and slightly higher complexity, before turning to observational data. We begin by training our hybrid model, incorporating only parts of the QG3 model, on data from the full QG3 model itself, to test whether the architecture is able to reconstruct missing model components and parametrizations. We then test the capacity of our hybrid model to model different dynamics with a data set generated from a primitive equation model, and a reanalysis data set. The latter therefore goes considerably beyond typical generalization tests.

As the QG3 model is formulated in the spectral space, our hybrid model inherits this property. Aside from NeuralGCM, which includes a spectral physics-based dynamical core as well, also other ML models demonstrated the usefulness of this approach even for purely data-driven models (Bonev et al., 2023; Li et al., 2021). The ACE model (Watt-Meyer et al., 2023), for example, already applied these techniques successfully to emulate GCMs at a considerable lower computational cost than the original GCM (C. Wang et al., 2024). demonstrate those technique to setup a data-driven AO-GCM.

We aim to model atmospheric dynamics with a hybrid model: part physics-based, part data-driven. The key research questions we address in this work are: (Q1) Can the data-driven part reduce biases of the physics-based part? (Q2) Can the physics-based part enable better generalization and stability than data-driven models, to enable simulations on climate time scales? (Q3) Does the hybrid model generalize to dynamics that are more complex than those of the physics-based part alone, stemming from different dynamics such as those from a primitive equation model? (Q4) Are hybrid models better suited than purely-data driven models when less training data is available?

Section 2 first gives an overview of the architecture of our hybrid model, before the training procedure is outlined in Section 3. Section 4 introduces all performance metrics and the data-driven baselines that we will use to verify the effect of adding a physics-based dynamical core. Finally, in Section 6 we discuss the results of training our

Table 1
Physics-Based Components of PseudospectralNet That Are Used for Each of the Experiments, Separated by Whether They Are Computed in the Spectral (Using as Basis Spherical Harmonics, SH) Space or the Grid Space, and Whether They Are Computed Online at Every Time Step or Static and Identical at Every Timestep

	SH space $\mathcal{K}_{l,m}$	
Experiment	Static	Online
QG3	l, m	$\Delta, \Delta^4, \partial_\lambda$
SpeedyWeather + ERA5	l, m, F	$\Delta, \Delta^4, \partial_\lambda, TR$
	Grid space $\mathcal{K}(x)$	
Experiment	Static	Online
QG3	$\cos\phi, \mu, C, \partial_\mu C, \partial_\lambda C, \Delta C$	$\Delta, \Delta^4, \partial_\mu, \partial_\lambda, J$
SpeedyWeather + ERA5	$\cos\phi, \mu, C, \partial_\mu C, \partial_\lambda C, \Delta C, h, \partial_\mu h, \partial_\lambda h, LS$	

Note. Table 2 Lists and names the components abbreviated here.

architecture on three different data sets: quasi-geostrophic model data, primitive equation model data, and reanalysis data. The appendix provides additional detail on the technical implementation of the model.

2. Methods: PseudoSpectralNet

Atmospheric models such as the QG3 model are discretized partial differential equations commonly solved in spectral space. It is therefore natural to formulate our hybrid model as a NDE (R. T. Q. Chen et al., 2018; Gelbrecht, Boers, & Kurths, 2021), so that we directly model the governing equation, or more specifically in this case the tendencies of potential vorticity q, with a model

$$\dot{q} = f_{\mathbf{p}}(q, t), \tag{1}$$

where $f_{\mathbf{p}}$ is (partially) parameterized by an ANN with parameters $\mathbf{p}$. The horizontal spectral space is expressed in real-valued spherical harmonics S^2 of degree l and order m denoting the corresponding spherical harmonic wavenumbers (see Supporting Information S1 for details on the transforms and spherical harmonics). The scalar q is therefore discretized in space as $q_{l,m,k}$ with k the vertical index $k = 1, \ldots, N$ on N_z equi-pressure surfaces. NDEs have proven to be suitable to learn spatiotemporally chaotic systems from data in previous work (Gelbrecht, Boers, & Kurths, 2021; Gelbrecht, Lucarini, et al., 2021) and are extendable to adhere to physical constraints as well (White, Büttner, et al., 2024; White, Kilbertus, et al., 2024).

Previous work (Bonev et al., 2023; Watt-Meyer et al., 2023) demonstrated that ML models that make use of spectral methods are a promising approach for modeling atmospheric dynamics. In this project we'll take a similar approach. However, we go beyond just using a spectral transform within our model as Bonev et al. (2023) and add a physics-based dynamical core to our model.

For this purpose, we mimic the pseudospectral solution approach, which many atmospheric models such as the QG3 model use. At each time step we transform between the spectral space with spherical harmonics as its basis functions and the grid (a regular longitude-latitude grid with Gaussian latitudes) space with transforms $\mathcal{S}$ (grid to spectral) and $\mathcal{S}^{-1}$ (spectral to grid) to compute operations in the space they are easiest to compute in:

$$\dot{q} = f_{\mathbf{p}_1}\left(q(t), \mathcal{S}\left\{g_{\mathbf{p}_2}\left(\mathcal{S}^{-1}\{q\}\right)\right\}, t\right), \tag{2}$$

with parameterized functions $f_{\mathbf{p}_1}$ in spectral space and $g_{\mathbf{p}_2}$ in grid space. Our approach is to use multi-layer perceptrons (MLPs) NN_{SH} and NN_{grid} in spherical harmonic (SH) or grid-point space ($grid$) as these functions, but also to use the physical parts of the QG3 model as additional inputs to those networks. Those parts are, for example, differential operators, drag coefficients, or orography (see Table 1 for a full list). Hence, we let the MLPs freely combine and parameterize the individual components of the physics-based model, and add a UNet (Ronneberger et al., 2015) as an additional purely data-driven correction to that. NN_{grid} is defined as a single-

Table 2
List of the Physical Model Components of PseudospectralNet

Static	l	Spherical harmonics degree
	m	Spherical harmonics order
	F	QG3 forcing, computed from long-term reanalysis data (see Supporting Information S1)
	$\cos\phi$	Cosine of latitude ϕ
	μ	$\sin\phi$—Sine of latitude ϕ
	C	QG3 drag coefficient
	h	Orography
	LS	Land-sea mask
Online	Δ	Laplace operator (on a sphere)
	Δ^4	Hyperdiffusion
	∂_λ	Longitudinal derivative
	∂_μ	Latitudinal derivate
	J	MM QG3 Advection operator
	TR	MM QG3 temperature relaxation

column model, that is, the same function is applied to every horizontal grid point but maps across the vertical. Similarly, the same NN_{SH} is applied to all spherical harmonics l, m (or channels), taking as input all vertical k. The UNet on the other hand uses convolutions to map between entire spatial fields.

