Deep Learning Emulators for Accessible Climate Projections

by

Björn Lütjens

B.S., Technical University of Munich (2017)
S.M., Massachusetts Institute of Technology (2019)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN
AERONAUTICS AND ASTRONAUTICS
with a major in Machine Learning and minor in Earth System Modeling

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Björn Lütjens

Department of Aeronautics and Astronautics
May 23, 2023

Certified by: Dava J. Newman

Apollo Program Professor of Astronautics; Director of the MIT Media Lab
Thesis Supervisor

Accepted by: Jonathan P. How

R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

This doctoral thesis has been examined by a Committee of the Department of Aeronautics and Astronautics as follows:

Prof. Dava J. Newman .....
Chair, Thesis Committee
Apollo Program Professor of Astronautics; Director of the MIT Media Lab

Dr. Catherine H. Crawford .....
Member, Thesis Committee
IBM Fellow, IBM Research

Prof. Youssef M. Marzouk .....
Member, Thesis Committee
Professor of Aeronautics and Astronautics, MIT

Dr. Mark S. Veillette .....
Reader, Thesis Committee
Senior Technical Staff, MIT Lincoln Laboratory

Dr. Christopher N. Hill .....
Reader, Thesis Committee
Principal Research Engineer, Department of Earth, Atmosphere and Planetary Sciences, MIT

Deep Learning Emulators for Accessible Climate Projections

by

Björn Lütjens

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2023, in partial fulfillment of the
requirements for the degree of
DOCTOR OF PHILOSOPHY IN
AERONAUTICS AND ASTRONAUTICS
with a major in Machine Learning and minor in Earth System Modeling

Abstract

Climate change has shifted from a purely scientific topic to a deeply politicized issue. To combat climate change we need to create mutual understanding on the links between policies, global warming, and city-scale impacts. Climate models have been incredibly helpful in generating this causal understanding, but running them requires supercomputers and is only accessible to the minority of researchers.

This thesis explores how emulating climate models with deep learning can make them more accessible and, at the same time, raise novel challenges in deep learning on physical, long-term time-series, and high-dimensional data. This dissertation shows that deep learning can decrease runtime in dynamical models, increase accuracy in local climate projections, and generate visualizations of climate impacts. Specifically, this thesis contributes a hybrid model, called multiscale neural operator, that corrects fast low-resolution simulations by learning a hard-to-model parametrization term. This achieves to cut runtime complexity from quadratic to quasilinear which can result in a 1000x faster model on selected equations in multiscale dynamics. This thesis also contributes satellite imagery of the future that visualizes climate data using physically-consistent deep generative vision models.

The thesis contributions are framed in an envisioned online tool that rapidly emulates the city-scale impacts of various climate policies. In the future, such an emulator could accelerate local climate risk analyses, attribution of extreme events, and the understanding of causal links between between impacts and policies.

Acknowledgments

I am grateful for my thesis committee. Dava has been believing in the value of tackling climate change with machine learning during a time where it was hard to find. More than this, she has been supporting me as a person and letting my research follow my heart. Thanks to Chris for introducing me to the world of climate science and HPC, and making me believe that academia still offers room for fun. Mark has checked my nitty gritty equations and answered all sort of questions with patience, hope, and inspiration; thank you Mark. Thank you Cait for taking so much time to help shape my future and critizice every plot I produced. Finally, thanks to Youssef for the encouraging words and sharing his wisdom in the world of surrogates.

Thanks to everybody in the HSL for creating such a welcoming and supportive lab. I am grateful to my colleagues in ACL in particular Mike, Dong-Ki, Shayegan, and Jon for initial guidance and co-shaping me as a rigorous and inquisitive researcher.

Thanks to every individual collaborator who has criticically reviewed or encouraged my work in all its stages. Early on, Dava recommended I focus on working with joyful and engaging people and I am looking back so gratefully to the collaborative time and hope to continue many relationships. To point out a view, David went through the same struggle of switching from AI to climate change and has pushed our forest work from a poster chat to actual measurable impact. I have learned so much from working with Matt, Ana, Lea, Thomas, Kenza, Vivian, Salva, Ernest, Suyash, Gyri, Rupa, Thomas, Phil, Aruna, Margaux, and Simona who's work and ideas have greatly contributed to the shape of this thesis. Especially, Salva and Suyash in ENSO forecasting, Matt and Rupa in downscaling, Ana, Aruna, Phil, and Margaux in the visualizations, and Lea, Thomas, and Vivian in the climate emulation. I really appreciate all my colleagues from FDL, IBM, and BC3; Campbell and Haruko have been inspiring mentors to me beyond the internship and thanks to Natalia for relentlessly nudging me to publish the visualization work.

A large thank you goes to all the advocates in GWAE, AeroAfro, QuASAR, and GA3, highlighting Chelsea, Regina, and Cadence, for making our department a more

welcoming place and putting so much of themselves into work that has and will primarily benefit others.

I am also grateful to the teams that have kept up social life at MIT: the intramurals organizers, sailing pavillon, MITOC, members of the Muddy Charles pub, and the GA3 social chairs. Thank you for creating one MIT and freely distributing laughters, joy, adventure, and peace of mind.

MIT has been a journey. I am grateful for all my roommates Nils, Nick, Manwei, and James for the bits, distraction, and daily lunch time seminar to keep us all healthy during covid. My friends and the quaranteam Cadence, Regina, Maya, and Katie who have been there in the tumultuous times. Zivvy, Océane, Jake and Jan for the art, techno, vibrations, and night life. I am forever grateful to Sruthi for giving me the courage to switch from autonomous vehicles to climate work.

Thank you to Michelle for being invaluable to me in finding balance, jokes, connection, learning, adventure, and remembering value outside academia.

Lastly, I am grateful for my closest family, Muttern, Meike, Lars, and friends from home-home for giving me a safety net where I know I will be forever welcome.

Acknowledgment of Support

Research was partially sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

The research was partially conducted during a research internship at Future of Climate, IBM Research, New York, USA.

This research was partially conducted at the Frontier Development Lab (FDL),

USA.

The authors also gratefully acknowledge support from Earthshot Labs for their work on the reforestation visualization. Further, we greatly appreciate the time, feedback, direction, and help from Patrick Leung.

The authors gratefully acknowledge support from the MIT Portugal Program, National Aeronautics and Space Administration (NASA), and Google Cloud for the Satellite Imagery from the Future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Motivation

1.1.1 Societal motivation for climate emulators

Our spaceship Earth has warmed by 1.1°C since the industrial revolution [128]. Globally, heat waves, flooding, wildfires, biodiversity loss, and human displacement has already become increasingly intense and unjust [128]. In the US, many believe that global warming will not harm them personally [126, 197]. But for example, my home state Massachusetts has warmed faster than average (1.9°C) [267] and there is an 18% chance that a 100-year flood will hit Boston within 2030-50 causing over USD20 billion damage indirectly affecting every resident [66].

Personally, I am most concerned about the unknowns. If global temperatures rise to 2°C until 2100, virtually all tropical coral reefs will likely face long-term degradation [272] with yet unknown ripple effects into the ocean ecosystem, fishing industry, and health sector. Further, there is increasing evidence that climate tipping points are more likely than thought: deforestation could turn the Amazon rainforest into a savanna-like ecosystem and a melting West Antarctic ice sheet could add over 3 meters of sea-level rise [160, 201, 228].

Thankfully, 95% of all nations signed the 2016 Paris agreement to limit global warming to 1.5°C [306]. Reaching 1.5° will require immediate collective action [128]

yet even if all nations meet their nationally determined contributions the Earth is projected to warm by 2.6°C by 2100 [120]. To limit warming to 1.5°C and adapt societies to the impacts that are already locked-in, nearly all sectors interface with Earth system models: Industries in energy, logistics, real estate, transportation, banking, agriculture, fishing, urban planning, and more are asking for more accurate local climate risk statistics to adapt to inevitable changes [329]. To mitigate these devastating consequences, politicians are asking how policy choices will impact their neighborhoods and media communication is asking for attribution analyses that establish probabilistic links between a recent extreme event and human-induced climate change [304].

This thesis contributes research in climate emulators: a fast subcategory of climate models. We define a climate emulator as an approximation of the whole or any component within an Earth system model. The overarching research question is if there exists an approximation method that is multiple orders of magnitude faster than the original climate model but only sacrifices unnoticable accuracy. Such an emulator could increase accessibility to applications that require ensembles, such as, attribution analyses, uncertainty quantification in climate risk analyses, or exploring the impact of climate policy choices.

While more generally applicable this thesis focuses on the application of emulators for policy simulations. Globally, climate change has shifted from a purely scientific topic to a deeply politicized issue. Technological innovations exist to limit climate change to 1.5°C, but political consensus does not [128]. Policy simulations have been shown to go beyond spreading climate information, create consensus, and motivate science-based action across people with different sociopolitical values [155, 264]. In particular, En-ROADS in Fig. 1-1 has been used by over 200.000 people in 1-3 hour workshops where a group of people is tasked to define policies that limit warming to 1.5°C [262, 263]. The backbone of the En-ROADS simulation is an emulator, called C-ROADS, which models the impact of climate policy choices onto global average temperature increase [92]. C-ROADS, however, cannot project local climate impacts at higher than global resolution and accumulates inaccuracies by reducing the dynmics

Figure 1-1: **The En-ROADS policy simulation** rapidly models the impact of climate policy choices (sliders) to global temperature increase (top-right). The simulation has been used in workshops by over 200.000 people to motivate science-based climate action and create political consensus [263]. This thesis contributes building blocks to extend policy simulations towards fast and accurate modeling of *local* climate impacts.

to a globally-averaged model. This thesis disentangles En-ROADS to its modules – dynamical core, downscaling, impact modeling, and visualization – and contributes to each as detailed in Section 1.2.

1.1.2 Earth system modeling motivation

The societal pressure for improved climate risk analyses requires a breakthrough moment in the accuracy, computational complexity, and accessibility of Earth system models (ESMs) [76]. Society is asking for climate projections that accurately model risk statistics and neighborhood-scale (1km) impacts [329]. While Earth system modeling has achieved numerous successes towards this goal there exist persistent challenges in modeling extreme event statistics, abrupt transitions, compounding extreme events, unknown processes, and subgrid-scale dynamics [76].

To solve these challenges, Earth system models have become increasingly complex, modeling more phenomena at higher spatiotemporal resolutions. But as a result, Earth system models have become computationally too expensive: running a climate model at 1km resolution can take 3 weeks on a 4888 GPU-node supercomputer, just to simulate one prediction year [98]. Even at 100km the computational cost is too high for the mentioned applications that require large ensemble runs: uncertainty

quantification, attributions, or policy simulations. One might suggest, that the issue of computational demand will be fixed in time through Moore's law. But, physical limitations of silicon and increasing cost of manufacturing chips suggests that Moore's law may be coming to and end.

As a result of the computational complexity, Earth system modeling is only accessible to a selected number of scientists with access to supercomputers. Beyond the computational complexity of running ESMs, the size of resulting datasets has also significantly grown. One of the most comprehensive climate datasets is the Coupled Model Intercomparison Project Phase 6 (CMIP6) dataset, which as an ensemble of runs from ~ 100 different ESMs for varying future scenarios. The full CMIP6 dataset contains $\approx 20PB$ of data and, as a result, generating insights from this dataset requires extensive study hindering progress in climate science and policy outcomes.

This thesis asks if methods from deep learning (DL) can overcome part of the computational issues. Deep learning has achieved breakthroughs in protein folding [136], natural language modeling [46], and computer vision [76]. In weather and climate modeling, DL has been used for nowcasting, weather modeling, (sub-)seasonal forecasting, data assimilation, uncertainty quantification, downscaling, and equation discovery [260, 142]. Specifically, this thesis investigates the following research question in climate emulation:

Can we reduce inference time while maintaining accuracy by approximating components in Earth system models with deep learning?

To answer this question, this thesis uses the main workflow in Fig. 1-2: First, a large dataset that maps initial conditions, forcing, parameters, and or boundary conditions to a solution is generated with a ground-truth numerical solver on a supercomputer. Second, a deep learning method is trained to recreate selected parts of the simulated data. Third, the deep learning model can be used for fast inference at slightly comprised accuracy. Towards this goal, this thesis shows how DL can contribute to emulate dynamical processes, create visualizations of climate data, and downscale data into higher resolution predictions.

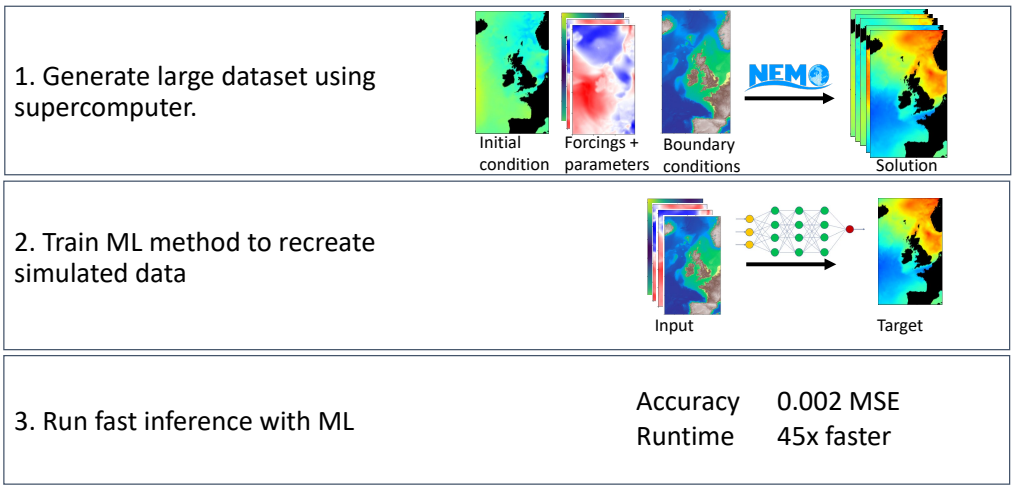


Figure 1-2: **Climate emulators** pose the research question if we can reduce inference time while maintaining accuracy by approximting components of Earth system models with deep learning (DL). The typical workflow in this thesis to answer this question is to generate a dataset, train a DL method to recreate it, and evaluate inference time and accuracy. The imagery is from a partial thesis contributions in emulating the flood model NEMO [134].

1.1.3 Deep learning motivation

Developing climate emulators benefits both Earth system modeling and deep learning research. Emulators pose novel challenges in high-dimensional data, long-term time-series models, risk sensitive modeling, and embedding physical constraints:

High-dimensional. Earth system data is high-dimensional in the spatial, temporal, and modal dimensions. For example, CMIP6 contains over 300 variables that describe the Earth state in hourly time-series from 1850-2100 on up to 2160x4320 pixel 10-100km horizontal grids with 1-100 vertical grid cells. All together, CMIP6 contains approximately 20PB of data which is 35-thousand times larger than the 570GB dataset that ChatGPT-3 was trained on [46]. Fitting deep learning models on such high-dimensional data will require intelligent subselection of learning tasks. In this thesis, I am proposing a multiscale learning scheme that is detailed in Chapter 3.

Long-term. Most deep learning-based time-series models are accurate for predictions of 10-100 time steps [107] with state-of-the-art models extending the range to 1K-16K steps [296]. Earth system models however project over 1M time steps in the case of hourly predictions ($250yr * 365d/yr * 24hr/d$). It is still unclear if an autoregressive, hierarchical, or time-average approach will be most successful to project

climate risk statistics with deep learning at such long time horizons.

Risk sensitive. Most applications in deep learning are fault-tolerant, for example when monitoring traffic flows or animal populations. Climate projections, however, are used in multi-billion dollar decisions that affect human health, such as, flood instrastructure development. Hence, deep learning methods in climate emulation ask for deep learning innovations in physical interpretability, robustness, uncertainty quantification, and robust domain-informed evaluation protocols. This thesis develops a new protocol to evaluate physical consistency in synthesized satellite imagery in Chapter 4.

Physical constraints. Lastly, these challenges would likely be insurmountable if scientists would not have discovered equations, correlations, and causal links that robustly describe parts of the Earth system. Incorporating this knowledge into deep learning methods poses new challenges in inductive biases, in- and equivariant architecture, causal modeling, and hybrid physics-deep learning architectures [255, 138]. This thesis proposes a new hybrid architecture that integrates deep learning into existing numerical solvers of multiscale chaotic dynamics in Chapter 3.

1.2 Thesis Contributions and Overview of a Climate Emulator

The contributions of this thesis are summarized under the umbrella of *the climate pocket*, an envisioned policy simulation that is displayed in Fig. 1-3. This thesis contributes to the components of such a climate emulator that would map climate policies to local impacts. Figure 1-4 gives a broad overview of the components: The user's climate policy choice is mapped by an energy policy model to greenhouse gas emissions over time. The emission data forces an emulated Earth system model to forecast low-resolution (100km) climate variables, such as temperatures, humidities, and pressure over time. Next, the low-resolution projections are downscaled onto higher-resolution grids (1km) and fed into a chain of impact models to predict, e.g., probabilities of

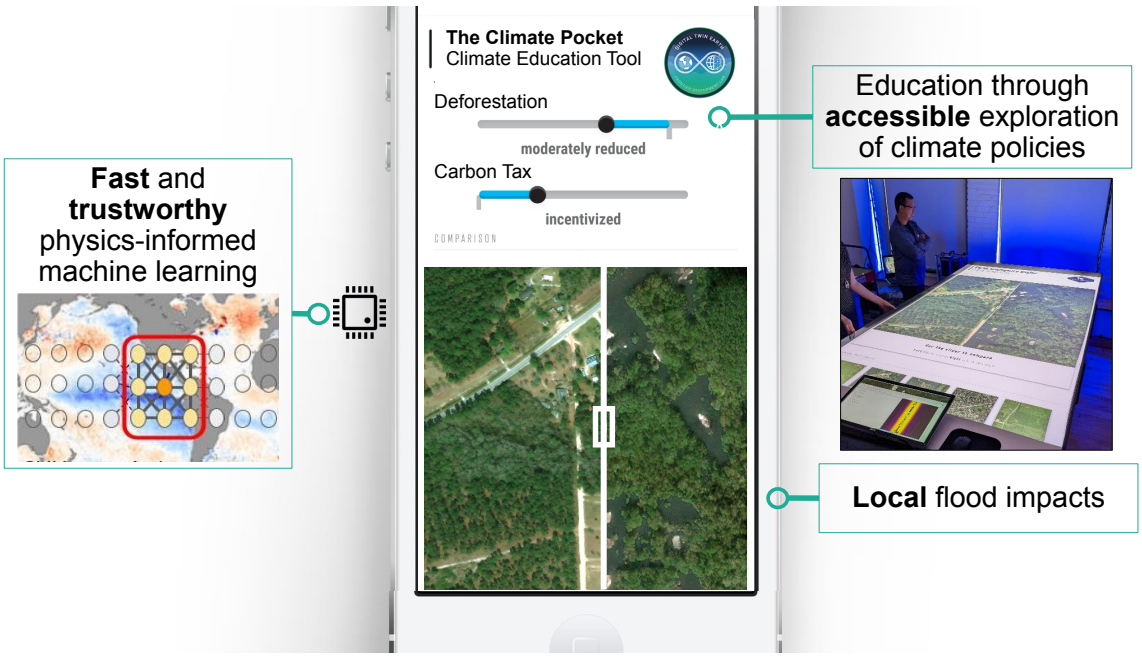


Figure 1-3: **This thesis envisions 'the climate pocket'.** The climate pocket would be an online policy simulation that visualizes local sub 1km-scale impacts (flood image) of chosen climate policies (sliders). A backbone hybrid ML-physics model (left) would quickly emulate the underlying climate dynamics. Collaborations with social scientists would bring the forecasts to policy makers as an engaging table top exercise (right). This thesis contributes to the components of such a high-resolution climate emulator.

storm surge heights. Lastly, the climate impact projections are presented in a tangible and engaging visualization, such as, the displayed satellite imagery of future flooding. The thesis contributes to the four areas: dynamical modeling, downscaling, impact modeling, and visualization.