The basic structure of *PseudospectralNet* (PSN) is therefore:

$$\dot{q} = NN_{SH}(q(t), \mathcal{K}_{l,m}, \mathcal{S}\{NN_{grid}(q(\mathbf{x},t), \mathcal{K}(q(\mathbf{x},t))), \mathrm{UNet}(q(\mathbf{x},t), \mathcal{K}(q(\mathbf{x},t)))\}). \quad (PSN)$$

The physics-based components of the model are split into two parts as well: one whose outputs are in spectral space $\mathcal{K}_{l,m}$ and another one whose outputs are in the grid space $\mathcal{K}(x)$. We note that some of the computations of $\mathcal{K}(x)$ are performed pseudospectrally, that is, with input in the spectral space, but output in the grid space. This applies for example, to meridional derivatives. Other parts of $\mathcal{K}(x)$ are computed purely in the grid space.

For both of these *knowledge layers* $\mathcal{K}_{l,m}$ and $\mathcal{K}(x)$, there are processes that are computed dynamically or *online* at each time step, for example, derivatives or the advection operator J, and parameters that remain constant and are thus pre-computed and given as additional inputs. Tables 1 and 2 give an overview of all individual processes considered. $\mathcal{K}_{l,m}$ and $\mathcal{K}(x)$ provide the neural networks with the advection and dissipation of the QG3 model, spatial derivatives of its inputs, and constant parameters such as orography and land sea mask. In a sense, the knowledge layers deconstruct the QG3 model into all of its individual expressions or contributions (see Table 1), and the MLPs NN_{SH} and NN_{grid} then learn to freely combine those and the additional purely data-driven UNet. The UNet architecture (Ronneberger et al., 2015) proved to be extremely capable at processing not only images, but also geospatial and atmospheric data (e.g., Hess et al., 2022; Weyn et al., 2020). In principle, all parameters (e.g., drag coefficient C) of the physics-based dynamical core can be made trainable as well, and can therefore be re-calibrated during training. In practice, this adds only a few parameters compared to a large number of weights in the neural networks and could avoid time-consuming manual re-tuning. Figure 1 shows an overview of the dataflow of the PSN, and the Supporting Information S1 goes into more detail on the implementation of the knowledge layers and setup of the neural networks.

A crucial requirement to enable this approach is that all physics-based components need to be differentiable by reverse-mode AD, and that they can run on GPUs. The differentiability is needed for *online* training of all parameters while the complete model integrates. This approach showed great success, increasing the stability of learned ML models (e.g., Gelbrecht et al., 2023; Kochkov et al., 2021). GPU compatibility significantly improves the computational efficiency of the model due to its ANN components and high dimensionality. We achieved both

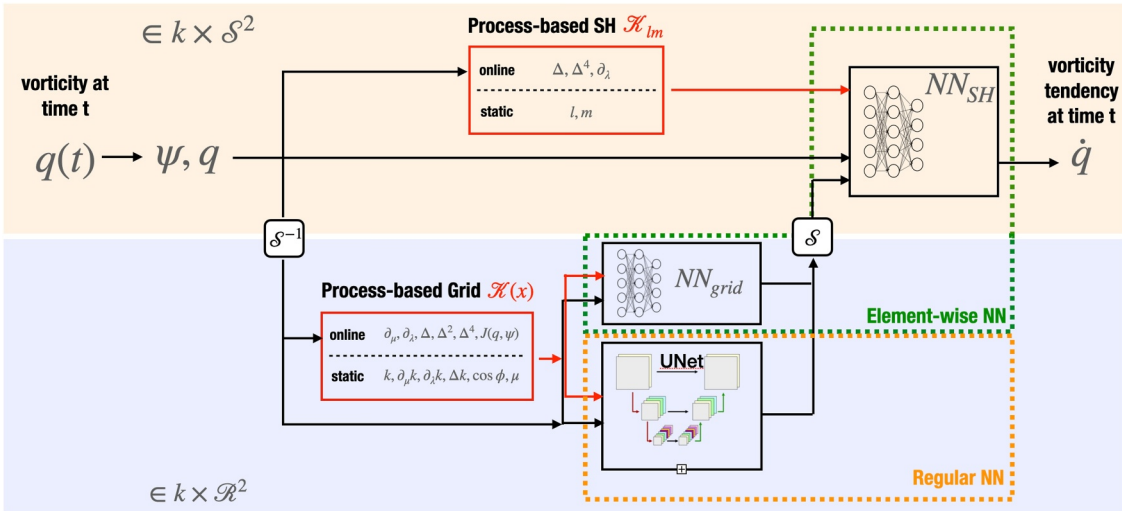

Figure 1. Overview of the data flow of PseudospectralNet, from left to right. The potential vorticity tendencies $\dot{q}$ are computed partially in the grid space (light blue) and partially in the spectral space (light orange) with transforms S and inverse transforms S^{-1} between the spaces. Process-based components $\mathcal{K}_{l,m}$, $\mathcal{K}(x)$ are marked in red, whereas data-driven neural network components are marked in black. NN_{SH} and NN_{grid} are elementwise-defined multi-layer perceptrons (green dashed line), whereas the UNet receives the complete spatial field at once (orange dashed line). The input q and output $\dot{q}$ are all three levels of the atmosphere that the model considers. The atmospheric levels are concatenated along the channel dimension together with all information the process-based layers add. All of these are concatenated together along the channel dimension when input into a neural network.

by implementing the QG3 model in the Julia language (Bezanson et al., 2017) and formulating it in matrix form to avoid array mutation, a current constraint of many AD libraries. Implementation details are described in the Supporting Information S1.

3. Training

PSN is trained on atmospheric data. The first experiments use different model data, the last experiment uses reanalysis data. We compute potential vorticity data sets for each of the individual experiments. These will serve as our ground truth data. Aside from the training data the gradient computation is the critical point of the training process. The chaotic nature of dynamical systems such as atmospheric circulation makes the direct computation of gradients of long trajectories infeasible due to the exponential accumulation of errors (Metz et al., 2021; Q. Wang et al., 2014). To mitigate this problem, we choose the same training strategy as in (Gelbrecht, Boers, & Kurths, 2021): we start training the model by minimizing the one-step ahead error for a set amount of epochs before slowly increasing the number of time steps $N_{\max}$ we integrate the model for. We integrate the NDE with collocation points at discrete time steps $t_i = t_0 + i_t\,\Delta t$. The training objective therefore follows as the mean square difference between the ground truth $q_{l,m,k}$ in the spectral space and the predicted trajectory of the potential vorticity by PSN $\hat{q}_{l,m,k}$, with parameters $\mathbf{p}$ from initial condition $q_{l,m,k}(t = 0)$:

$$L(\mathbf{p}) = \sum_{i_t=1}^{N_{\max}} \sum_{l,m,k}^{l=l_{\max}-i_t+1} \left(q_{l,m,k}(i_t) - \hat{q}_{l,m,k}(i_t; q_{lmk}(0), \mathbf{p})\right)^2. \tag{3}$$

We apply a progressive truncation to the output field (see the upper limit of the second sum). For $i_t = 1$ the full spectral fields are considered, for $i_t = 2$ the maximum wavenumber $l_{\max}$ is reduced by one, and subsequently further for larger i_t. This is motivated to focus on increasingly larger spatial scales with increasing forecast lead time (demonstrated before e.g., by Kochkov et al. (2024)). All results presented in this article use a truncation with $l_{\max} = 42$. When training the one-step ahead prediction a Runge-Kutta 4 solver is used and gradients $\nabla_{\mathbf{p}} L$ with respect to the model parameters $\mathbf{p}$ are computed directly by AD (discritize-then-optimize). This one-step ahead prediction is performed batched with 14 trajectories and their gradients are computed at once. The further optimization for longer rollouts is then performed with a Tsitouras solver (Tsitouras, 2011) and adjoint sensitivity analysis to compute the gradients (optimize-then-discritize) on single trajectories using an interpolating adjoint algorithm (Ma et al., 2021). The switching of the gradient computation method was done primarily to speed up the

training processes, as the adjoint sensitivity analysis is significantly slower than the direct AD computation of the gradients for our model implementation. We start optimizing the model for $N_{\max} = 1$ for 300 epochs, before increasing the trajectory length stepwise by one, continue training and increase the trajectory length again until $N_{\max} = 6$. The optimization itself was then performed with an AdamW optimizer (Loshchilov & Hutter, 2019) that includes a weight decay for the parameters and with learning rate scheduled to be sinusoidally decaying to avoid local minima.

4. Evaluation Metrics and Baselines

In this study, we want to investigate whether hybrid models for atmospheric dynamics as described above can achieve short-term predictability, long-term numerical stability and generalizability to more complex dynamics than those of the physics-based dynamical core of the model as defined in our research questions Q1–Q4. In order to verify the model with respect to these goals, we train and test PSN on different types of atmospheric data. First, we show that PSN can learn dynamics similar to those of its physics-based dynamical core by learning QG3 data while withholding information on some of the dynamics from the model's core. Second, we test how well PSN generalizes on data from the primitive equation model SpeedyWeather.jl (Klöwer et al., 2024). Third, we investigate how well PSN can predict the ERA5 reanalysis data.

The data set used or generated for each applications is a single, long trajectory of the system that is split into training and validation sets. To measure short term predictability we compute an average normalized error of $N_{samples} = 25$ predicted trajectories in grid space, each starting from a different initial condition $q(\mathbf{x}, t = 0)$ outside of the training set, against the ground truth $q(\mathbf{x}, t)$ in grid space:

$$e(t) = \left\langle \frac{\|q(\mathbf{x},t) - \mathcal{S}^{-1}\{\hat{q}(t_i; q(\mathbf{x},0), \mathbf{p})\}\|_2}{<\|q(\mathbf{x},t)\|_2^2>_t^{1/2}} \right\rangle_{N_{samples}}. \tag{4}$$

We additionally report the forecast horizon defined by the times at which $e(t)$ first crosses 0.01 and 0.1 as $t(e > 0.01)$ and $t(e > 0.1)$, comparable to the *valid times* used by Pathak et al. (2018). It is important to note that here we assume perfect knowledge of the initial condition. We do not use any data assimilation techniques that would need to be used for an operational weather forecast. The chaoticity of the system still plays a crucial role though, as every smallest discrepancy of our model to the ground truth system will lead to exponentially accumulating errors over time.

To evaluate long-term numerical stability of PSN, we integrate each trained model for up to 100 model years from 50 different initial conditions from the test data set (hence outside of the training data set). When the solution becomes numerically unstable, we denote that time as the instability time t_s and report the 5%, 50%, and 95% percentile of all trials. We also test the trained models for individual initial conditions for up to 100 years. To detect the instability we use the automatic detection of the Tsitouras solver implemented in DifferentialEquations.jl that is also used when training the model via adjoint sensitivity analysis (Rackauckas & Nie, 2017; Tsitouras, 2011). Additionally, we also monitor the average kinetic energy during the long integrations and the average power spectra (see Supporting Information S1 for more details).

We compare the performance of PSN with regards to the above metric with several baselines, both physics-based and purely data-driven. As a basic baseline we use the constant climatology. We also use the full QG3 model itself and the horizontal advection operator J alone as baselines. On the other hand, we show how the purely data-driven components of the PSN perform, test a UNet with the same hyperparameters as the one of PSN in a recursive neural network setup (Grid UNet), and test the PSN without its process-based components, that is, a UNet and two small MLPs in an NDE using a pseudospectral approach (PS UNet). The latter is therefore a direct baseline for the impact of adding the process-based components to the model.

The models are trained in the same manner as the PSN. We can therefore regard the baselines, especially PS UNet, as a way to investigate whether adding the process-based components of the QG3 model improves the forecasts and stability. The Supporting Information S1 provides more details on the implementation of these baselines.

5. Applications

In order to study the effect of the physics-based dynamical core compared to our baselines and answer our research questions, we perform three initial experiments with PSN in increasing order of complexity. First, we train on data with similar dynamics to its physics-based core. For this purpose we test a reduced version of PSN with some components withheld on data from the full QG3 model. With this approach we test if our model can emulate the data faithfully and achieve very long stable integrations. Subsequently, we test how well our PSN generalizes to more complex data: first on data from a primitive equation model, and then ultimately on reanalysis data.