The three main thesis contributions are summarized as:

- A novel surrogate model that learns subgrid parametrizations with neural operators, titled multiscale neural operator [192], and
- A novel deep learning pipeline to generate physically-consistent visualizations of climate data with deep generative vision models, titled satellite imagery of the future [188], and
- Contributions towards improving El Niño forecasts with graph neural networks [50, 266], speeding up flood models with neural operators [134], and downscaling wind data with deep generative vision models [154].

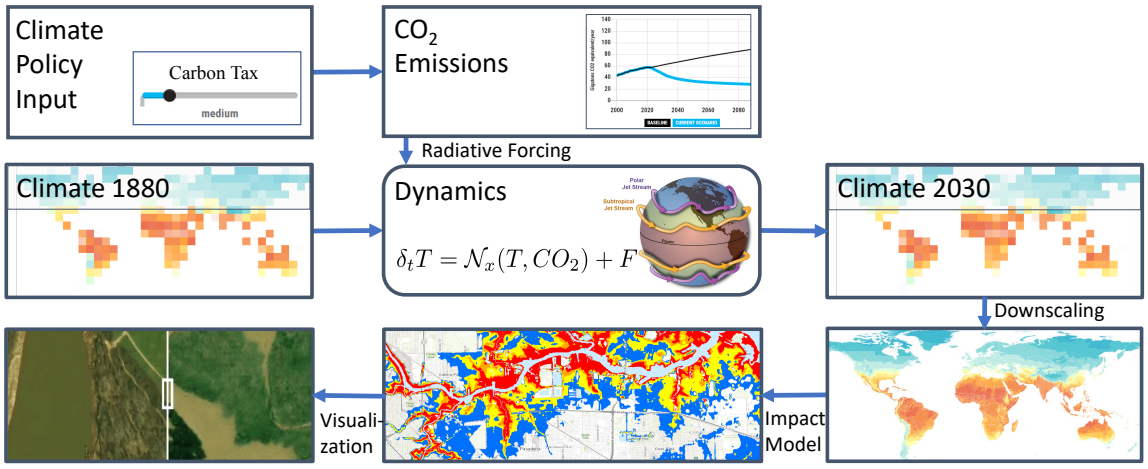


Figure 1-4: **The system architecture** of a climate emulator contains a dynamics, downscaling, impact modeling, and visualization component. The thesis contributes within these four areas.

1.2.1 Dynamics

Within the dynamics module lies the first main thesis contribution. I contribute a novel hybrid physics-deep learning architecture that is inspired by climate model parametrizations. Parametrizations are terms in climate models that capture the influence of subgrid-scale processes, such as cloud formation, onto large-scale patterns and are one of the largest sources of uncertainty in current climate models [274]. Specifically, I propose *Multiscale Neural Operator: Creating Fast PDE Surrogates by Learning Resolution-variable Subgrid Parametrizations* [192, 189]. Multiscale Neural Operator (MNO) is detailed in Chapter 3 and contributes:

- the first parametrization with neural operators,
- a learning-based multiscale PDE surrogate that
  - has quasilinear runtime complexity,
  - leverages known large-scale physics,
  - is resolution-variable,
  - and does not require autodifferentiable solvers, and

- a demonstration on the chaotic, multiscale equations multiscale Lorenz96 and preliminary results on quasigeostrophic turbulence.

Towards the goal of deep learning-based emulators of dynamical systems, I also contributed to *Improving El Niño Forecasts with Graph Neural Networks* [50], which is summarized in Chapter 5, and a work in progress on *Harnessing data-driven ocean emulators for fast adjoints* [28].

1.2.2 Downscaling

Microclimates are prominently found in urban areas, mountains, or forests. Downscaling methods project the $\sim 10 - 100km$ output of global climate models onto microclimates. Deep learning-based downscaling methods can increase the accuracy of current statistical downscaling or decrease the runtime of dynamical downscaling approaches as detailed in Chapter 5.

Towards deep-learning-based downscaling methods I have contributed to:

- *WiSoSuper: Benchmarking Super-Resolution Methods on Wind and Solar Data*, described in [154] and Chapter 5,
- work in progress on *Diffusion-based models for Probabilistic Downscaling of Wind Data* described in Chapter 6, and
- work in progress on *DailyMelt: Diffusion-based Models for Spatiotemporal Downscaling of (Ant-)arctic Surface Meltwater Maps* in Chapter 6.

1.2.3 Impact modeling

Impact models predict stakeholder-relevant climate impact statistics from downscaled climate projections. In the case of coastal floods, a chain of hurricane track, flood, and storm surge model predict storm surge hazard maps [131]. Deep learning-based emulators have the potential to decrease runtime of these numerical models.

As part of this thesis, I contributed to *Digital Twin Earth – Coasts: Developing a fast and physics-informed surrogate model for coastal floods via neural operators*

in [134] and Chapter 5. Here, we leveraged neural operators to emulate the NEMO flood model and decreased runtime by a factor of 45x.

1.2.4 Visualization

Lastly, climate impact statistics are communicated to stakeholders in industry, policy, or the general public. Various communication media and audiences exist. The visualization of flood risks as storm surge hazard maps, for example, is used in community discussion groups to debate flood instratructure development.

The second main contribution of this thesis is a novel visualization called *Satellite Imagery from the Future: Creating Physically-Consistent Visualizations of Climate Data with Deep Generative Vision Models* [188, 182]. This work leverages deep generative vision models to synthesize satellite imagery. To do so, I create a novel mode to visualize climate data and a framework to evaluate physical consistency. Speifically, the work contributes:

- a novel framework to measure physical consistency in synthetic satellite imagery,
- the first physically-consistent and photorealistic visualization of flood risks as satellite imagery, and
- the first visualization of planned reforestation projects as satellite imagery.

Chapter 2

Background and Related Works

2.1 Terminology

This section introduces the common terminology and essential background in function approximation with deep learning. For the most part this thesis assumes familiarity with deep learning and references additional resources for further background. I recommed [29] for concepts in machine learning and pattern recognition, [107] for deep learning, and [283] for surrogate modeling.

2.1.1 Emulation and function approximation

All contributions in this thesis have been formulized as supervised learning problems [158]. In a supervised learning problem the goal is to find a function, f_θ, with weights, θ, that map an input, x, to a prediction, $\hat{y}$, such that, $f_\theta(x) = \hat{y}$ [158]. The function and weights are found by minimizing the expected loss over the training dataset:

$$\mathbb{E}_{x,y\sim X,Y}\mathcal{L}(f_\theta(x), y) \tag{2.1}$$

where x, y are randomly sampled pairs from the dataset, $D = (X, Y)$, and the loss function, $\mathcal{L}$, is computed between the predictions, $\hat{y} = f_\theta(x)$, and targets, y.

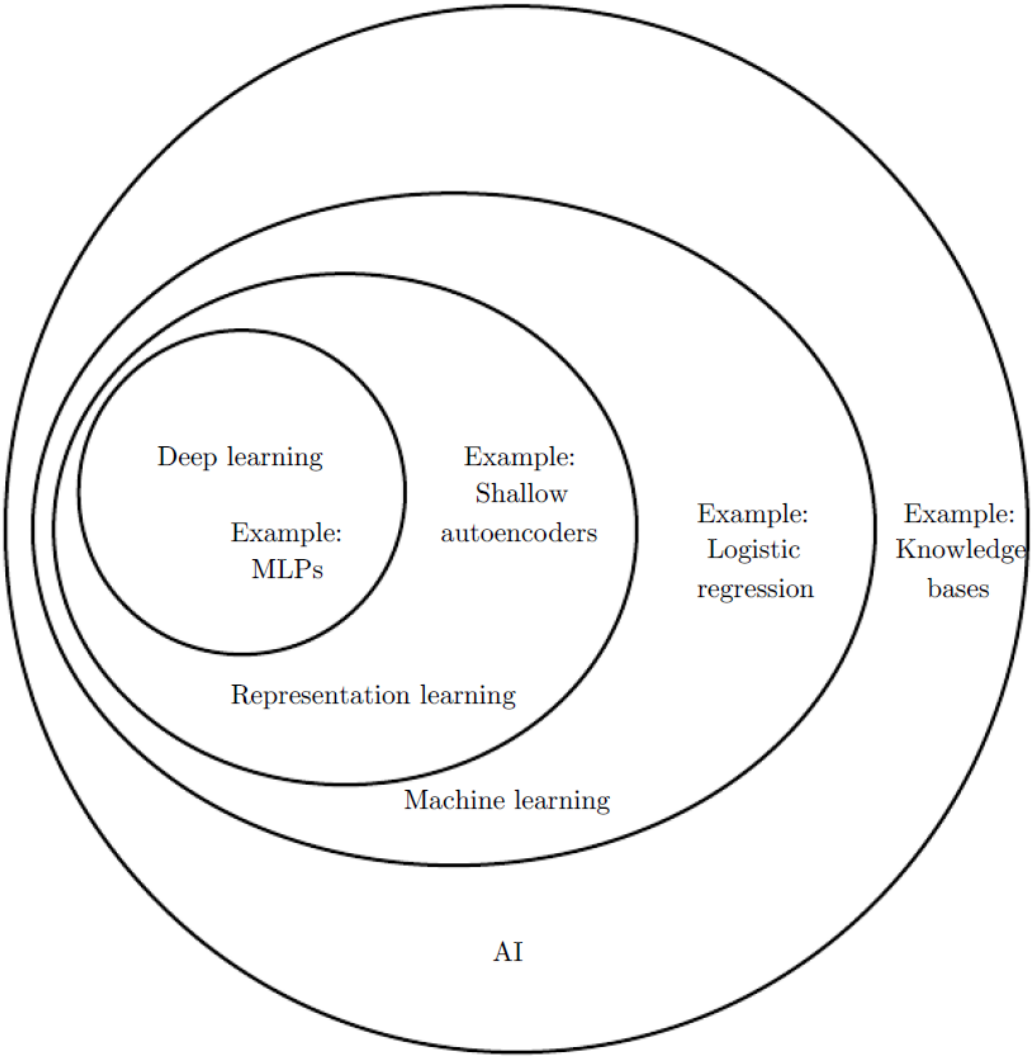


Figure 2-1: **Deep learning** is a subset of machine learning that refers to the study of computational models that have multiple processing layers that learn data representations at multiple levels of abstraction [158]. Image taken from [107].

2.1.2 Deep learning

As visualized in Fig. 2-1, deep learning is a subset of machine learning that refers to the study of computational models that have multiple processing layers that learn data representations at multiple levels of abstraction [158]. Deep neural networks (DNNs) are one of the most commonly used computational model in the field of deep learning [106].

This thesis uses DNNs as nonlinear function approximators that can learn more complex functions than other machine learning models such as nonlinear regression,

support vector machines, Gaussian processes, or random forests [29]. Indeed, neural networks with infinite units, $n_k \to \infty$, are universal function approximators [125]. Further, DNNs are easy to implement through open-source libraries [233] and can be trained via efficient optimization algorithms to approximate high-dimensional functions [323].

A basic building block in DNNs is called multi-layer perceptron, artificial neural network, or fully-connected neural network (FCNN). FCNNs, f_θ, map an input vector with n_0 elements, $x \in X \subset \mathbb{R}^{n_0}$, that is also called feature vector or predictor, to an output vector with n_K elements, $\hat{y} \in \mathbb{R}^{n_K}$, that is also called prediction. The FCNN is characterized by the concatenation of nonlinear functions, z_k, as follows:

$$\begin{aligned} f_\theta(x) &= z_K(x), \\ z_{k+1}(x) &= \sigma(W_k z_k(x) + b_k), \\ z_0(x) &= x, \end{aligned} \tag{2.2}$$

where $k \in \{0, ..., n_K\}$ indexes the layers, $W_k \in \mathbb{R}^{n_k \times n_{k-1}}$ is the learned matrix of weights or parameters, and $b_k \in \mathbb{R}^{n_k}$ is the learned bias vector. Further, $z_k(x) \in \mathbb{R}^{n_k}$ is the hidden state in layer k that is also called latent state, learned representation, or learned features. The FCNN architecture is described by the hyperparameters $n_k \in \mathbb{N}$ which is the number of units in each layer, and $K \in \mathbb{N}$ which is the number of layers, σ which is the nonlinear activation function. Hyperparameters are parameters that are not optimized via backpropagation of the loss function and often found via domain expertise and/or random grid search. A common choice of activation function is the rectified linear unit (relu), $\sigma : \mathbb{R} \to \mathbb{R}^+$ with $\sigma(x) = \max(0, x)$.

The FCNN is optimized to approximate the target, $y \in Y \subset \mathbb{R}^{n_K}$, which is also called predictand or ground-truth. Training or fitting the FCNN refers to finding the optimal weights, $\theta^* = \{W_k, b_k\}_{\forall k \in \{0,...,n_K\}}$, that minimize the loss function in. A

common choice for the loss function, $\mathcal{L}$, is the mean squared error (MSE),

$$\mathcal{L}(f_\theta(x), y) = \frac{1}{n_K} \sum_{i=1 \in \{1,\ldots,n_K)\}} (y_i - f_\theta(x)_i)^2. \tag{2.3}$$

The weights are found via an optimization algorithm, such as stochastic gradient descent or ADAM [146]. An essential step during optimization is to calculate the gradient of the loss function with respect to the weights in each layer, $\frac{\partial L(f_\theta(x),y)}{\partial W_k}$, which is usually done via application of the chain rule and automatic differentation [18] in process called backpropagation [323].

For a high number of layers, e.g., $k > 3$, one typically refers to the neural networks as deep neural networks. Over the years, neural networks have been extended to various model architectures that embed biases in spatial, temporal, latent, or graph structure via convolutional neural networks (CNNs), recurrent neural networks (RNNs), auto encoder-decoders (AEs), or graph neural networks (GNNs), respectively. Other model architectures focus on modifications of the loss function, such as generative adversarial networks (GANs), normalizing flows (flows) or diffusion-based models.

In climate data, many spatial patterns are known and Section 2.3.2 reviews how to incorporate this knowledge into deep learning architectures. A thorough review of deep neural networks and their variations, properties, optimization, and applications until 2016 can be found in [106].

2.2 Overview of Relevant Research Fields and Broad Research Gaps

2.2.1 Emulation, surrogate, and reduced-order modeling

This thesis uses deep neural networks for surrogate modeling. Surrogate models can be broadly categorized into interpolation-based, projection-based, or hierarchical

models [283]. In interpolation-based of data-fit models, the original model is usually approximated by fitting a regression model, such as linear regression, polynomial approximation, Gaussian processes, random forests, or deep neural networks (DNNs), to the input-output relationship [208]. Purely data-driven models are non-intrusive, i.e., can fit black-box models, and hence require little domain expertise, but often do not respect physical constraints and can fail to generalize beyond the training data [252, 40, 321, 4]. This thesis considers purely data-driven interpolation-based surrogates as too computationally expensive to emulate high-resolution (1km) climate models and thus uses a hierarchical approach.

Most works in climate modeling rely on hierarchical models [274]. These models use subgrid parametrizations in coarse-grid simulations to capture small-scale processes, such as cloud formation [274]. As subgrid parametrizations are traditionally based on heuristics or physical intuition, novel works propose machine learning models to fit the subgrid parametrizations of coarse-grid to high-resolution simulations [252, 337, 244]. The DNN-based parametrizations show great promise, reducing the computational cost of numerical simulations one to two orders of magnitude in comparison to fully-resolved simulations [338, 104], but often suffer from stability [252, 64], memory-complexity [8], computational cost of the training data [152], locality [312], generalization [40], and lack quantified uncertainties. This thesis contributes a new hierarchical surrogate model that uses neural operators to target the issues of training data and locality in Chapter 3.

Projection-based or reduced order models (ROMs), such as proper orthogonal decomposition, empirical orthogonal functions, or polynomial chaos expansion (PCE), project the model evaluations into a lower-order subspace [283, 289, 241]. Projection-based models can enforce physical-constraints, but often involve expensive computations or domain expertise [121]. Recent works explore DNNs to replace the computationally expensive sections, but have mostly been demonstrated on simple, elliptic or parabolic, non-coupled differential equations [340]. I explored applications of deep learning to polynomial chaos expansion in [187], but this thesis focuses instead on leveraginmg ideas from projection-based methods to representing subgrid

parametrizations in Chapter 3.

2.2.2 Links to climate communication

Climate communication is faced with persistent challenges to communicate in deeply polarized and politicized environments, manage the growing degrees of overwhelm and hopelessness, and transitioning from awareness and concern into action [207]. Visualizations have become a dominent mode of climate communication. But, many climate visualizations feature the typical iconography of polar bears, melting glaciers, extreme weather, globes, and politications [224]. Many of these visualizations can create psychological distance, e.g., via portraying distant polar bears, cathartic climate catastrophes, alienating politicians or protesters, or strongly politized art [313].

Instead, research suggests that modern visualizations of climate change need to: 1) rephrase climate change from an issue of abstract politics and science, to a people-centered issue, 2) illustrate the link between causes and impacts, 3) and emphasize solutions [313]. Polls across the US support the misunderstanding in causal links: 72% of Americans believe that global warming is happening but only 57% believe that is is caused mostly by human activities [197]. Further, solutions need to emphasized as only 35% of Americans take action by discussing global warming at least occassionally [126]

Policy simulations such En-ROADS are one possibility to increase causal understanding and action [264]. The causal understanding is created by letting users change climate policies and inspect the impacts in real-time. However, existing policy simulation only simulate global impacts which are more abstract and personally-irrelevant than locally tailored climate impact projections. This thesis lays the foundations for policy simulations with modeling backends that can simulate at a finer than global resolution and can visualize local climate impacts, specifically, increased flood risks.

Another important element in communicating causal relationships are climate attribution analyses [296]. Especially, extreme events raise the public question if the event was more likely due to climate change. Attribution analyses require multiple climate or weather model runs to accurately calculate extreme event occurence proba-

bilities with and without climate change. Thus deep learning-based climate emulators could contribute towards reducing the computational cost and make attribution analyes more accessible.

2.3 Expressing Domain Knowledge from Climate Science in Deep Learning

2.3.1 Physics-informed deep learning

Deep learning emulators can be significantly faster than numerical solvers during inference time. For example, [236] proposes a deep learning emulator of numerical weather predictions that is multiple orders of magnitude faster than a US operational weather model. It is less clear how to apply emulators in climate modeling (not weather). Deep learning-based climate emulators are challenged by high-dimensional datasets, long-term time series, and risk sensitive applications that require trust. To complicate things further, the large quantities of data are often correlated and i.i.d. data is scarce. As a result, deep learning emulators can be fast, but not trusted for extrapolation tasks, hard to interpret, and too data-hungry [255, 129]. Out of this forms a wide-spread belief that successful applications of deep learning in Earth system modeling will be hybrid physical and deep learning methods [255, 141, 129, 142]. We call the broad area of hybrid modeling approaches physics-informed machine learning (ML).