5.1. Training on the QG3 Model

For the first experiment, we test how well PSN performs on data similar to that of its physics-based core. For this purpose, we use data from the full QG3 model but set up a version of PSN that does not include all components of the QG3 Model. Specifically, we withhold information about the forcing F that was computed from long-term climatological data following (Corti et al., 1997), and the temperature relaxation TR from this version of PSN. The forcing in the QG3 is constant in time and used to inject potential vorticity into the system (see Supporting Information S1 for a detailed derivation). This setup of withholding the components mimics the situation for more complex models, where some terms such as the advection are more certain but others such as the parameterizations are less certain, which we therefore aim to learn from data. We therefore test whether the ANNs can learn these functions from a short trajectory and fit the complete model sufficiently well. Table 1 shows an overview of which parts of the model are used for this experiments, and Table S2 in Supporting Information S1 lists the used hyperparameters. In previous studies (Gelbrecht, Boers, & Kurths, 2021; Liu et al., 2024) we have seen already that one of the advantages of hybrid models is the ability to train with comparably small amounts of data. Therefore, we also train on short trajectories only here, but also explore the effect of increasing the length of the training data set. We train two models, one with a trajectory of 4 days (PSN short), and one with a 1 month long trajectory (PSN), both at a time step of about 0.8 h.

All pseudospectral NDE approaches (PSN, PSN short, PS UNet) significantly outperform the grid-based UNet-RNN (Figure 2) even though the same UNet hyperparmeters are used in all of these models. The PSN achieves a forecast horizon $t(e > 0.01) = 64.1$ h, whereas both the PSN short and the PS UNet achieve significantly shorter forecast horizon with $t(e > 0.01) = 47.5$ h and $t(e > 0.01) = 22.5$ h, respectively. The RNN-based Grid UNet baseline only achieves $t(e > 0.01) = 2.5$ h and therefore fails to predict longer trajectories well. Table 3 gives an overview over these results. The snapshots in Figure 2 also show that the difference to the ground truth emerges slowly, spread across all latitudes.

We test the numerical stability of the best performing models, PSN and PS UNet, by integrating them for 100 years from 50 different, randomly selected initial conditions from the test data set. Whereas PS UNet became unstable in all experiments after about 48 years (with 5%, 50%, and 95% percentiles $t_s = [48.1, 48.2, 48.4]$ y), PSN integrated for 100 years in all experiments. The PSN somewhat overestimates the energy and variance of the energy of the ground truth, but remains stable for the whole time span of the simulation (see Supporting Information S1). It seems therefore promising to incorporate explicit energy constraints (White, Kilbertus, et al., 2024) or correction schemes in future versions of the method. An analysis of the average power spectra (see Supporting Information S1 for details) also shows that the spectra averaged over 4-year windows remain stable and virtually indistinguishable for PSN. In contrast, we can see a clear trend in the spectra of the PS UNet model, losing energy particularly at high wavenumbers over time. We do note though that during hyperparameter optimization, while all PSN models outperform the PS UNet models in terms of stability, not all of them were as stable as our finally reported version. The Swish activation (Ramachandran et al., 2017) for example, achieves much better results in this respect than a ReLU activation.

With this experiment we therefore showed that even with just very small amounts of training data, we can learn the dynamics of the QG3 model, and in this way also learn TR and the forcing F that is otherwise computed from long-term averages. Applying PSN to data similar to that of its physics-based core thus yields stable simulations at decadal to centennial time scales with good short-term predictability, significantly outperforming the baselines, even when it is only trained on a short training data set.

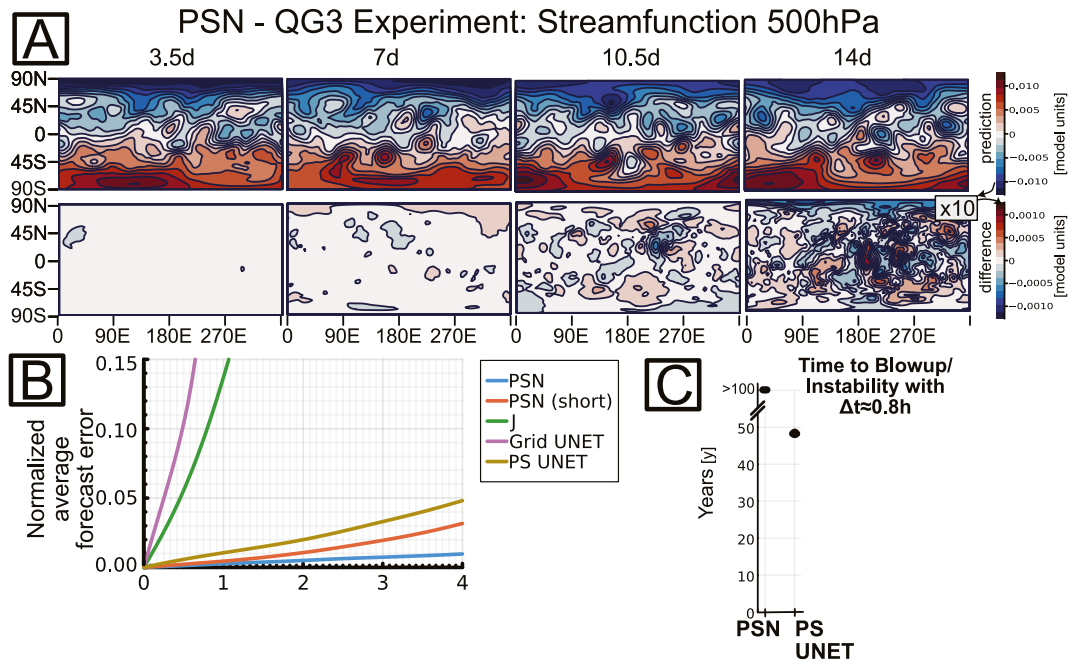

Figure 2. Overview of the results of training a PseudospectralNet (PSN) with incomplete process-knowledge on data from the full quasi-geostrophic model (QG3). (a) Snapshots of the global streamfunction ψ at the 500 hPa layer in normalized units of a single trajectory, integrated from initial conditions outside of the training set at different time steps. The upper four panels display the prediction made by the PSN. The lower four panels display the difference of that prediction to the ground truth. Compared to the upper panels, the colorscale is magnified by a factor of 10 to be able to visualize the relatively small differences. (b) Evolution of the normalized average forecast error (Equation 4) of the PSN trained on a month of model data, another PSN trained on just 4 days of data (PSN short) and the baselines described in Section 4 and the Supporting Information S1. The climatology baseline is outside of the depicted area, constant at ≈ 0.32 (c) Stability time of the PSN and the PS UNet baseline measured from 50 different initial conditions as outlined in Section 4. The bar indicates the range between the 5th and 95th percentiles, and >100 indicates that the trajectory did not become unstable within 50 model years.

5.2. Training on a Primitive Equation Model

With the QG3 application, we showed that PSN is able to model quasi-geostrophic data well, but how well does PSN generalize to significantly more complex data? For this purpose, we train PSN on data generated with the primitive equation dry core (meaning no humidity) from SpeedyWeather.jl (Klöwer et al., 2024). The primitive equations of atmospheric circulation use as prognostic variables the vorticity-divergence formulation (relative vorticity $\zeta = \nabla \times \mathbf{u}$, divergence $\mathcal{D} = \nabla \cdot \mathbf{u}$), to represent the horizontal wind $\mathbf{u} = (u, v)$. Further prognostic variables are the surface pressure p_s (specifically its logarithm $\ln p_s$) and the temperature T (Hoskins & Simmons, 1975; Kucharski et al., 2013; Molteni, 2003; Simmons & Burridge, 1981; Simmons et al., 1978):

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= \nabla \times \left(\mathcal{P}_{\mathbf{u}} + (f+\zeta)\mathbf{u}_\perp - W(\mathbf{u}) - R_d T \nabla \ln p_s\right) \\
\frac{\partial \mathcal{D}}{\partial t} &= \nabla \cdot \left(\mathcal{P}_{\mathbf{u}} + (f+\zeta)\mathbf{u}_\perp - W(\mathbf{u}) - R_d T \nabla \ln p_s\right) - \nabla^2\left(\frac{1}{2}(u^2+v^2) + \Phi\right) \\
\frac{\partial \ln p_s}{\partial t} &= -\frac{1}{p_s}\nabla \cdot \int_0^{p_s} \mathbf{u}\, dp \\
\frac{\partial T}{\partial t} &= \mathcal{P}_T - \nabla \cdot (\mathbf{u}T) + T\mathcal{D} - W(T) + \kappa T \frac{D \ln p}{Dt},
\end{aligned}
\tag{5}
$$

with the thermodynamic constants R_d, κ, the geopotential Φ, the vertical advection operator W and the Coriolis parameter f. The primitive equations are used for numerical weather prediction but here simplified with prescribed

Table 3
Results for the Forecast Horizon, the Times at Which the Forecast Error According to Equation 4 First Crosses the Given Threshold

Experiment	Metric	PseudospectralNet	PS UNet	Grid UNet
QG3	$t(e>0.01)$	64.1 h	22.5 h	2.5 h
SpeedyWeather.jl	$t(e>0.1)$	26.6 h	25 h	16.6 h
ERA5	$t(e>0.1)$	8 h	8 h	6 h

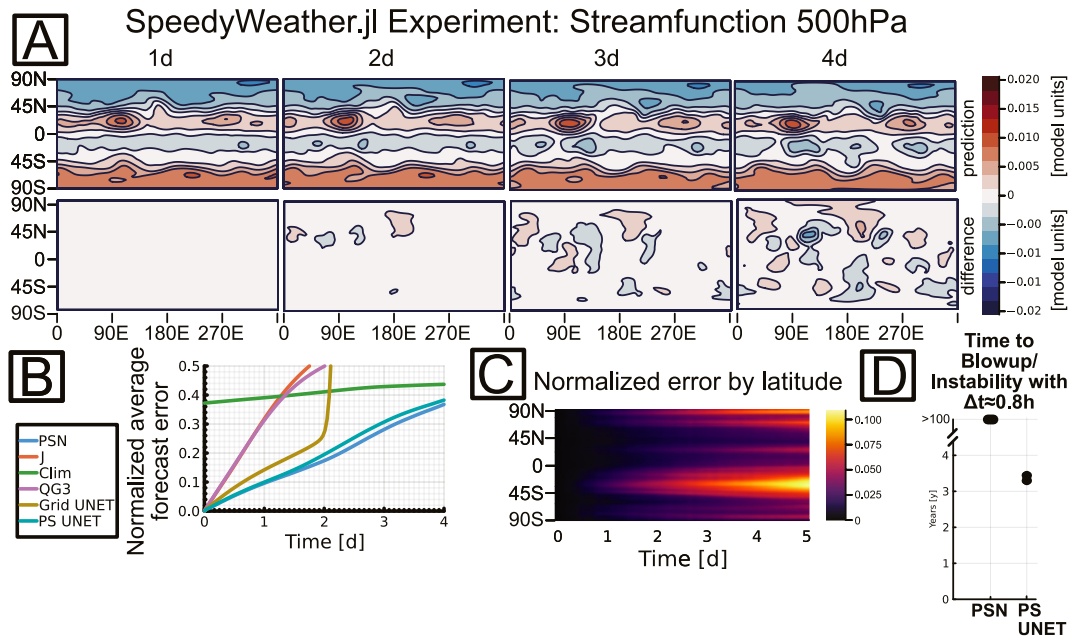

Figure 3. Overview of the results when training PseudospectralNet (PSN) on data from SpeedyWeather.jl. (a) Snapshots of the global streamfunction ψ at the 500 hPa layer in normalized units of a single trajectory, integrated from an initial condition outside of the training set at different time steps. The upper four panels display the prediction made by the PSN. The lower four panels display the difference of that prediction to the ground truth. (b) Evolution of the normalized average forecast error (Equation 4) of PSN trained on 200 days of model data, and the baselines described in Section 4 and the Supporting Information S1. (c) Evolution of the forecast error, similar as in (b) but resolved by latitude to visualize any potential biases due to the quasi-geostrophic process-based core of the model. (d) Stability time of PSN and the PS UNET baseline measured from 50 different initial conditions as outlined in Section 4. The bar indicates the range between the 5% and 95% percentile.

ocean and land surface temperatures. The parameterizations $\mathcal{P}_\mathbf{u}, \mathcal{P}_\mathbf{T}$ represent unresolved physical processes such as convection, surface fluxes and diffusion, and are much simpler and consequently less accurate in comparison to operational weather forecast models.