The motivation for physics-informed machine learning is to combine the trust and interpretability of physics with the runtime complexity and flexibility of machine learing methods. Yet, it is largely unknown what the best way to combine physical knowledge with deep learning methods is [138]. To answer this, many recent works have integrated physical laws in the input features, model architecture, loss function, postprocessing, or evaluation. A full review has been gratefully compiled for physics in general in [138] and weather in [142, 300]. Here, I extend the reviews to discuss how to incorporate domain knowledge from the climate sciences in spatial inductive

biases and soft or hard constraints.

2.3.2 Inductive bias, incorporating known spatial correlations, and how to choose a deep learning architecture

One way to incorporate domain knowledge into deep learning models are inductive biases [15]. An inductive bias is the set of assumptions that a machine learning algorithm makes on the observed data [202]. In a Bayesian model, inductive biases are typically incorporated via the choice of prior distribution [13]. Incorporating inductive biases will reduce flexibility, but also reduce the amount of required training data [103]. In deep learning, inductive biases can be expressed by the choice of input feature selection, feature transformations, data augmentation, weight initialization, activation functions, regularization, normalization layers, model architecture, optimization algorithm and others. Much about inductive biases in deep learning is still unknown. Here, I focus on a very relevant question for climate emulators: 'How can we incorporate known spatial patterns, modes, and correlations of Earth data into the model architecture choice?'

To answer this question this subsection and Table 2.1 reviews the most common model architectures in deterministic supervised learning under the lens of climate emulation: fully-connected neural networks (FCNNs) [107], convolutional neural networks (CNNs) [107], graph neural networks (GNNs) [115], neural operators (NOs) [166], and transformers [83]. This section assumes basic familiarity with each architecture and otherwise recommends reading the linked reference.

Fully-connected neural networks are often chosen if there are only a few in- and output features, most commonly in PINNs to map a scalar location, x, and time, t, to a scalar solution, u, of a PDE, such that, $f_{\text{fcnn}} : (x, t) \rightarrow u$ [249]. By default, FCNNs have no spatial inductive bias across the feature dimension as all units in a layer are connected to all units in the next layer [115]. However, with FCNNs formulated as, $u = f(x, t)$, they will have a *spectral bias towards learning functions with low-frequency features* [247, 13, 53] (more specifically towards large

Table 2.1: **Spatial inductive biases**. Domain knowledge from the Earth sciences can be expressed in the choice of deep learning architecture. For example, CNNs can be chosen to learn local patterns, GNNs for non-local teleconnections, and neural operators for known modes.

Architecture	Relations	Spatial Inductive Bias	In- or Equivariance	Thesis Application in Climate
Fully-connected	all-to-all	low-frequency	-	[252]
Convolutional	local convolutions	locality and shape or texture	translation	Chapter 4
Graph	message passing	arbitrary relations	permutation	Chapter 5
Neural Operator	spectral transforms	spectral	translation bias	Chapter 3
Transformer	attention	none	positional (opt.)	Chapter 6

eigenvectors of the neural tangent kernel [315]) and, with ReLu activations, to learn piecewise linear functions. Some options exist to remove the spectral bias via Fourier feature transforms [295]. In climate emulation, FCNNs have been chosen to learn local parametrizations in climate models [252] or downscale climate projections [269] and it is unclear if the spectral bias contributed to the accuracy or not. Beyond inductive bias FCNNs are computationally too expensive for high-dimensional in- and outputs, because the number of weights scales quadratically with the dimensionality of the (flattened) input. As a result, FCNNs are scarcely used in image, video, or other high-dimensional data representations.

Convolutional neural networks are currently the go-to choice for deep learning on images partially due to them scaling linearly with the dimensionality of the (flattened) input [107]. The learned convolutions are shared across the image. This gives CNNs a *locality bias* towards learning local feature correlations and *translation equivariance*, i.e., the assumption that translating the input will translate the output by an equal amount [107]. CNNs are also *biased towards either shape or texture* depending on the dataset [164] and can be extended to other in- or equivariances [45]. In climate emulation, CNNs have been chosen for nowcasting [310], seasonal forecasting [113], weather prediction [89], climate modeling [319], and remote sensing [212]. I have

chosen CNNs in the Earth Intelligence Engine because of the problem's local nature as detailed in Chapter 4. The locality bias and translation equivariance, however, are not always desired in climate applications. Many climate phenoma are non-local including, for example, large-scale wave patterns or teleconnections which are links between weather phenomena at widely separated locations. Further, most climate phenomena are not translationally equivariant as they are affected by location, for example, cyclones have different dynamics in higher than lower latitudes. There exist ways to reduce the locality bias in CNNs such as choosing dilated convolutions [180], but in this thesis we have chosen other architectures: GNNs to capture teleconnections in seasonal forecasting in Section 5.1 and neural operators for learning non-local parametrizations in Chapter 3.

Graph neural networks represent data as nodes and edges and learn graph convolutions [115]. They have an *arbitrary relation bias*, i.e., an inductive bias towards learning relationships between an arbitrary selection of node pairs [115]. Further, GNNs have an *permutation invariance bias*, i.e., the prediction is biased maintain constant under permutations of input nodes [115]. In climate emulation, GNNs are a powerful method to express knowledge of non-local teleconnections, point observations of ocean buoys and weather stations, climate networks, and other graph patterns. Part of this thesis work was the first study of graph neural networks in (sub-)seasonal forecasting detailed in Chapter 5. Our work was followed by impactful studies of GNNs in weather forecasting [143] and fluid dynamics [38]. It is unclear for which climate phenomena the arbitrary relation and permutation invariance bias are suitable and details are discussed in Chapter 5.

Neural operators apply layers with spectral transformations, such as Fourier transforms [166], PCA [26], wavelet transforms [111], spherical harmonics, or others. Inductive biases in neural operators have not yet been studied to a large extent [300]. But, the spectral transformation can be interpreted as feature selection and will introduce inductive biases according to the selection of transform. A Fourier transform, for example, likely expresses a periodicity bias and a low-frequency bias, if high-frequencies are cut-off. PCA with eigendecompositions will likely introduce a bias

towards learning modes that express the high variances first. In climate emulation, I hypothesize that neural operators will be a very impactful tool due to the many studies suggesting the existence of spectral modes, such as ENSO, MJO, or other in climate data. I have use Fourier neural operators to learn parametrizations in chaotic multiscale dynamics, see Chapter 3.

Transformers are designed to express no spatial inductive biases [82]. Unless location encodings are used transformers are also positionally invariant, i.e., the position of patches within an image can be permuted without changing the output [82]. Thus, transformers are known to require the largest amount of data within the mentioned model architectures. Transformers have achieved unparalled performance in natural language modeling, e.g., ChatGPT [46], likely due to the sparsity of inductive biases and abundance of data. For climate emulation, however, much domain knowledge exists and independent and identically distributed (i.i.d.) data is more limited. As a result, transformers coudl be too data inefficient and architectures that express climate knowledge via inductive biases could be more successful. The full utility of transformers in Earth system modeling is yet undiscovered and as a part of this thesis I have been helping to explore the combination of transformers and neural operators for ocean data assimilation [28].

Inductive biases are also incorporated in the selection of temporal (e.g., RNNs, LSTMs, GRUs, or neural ODEs), probabilistic (e.g., Bayesian neural networks), unsupervised (PCA, VAEs, etc.), semi-supervised learning (metric and constrastive learning) or other architectures which goes beyond the scope of this thesis and I refer to [107] for an introduction.

2.3.3 Incorporating known equations via hard or soft constraints

Known ordinary (ODEs) or partial differential equations (PDEs) can be incorporated in deep learning methods via soft and hard constraints. A popular approach to embed equations as soft constraints is via the loss function [74]. The term physics-informed

neural networks (PINNs) has been coined to refer to neural networks that predict a ODE or PDE solution, $\hat{u}$, and then calculate a residual by plugging the solution into the known equation [249]. The distance of the residual to zero will be used as loss function, $L = f(\hat{u})$. I have adopted this approach for exploratory work in uncertainty quantification in [187]. Outside of this thesis, PINNs have been extended by over 2000 papers as extensively reviewed in [74], and theoretical proofs for consistency, stability or convergence [278] exist for select equations. In theory, infinite-width fully connected neural networks can approximate any continuous function [75]. In practice, however, there is a non-zero global error between the approximated and ground-truth solution due to the sum of errors that result from the optimization process not converging to the global minimum, expressivity of the chosen deep learning architecture, and domain shift between train and test data [74]. As a result, incorporating equations via soft constraints with PINNs is generally not guaranteed to conserve energy or converge.

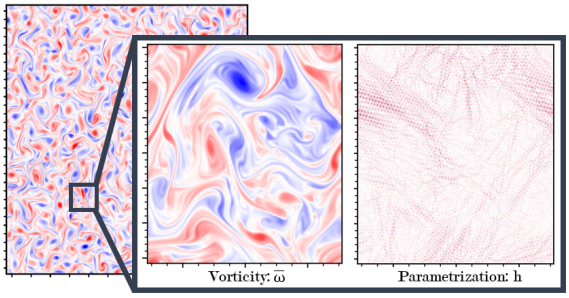
The alternative is to incorporate known physical equations as hard constraints. Two methods for incorporating hard constraints is via the choice of neural network output layer or postprocessing. In the output layer, choosing a sigmoid activation function will bound the output to $[0, 1]$ which can be scaled to match the desired bounds of the target variable [118]. In postprocessing, neural networks can predict intermediate physical variables from which other consistent variables can be computed via known equations [338, 22]. This thesis uses the idea of hard constraints in multiscale neural operator by predicting an intermediate variable, the streamfunction, and then calculating velocities and vorticity from the streamfunction. More intricate ways of incorporating hard constraints, e.g., in intermediate layers [81] are still an open research question.

Chapter 3

Multiscale Neural Operator: Creating Fast PDE Surrogates by Learning Resolution-variable Subgrid Parametrizations

Chapter abstract: Numerical simulations in climate, chemistry, or astrophysics are computationally too expensive for uncertainty quantification or parameter-exploration at high-resolution. Reduced-order or surrogate models are multiple orders of magnitude faster than high-resolution simulations, but traditional surrogates are inflexible or inaccurate and pure machine learning (ML)-based surrogates often uninterpretable. We propose a hybrid, flexible surrogate model that leverages known physics for simulating large-scale dynamics and limits learning to the hard-to-model term. Here, we learn the parametrization or closure which captures the effect of fine- onto large-scale dynamics. This work is the first to learn parametrizations with neural operators. Using neural operators creates the first resolution-variable, non-local, and flexible parametrizations. Our *multiscale neural operator* (MNO) is motivated by a rich literature in multiscale modeling and has quasilinear instead of quadratic runtime complexity. MNO is more accurate than a pure machine learning model and base-

(a) Multiscale Neural Operator



(b) Quasi-Geostrophic Turbulence

Figure 3-1: **Left: Multiscale neural operator (MNO).** Explicitly modeling all scales of Earth's weather is too expensive for traditional and learning-based solvers [227]. MNO dramatically reduces the computational cost by modeling the large-scale explicitly and learning only the effect of fine- onto large-scale dynamics; such as turbulence slowing down a river stream. We embed a resolution-variable neural operator in the large-scale physical simulations as a "parametrization", conceptually similar to stacking dolls [284]. **Right: Quasi-Geostrophic (QG) turbulence.** We choose QG turbulence to illustrate MNO on high-dimensional and turbulent climate data. MNO predicts the coarse-grained parametrization (right) from a low-resolution vorticity input (mid). The non-local and periodic nature of the parametrization motivates using neural operators.

line parametrizations as demonstrated on two chaotic equations from atmospheric modeling: multiscale Lorenz96 and quasi-geostrophic turbulence.

3.1 Introduction and Contributions

Climate change increases the likelihood of storms, floods, wildfires, heat waves, biodiversity loss and air pollution [127]. Decision-makers rely on climate models to understand and plan for changes in climate, but current climate models are computationally too expensive: as a result, they are hard to access, cannot predict local changes ($< 10km$), fail to resolve local extremes (e.g., rainfall), and do not reliably quantify uncertainties [227]. For example, running a global climate model at $1km$ resolution can take ten days on a 4888×GPU node supercomputer, consuming the same electricity as a coal power plants generates in one hour [98]. Similarly, in molecular dynamics [16], chemistry [19], biology [335], energy [342], astrophysics or fluids [85], scientific progress is hindered by the computational cost of solving partial differential equations (PDEs) at high-resolution [138]. We are proposing a surrogate in Fig. 3-1

that runs at low-resolution and then learns a corrective error term.

Surrogate or emulator models are fast, reduced-order approximations of numerical simulations [243] and of significant interest in physics-informed machine learning [142, 255, 141, 101]. Machine learning (ML)-based surrogates have simulated PDEs up to $1-3$ orders of magnitude faster than traditional numerical solvers and are more flexible and accurate than traditional surrogate models [138]. However, pure ML-based surrogates are too data-hungry [253]; so, hybrid ML-physics models are created, for example, via incorporating known symmetries [45, 16] or equations [325]. Most hybrid models represent the solution at the highest possible resolution which becomes computationally infeasible in multiscale or very high-resolution physics; even at optimal runtime [237, 238].

As depicted in Figs. 3-1 and 3-2, we simulate multiscale physics by running easy-to-acces large-scale models and focusing learning on the challenging task: *How can we model the influence of fine- onto large-scale dynamics*, i.e., what is the subgrid parametrization term? The lack of accuracy in current subgrid parametrizations, also called closure or residual terms, is one of the major sources of uncertainty in multiscale systems, such as turbulence or climate [227, 104]. Learning subgrid parametrizations can be combined with incorporating equations as soft [249] or hard [23] constraints. Various works learn subgrid parametrizations, but are either inaccurate, hard to share or inflexible because they are local [104], resolution-dependent [157], or domain-specific [20], respectively as detailed in Section 3.2. Here, we formulate the parametrization problem as learning non-local, resolution-variable neural operators [5].

We propose, *multiscale neural operator* (MNO), a novel learning-based PDE surrogate for multiscale physics with the key contributions:

- The first surrogate to approximate parametrizations via neural operators.
- A learning-based multiscale PDE surrogate that has quasilinear runtime complexity, leverages known large-scale physics, is resolution-variable, and does not require autodifferentiable solvers.
- Demonstration of the surrogate on the chaotic, coupled, multiscale PDEs: mul-

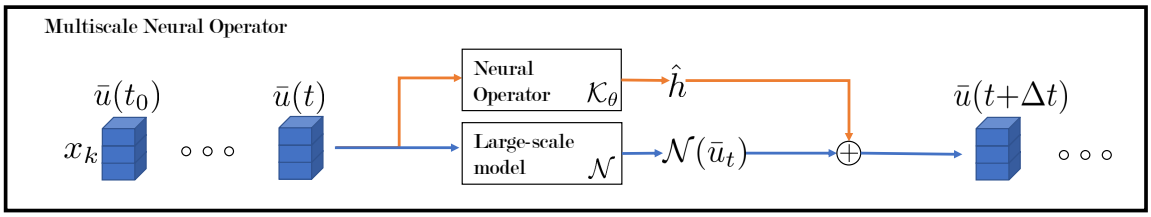
tiscale Lorenz96 and quasi-geostrophic turbulence.

3.2 Related Works

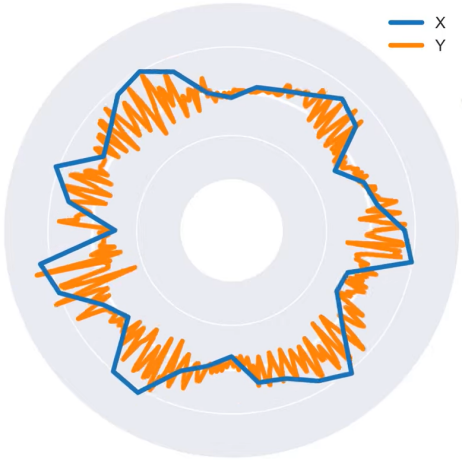
We embed our work in the broader field of physics-informed machine learning and surrogate modeling. We propose the first surrogate that corrects a coarse-grained simulation via learned, resolution-variable, non-local parameterizations.

Direct numerical simulation. Despite significant progress in simulating physics numerically it remains prohibitively expensive to repeatedly solve high-dimensional partial differential equations (PDEs) [138]. For example, finite difference, element, volume, and (pseudo-) spectral methods have to be re-run for every choice of initial or boundary condition, grid resolution, or parameters [91, 36]. The issue arises if the chosen method does not have optimal runtime, i.e., does not scale linearly with the number of grid points, which renders it infeasibly expensive for calculating ensembles [36]. Select methods have optimal or close-to-optimal runtime, e.g., quasi-linear $O(N \log N)$, and outperform machine learning-based methods in runtime and accuracy, but their implementation often requires significant problem-specific adaptations; for example multigrid [43] or spectral methods [36]. We define a 'flexible' method as a method that requires few or no domain-specific adaptations. We acknowledge the existence of impressive resarch directions towards optimal and flexible non-ML solvers, such as the spectral solver called Dedalus [48], but advocate to simultaneously explore easy-to-adapt ML methods to create fast, accurate, and flexible surrogate models.

Surrogate modeling. Surrogate models are approximations, lightweight copies, or reduced-order models of PDE solutions, often fit to data, and used for parameter exploration [263] or uncertainty quantification [283, 243, 187]. Surrogate models via SVD/POD [59], Eigendecompositions/KLE [100], Koopman operators/DMD [327], take simplying assumptions to the dynamics, e.g., linearizing the equations, which can break down in high-dimensional or nonlinear regimes [243]. Instead, our work leverages the expressiveness of neural operators as universal approximations [62] to



(a) Model Architecture



(b) Multiscale Lorenz96

Figure 3-2: **Left: Model Architecture.** A physics-based model, $\mathcal{N}$, can quickly propagate the state, $\bar{u}_t$, at a large-scale, but will accumulate the error, $h = \overline{\mathcal{N}(u)} - \mathcal{N}\bar{u}$. A neural operator, $\mathcal{K}_\theta$, wraps the computational and implementation complexities of unmodeled fine-scale dynamics into a non-local and resolution-variable term, $\hat{h}$, that iteratively corrects the large-scale model. **Right: Multiscale Lorenz96.** We demonstrate multiscale neural operator (MNO) on the multiscale Lorenz96 equation, a model for chaotic atmospheric dynamics. Image: [251]

learn fast high-dimensional surrogates that are accurate in nonlinear regimes [186, 338, 72, 220]. **Pure ML-based** surrogate models have shown impressive sucess in approximating dynamical systems from ground-truth simulation data – for example with neural ODEs [244, 64, 119], GNNs [37, 50], CNNs [287], neural operators [166, 5, 236, 177, 134], RNNs [137, 253], GPs [57], reservoir computing [234, 220], or transformers [60] – but, without incorporating physical knowlege become data-hungry or poor at generalization (see [138, 24] and Fig. 3-7).

Physics-informed machine learning. Two main approaches of incorporating physical knowledge into ML systems is via known symmetries [45] or equations [138]. Our approach leverages known equations for computing a coarse-grid prior; which is complementary to using known equations as soft [249, 159, 339, 331, 341, 335]

or hard constraints [108, 190, 23, 81, 22, 135] as these methods can still be used to constrain the learned parametrization. In terms of symmetry, our approach has a translational equivariance bias via Fourier transformations [166], but can be extended to other frameworks that embed in- or equivariance of PDEs [223] to rotational [97, 298], Galilean [330, 239], scale [24], translational [293], reflectional [70] or permutational [344] transformations.