The primitive equations are discretized using a similar pseudo-spectral method as used in QG3.jl. The time integration is based on a semi-implicit Leapfrog scheme (Hoskins & Simmons, 1975) with Robert-Asselin-Williams filter (Amezcua et al., 2011; Williams, 2011). The vertical is discretized with 3 equispaced σ-levels with $\sigma = \frac{p}{p_s}$, the fraction of pressure p (the vertical coordinate) over surface pressure p_s to mirror with $\sigma = 0.8, 0.5, 0.2$ the setup of the QG3 model. Despite the additional variables in the primitive equation we restrict learning and comparison to the wind field expressed in terms of the streamfunction, from which we compute the potential vorticity q (in the formulation of the QG3 model) to train PSN.

We generate training data by integrating the primitive equation model for 200 days, with the same time step used as before in the QG3 experiment $\Delta t = 0.8$ h. In contrast to the previous experiment, in this case we setup PSN to include all components of the QG3 model, as also shown in Table 1. Otherwise the training procedure is the same as outlined before.

PSN achieves a forecast horizon $t(e > 0.1) = 26.6$ h and $t(e > 0.2) = 55.8$ h for short-term predictability. It thus outperforms the data-driven pseudospectral baseline PS UNet, which achieves $t(e > 0.1) = 25.0$ h and $T(e > 0.2) = 50.0$ h (Figure 3). This is, in turn, significantly longer than the forecast horizon of the Grid UNet at $t(e > 0.1) = 16.6$ h and $t(e > 0.1) = 38.3$ h and of directly using the QG3 itself as a predictor. Table 3 gives an overview over these results. Those forecast horizons are shorter than for the QG3 applications, which is expected because the data from SpeedyWeather.jl is substantially more complex, for example, by involving additional prognostic variables that we do not explicitly resolve in the current version of PSN. Remarkably, while quasi-geostrophic models inherently have biases in the tropics, PSN does not inherit this bias. This points to the UNet as the key contributor for this prediction. Despite this, the hybrid PSN model still outperforms the PS UNet

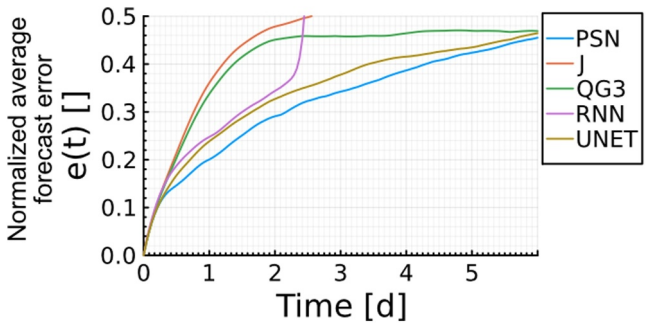

Figure 4. Evolution of the normalized average forecast error (Equation 4) of PseudospectralNet trained on 1 year of ERA5 reanalysis data, and the baselines described in Section 4 and the Supporting Information S1.

in terms of short-term predictability, albeit not as strongly as in the QG3 application. The PSN outperforms the PS UNet in terms of numerical stability clearly though. While PSN is stable for the full tested 100 years, PS UNet achieves only 5%, 50%, 95% percentiles $t_s = [3.2, 3.3, 3.5]$ y. In the analysis of the average power spectra PSN slightly overestimates the tails of spectra in longer rollouts and shows a very small trend over the long rollout. PS UNet on the other hand is never stable in the first place and becomes numerically unstable quickly, so that the rollout can't be continued. Adding the physics-based components therefore significantly increases the stable integration time in this experiment. The PSN is able to capture the kinetic energy of the primitive equation well (see Supporting Information S1). The kinetic energy of PS UNet on the other hand increases very quickly after a few weeks of integration time.

Hence, even for dynamics more complex than its physics-based core, PSN outperforms the data-driven baseline. Adding the process-core yields improved short-term predictability, and more importantly significantly longer stable integration than with the purely data-driven baseline.

5.3. Training on Reanalysis Data

Lastly, we investigate how PSN performs when applied to ERA5 reanalysis data (Hersbach et al., 2020). The dynamics of reanalysis data are substantially more complex than the dynamics of a three-level quasi-geostrophic model, or the primitive equation model used above. As such, we do not expect to be competitive with either traditional NWP, nor ML-based weather models. Instead, we intend to investigate whether including physics-based components leads to any improvements against the purely data-driven approaches in terms of short-term predictability and long-term stability, even when the underlying dynamics are much more complex. For that comparison we use the same baselines as before. Similar to the experiment on the primitive equation model, we restrict the training data to the three equi-pressure levels at which the QG3 is defined as well (200, 500, 800 hPa), and compute the potential vorticity from the wind field of one year of hourly reanalysis data. PSN for this experiment has the same architecture and hyperparameters as in the previous primitive equation example. It includes all physics-based components of the QG3 model. The same training procedure as for the other applications is applied here as well. Figure 4 shows the evolution of the normalized average forecast error of all trained models. While those errors are considerably larger than in our previous applications, and nowhere near competitive weather prediction models, we still notice that the hybrid model performs best, and better than our purely data-driven baselines.

6. Discussion

We have proposed a novel architecture for hybrid atmospheric models, which combines a physics-based core—here from a quasi-geostrophic model—with artificial neural networks. PSN deconstructs the individual components of a physics-based atmospheric model, allows them to be potentially reparameterized and adaptively coupled via AD, and adds data-driven capacities to it. With the physics-based dynamical core still comparably simple, our goal was to investigate whether the approach works to model data from atmospheric models of similar and higher complexity. To do so we compared PSN to baselines that remove the physics-based parts from the model. In the introduction, we outlined four research questions that we intended to answer in this study on our model: (Q1) do we reduce biases of the physics-based part, (Q2) does the inclusion of physics-based parts enable better generalization and stability than data-driven models, (Q3) can the hybrid model generalize to dynamics that are more complex than of those of the physics-based part alone, (Q4) can the hybrid model successfully learn from less training data than a purely data-driven model?

The QG3 experiment shows that PSN is able to model the dynamics of the full QG3 model remarkably well. It strongly outperforms the purely data-driven baseline and achieves high accuracy for short-term forecasts and stable long-term integration. Furthermore, we showed that the architecture can generalize to atmospheric data of higher complexity, for which the physics-based dynamical core still provides benefits, especially for the numerical stability of the model. When training our model on data from a primitive equation data, our hybrid model

did not show the typical biases of quasi-geostrophic models in the equatorial regions, corroborating Q1 and Q3. We are also in these cases able to achieve long-term numerical stability when the purely data-driven baselines fail to do so, verifying Q2. Furthermore, we demonstrated as well that our model is able to learn from very small training data, with just months up to a year used in the experiments we presented, answering Q4.

The key advantages that our study corroborates is that adding physics knowledge lowers the training error below a level that is achieved with the purely data-driven model on short training data. This saves both memory and compute during training, and might translate well to purely training on observational data, which is sparse in space. In the case of the here proposed PSN we do add computational complexity at inference compared to our purely data-driven baselines as we intended to directly study the effect of adding the process-based components. But in principle, hybrid atmospheric models can be set up to actually save computational time, for example, by emulating costly processes.

While PSN in the form presented here is not competitive with much more comprehensive models, both physics-based and ML, we used it to study the possibilities and properties of setting up such a hybrid model, and if adding physics-based parts does results in any benefits, in a significantly more comprehensive manner than previous studies on hybrid models that we are aware of, PSN does not resolve all prognostic variables of the primitive equations in SpeedyWeather.jl, and not nearly all dynamics and processes necessary to represent a reanalysis data set such as ERA5. Despite this, we showcase the potential of hybrid models, bridging between ML models and classical models, by examining their numerical stability, prediction accuracy and decreased training data needs compared to pure ML models.

Future versions of our model could include more variables in either a purely data-driven or a hybrid approach as well. Aside from that, there are many other potential pathways to further advance our approach. A significantly more complex physics-based core should be considered, that also includes important parameterizations. Adding further physics-based components as explicit constraints to the model to enhance its stability seems very promising as well given our results. When increasing the physics-based complexity, a deeper ANN architecture, trained on more data, also only seems logical.

Stability and long-term predictions are an issue for many ML models. We have shown that adding a physics-based dynamical core to those models helps to decrease biases while achieving increased generalizability across dynamics of different complexity.

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Data Availability Statement

This article is accompanied by several Julia packages: NeuralDELux.jl, QG3.jl (Gelbrecht, 2024) and NeuralQG3.jl (Gelbrecht, 2025) that are detailed in the Supporting Information S1. The third application uses the ERA5 reanalysis data set from ECMWF (Hersbach et al., 2020).

Acknowledgments
We thank the three anonymous reviewers for their valuable feedback that improved this paper. This work received funding from the Volkswagen Foundation. MK acknowledges funding from the National Science Foundation and Schmidt Sciences through an Eric & Wendy Schmidt AI in Science Postdoctoral Fellowship. NB acknowledges further funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101137601 (ClimTip) and under the Marie Sklodowska-Curie grant agreement No. 956170. The authors gratefully acknowledge the Ministry of Research, Science and Culture (MWFK) of Land Brandenburg for supporting this project by providing resources on the high performance computer system at the Potsdam Institute for Climate Impact Research. Open Access funding enabled and organized by Projekt DEAL.

References

Amezcua, J., Kalnay, E., & Williams, P. D. (2011). The effects of the RAW filter on the climatology and forecast skill of the SPEEDY model. *Monthly Weather Review*, *139*(2), 608–619. https://doi.org/10.1175/2010MWR3530.1

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

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. arXiv:2211.02556.

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. In *Proceedings of the 40th international conference on machine learning*. JMLR.org.

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. arXiv:2304.02948.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. *arXiv*. https://doi.org/10.48550/ARXIV.1806.07366

Corti, S., Giannini, A., Tibaldi, S., & Molteni, F. (1997). Patterns of low-frequency variability in a three-level quasi-geostrophic model. *Climate Dynamics*, *13*(12), 883–904. https://doi.org/10.1007/s003820050203

Gelbrecht, M. (2024). Qg3.jl. *Zenodo*. https://doi.org/10.5281/zenodo.14547916

Gelbrecht, M. (2025). maximilian-gelbrecht/neuralqg3.jl: Initial paper release. *Zenodo*. https://doi.org/10.5281/zenodo.14711058

Gelbrecht, M., Boers, N., & Kurths, J. (2021). Neural partial differential equations for chaotic systems. *New Journal of Physics*, *23*(4), 043005. https://doi.org/10.1088/1367-2630/abeb90

Gelbrecht, M., Lucarini, V., Boers, N., & Kurths, J. (2021). Analysis of a bistable climate toy model with physics-based machine learning methods. *The European Physical Journal Special Topics*, *230*(14), 3121–3131. https://doi.org/10.1140/epjs/s11734-021-00175-0

Gelbrecht, M., White, A., Bathiany, S., & Boers, N. (2023). Differentiable programming for earth system modeling. *Geoscientific Model Development*, *16*(11), 3123–3135. https://doi.org/10.5194/gmd-16-3123-2023

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

Hess, P., Drüke, M., Petri, S., Strnad, F. M., & Boers, N. (2022). Physically constrained generative adversarial networks for improving precipitation fields from earth system models. *Nature Machine Intelligence*, *4*(10), 828–839. https://doi.org/10.1038/s42256-022-00540-1

Hoskins, B. J., & Simmons, A. J. (1975). A multi-layer spectral model and the semi-implicit method. *Quarterly Journal of the Royal Meteorological Society*, *101*(429), 637–655. https://doi.org/10.1002/qj.49710142918

Innes, M. (2019). Don't unroll adjoint: Differentiating SSA-form programs. arXiv:1810.07951.

Klöwer, M., Gelbrecht, M., Hotta, D., Willmert, J., Silvestri, S., Wagner, G. L., et al. (2024). SpeedyWeather.jl: Reinventing atmospheric general circulation models towards interactivity and extensibility. *Journal of Open Source Software*, *9*(98), 6323. https://doi.org/10.21105/joss.06323

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, *118*(21), e2101784118. https://doi.org/10.1073/pnas.2101784118

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., et al. (2024). Neural general circulation models for weather and climate. *Nature*, *632*(8027), 1060–1066. https://doi.org/10.1038/s41586-024-07744-y

Kondrashov, D., Ide, K., & Ghil, M. (2004). Weather regimes and preferred transition paths in a three-level quasigeostrophic model. *Journal of the Atmospheric Sciences*, *61*(5), 568–587. https://doi.org/10.1175/1520-0469(2004)061⟨0568:WRAPTP⟩2.0.CO;2

Kucharski, F., Molteni, F., King, M. P., Farneti, R., Kang, I.-S., & Feudale, L. (2013). On the need of intermediate complexity general circulation models: A "SPEEDY" example. *Bulletin of the American Meteorological Society*, *94*(1), 25–30. https://doi.org/10.1175/BAMS-D-11-00238.1

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., et al. (2023). Graphcast: Learning skillful medium-range global weather forecasting. arXiv:2212.12794.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. arXiv:2010.08895.