The field of physics-informed machine learning is very broad, as reviewed most recently in [325] and [138, 54, 139]. We focus on the task of learning fast and accurate surrogate models when a fine-scale simulated dataset and a fast and approximate coarse-grained simulation is available. This task differs from other interesting research areas in equation discovery or symbolic regression [47, 173, 174, 171, 242], downscaling or superresolution [332, 31, 154, 288, 309, 109], design space exploration or data synthesis [65, 58], controls [27] or interpretability [302, 199]. Our work is complementary to data assimilation or parameter calibration [132, 133, 140, 342, 33] which generally fit to observational data instead of models and differs from inverse modeling and parameter estimation [230, 114, 336, 172] which usually fit parametrizations that are independent of the previous state.

Correcting coarse-grid simulations via parametrizations. Problems with large domains are often solved via multiscale methods, as defined in [237]. Multiscale methods simulate the dynamics on a coarse-grid and capture the effects of small-scale dynamics that occur within a grid cell via additive terms, called subgrid parametrizations, closures, or residuals [237, 200]. The term multiscale thus also refers to models with only two scales [237]. Existing subgrid parametrizations for many equations are still inaccurate [320] and ML outperformed them by learning parametrizations directly from high-resolution simulations; for example in turbulence [85], climate [104], chemistry [116], biology [238], materials [170], or hydrology [21]. The majority of ML-based parametrizations, however, is local [104, 221, 39, 41, 42, 338, 51, 21, 116, 170, 239, 169, 230, 330, 251], i.e., the in- and output are variables of single grid points, which assumes perfect scale separation, for example, in isotropic homogeneous turbu-

lent flows [226]. However, local parametrizations are inaccurate; for example in the case of anisotropic nonhomogeneous dynamics [226, 312], for correcting global error of coarse spectral discretizations [36], or in large-scale climate models [84, 234]. More recent works propose non-local parametrizations, but their formulations either rely on a fixed-resolution grid [312, 30, 157, 347, 61], an autodifferentiable solver [305, 281, 95], or are formulated for a specific domain [20]. A single work proposes non-local and resolution-variable parametrizations [235], but requires the explicit representation of a high-resolution state which is computationally infeasible for large domains, such as in climate modeling. We propose resolution-variable and non-local parametrizations via neural operators to create fast and accurate surrogates.

Neural operators for resolution-variable, non-local parametrizations. Most current learning-based non-local parametrizations rely on FCNNs, CNNs [157], or RNNs [61], which are mappings between finite-dimensional spaces and dependent on the spatial resolution. As a result, CNNs for example require by default to be retrained after changing the grid resolution. Neural operators use a global kernel to learn mappings in between infinite-dimensional function spaces [153] similar to Laplacian, Hessian, or Jacobian. Neural operators lift the input from a spatial into a spectral state via a Fourier [166], Eigen- [25], graph kernel [165, 5] or other transform [177]. As a result, Fourier neural operator (FNO) for example is resolution-variable, as defined in Appendix A.6.1. We are the first to formulate neural operators for learning parametrizations.

3.3 Approach

We propose *multiscale neural operator* (MNO): a surrogate model with quasilinear runtime complexity that corrects known coarse-grained simulations via a learned resolution-variable, non-local parametrization. Similar to other concurrently devel-

oped parametrizations [162], MNO propagates the dynamics according to:

$$\underbrace{\frac{\partial \bar{u}}{\partial t}}_{\text{Corrected Large-scale Dyn.}} = \underbrace{\mathcal{N}(\bar{u})}_{\text{Large-scale Dyn.}} + \underbrace{\mathcal{K}_\theta(\bar{u})}_{\text{Parametrization}} \tag{3.1}$$

3.3.1 Multiscale neural operator

Partial differential equations. We focus on partial differential equations (PDEs) that can be written as initial value problem (IVP) via the method of lines [326]. The PDEs in focus have one temporal dimension, $t \in [0, T] =: D_t$, and (multiple) spatial dimensions, $x = [x_1, ..., x_d]^T \in D_x$, and can be written in the iterative, explicit, symbolic form [91]:

$$\begin{gathered} \frac{\partial u}{\partial t} - \mathcal{N}(u) = 0 \text{ with } t, x \in [0, T] \times D_x \\ u(0, x) = u^0(x),\ \mathcal{B}[u](t, x) = 0 \text{ with } x \in D_x,\ (t, x) \in [0, T] \times \partial D_x \end{gathered} \tag{3.2}$$

In our case, the (non-)linear operator, $\mathcal{N}$, encodes the **known** physical equations; for example a combination of Laplacian, integral, differential, etc. operators. Further, $u : D_t \times D_x \to D_u$ is the solution to the initial values, $u^0 : D_x \to D_u$, and Dirichlet, $\mathcal{B}_D[u] = u - b_D$, or Neumann boundary conditions, $\mathcal{B}_N[u] = n^T \partial_x u - b_N$, with outward facing normal on the boundary, $n \perp \partial B$.

Scale separation. We transfer a concept from the rich and mathematical literature in multiscale modeling [237] to consider a filter kernel operator, $\mathcal{G}*$, that creates the large-scale solution, $\bar{u}(x) = u(x) + u'(x)$, where u' are the small-scale deviations and $\bar{\cdot}$ denotes the filtered variable, $\bar{\phi}(x) = \mathcal{G} * \phi = \int_{D_x} G(x, x')\phi(x')dx'$. Assuming the kernel, G, 1) preserves constant fields, $\bar{a} = a$, 2) commutes with differentiation, $[\mathcal{G}*, \frac{\partial}{\partial s}]$ with $s{=}x, t$, and 3) is linear, $\overline{\phi + \psi} = \bar{\phi} + \bar{\psi}$ [226], we can compute the coarse-grained version of Eq. (3.2) as:

$$\begin{aligned} \mathcal{G} * \frac{\partial u}{\partial t} = \frac{\partial \bar{u}}{\partial t} &= \mathcal{G} * \mathcal{N}(u) \\ &= \mathcal{N}(\bar{u}) + [\mathcal{G}*, \mathcal{N}](u) \end{aligned} \tag{3.3}$$

where $\mathcal{N}(\bar{u})$ are the dynamics from plugging the coarse-grained solution into the known large-scale equations. Propagating the coarse-grained solution introduces an error, $h(u) := [\mathcal{G}*, \mathcal{N}](u) = \mathcal{G} * \mathcal{N}(u) - \mathcal{N}(\mathcal{G} * u)$, which is called the subgrid(-scale) parametrization, closure, or commutation error.

Approximations of the subgrid parametrization as an operator that acts only on the coarse-grained solution, $\bar{u}$, require significant domain expertise and are derived on a problem-specific basis. In the case of isotropic homogeneous turbulence, for example, the subgrid parametrization can be approximated as the spatial derivative of the subgrid stress tensor, $[\mathcal{G}*, \mathcal{N}](\bar{u})_{\text{turbulence}} \approx \frac{\partial \tau_{ij}}{\partial x_j} = \frac{\partial \overline{u_i' u_j'}}{\partial x_j}$ [226]. The goal in ML-based parametrizations is to improve accuracy and reduce the derivation time for new problems [239, 169, 230, 330].

Multiscale neural operator. We approximate the subgrid parametrization, $[\mathcal{G}*, \mathcal{N}] \approx: h$, via learning a neural operator on training data from a high-resolution simulation. Let $\mathcal{K}_\theta$ be a neural operator with the mapping:

$$[\mathcal{G}*, \mathcal{N}] \approx \mathcal{K}_\theta : \bar{U}(D_x; \mathbb{R}^{d_u}) \to H(D_x; \mathbb{R}^{d_u}) \tag{3.4}$$

where θ are the learned parameters and $\bar{U}, H$ are separable Banach spaces of all continuous functions, $\bar{u}, h$, taking values, $\mathbb{R}^{d_u}$, defined on the bounded, open set, $D_x \subset \mathbb{R}^{d_x}$, with norm $||f||_{\bar{U}} = ||f||_H = \max_{x \in D_x} |f(x)|$. We embed the neural operator as an autoregressive model such that the final *multiscale neural operator* (MNO) model is:

$$\bar{u}(t + \Delta t) = \int_t^{t+\Delta t} \mathcal{N}(\bar{u}(\tau)) + \mathcal{K}_\theta(\bar{u}(\tau)) d\tau \tag{3.5}$$

where the integral $\int_t^{t+\Delta t} \tau$ is solved with a finite difference method, such as the Euler or 4th order Runge-Kutta method. MNO uses the mean-square error (MSE) between the ground-truth, $h(u)$, and predicted parametrization, $\hat{h} := \mathcal{K}_\theta(\bar{u}) = \mathcal{K}_\theta(\mathcal{G} * u)$, as loss function, $\mathcal{L}$. We discuss alternative probabilistic, adversarial, physics-informed, or temporal loss functions in Appendix A.1.

$$\text{Loss} = \mathbb{E}_t \mathbb{E}_{\bar{u}|u(t)\sim p(t)} \mathcal{L}\left[\mathcal{K}_\theta(\bar{u}(t)), h(u(t))\right] \tag{3.6}$$

The ground-truth data, $u(t) \sim p(t)$, is generated by integrating a high-resolution simulation with varying parameters, initial or boundary conditions and uniformly sampling time snippets according to the distribution $p(t)$. During training, the model inputs are calculated by coarse-graining the high-resolution simulation at every time-step, $\bar{u}(t) = \mathcal{G} * (u(t))$. The ground-truth parametrizations, $h(u(t))$, are calculated as functions of the high-resolution solution as detailed in Eq. (3.7) and Eq. (3.9). During inference MNO is initialized with a large-scale state and integrates the dynamics in time via coupling the neural operator and a large-scale simulation.

Choice of neural operator. Our formulation is general enough to allow the use of many operators, such as Fourier [166], PCA-based [25], low-rank [145], Graph [165] operators, or DeepOnet [314, 177]. Because DeepONet [177] focuses on interpolation and assumes fixed-resolution sensor data, we decided to embed Fourier Neural Operator (FNO) [166] in MNO. We reimplemented the same FNO architecture as in [166] to be parallelizable on large-scale datasets and tuned hyperparameters, as detailed in Appendix A.5.1. FNO is a universal approximator of nonlinear operators [153, 62] and can be formulated as autoregressive model [166, 236]. FNO is resolution-variable, i.e., it can be trained and tested on equispaced grids of different size and resolution in spatial space. The spectral resolution is fixed once the model is trained and error is constant across different spatial resolutions, as discussed in Appendix A.6.1 and [166]-Fig.3). We chose FNO due to the inductive biases of a Fourier transform; in climate modeling many spatiotemporal phenomena can be described as recurring frequencies or modes [290] which is implicitly assumed in the FNO architecture [300]. In comparison, a CNN would have introduced a locality bias [107]. A Fourier transform is translationally symmetric on a torus and, in our case, introduces an appropriate translational symmetry bias for multiscale Lorenz96 and QG turbulence. There are many known equi- and invariances of the subgrid parametrization term [226, 239] which could be incorporated in future design of neural operators.

3.3.2 Illustration of MNO via multiscale Lorenz96

We illustrate the idea of MNO on a canonical model of atmospheric dynamics, the multiscale Lorenz96 equation [175, 299]. This PDE is multiscale, chaotic, time-continuous, space-discretized, 1D, nonlinear, displayed in Fig. 3-2-right and detailed in Appendix A.3. Most importantly, the large- and small-scale solutions, $X_k \in \mathbb{R}, Y_{j,k} \in \mathbb{R} \; \forall \; j \in \{0, ..., J\}, k \in \{0, ..., K\}$, demonstrate the *curse of dimensionality*: the number of the small-scale states grows exponentially with scale and explicit modeling becomes computationally expensive. Solving the full small-scale dynamics in Eq. (A.3), for example, is quadratic: $O(N^2) = O(JK)$. The large-scale portion is as follows:

$$\frac{\partial X_k}{\partial t} = \underbrace{X_{k-1}(X_{k+1} - X_{k-2}) - X_k + F}_{\text{Large-scale Dyn.: } \frac{\partial \bar{X}_k}{\partial t}} \underbrace{- \frac{h_s c}{b} \sum_{j=0}^{J-1} Y_{j,k}(X_k)}_{\text{Parametrization: } h}, \tag{3.7}$$

where F is the forcing, h_s the coupling strength, b the relative magnitude of scales, and c the evolution speed. With the multiscale framework from Section 3.3.1, we define the solution, $u(x)$, operator, $\mathcal{N}(u)(x)$, and kernel, $G(x, x')$, as detailed in Appendix A.3.1.

MNO then learns the parametrization term via a neural operator, $\mathcal{K}_\theta = \hat{h} \approx h$, and models:

$$\partial_t \hat{\bar{X}}_k = \partial_t \bar{X}_k + \mathcal{K}_\theta(\hat{\bar{X}}_{0:K})(k) \tag{3.8}$$

with the abbreviation, $\partial_t = \frac{\partial}{\partial t}$. The known large-scale dynamics without parametrization are, $\partial_t \bar{X}_k$, and the corrected large-scale dynamics, $\partial_t \hat{\bar{X}}_k$. The ground-truth parametrization is $h(x) = \{-\frac{h_s c}{b} \sum_{j=0}^{J-1} Y_{j,k}(X_k)$ if $x = k(J+1) \; \forall k \in \{0, \ldots, K\}$ and 0 otherwise$\}$. See Appendix A.4 for all terms.

The parametrization, $\mathcal{K}_\theta$, accepts inputs that are sampled anywhere inside the spatial domain, which differs from previous local [251] or resolution-dependent [61] multiscale Lorenz96 parametrizations.

We create the ground-truth data via randomly sampled initial conditions, periodic boundary conditions, and integrating the coupled equation with a 4th-order Runge-Kutta solver. After a Lyapunov timescale the state is independent of initial conditions

and we extract 4K snippets with $T/\Delta t = 400$steps length, corresponding to 10 Earth days, for 1-step training. During test the model is run autoregressively on 1K samples from a different initial condition, as detailed in Appendix A.3.

3.3.3 MNO for 2D Quasi-Geostrophic Turbulence

We further demonstrate MNO on a turbulent set of 2D Navier-Stokes equations. One-layer quasi-geostrophic (QG) turbulence shown in Fig. 3-1b is a widespread model for atmospheric turbulence at the equator and derived in Appendix A.2 and [196, 95]. Filtering out the small-scales lets us write QG turbulence for simulating vorticity, $\bar{\omega}$, in the parametrized large-scale equation:

$$\partial_t \bar{\omega} = \underbrace{-J(\bar{\psi}, \bar{\omega}) + \nu \nabla^2 \bar{\omega} - \mu \bar{\omega} - \beta \partial_x \bar{\psi} + \bar{F}}_{\text{Large-scale Dyn: } \mathcal{N}(\bar{\omega})} + \underbrace{J(\bar{\psi}, \bar{\omega}) - \overline{J(\psi, \omega)}}_{\text{Parametrization: } h(\omega)}, \tag{3.9}$$

where $\bar{\psi}$ is the streamfunction with $\bar{\omega} = \nabla^2 \bar{\psi}$ and $\mathcal{N}$ the nonlinear operator of the known large-scale dynamics. The constant parameters are turbulent viscosity, ν, linear drag coefficient, μ, and Rossby parameter, β. The source term, F, is chosen to induce significant turbulence with Reynolds number $\text{Re} = 22 \times 10^4$.

In the case of QG turbulence, calculating the nonlinear Jacobian operator, $\overline{J(\psi, \omega)}$, from a filtered state is analytically infeasible. Hence, we choose a neural operator, $\mathcal{K}_\theta$, to learn the mapping from the filtered state, $\bar{\omega}$, to the ground-truth subgrid parametrization, h, at $128 \times 128 \times 1$ large-scale resolution. The final multiscale neural operator for QG turbulence is:

$$\partial_t \bar{\omega}(x, y) = \mathcal{N}(\bar{\omega})(x, y) + \mathcal{K}_\theta(\bar{\omega})(x, y) \tag{3.10}$$

3.4 Results

Our results demonstrate that multiscale neural operator (MNO) is faster than direct numerical simulation, generates stable solutions, and is more accurate than current parametrizations. We now proceed to discussing each of these in more detail.

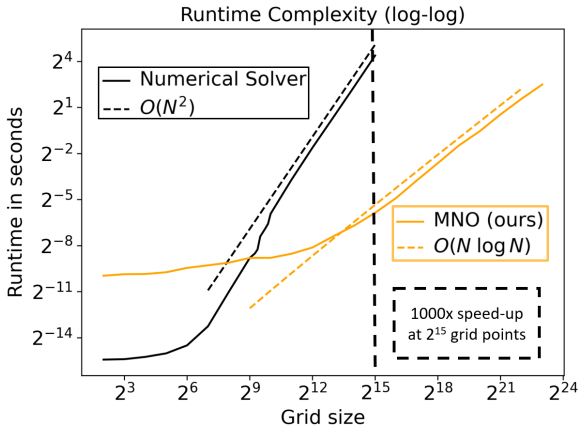
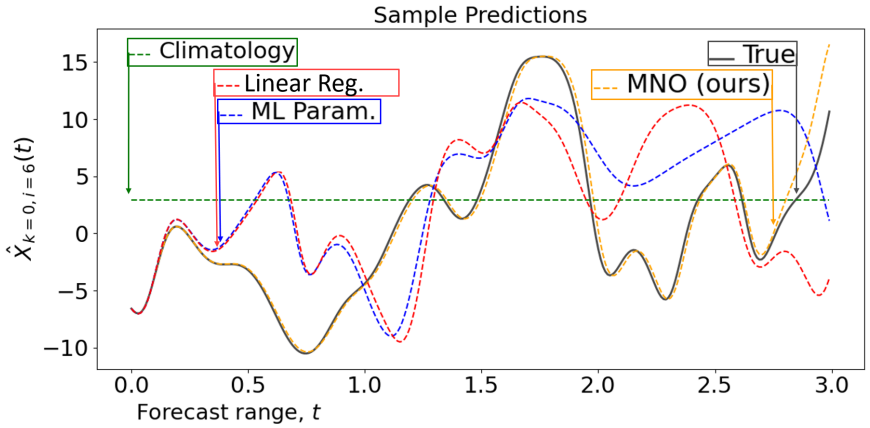


Figure 3-3: **MNO is faster than direct numerical simulation.** Our proposed multiscale neural operator (orange) can propagate multiscale PDE dynamics in quasilinear complexity, $O(N \log N)$. For a grid with $K = 2^{15}$, MNO is $\sim$ 1000-times faster than direct numerical simulation (black) which has quadratic complexity, $O(N^2)$

3.4.1 Runtime complexity: MNO is faster than a traditional PDE solver

MNO (orange in Fig. 3-3) has quasilinear, $O(N \log N)$, runtime complexity in the number of large-scale grid points, $N{=}K$, in the multiscale Lorenz96 equation. The runtime is dominated by a lifting operation, here a fast Fourier transform (FFT), which is necessary to learn spatial correlations in the spectral space. In comparison, the direct numerical simulation (black) has quadratic runtime complexity, $O(N^2)$, because of the explicit representation of $N^2{=}JK$ small-scale states. Both models are linear in time, $O(T)$. Local parametrizations can achieve optimal runtime, $O(N)$, but it is an open question if there exists a spectral transform other than FFT that yields a non-local model in optimal runtime.

We ran MNO up to a resolution of $K = 2^{24}$, which would equal $75cm/px$ in a global 1D (space) climate model and only took $\approx 2s$ on a single CPU. MNO is three orders of magnitude (1000-times) faster than DNS, at a resolution of $K = 2^{15}$ or $200m/px$. For 2D or 3D simulations the gains of using MNO vs. DNS are even higher with $O(N^2 \log N)$ vs. $O(N^4)$ and $O(N^3 \log N)$ vs. $O(N^6)$, respectively [144]. The runtime comparison is detailed in Appendix A.5.2. We focus on a runtime comparison, but MNO might also have significant savings in memory: representing the state at $K = 2^{17}$ in double precision occupied 64GB RAM in our DNS and 0.5MB in MNO.