Liu, N., Fan, Y., Zeng, X., Klöwer, M., Zhang, L., & Yu, Y. (2024). Harnessing the power of neural operators with automatically encoded conservation laws. Retrieved from https://arxiv.org/abs/2312.11176

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. arXiv:1711.05101.

Lucarini, V., & Gritsun, A. (2020). A new mathematical framework for atmospheric blocking events. *Climate Dynamics*, *54*(1), 575–598. https://doi.org/10.1007/s00382-019-05018-2

Ma, Y., Dixit, V., Innes, M. J., Guo, X., & Rackauckas, C. (2021). A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. In *2021 IEEE high performance extreme computing conference (HPEC)* (pp. 1–9). https://doi.org/10.1109/HPEC49654.2021.9622796

Marshall, J., & Molteni, F. (1993). Toward a dynamical understanding of planetary-scale flow regimes. *Journal of the Atmospheric Sciences*, *50*(12), 1792–1818. https://doi.org/10.1175/1520-0469(1993)050⟨1792:TADUOP⟩2.0.CO;2

Metz, L., Freeman, C. D., Schoenholz, S. S., & Kachman, T. (2021). Gradients are not all you need. *arXiv*. https://doi.org/10.48550/ARXIV.2111.05803

Molteni, F. (2003). Atmospheric simulations using a GCM with simplified physical parametrizations. I: Model climatology and variability in multi-decadal experiments. *Climate Dynamics*, *20*(2), 175–191. https://doi.org/10.1007/s00382-002-0268-2

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. arXiv:2202.11214.

Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B. R., Girvan, M., & Ott, E. (2018). Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *28*(4), 041101. https://doi.org/10.1063/1.5028373

Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T. R., El-Kadi, A., Masters, D., et al. (2024). Gencast: Diffusion-based ensemble forecasting for medium-range weather. Retrieved from https://arxiv.org/abs/2312.15796

Rabier, F., Järvinen, H., Klinker, E., Mahfouf, J.-F., & Simmons, A. (2000). The ecmwf operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, *126*(564), 1143–1170. https://doi.org/10.1002/qj.49712656415

Rackauckas, C., & Nie, Q. (2017). Differential equations.jl—A performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, *5*(1), 15. https://doi.org/10.5334/jors.151

Rackow, T., Koldunov, N., Lessig, C., Sandu, I., Alexe, M., Chantry, M., et al. (2024). Robustness of AI-based weather forecasts in a changing climate. Retrieved from https://arxiv.org/abs/2409.18529

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv:1710.05941.

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., Russel, 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

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention (miccai)* (Vol. 9351, pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

Simmons, A. J., & Burridge, D. M. (1981). An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates. *Monthly Weather Review*, *109*(4), 758–766. https://doi.org/10.1175/1520-0493(1981)109⟨0758:AEAAMC⟩2.0.CO;2

Simmons, A. J., Hoskins, B. J., & Burridge, D. M. (1978). Stability of the semi-implicit method of time integration. *Monthly Weather Review*, *106*(3), 405–412. https://doi.org/10.1175/1520-0493(1978)106⟨0405:SOTSIM⟩2.0.CO;2

Tsitouras, C. (2011). Runge–Kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, *62*(2), 770–775. https://doi.org/10.1016/j.camwa.2011.06.002

Um, K., Brand, R., Fei, Y. R., Holl, P., & Thuerey, N. (2020). Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. *arXiv*. https://doi.org/10.48550/ARXIV.2007.00016

Vannitsem, S. (2017). Predictability of large-scale atmospheric motions: Lyapunov exponents and error dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *27*(3), 032101. https://doi.org/10.1063/1.4979042

Wang, C., Pritchard, M. S., Brenowitz, N., Cohen, Y., Bonev, B., Kurth, T., et al. (2024). Coupled ocean-atmosphere dynamics in a machine learning earth system model. Retrieved from https://arxiv.org/abs/2406.08632

Wang, Q., Hu, R., & Blonigan, P. (2014). Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, *267*, 210–224. https://doi.org/10.1016/j.jcp.2014.03.002

Watt-Meyer, O., Dresdner, G., McGibbon, J., Clark, S. K., Henn, B., Duncan, J., et al. (2023). Ace: A fast, skillful learned global atmospheric model for climate prediction. arXiv:2310.02074.

Weyn, J. A., Durran, D. R., & Caruana, R. (2020). Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, *12*(9), e2020MS002109. https://doi.org/10.1029/2020MS002109

White, A., Büttner, A., Gelbrecht, M., Duruisseaux, V., Kilbertus, N., Hellmann, F., & Boers, N. (2024). Projected neural differential equations for learning constrained dynamics. Retrieved from https://arxiv.org/abs/2410.23667

White, A., Kilbertus, N., Gelbrecht, M., & Boers, N. (2024). Stabilized neural differential equations for learning dynamics with explicit constraints. arXiv:2306.09739.

Williams, P. D. (2011). The RAW filter: An improvement to the Robert–Asselin filter in semi-implicit integrations. *Monthly Weather Review*, *139*(6), 1996–2007. https://doi.org/10.1175/2010MWR3601.1

References From the Supporting Information

Innes, M. (2018). Flux: Elegant machine learning with Julia. *Journal of Open Source Software*, *3*(25), 602. https://doi.org/10.21105/joss.00602

Pal, A. (2023). Lux: Explicit parameterization of deep neural networks in Julia. *Zenodo*. https://doi.org/10.5281/zenodo.7808904

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., & Edelman, A. (2021). Universal differential equations for scientific machine learning.

Roads, J. O. (1987). Predictability in the extended range. *Journal of the Atmospheric Sciences*, *44*(23), 3495–3527. https://doi.org/10.1175/1520-0469(1987)044⟨3495:PITER⟩2.0.CO;2