Method	RMSE
Climatology	6.902
Traditional parametrizations	2.326
ML-based parametrization [252]	2.053
MNO (ours)	0.5067

Figure 3-4: **Left: MNO is more accurate than traditional parametrizations.** A sample trajectory illustrates how MNO (yellow/orange-dotted) can forecast the large-scale physics (black-solid), $X_{k=0}(t)$. In comparison, ML-based blue-dotted) and traditional (red-dotted) parametrizations quickly start to diverge. Note that the system is chaotic and small deviations are rapidly amplified; even inserting the exact parametrizations in float32 instead of float64 quickly diverges. **Right: Accuracy.** MNO is more accurate than traditional parametrizations as measured by the root mean-square error (RMSE).

3.4.2 MNO is more accurate than traditional parametrizations

Figure 3-4-left shows a forecasted trajectory of a sample at the left boundary, $k = 0$, where MNO (orange-dotted) accurately forecasts the large-scale dynamics, $X_0(t)$, (black-solid) while current ML-based (blue-dotted) [104] and traditional parametrizations (red-dotted) quickly diverge. The quantitive comparison of RMSE and a mean/std plot Fig. 3-4 over $1K$ samples and 200steps or 10days ($\Delta t = 0.005 = 36$min) confirms that MNO is the most accurate in comparison to ML-based parametrizations, traditional parametrizations, and a mean forecast (climatology). Note, the difficulty of the task: when forecasting *chaotic* dynamics even numerical errors rapidly amplify [226].

ML-based parametrizations is a state-of-the-art (SoA) model in learning parametrizations and trains a ResNet to forecast a local, resolution-variable parametrization,

$h_k = \text{NN}(X_k)$, similar to [104]. The **traditional parametrizations** (trad. param.) are often used in practice and use linear regression to learn a local, resolution-variable parametrization [200]. It was suggested that multiscale Lorenz96 is too easy as a test-case for offline evaluation because traditional parametrizations already perform well [250], but the significant difference between MNO and Trad. Params. shows that online evaluation is still interesting. The **climatology** forecasts the mean of the training dataset, $X_k(t) = 1/T \sum_{t=0}^{T} 1/N \sum_{i=0}^{N} X_{k,i}(t)$. The full list of hyperparameters and model parameters can be found in Appendix A.5.3. The accuracy gain likely stems from a change from local to global receptive fields of the parametrization. A comparison with CNNs or physics-constrained architectures [249, 81] are interesting, but would evaluate different aspects which were out of scope, as detailed in Appendix A.5.3.

3.4.3 MNO on time-series

Figure 3-5 shows that predicting Lorenz96's large-scale dynamics with MNO is empirically stable, i.e., the error does not diverge to infinity over time. We first plot a randomly selected sample of the first large-scale state, $X_{k=0}(t)$ (left-black), to illustrate that the prediction is bounded. The MNO prediction (left-yellow) follows the ground-truth up to an approximate horizon of, $t = 1.8$ or 9 days, then differs from the ground-truth solution, but stays within the bounds of the ground-truth prediction and does not diverge to infinity. The RMSE over time in Figure 3-5-right shows that MNO (yellow) is approximately more accurate than current ML-based (blue) and traditional (red) parametrizations for $\approx 100\%$-longer time, measuring the time to intersect with climatology. Despite the difficulty in predicting chaotic dynamics, the RMSE of MNO reaches a plateau, which is slightly above the optimal plateau given by the climatology (black). The RMSE is defined in Appendix A.5.4.

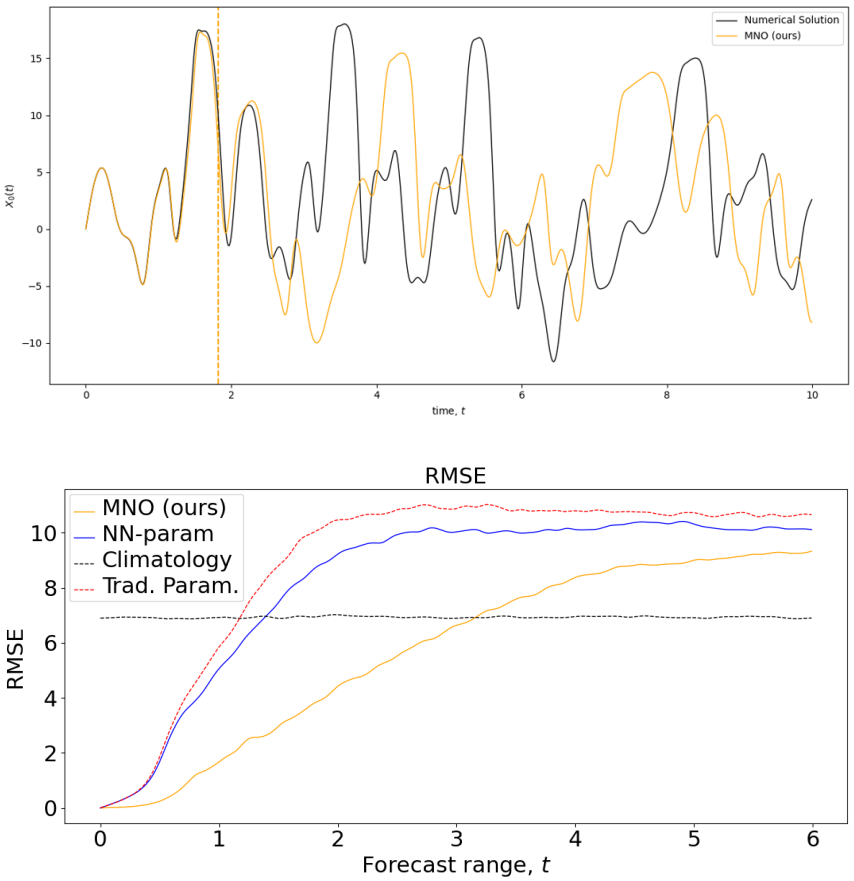


Figure 3-5: **MNO on long-term time series.** MNO can propagate a sample state, $X_{k=0}(t)$, over a long time horizon without diverging to infinity (left). The right plot shows that the RMSE of MNO plateaus for long-term forecasts, further confirming stability. Further, MNO (yellow) maintains accuracy longer than ML-based parametrizations (blue) and a climatology (black).

3.4.4 MNO in higher dimensions: quasi-geostrophic turbulence

Figure 3-6 shows that MNO learns to predict large- and small-scale patterns in the subgrid parametrization. Quantitative analysis on the test set confirms that MNO is more accurate than a baseline parametrization that predicts the mean parametrization of the training set, called null param.: RMSE of 8.3723vs.74.3126. Integrating the dynamics over time in Fig. 3-7 shows that the hybrid models, MNO and null param., are more accurate than a pure machine learning (ML)-based surrogate. For the pure ML surrogate, we trained a Fourier Neural Operator for autoregressive forecasting of the coarse-grained vorticity: $\bar{\omega}_{t+1} = FNO(\bar{\omega}_t)$, as detailed in Appendix A.5.3. While

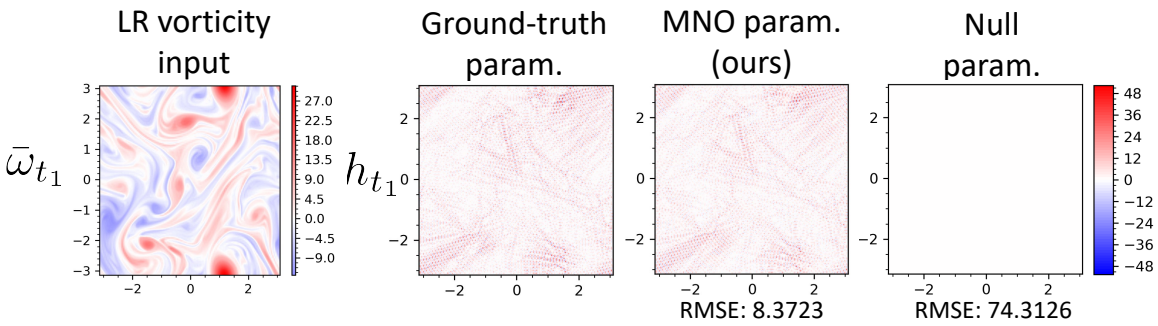


Figure 3-6: **MNO for Quasi-Geostrophic Turbulence.** MNO can predict most large- and small-scale patterns of the ground-truth subgrid parametrization, given the low-resolution vorticity input. The RMSE is 9x lower than a baseline that predicts the mean parametrization of the training set which is zero.

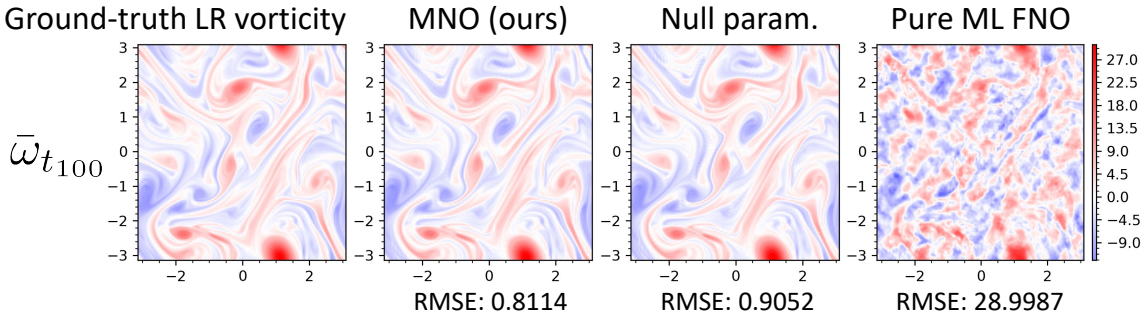


Figure 3-7: **If rolled out over time**, a pure machine learning-based model, here Fourier Neural Operator (FNO), diverges from the ground-truth vorticity, $\bar{\omega}_{t_{100}}$, after $\approx$ 100 time steps whereas the hybrid models, MNO and Null Param., follow it more accurately.

MNO slightly outperforms a baseline parametrization in the integrated forecast, the difference is only marginal. The authors presume that extending the underlying neural operator architecture could improve the integrated forecast, for example, by incorporating temporal information or multiresolution spectral transforms, such as multiwavelets. Appendix A.5.1 shows more results over larger predictions horizons.

3.5 Limitations and Future Work

We demonstrated the accuracy, speed, and stability of MNO on chaotic multiscale Lorenz96 and accuracy on quasi-geostrophic turbulence. MNO can be extended to parametrize non-local processes in global climate models, such as squall lines [312], clouds [227], or gravity waves [90].

MNO embeds the original FNO architecture where the spatial resolution and grid size can be varied between train and test, but assumes an equispaced grid, fixed spectral resolution, and a certain subset of initial and boundary conditions. Future work could capture local discontinuities, e.g., along coastlines [134] with multiwavelets [110], incorporate non-periodic boundaries via Chebyshev polynomials, or use Fourier transforms for non-equispaced grids [86], as detailed in Appendix A.6.

We implemented MNO to be myopic, i.e., we do not incorporate time in the feature or latent space, because it might theoretically suffice for chaotic dynamics [167, 162]. However, we believe that combining MNO with temporal learning schemes, such as, a posteriori learning [95], Neural ODEs [64], CNN-LSTMs [204], reservoir computing [234], or FourCastNet [236], shows significant promise. We are particularly interested in combining MNO with the pushforward trick [37] to create a temporal parametrization that does not require an autodifferentiable solver. Combining MNO with hard constraint models could also extend the prediction horizon; this is exciting as there exist known symmetries for many parametrizations [239].

3.6 Conclusion

We proposed a hybrid physics-ML surrogate of multiscale PDEs that is quasilinear, accurate, and stable. The surrogate limits learning to the influence of fine- onto large-scale dynamics and uses neural operators for a resolution-variable, non-local subgrid parametrization. We demonstrated multiscale neural operator on the chaotic systems: multiscale Lorenz96 and quasi-geostrophic turbulence. MNO is faster than direct numerical simulation ($O(N \log N)$ vs. $O(N^2)$) and more accurate ($\approx$ 100% longer prediction horizon) than baseline parametrizations for multiscale Lorenz96. With the reduction in runtime MNO could enable rapid parameter exploration and robust uncertainty quantification in complex climate models.

3.6.1 Ethics Statement

Climate change is a defining challenge of our time and environmental disasters will become more frequent: from storms, floods, wildfires and heat waves to biodiversity loss and air pollution [127]. The impacts of these environmental disasters will likely be unjustly distributed: island states, minority populations, and the Global South are already facing the most severe consequences of climate change, while the Global North is responsible for the most emissions since the industrial revolution [3]. Decision-makers require more accurate, accessible, and local tools to understand and limit the economic and human impact of a changing climate [227]. We propose multiscale neural operator (MNO) to improve the parametrizations in climate models, thus leading to more accurate predictions. Related techniques to MNO, specifically neural operator-based surrogate models, could help reduce computational complexity of large-scale weather and climate models. The reduced computational complexity would make them more accessible to low-resource countries or allow for higher resolution predictions. Unfortunately, discoveries for faster differential equations solvers can and likely will be leveraged in ethically questionable fields, such as missile development or oil discovery. We acknowledge the possible negative impacts and hope that our targeted discussion and application to equations from climate modeling can steer the our work towards a positive impact.

3.6.2 Reproducibility Statement

The code will be open-sourced. A list of hyperparameters per model is in Appendix A.5.3; data splits are explained in results and Appendix A.2; all simulation details in Appendices A.2 and A.3; background for the neural operator in Appendix A.7; and ethics statement in Section 3.6.1. The full study used approximately 15K CPU hours on a cluster.

Chapter 4

Satellite Imagery from the Future: Creating Physically-Consistent Visualizations of Climate Data with Deep Generative Vision Models

Chapter abstract: Long-term impacts of climate change are difficult to communicate to a wide audience. Maps and other visualizations of geospatial data have helped in this regard. But, most visualizations still focus on a mostly scientific audience and omit the general public. To address this issue, we have developed a new approach that creates 'satellite imagery from the future'. Our approach leverages deep generative vision models to transform the outputs of Earth system models or human-induced landscape changes into satellite imagery. To ensure trustworthy visualizations, we condition and evaluate our model on segmentation maps of projected ground-truth impacts. We demonstrate the effectiveness of our approach using two cases: flooding and reforestation. To promote transparency and reproducibility, we are sharing our code and dataset that includes a third case study of melting Arctic sea ice and over 25,000 labeled image triplets which can be used to study segmentation guided

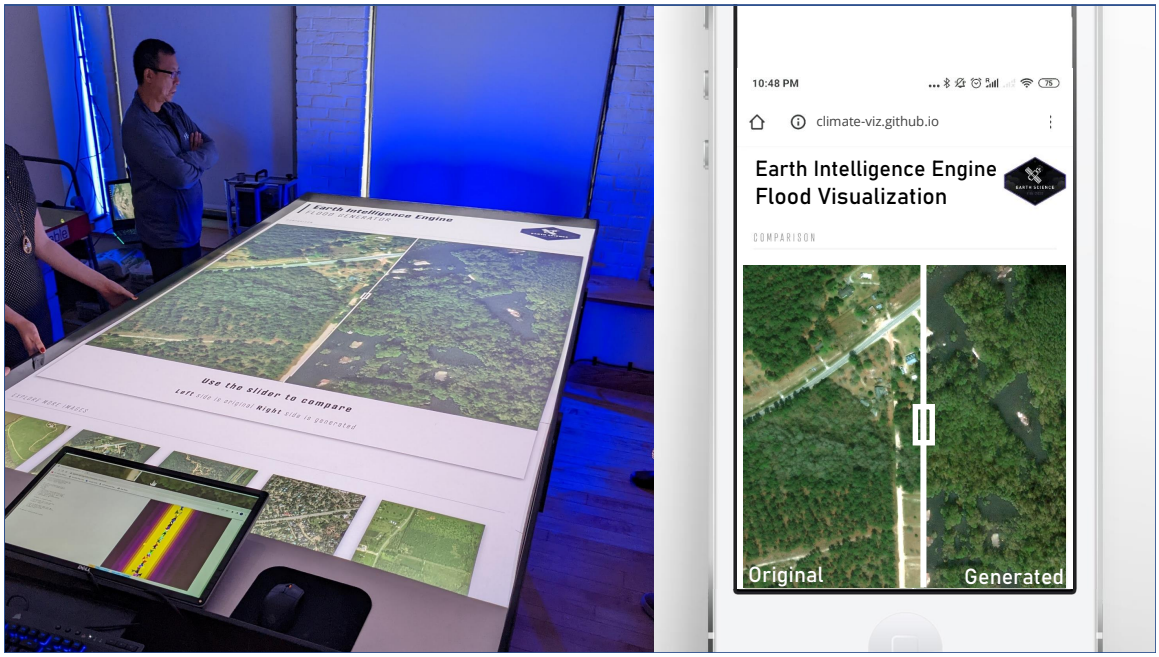


Figure 4-1: **We synthesize satellite imagery that visualizes flooding** (right). The new visualizations facilitate intuitive communication of climate risks, for example, via tabletop exercises as seen on the left. Explore more results at climate-viz.github.io.

image-to-image translation in Earth observation [1].

4.1 Introduction

With climate change, natural disasters are becoming more intense [127]. Floods are the most frequent weather-related disaster [56] and already cost the U.S. 4.1B USD per year [217]; this damage is projected to grow over the next decades [127].

Visualizations of climate impacts are widely used by policy and decision makers to raise environmental awareness and facilitate dialogue on long-term climate adaptation decisions [277]. Visualizations of flood risks, for example, are used in local policy making and community discussion groups as decision-aids for flood infrastructure investments [277]. Current visualizations of flood impacts, however, are limited to color-coded flood maps [218, 68, 216] or synthetic street-view imagery [291, 273], which do not convey city-wide flood impacts in a compelling manner, as shown

[1] Code and data will be open-sourced at github.com/blutjens/earth-eie; an interactive demo is available at climate-viz.github.io.

in Fig. 4-3 and [276]. Our work generates synthetic satellite imagery of future coastal floods that are informed by the projections of expert-validated flood models, as shown in Fig. 4-1. As a map, this imagery will enable a more engaging communication of city-wide flood risks to governmental offices. We call our visualization tool the *Earth Intelligence Engine*. While visualizing climate risks such as flooding can encourage pro-environmental behavior [297] it can also feel overwhelming [225]. Hence, we extend the *Earth Intelligence Engine* to also communicating hopeful positive action, specifically, the effect of reforestation projects in Fig. 4-2.

We focus on deep generative vision models, such as generative adversarial networks (GANs) [105], as our method for visualization. Generative vision models have generated impressive imagery of faces [130, 316], animals [348, 44], street-level flood imagery [273], and recently even satellite observations [257, 96, 206, 280, 7]. Synthetic satellite imagery, however, needs to be trustworthy [12]. Many complementary approaches exist to increase the trustworthiness of generative vision models, including interpretable networks [17], adversarial robustness [195, 271, 184], or probabilistic predictions with uncertainty [301, 179, 183]. Here, we raise a new question of 'How can we increase trust in synthetic satellite imagery through physical-consistency?'.

We define a synthetic image to be physically-consistent if the depictions in the image are consistent with the output of a physics-based model, as detailed in Section 4.5.3. Our definition of physical-consistency relates to the field of physics-informed machine learning (ML) in which researchers find novel ways to embed physics domain knowledge into deep learning methods [138, 248, 289, 252, 185]. We considered various physics-informed ML methods to incorporate the physics of floods in a generative vision model as inputs [255], constrained representation [181, 108, 17], training loss [249], hard output constraints [205, 81, 118], or evaluation function [161]. Vice versa, we also considered to embed a generative vision model in physics, specifically, in the differential equations of floods as learned parameters [102, 249], dynamics [64], residual [140, 338], differential operator [248, 174], or solution [249]. Finally, we decided to incorporate physics as input and evaluation functions. Specifically, we use the 1-channel projections of a physics-based coastal flood model and 3-channel satel-

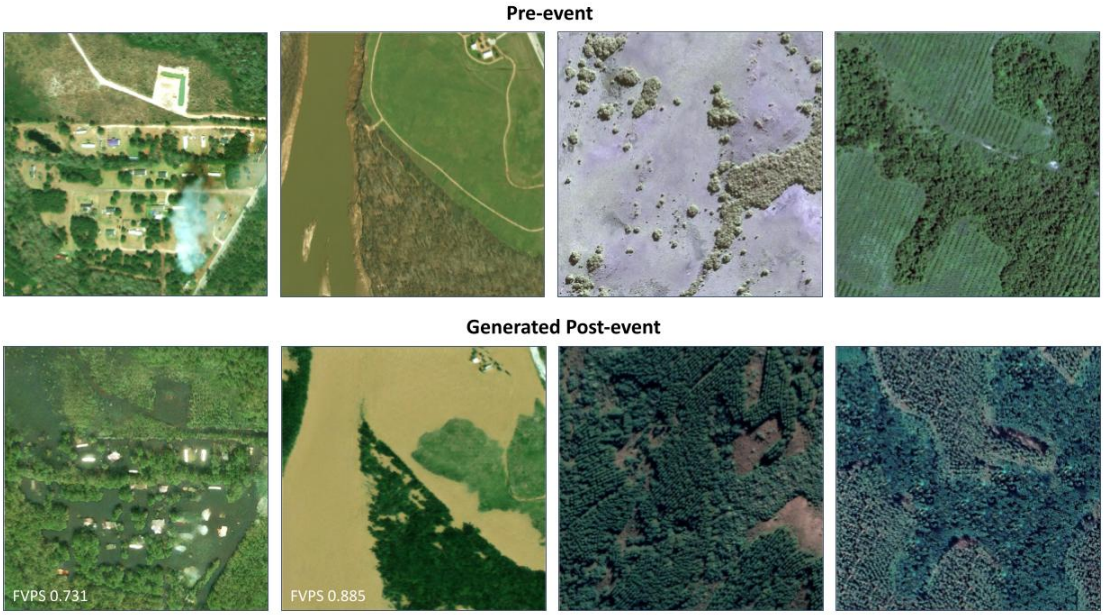


Figure 4-2: **The *Earth Intelligence Engine* generates 'satellite imagery from the future'.** Specifically, we visualize how flooding (left) or reforestation (right) would impact the landscape as seen from space.

lite imagery as inputs to a deep generative vision model and evaluate the intersection over union (IoU) of the generated image and flood input, as detailed in Section 4.5.

Our work makes the following contributions:

- a novel framework to measure physical-consistency in synthetic satellite imagery,
- the first physically-consistent and photorealistic visualization of flood risks as satellite imagery,
- the first visualization of planned reforestation projects as satellite imagery, and
- an open-source dataset with over $25k$ labeled high-resolution image-triplets that can be used to image-to-image translation in Earth observation.

4.2 Related Work

We generate trustworthy visualizations of climate adaptations and impacts by formulating a semantic image synthesis task and applying deep generative vision models to

solve it.

4.2.1 Generative vision modeling

We formulate the generation of satellite imagery as an image-to-image (im2im) translation problem: learn a mapping from satellite image and segmentation mask to another satellite image [130]. Deep generative vision models have been most successful at solving im2im problems [130]. Generative adversarial networks (GANs) have been successfully used in semantic image synthesis, a subproblem in im2im, to generate photorealistic street scenery from semantic segmentation masks: DCGAN [245], Pix2pixHD [316], DRPAN [316], SPADE [231], or OASIS [275]. Similarly, probabilistic normalizing flows (NFs) [258, 179], variational autoencoders (VAEs) [147, 349], autoregressive models [232], and – concurrently to this study – also diffusion-based models ([268] and Fig. B-4) have been adapted to im2im translation. Our use-case requires a deterministic semantic image synthesis model that is capable of generating realistic high-resolution (i.e., 1024x1024px) images. We decided to focus on GANs, because VAEs generate less realistic images ([82, 349] and Fig. 4-4) and NFs, autoregressive, and diffusion-based models were not accessible for high-resolution semantic image synthesis during this study. Further, our use-case only requires deterministic outputs. So, we decided to extend the high-resolution semantic image synthesis model, pix2pixHD [316], to take in 4-channel images that include physical information and to generate satellite imagery that is both photorealistic and physically-consistent.

4.2.2 Climate change visualization tools

Visualizations of climate change are commonly used in policy making and community discussions on climate adaptation [277, 69]. Landscape visualizations are used to raise environmental awareness in the general public or policy [276, 273], because they can convey the impacts of climate change, such as rising sea levels or coastal floods, in a compelling and engaging manner ([276], Fig. 4-3b). Most landscape visualizations, however, are limited to regional information [291]. Additionally, most landscape vi-

a) color-coded b) street view c) satellite (ours)

Figure 4-3: **Physically-consistent satellite imagery (c) could enable more engaging and relatable communication of city-scale flood risks [277].** Most existing visualizations of coastal floods or sea-level rise that are aimed towards the public rely on color-coded geospatial rasters (a), that can be unrelatable or impersonal [218, 93, 67]. Alternative photorealistic visualizations are often limited to local street-level imagery (b) [291, 273] and lack further spatial context. Images, left-to-right: [218, 218, 291, 291, 112], ours.

sualizations require expensive physics-based renderings and/or high-resolution digital elevation models [291]. Alternative visualization tools of coastal floods or sea-level rise are color-coded maps, such as [214, 218, 67]. Color-coded maps convey the flood extent on a city-wide scale, but are less engaging than a photorealistic image [277]. We are generating compelling visualizations of future climate visualizations as satellite imagery to aid in policy and community discussion on climate adaptation.

4.3 Experimental Results

As detailed in Section 4.5, we trained a deep generative vision model to synthesize satellite imagery that depicts floods. Specifically, we trained a pix2pixHD GAN [316] to translate a 'pre-flood' satellite image of size (1024,1024,3) and a corresponding 'flood mask' segmentation map that is derived from a ground-truth flood image of size (1024,1024,1) to a 'post-flood' satellite image tile that depicts floods of size (1024,1024,3). The dataset is described in Section 4.5.1. This section analyses the visual quality and physical-consistency of the generated imagery and underlying flood segmentation model. We also evaluate the generalization performance to other locations, extend the model to visualize reforestation, and discuss a dataset for future extension to melting Arctic sea ice.

4.3.1 Evaluation Metrics.

Evaluating imagery generated by a GAN is difficult [333, 34]. Most evaluation metrics measure photorealism or sample diversity [34], but not physical consistency [254] (see, e.g., SSIM [318], MMD [35], IS [270], MS [303] , FID [123, 346], or LPIPS [343]).

To evaluate physical consistency we propose using the intersection over union (IoU) between water in the generated imagery and water in the flood extent map. This method relies on flood masks, but because there are no publicly available flood segmentation models for Maxar RGB satellite imagery, we trained our own model on ∼100 hand-labeled flooding images (Section 4.3.3). This segmentation model produced flood masks of the generated and ground-truth flood image which allowed us to measure the overlap of water in between both. When the flood masks overlap perfectly, the IoU is 1; when they are completely disjoint, the IoU is 0.

To evaluate photorealism, we used the state-of-the-art perceptual similarity metric Learned Perceptual Image Patch Similarity (LPIPS) [343]. LPIPS computes the feature vectors (of an ImageNet-pretrained AlexNet CNN architecture) of the generated and ground-truth tile and returns the mean-squared error between the feature vectors (best LPIPS is 0, worst is 1).

Because the joint optimization over two metrics poses a challenging hyperparameter optimization problem, we propose to combine the evaluation of physical consistency (IoU) and photorealism (LPIPS) in a new metric (FVPS), called Flood Visualization Plausibility Score (FVPS). The FVPS is the harmonic mean over the submetrics, IoU and $(1-\text{LPIPS})$, that are both $[0,1]$-bounded. Due to the properties of the harmonic mean, the FVPS is 0 if any of the submetrics is 0; the best FVPS is 1. In other words, the FVPS is only 1 if the imagery is both photorealistic and physically-consistent.

$$\text{FVPS} = \frac{2}{\frac{1}{\text{IoU}+\epsilon} + \frac{1}{1-\text{LPIPS}+\epsilon}}. \tag{4.1}$$

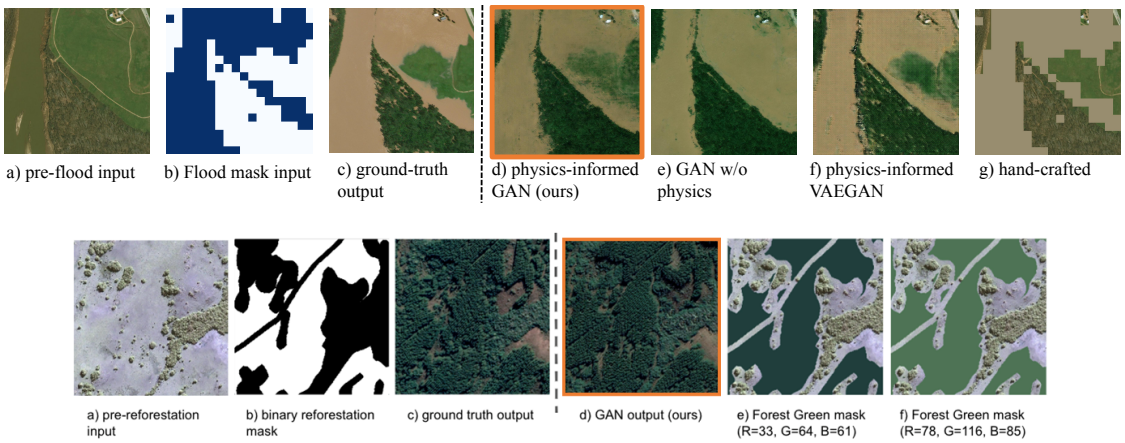


Figure 4-4: **Top: Flooding.** The proposed physics-informed GAN, (d), generates photorealistic and physically-consistent flood imagery from the inputs, (a,b), outperforming all other models, (e,f,g). The baseline GAN, pix2pixHD [316] (e), in comparison, receives only a pre-flood image and no physical input. The resulting image, (e), is fully-flooded, rendering the model untrustworthy. The VAEGAN, BicycleGAN [349] (f), creates glitchy imagery (zoom in). A handcrafted baseline model (g), as used in common visualization tools [67, 214], visualizes the correct flood extent, but is pixelated and lacks photorealism. **Bottom: Reforestation.** The proposed pix2pixHD [316] GAN, (d), generates photorealistic and consistent reforestation imagery from the inputs, (a,b), outperforming baseline models (e,f).

4.3.2 Physical-consistency and photorealism.

We evaluate the comparative physical-consistency and photorealism of our physics-informed GAN via an ablation study of the flood mask (baseline GAN) and comparing to a photoshopped baseline model. Figure 4-4 qualitatively shows that our conditioned GAN outperforms a baseline GAN, as well as a handcrafted baseline model in terms of both physical-consistency and photorealism.

A GAN without physics information generates photorealistic but non physically-consistent imagery.

Figure 4-4e-top shows that the baseline GAN visualizes floods at locations where there should be no flooding according to the flood mask in Fig. 4-4b-top. The inaccurately modeled flood extent illustrates the physical-inconsistency of a baseline GAN. To quantitatively evaluate the physical-consistency, we calculate a flood mask of the predicted image via a segmentation model and measure the intersection over union

(IoU) of the predicted and ground-truth flood mask. The IoU over the test set which is detailed in Appendix B.1 is low (0.226) as shown in Table 4.1. Despite the photorealism of the baseline GAN (LPIPS = 0.293), the physical-inconsistency renders the model non-trustworthy for critical decision making as confirmed by the low FVPS of 0.275. The baseline GAN is the default pix2pixHD [316], which only uses the pre-flood image and no flood mask as input.

A handcrafted baseline model generates physically-consistent but not photorealistic imagery.

Similar to common flood visualization tools [67], the handcrafted model overlays the flood mask input as a hand-picked flood brown (#998d6f) onto the pre-flood image, as shown in Fig. 4-4g. Because typical storm surge models output flood masks at low resolution ($30m/px$ [218]), the handcrafted baseline generates pixelated, non-photorealistic imagery. Combining the relatively high IoU of 0.361 and the poor LPIPS of 0.415, yields a low FVPS score of 0.359, highlighting the difference to the physics-informed GAN in a single metric.

The proposed physics-informed GAN generates physically-consistent and photorealistic imagery.

To create the physics-informed GAN, we trained pix2pixHD [316] from scratch on our dataset (200 epochs in ∼7hrs on 8×V100 Google Cloud GPUs). This model successfully learned how to convert a pre-flood image and a flood mask into a photorealistic post-flood image, as shown in Fig. 4-2. The model improves over all other models in terms of IoU, LPIPS, and FVPS (Table 4.1). The learned image transformation "in-paints" the flood mask in the correct flood colors and displays an average flood *height* that does not cover structures (e.g., buildings, trees), as shown in 64 randomly sampled test images in Fig. B-3. Occasionally, city-scenes show scratch patterns, e.g., Fig. B-3 (top-left). This could be explained by the unmodeled variance in off-nadir angle, sun inclination, GPS calibration, color calibration, atmospheric noise, dynamic objects (cars), or flood impacts, which is partially addressed in Sec-

Table 4.1: **In terms of photorealism (LPIPS) and physical consistency (IoU), our physics-informed GAN outperforms three benchmarks**: the baseline GAN without physics; a physics-informed VAEGAN; and a handcrafted baseline. The proposed Flood Visualization Plausibility Score (FVPS) trades-off IoU and LPIPS as a harmonic mean and highlights the performance differences between the GAN with and without physics on low-resolution flood mask inputs.

	LPIPS ↓ high res.	LPIPS ↓ low res.	IoU ↑ high res.	IoU ↑ low res.	FVPS ↑ high res.	FVPS ↑ low res.
GAN w/ phys. (ours)	**0.265**	**0.283**	**0.502**	**0.365**	**0.533**	**0.408**
GAN w/o phys.	0.293	**0.293**	0.226	0.226	0.275	0.275
VAEGAN w/ phys.	0.449	-	0.468	-	0.437	-
Handcrafted baseline	0.399	0.415	0.470	**0.361**	0.411	0.359

tion 4.3.4. While our model also outperforms the VAEGAN (BicyleGAN), the latter has the potential to create ensemble forecasts over the unmodeled flood impacts, such as the probability of destroyed buildings.

4.3.3 Flood segmentation model.

Our approach requires a flood segmentation model to generate ground-truth flood masks and evaluate the predicted imagery, but there does not exist any open-source model that segments floods in high-resolution (<1m/px) satellite imagery. Hence, we trained our own flood segmentation model. We chose a pix2pix segmentation model [130] which uses a vanilla U-net as generator and trained it from scratch to minimize a modified loss function that minimizes $L1$-loss and IoU in addition to adversarial loss and finetuned the last layers on L1-loss. We hand-labelled pixel-wise flood maps of 109 post-flood images to train the model. A four-fold cross validation was performed leaving 22 images for testing in each split. The selected segmentation model has a mean IoU of 0.343 which matches the expected performance of pix2pix [130]. The IoU is lower than state-of-the-art performance of segmentation models [328] due to the size of our dataset. Further, many of our ground-truth labels only have very few flooded pixels as shown in Fig. B-7 which skews the IoU towards zero if not perfectly predicted.

Our labelled imagery is the first dataset for flood segmentation on visual high-

resolution ($< 1m/px$) satellite imagery and will be made available as part of the dataset. The full details on the flood segmentation model are described in Appendix B.2.

4.3.4 Generalization performance.

So far, all images were from Maxar-owned satellite imagery of hurricane Harvey in Houston, TX. Hence, our results have shown that our pipeline can generate post-flood imagery at select locations and train/test datasets that stem from the same remote sensing instrument. The *Earth Intelligence Engine*, however, aims to visualize global impacts of the climate challenge as seen from space. This subsection evaluates the generalization capability across location, remote sensing instrument, and climate phenomena.

Generalization across location and remote sensing instruments.

To visualize coastal flooding across the U.S. East coast our model needs to generalize across location and remote sensing instrument. The original model in Section 4.3.2 used pre-flood images of Houston, TX from Maxar. Maxar imagery, however, is not freely available across the full U.S. East Coast. Hence, we assembled another dataset of pre-flood image tiles from the open-access U.S.-wide mosaic of $1.0m/px$ visual aerial imagery from the 2019 National Agriculture Imagery Program (NAIP) [308]. The pre-flood NAIP image tiles are paired with open-access Maxar post-flood satellite imagery and a generated pixelwise flood segmentation mask. This creates a dataset of 6500 clean image-triplets that we are releasing as the flood-section of our open-source dataset to study segmentation guided image-to-image (im2im) translation in Earth observation.

The im2im translation task from NAIP aerial to Maxar satellite imagery is significantly more challenging than the Maxar→Maxar task, because the learned image-transformation needs to account for all sources of variance in differing remote sensing instruments. These are, for example, resolution, atmospheric noise, color calibration,

inclination angle, and more. Indeed, applying the physics-informed GAN from Section 4.3.2 without retraining generates unintelligible imagery in Fig. B-1-top. To reduce the learning task complexity, we removed the variation within Maxar data, via sourcing post-flood tiles from a single satellite pass over west Houston, TX on 8/31/2017, post-hurricane Harvey [198]. To re-run our pipeline, we labelled an additional 260 flood segmentation masks (in $\sim$ 2hrs), retrained the flood segmentation model, and re-trained the physics-informed GAN from scratch on the new dataset (for 15 epochs in $\sim$ 5hrs on $1 \times V100$ GPU). The resulting model did not outperform the baseline in physical-consistency, which is likely due to the suboptimal performance of this dataset's segmentation model (IoU=0.23). However, the resulting model still outperforms the baseline in photorealism (LPIPS=0.369 vs. 0.465) on a 20% test split. While these flood maps are not ready for production our research shows that im2im translation across remote sensing instruments is feasible, as well as, the potential to visualize coastal flooding along the full U.S. East Coast in follow-up work.

Generalization across climate phenomena – visualizing reforestation

Visualizing negative climate impacts can evoke anger, fear, or guilt which can encourage pro-environmental behavior [297, 282, 9], respectively. These emotions can, however, also feel overwhelming [225] and hope is needed to maintain environmental engagement [222]. Here we extend the *Earth Intelligence Engine* to visualize the impact of positive action, specifically, to encourage policymakers, carbon finance investors [256, 191], and landowners to implement reforestation.

To synthesize satellite imagery of reforested land, the *Earth Intelligence Engine* uses an image of a deforested area along with a binary mask of where trees will be planted as input. To train and evaluate the model we collected image-triplets of an RGB pre-reforestation image, a binary reforestation area mask (1=reforestation), and an RGB post-reforestation image, as illustrated in Fig. B-2. We assembled a total of 1026 train and 246 test high-resolution (50cm/px) 1024x1024 image-triplets via Google Earth Pro (Map data: Google, Maxar Technologies, CNS/Airbus). The dataset spans four different countries: Uruguay, Sierra Leone, Peru, and Mexico, as

Table 4.2: **Reforestation accuracy**. Our model quantitatively outperforms two baseline models that apply a color mask in a random and spatial split.

	LPIPS ↓ random split	LPIPS ↓ spatial split
GAN (ours)	**0.503**	**0.574**
Green mask (RGB=33,64,61)	0.794	0.848
Green mask (RGB=78,116,85)	0.845	0.957

detailed in Appendix B.1.

We trained the generative vision model that performed best on floods, pix2pixHD, using several augmentation techniques and the default [316] hyperparameters, as detailed in Appendix B.3.1. We evaluated the model on a random split and a spatial split – testing on 107 held-out images from Guatemala) using LPIPS loss and compare it to a baseline model that applies uniformly colored masks.

Figure 4-4d) shows how our model generates photorealistic visualizations of reforestation projects. The generated imagery looks more realistic than handcrafted baseline models (e,f), where the reforested area pixels are set to a mean forest color. Our quantitative analysis in Table 4.2 confirms that our model outperforms the baselines in both random and spatial split. Note however that LPIPS has limits as distance metric, because the ground-truth post-reforestation image often contained unknown features such as newly built houses.

Visualizing Arctic sea ice melt

The retreat of Arctic sea ice is one of the most important and imminent consequences of climate change [127]. However, visualizations of melting Arctic sea ice are limited to physics-based renderings, such as [209]. There also does not exist daily high-resolution (less than 500m) visual satellite imagery due to satellite revisit rate and cloud cover. To enable the extension of the *Earth Intelligence Engine* for visualizing Arctic sea ice melt, we publish a dataset of $\approx 20k$ image pairs of Winter image, Summer image, and ice segmentation mask, as detailed in Appendix B.1.

4.4 Discussion and Future Work

Limitations.

Although our pipeline outperformed all baselines in the generation of physically-consistent and photorealistic imagery of coastal floods, there are areas for improvement in future works. For example, our flood datasets only contained 3 or 6.5k samples and were biased towards vegetation-filled satellite imagery; this data limitation likely contributes to our model rendering human-built structures, such as streets and out-of-distribution skyscrapers in Fig. B-3 top-left, as smeared. Although we attempted to overcome our data limitations by using several state-of-the-art augmentation techniques, this work would benefit from more public sources of high-resolution satellite imagery (augmentation details in Appendix B.3). Apart from the data limitation, smeared features are still a current concern in state-of-the-art GAN architectures [275]. Furthermore, the computational intensity of training GANs made it difficult to optimize the models on new data. Improved transfer learning techniques or foundation models as mapped out in Section 6.2.2 could address this challenge. Lastly, satellite imagery is an internationally trusted source for analyses in deforestation, development, or military domains [117, 6]. With the increased capability of data-generating models, more work is needed in the identification of and the education around misinformation and ethical and trustworthy AI [271, 12]. We point out that our satellite imagery is synthetic, should only be used as a scientific communication aid to better explain our results to decision makers or the general public [277], and we take first steps towards guaranteeing trustworthiness in synthetic satellite imagery.

Cloud-penetrating satellite imagery.

Remote sensing commonly faces the problem of missing frames, due to cloud-cover, orbital alignment, or cost of high-resolution imagery [350, 311]. The *Earth Intelligence Engine* can be seen as a gap-filling model that combines the information from low-resolution flood maps and high-resolution pre-flood image mosaics to infer the missing high-resolution post-flood satellite imagery. For example after floods, the

arrival of the first visual images is often delayed until clouds pass or expensive drone surveys are conducted. Synthetic-aperture radar (SAR) is cloud-penetrating and often returns the first available medium-resolution flood maps (at $\sim$ 10m/px) [78]. The *Earth Intelligence Engine* could visualize the medium-resolution SAR-derived flood extent maps. However, future work will be necessary to extend the trustworthiness of generated flood visualizations in disaster response, for example, via incorporating information on the flood height, building damage, or the raw SAR signal. The current visualizations are aimed towards media or policy to communicate the possible extent of future floods in a compelling manner [277].

Vision for the future.

We envision a global visualization tool of climate impacts and adaptation techniques. By changing the input data, future work can visualize impacts of other well-modeled, climate-attributed events, including Arctic sea ice melt, hurricanes, wildfires, or droughts. Non-binary climate impacts, such as inundation height, or drought strength could be generated by replacing the binary flood mask with continuous model predictions. Opportunities are abundant for further work in visualizing our changing Earth. This work opens exciting possibilities in generating realistic imagery with the potential impact of improving climate mitigation and adaptation.

4.5 Methods

The proposed pipeline uses a deep generative vision model to synthesize post-event images from pre-event images and a focus-indicating segmentation mask. Specifically, we synthesize

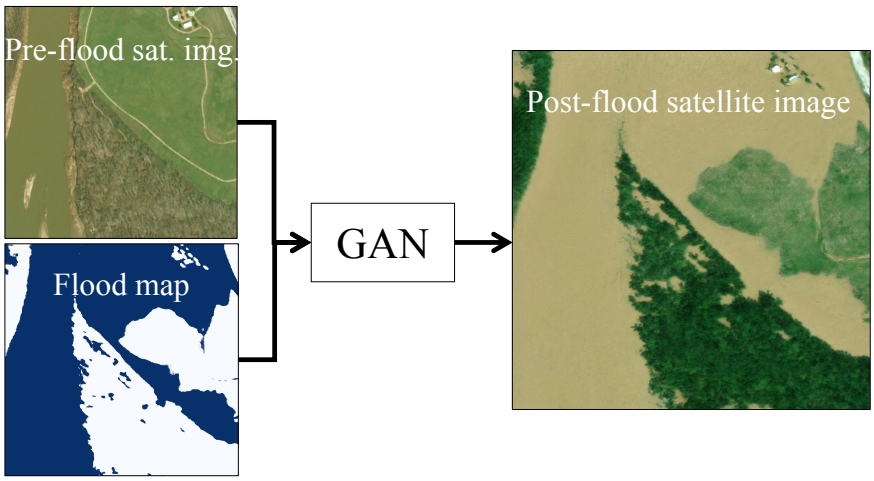
- post-flood images from pre-flood images and a flood extent map, as shown in Fig. 4-5, and
- post-reforestation images from deforested land images and a reforestation area map, as shown in Fig. B-2.

4.5.1 Data overview.

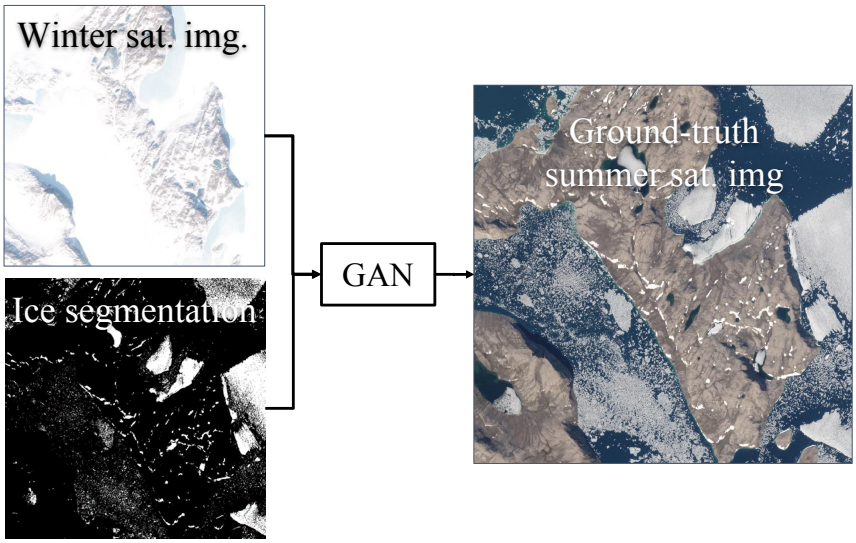
Flooding

Obtaining ground-truth post-flood images that display standing water is challenging due to cloud-cover, time of standing flood, satellite revisit rate, increased atmospheric noise, and cost of high-resolution imagery. This work leverages the xBD dataset [112], a collection of pre- and post-disaster images from events like Hurricane Harvey or Florence, from which we obtained ∼3k pre- and post-flood image pairs with the following characteristics: ∼.5m/px, RGB, 1024×1024px/img, Maxar DigitalGlobe.

The coastal flood model is the *Sea, Lake and Overland Surges from Hurricanes* (SLOSH) model [131], developed by the National Weather Service (NWS). SLOSH estimates storm surge heights from atmospheric pressure, hurricane size, forward speed, and track data, which are used as a wind model driving the storm surge. The SLOSH model consists of shallow water equations, which consider unique geographic locations, features, and geometries. The model is run in deterministic, probabilistic, and composite modes by various agencies for different purposes, including NOAA, National Hurricane Center (NHC) and NWS. We use outputs from the composite approach – that is, running the model several thousand times with hypothetical hurricanes under different storm conditions. As a result, we obtain a binary flood hazard map from [218] as displayed in Fig. 4-3a which are storm-surge, height-differentiated, flood extents at 30m/px resolution. The flood hazard maps do not intersect with the locations of existing post-flood imagery. To get around the data limitation, we generate and coarse-grain segmentation maps of the post-flood imagery to 30m/px for training and evaluation and use binarized flood hazard maps during test. Future works will extend the state-of-the-art *ADvanced CIRCulation* model (ADCIRC) [178] model, which is described in [93] and has a stronger physical foundation with better accuracy, and higher resolution than SLOSH.



(a) Model Architecture



(b) Extension to Arctic sea ice melt

Figure 4-5: **Top: Model Architecture.** Our model leverages the semantic image synthesis model, Pix2pixHD [316], and combines a pre-flood satellite image with a physics-based flood map to generate post-flood satellite imagery. **Bottom: Arctic sea ice melt.** We publish a dataset of 25,000 labeled image triplets for segmentation guided image-to-image translation which in includes an additional case study on melting Arctic sea ice.

4.5.2 Model architecture.

The central model of our pipeline is a generative vision model that learns the physically-conditioned image-to-image transformation from pre-flood image to post-flood image. We leveraged the existing implementation of pix2pixHD [316]. Pix2pixHD is a state-of-the-art semantic image synthesis model that uses multi-scale generator and discriminator architectures to generate high-resolution imagery. We extended the input dimensions to 1024×1024×4 to incorporate the focus area mask. The resulting pipeline is modular, such that it can be repurposed for visualizing other climate impacts.

4.5.3 Trust in flood images through physical-consistency.

We define a *physically-consistent* model as one that fulfills laws of physics, such as, conservation of momentum, mass, and energy [289]. For example, most coastal flood models consist of numerical solvers that resolve the conservation equations to generate flood extent predictions [131]. Here, we consider a flood image to be physically-consistent if it depicts the predictions of a physically-consistent model.

Specifically, we define our generated satellite imagery, $I_G \in \mathcal{I} = [0,1]^{w \times h \times c}$ with width, $w = 1024$, height, $h = 1024$, and number of channels, $c = 3$, to be physically-consistent if it depicts the same flood extent as the binary flood map, $F \in \mathcal{F} = \{0;1\}^{w \times h}$. We implemented a flood segmentation model, $m_{\text{seg}} : \mathcal{I} \rightarrow \mathcal{F}$, to measure the depicted flood extent in the generated image. If the flood extent of a generated image and the coastal flood model match within a margin, the image is in the set of physically-consistent images, i.e,. $I_{\text{phys}} \in \mathcal{I}_{\text{phys}} = \{I_G \in \mathcal{I} : ||m_{\text{seg}}(I_G) - F|| < \epsilon\}$. The generated image is considered photorealistic, if it is contained in the manifold of naturally possible satellite images, $I_{\text{photo}} \in \mathcal{I}_{\text{photo}} \subset \mathcal{I}$. Hence, we are looking for a conditional image generation function, g, that generates an image that is both, physically-consistent and photorealistic, i.e, $g : \mathcal{I}_{\text{photo}} \times \mathcal{F} \rightarrow \mathcal{I}_{\text{photo}} \cap \mathcal{I}_{\text{phys}}$. Here, we condition the GAN on the flood map, F, and use a custom evaluation function to identify the generation function, g.

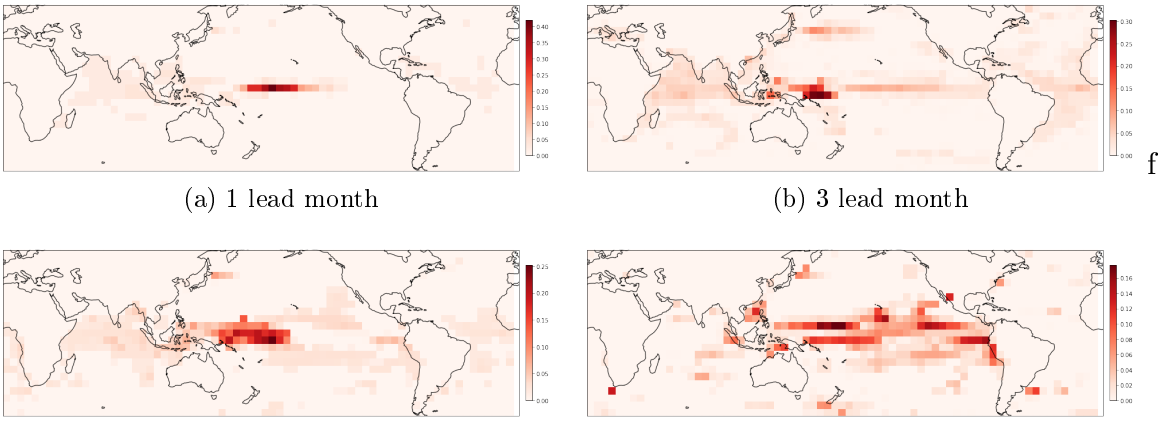
Chapter 5

Deep Learning-based Emulators in Downscaling and Impact Modeling

Chapter abstract: The previous chapters have described the main contributions in the dynamics and visualization section of the envisioned climate emulator. This section outlines various contributions in dynamics, downscaling, and impact modeling that I have contributed to during this thesis.

5.1 The World as a Graph: Improving El Niño Forecasts with Graph Neural Networks

Deep learning-based models have recently outperformed (sub-)seasonal forecasting models in terms of accuracy, such as for predicting El Niño-Southern Oscillation (ENSO) [113]. However, most deep learning-based ENSO forecasts are based on convolutional neural networks which can be difficult to interpret and are biased to learn small- instead of large-scale atmospheric patterns, as mentioned in Table 2.1. In comparison, graph neural networks (GNNs) can be biased to learn large-scale spatial dependencies and can be interpreted via the learned edge connections [15]. In this work, I contributed towards the first application of graph neural networks to seasonal forecasting. We designed a novel graph connectivity learning module that enabled



(a) 1 lead month (b) 3 lead month

(c) 6 lead month (d) 9 lead month

Figure 5-1: **The world as a graph.** This thesis contributed towards the first application of graph neural networks for improving ENSO forecasts in [50]. We interpreted the learned edge connections of a graph neural network via plotting the eigenvector centrality of each node as a heatmap. The heatmap visualizes the influence of each node onto the learned graph. As expected, nodes with the highest eigenvector centrality (dark red) for 1 lead months are in the ONI region (5N-5S, 120W-170W), shifting to the Western Pacific for 3-6 lead month, and become more global with 9 lead months. The full discussion is in [50]. Interpreting learned weights in deep learning-based emulator is a future direction for collecting evidence for climate theories.

a GNN model to learn large-scale spatial interactions jointly with the actual ENSO forecasting task. Our model, Graphiño, is more accurate than a state-of-the-art CNN-based model [113] for forecasts up to six months ahead. Additionally, we show that our model can be interpreted via the learned connectivity structures in Fig. 5-1 that correlate with the ENSO anomaly pattern. The full analysis is documented in [50].

5.2 WiSoSuper: Deep Generative Vision Models for Downscaling Wind and Solar Fields

The climate can vary substantially within the 10-100km output of a global climate model [334]. These variations are called microclimates and are prominently found in urban areas, forests, mountaineous areas, or water bodies. Downscaling or superresolution methods correct the local biases in large-scale climate projections [79]. To do so, downscaling methods map large-scale climate projections and fine-scale con-

text information onto fine-scale projections at 100m-1km [124]. The impact of deep learning on downscaling methods differs between the two main approaches: statistical downscaling and dynamical downscaling.

In statistical downscaling, statistical models are fit to interpolate large- and onto fine-scale projections. Default methods are mostly based on random forests, XGBoost, or polynomials that are fit to correct biases in local 3x3 stencils [124]. The learned local stencil is independent of space and applied equally to all areas of the image. Deep learning methods have a wider receptive field and more parameters than the default statistical methods. With this, deep learning methods could outperform current statistical downscaling approach in terms of accuracy.

Dynamical downscaling methods, in opposition run a dynamical regional weather/climate model to downscale the output of a global climate model [334]. This approach is often considered more accurate than statistical downscaling, but comes at higher computational cost and set-up time. Deep learning models could be trained to emulate the mapping in dynamical downscaling datasets from large- to fine-scale projections. With this, deep learning methods could significantly decrease computational cost for applying the downscaling approach to new regions or climate projections.

Within this thesis, I contributed to a study that evaluates deep generative vision models for statistical downscaling of weather forecasts [154]. We focused on wind and solar forecasts which are needed in the green energy transition for optimized siting and scheduling of renewable power plants [307]. Operational forecasts from numerical weather prediction models only have a spatial resolution of 10 to 20-km [294], which leads to sub-optimal usage and development of renewable energy farms [73]. Recently, deep learning-based models have outperformed traditional downscaling methods, for example, the physics-informed resolution-enhancing generative adversarial network (PhIREGAN) [288]. In this contribution, we provide a thorough and extensible benchmark of leading deep learning-based super-resolution techniques, including the enhanced super-resolution generative adversarial network (ESRGAN) [317] and an enhanced deep super-resolution (EDSR) network [168], on the wind and solar data in Fig. 5-2. We evaluate the accuracy on physics-informed

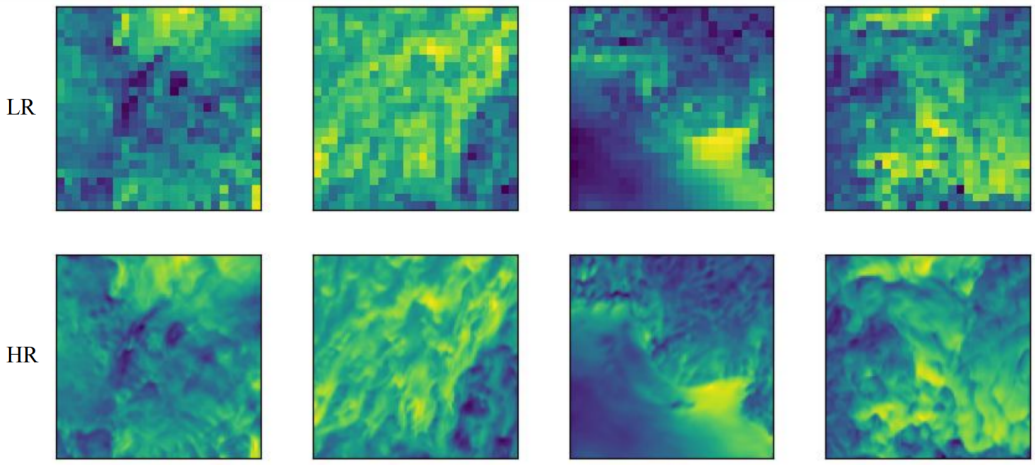


Figure 5-2: **This thesis contributes to WiSoSuper** in which we benchmarked various deep generative vision models on downscaling wind and solar data. Specifically, we trained PhiREGAN, EDSR, and ESRGAN to predict a high-resolution wind field (bottom) from an artificially generated (via bicubic upsampling) low-resolution image (top). We compare model performances on RMSE and physics-informed metrics.

metrics such as the wind kinetic energy spectrum and solar normalized variogram. Future work, could integrate the physics-informed metrics into the loss function, extend the benchmark to normalizing flows and diffusion-based models, or compile a benchmark dataset on real instead of synthetic data. The full analysis can be found in [154].

5.3 Digital Twin Earth – Coasts: Developing a Fast and Physics-informed Surrogate Model for Coastal Floods via Neural Operators

One example in impact modeling are models that predicting coastal flood risks. Rising sea levels are one of the most significant results of climate change, potentially threatening lives and damaging infrastructure in the coastal regions [150]. The accelerating rate of sea level rise will exacerbate coastal flooding, particularly under the increasing coastal populations and in some regions an increase in the severity of extreme storms [127, 88]. Physics-based numerical models, such as Nucleus for Eu-

ropean Modelling of the Ocean (NEMO) [194], have been developed to simulate and predict dynamics of sea surface height. These physical models – driven by wind speed and mean sea level atmospheric pressure – simulate the dynamics of water velocity and sea surface height by solving the mass and momentum conservation equations. Yet running these physics-based models can be extremely computationally expensive depending on the simulation time, the domain size and resolution, due to the need to numerically resolve multi-physics and multi-scale dynamics represented through coupled nonlinear equations in large spatial domains [240]. In particular, these complex simulators are not fast enough for reliable risk estimation, uncertainty quantification, or real-time predictions [240], and are replaced by models with physical approximations that sacrifice accuracy for computational efficiency [14].

Hybrid physics-ML models could potentially replace expensive forward simulations by statistical representations through regression [138]. But, training classical physics-informed neural networks is difficult due to the need to resolve the discretized PDE in the loss function [249]. Indeed, researchers found that these approaches are unable to represent dynamics of simple cases such as a one-dimensional two-phase flow model [99]. On the other hand, the recently proposed Fourier Neural Operator (FNO) [166] shows a promising alternative by learning the dynamics in the frequency domain. In doing so, FNO is not limited to one specific instance of a PDE but directly learns the solution operator of the PDE, which makes it resolution-variable in the physical space [166].

Here, I contributed towards the first "coastal digital twin," an emulator built on state-of-art physics-informed ML techniques to produce computationally lightweight surrogate models that provide fast and accurate predictions of sea surface heights in coastal regions. As a proof-of-concept experiment, we developed the emulator in Fig. 5-3 for NEMO simulations in northwestern Europe using an improved version of FNO and published the code at gitlab.com/frontierdevelopmentlab/fdl-2021-digital-twin-coasts/coastaltwin.

Our results show:

- a 45x acceleration achieved by FNO compared with NEMO simulation.

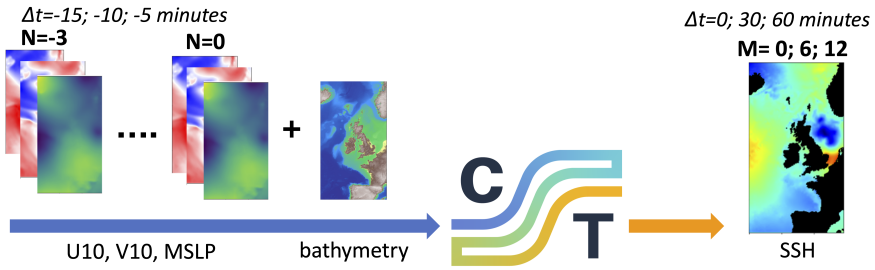


Figure 5-3: **Digital twin Earth – Coasts.** Within impact modeling, we applied a deep learning to emulate sea surface height (SSH) based on NEMO atmospheric forcings, including both the bathymetry profile and the dynamics of mean sea level pressure (MSLP), horizontal wind speed (U10), and vertical wind speed (V10).

- an overall improvement of FNO in comparison to a baseline UNet model [261] in emulating sea surface height,
- the adverse impact of non-periodic boundary conditions on FNO accuracy, and
- the extension of FNO to learn multivariate dynamics (note that FNO was used for univariate cases in its original development [166]),

The full analysis can be found in [134].

Chapter 6

Future Work

Section 3.5 and Section 4.4 have described limitations and next steps of the main thesis contributions. This section maps out concrete follow up work and larger existing research challenges.

6.1 Climate Communication: Integrating Climate Emulators in Policy Simulations

Climate change continues to be a politically polarizing issue [197]. Gratefully, most people in the US are aware and many even concerned about climate change [197]. But, converting awareness into action across all political and systemic levels remains one of the most fundamental societal challenges [128]. Climate policy simulations such as En-ROADS help create political consensus by increasing the sense of hope and urgency, as detailed in [264] and Section 2.2.2. This thesis has been written under the umbrella of an envisioned higher-resolution policy simulation, called *the climate pocket*. The climate pocket would map climate policy choices to locally relevant climate impacts.

This thesis contributed submodules towards the climate pocket: I synthesized a new visualization of floods as satellite imagery [188], contributed foundational advances towards fast hybrid physics-ML climate models [192], and contributed to works

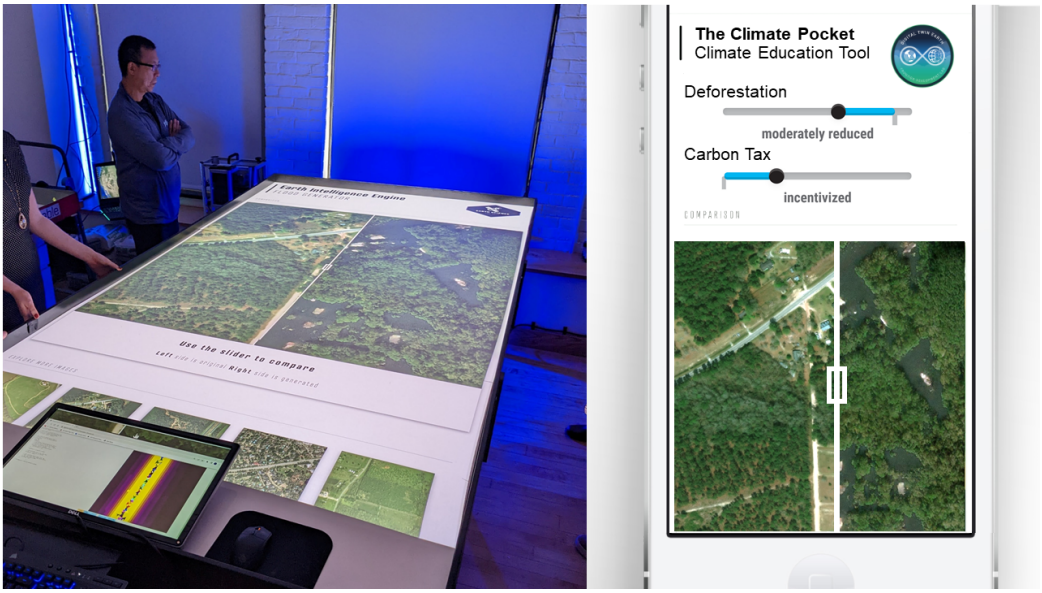


Figure 6-1: **The climate pocket** is an envisioned end-to-end climate emulator that maps emissions to local climate projections.

in downscaling [154] and impact models [134]. Integrating these advances into policy simulations will pose at least two projects: 1) an end-to-end emulator that maps emissions to local (100km) climate projections and 2) global visualizations of flood hazard maps.

6.1.1 The climate pocket: developing an end-to-end climate emulator from emissions to climate impacts

This subsection maps out next steps and speculative challenges towards creating the climate pocket, an end-to-end climate emulator from emissions to local climate projections in Fig. 6-1. The subsection is structured into data generation, model development and evaluation, and dissemination.

Data generation

During data generation we will have to take choices in: data sources, in- and output variables, data generating model, and spatiotemporal resolution.

- **Data sources:** We can choose between existing modeled data [2], modeling

new data [211], reanalysis [122] and observational data [215]. It most likely the easiest to start with existing modeled data and start modeling new data once existing data sources are insufficient. Fitting a deep learning model on only observatial data would likely overfit and not be trustworthy for future predictions; fitting a model on observational and modeled data would be more akin to climate data assimilation than climate emulation, but a worthy approach for data augmentation in (sub-)seasonal forecasts [50]. To get started with existing modeled data, I have co-authored a tutorial on how to download data of one model from the CMIP6 data archive at https://github.com/blutjens/cesm-emulator/blob/main/code/explore_cesm_data.ipynb.

- **In- and output variables:** The ESGF repository of CMIP6 data contains over 20PB of data and 300 distinct variables [87]. The decision which in- and output variables to download from this stack is a function of stakeholder relevance, data availability, and necessity for successful emulation. The inputs to the climate pocket should be the outputs of the climate energy policy models in En-ROADS which are global greenhouse gas emissions or concentrations over time [92]. It is likely preferable to work with emission data as the mapping from emissions to concentrations is better represented in CMIP6 climate models than En-ROADS. Choosing CO2 emission data over time would be a good start because there is more CO2 data than other greenhouse gases. For the first iteration, the climate pocket could output variables with high skill such as extreme temperatures [322]. Then, it can be extended towards variables with lower predictive skills such as extreme precipitation or variables from downstream climate impact models, such as flood, wildfire, compound drought, or hurricane risk models.

- **Data generating model:** The CMIP6 dataset contains approximately 100 climate models from different modeling centers [2]. Emulating a single model rather than the ensemble of models will allow for better interpretability of the emulator. This is because different climate models are known to have higher/lower accuracy in different regions or physical phenomena [322]. The

decision which model to pick will be a function of data availability, model skill, and accessibility of model documentation, among others. For example, NCAR CESM2 would be a suitable choice to start [211].

- **Spatiotemporal resolution:** The envisioned climate pocket should have spatial resolution $<= 100km$ resolution to allow for state-scale climate policy planning. The temporal resolution desired is unclear. On the one hand seasonally aggregated projections of extreme statistics in landmark years, e.g., 2025, -30, -40, and -50, would likely suffice for most stakeholders. On the other hand, the ground-truth simulation is run at $30min - 1hr$ resolution and it is unclear how much of this temporal accuracy is needed to accurately emulate statistics on landmark years.

Model development and evaluation

For model development, various temporal mappings are possible:

- **An Instantaneous mapping** from global greenhouse gas concentrations, $co2_{global-mean}(t) \in \mathbb{R}$, to local surface temperatures, $tas_{0:Lat,0:Lon}(t) \in \mathbb{R}^{(Lat,Lon)}$, would capture annual to decadal trends, but average over variations of shorter time-scale.

- **A myopic autoregressive mapping** from emissions and state at time, t, $[co2_{global-mean}(t), tas_{0:Lat,0:Lon}(t)] \in \mathbb{R} \times \mathbb{R}^{(Lat,Lon)}$, to the next steps state, $tas_{0:Lat,0:Lon}(t+1) \in \mathbb{R}^{(Lat,Lon)}$, would take a Markovian assumption. With the Markovian assumption this method would require that the state contains more than, tas, and rather captures all inputs necessary to uniquely describe the current state of the climate, i.e., temperatures, pressure, humidities, etc. However, describing the full state of the climate in a state vector description is likely infeasible.

- **An autoregressive mapping** with memory from emissions and states from previous time steps, $[co2_{global-mean}(t-t_h:t), tas_{0:Lat,0:Lon}(t-t_h:t)] \in \mathbb{R}^{(t_h)} \times$

$\mathbb{R}^{(Lat,Lon,t_h)}$, to the next time step, $tas_{0:Lat,0:Lon}(t+1) \in \mathbb{R}^{(Lat,Lon)}$, could overcome some issues of explicitly encoding a state vector. The memory or latent state in this autoregressive model would be expected to encode the current state of the climate.

- **A functional mapping** from all time steps of emissions, $co2_{global-mean}(t_0 : T) \in \mathbb{R}^{(T)}$, to all time steps of climate, $tas_{0:Lat,0:Lon}(0 : T) \in \mathbb{R}^{(Lat,Lon,T)}$ is also possible. This set-up would encode all temporal information, but would be computationally very demanding especially with a higher-dimensional climate state than surface temperatures. This set-up is also counterintuitive, because the climate evolves forward in time and not backward.

A subset of these temporal mappings is reviewed in [300]. Regardless, identifying the best temporal mapping will be an aspect of deeper theoretical and empirical study.

The deep learning architecture would likely be chosen to learn both temporal and spatial correlations. We summarized the choice for spatial architectures in Section 2.3.2. For capturing temporal relations, one can consider gated recurrent units (GRUs), recurrent neural networks (RNNs), long-short term memory networks (LSTMs), transformers, or state-of-the-art architectures such as FourCastNet [236] or MetNet-2 [89]. The models will likely not work out of the box on the given climate data. A successful deep learning architecture will need to answer questions on how to conserve energy over time, how to remain computationally tractable, how to reduce training data requirements for large-scale patterns, how to embed assumptions on multiscale spatiotemporal patterns, and others.

The evaluation of the model would combine metrics from machine learning, such as root mean square error (RMSE) or anomaly correlation coefficient (ACC) over time, with metrics from climate science, such as the existence of a double intertropical convergence zone (ITCZ) [252] and skill [322]. The evaluation also needs to take extreme event statistics into account, e.g., by measuring accuracy of Gaussian or binned probability distributions of predicted daily maximum temperature. A fair evaluation will baseline the deep learning-based emulator against non-neural network approaches,

such as, physical simplifications [92] or linearizing Green's function approach [94].

Dissemination

Lastly to disseminate the emulator researchers would need to collaborate with existing NGOs, non-profit, start-ups, or industry. If distributed by a non-profit such as, Climate Interactive, the emulator could reach beyond 200.000 people and help decision makers in industry and policy understand pathways for climate mitigation and adaptation [263]. To engage machine learning researchers in climate emulation, I have co-started to develop a tutorial that introduces DL in climate emulation github.com/blutjens/climate-emulator-tutorial.

6.1.2 Creating global visualizations of flood hazard maps

Data collection

The generated flood visualizations in Chapter 4 visualize floods only in Houston, TX and do not generalize to other regions. Scaling the model to a global scale will require the following data products: (1) a diverse set of high-resolution aerial imagery of past floods with (2) associated pre-flood aerial imagery to train a global model and (3) a global flood hazard map of all areas of interest with (4) open-source global high-resolution aerial imagery of all areas of interest.

Assembling (1) is likely the largest challenge: the Maxar Open data program publishes $\approx 1 m/px$ satellite imagery after major global flood events for the past ≈ 10 years [80]. However, there is no open-source GUI to explore the imagery, no open-source API to download the imagery, most areas are cloud-covered, and cloud masks are not published. Alternatively, the NOAA National Geodetic Survey Emergency Response Imagery Database provides aerial $\approx 0.5 m/px$ post-flood imagery after major US flood events since 2003 [210]. The NOAA imagery is mostly cloud-free, explorable via a map viewer, and higher resolution, but only available for the US. Developing a global visualization of floods will likely use a combination of both datasources and possibly additional datasets. Finding visibly flooded areas has been an arduous

manual task and only $< 0.1\%$ of all pixels are flooded. An active learning process with a deep learning-based flood segmentation model might be very helpful to speed-up this task. It would also be very helpful if a satellite company, such as Planet or Maxar, publishes a set of high-resolution flood imagery that contain ground-truth flood segmentation maps.

For (3), Climate Central's flood hazard map at 10m/px is the highest resolution openly accessible flood map [292]. For (2) and (4) OpenStreetMap imagery is likely the best resource although selected countries offer higher-resolution aerial imagery, e.g., USA with NAIP at $1m/px$ [308] or Portugal with SNIG $0.25m/px$ [1].

Model development and evaluation

After data collection, a segmentation model will need to be trained to create flood segmentation masks for every post-flood image. Next, a deep generative vision model will need to be trained to predict post-flood from pre-flood image and flood segmentation mask. Running inference on the global satellite mosaic and tiled flood map will still be a significant research and engineering challenge. This is mostly due to the computational cost, but it also challenging to create a homogeneous map because all generated tiles will have slightly different color, contrast, and resolution [96]. As intermediate step, it will be easier to generated flood maps at select global locations, including flood-prone areas such as Pakistan; Puerto Rico; Coimbra, Portugal and regions of large economic impact, including Shanghai, Shenzhen, the Ganges delta, and New Orleans [292].

The segmentation and generative model will need to be evaluated and trained to reduce regional biases of (in-)accuracy [151]. Further, the generated imagery would be evaluted in terms of (1) physical-consistency via comparing the intersection over union (IoU) of the generated and ground-truth flood extent, (2) perceptual quality via PSNR, LPIPS [343], and (3) human-centered evaluation metrics. Human-centered metrics include subjective evaluation of perceptual quality [345], self-report measures of flood-related beliefs and attitudes before and after seeing the generated imagery [264], and behavioral measures to taking action [313].

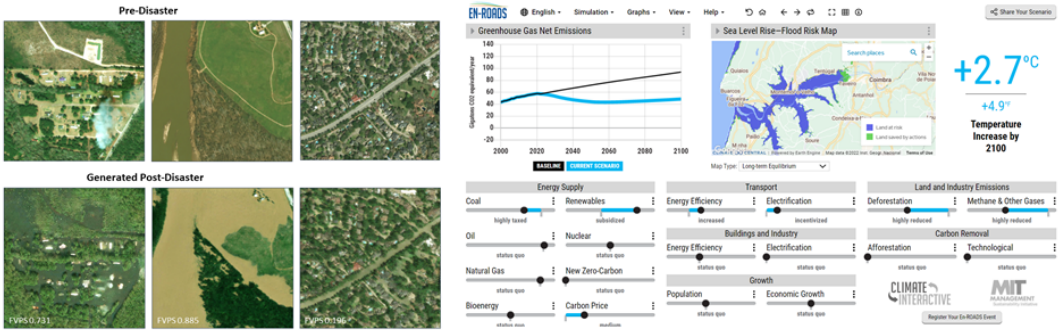


Figure 6-2: **Integrating flood visualizations.** Future work could extend the contributed flood visualizations (left) to global visualization layers and integrate them in existing policy simulations, such as the En-ROADS model (right).

Dissemination

After having generated a single global flood visualization layer, the next step, would be to generate multiple global maps that match multiple CO2 emission scenarios. These maps would enable the visual exploration of local flood impacts as a result of multiple climate scenarios. The model behind En-ROADS, called C-ROADS, already projects global mean temperature and sea level rise among others [92]. Recently, the projected global sea level rise of C-ROADS was linked with Climate Central's 10m resolution flood hazard maps Fig. 6-2. Generating a layer of visual satellite imagery for every flood hazard map that is currently used in En-ROADS would allow easy integration. Finaly, the integrated visualizations would enable a more engaging communication of city-wide flood risks to community-discussion groups, governmental offices, or the general public, especially if the decision-exploration tool is supported by experts in flood risk communication, such as Climate Central.

6.2 Deep Learning in Earth System Modeling

6.2.1 DailyMelt: diffusion-based models for spatiotemporal downscaling of (Ant-)arctic surface meltwater maps

Successful adaptation to the increasing climate and extreme weather risks requires high-resolution (HR) forecasts. As running HR models is expensive, models are run

at low-resolution (LR) and later downscaled, interpolated, or superresolved onto HR grids, as mentioned in Section 1.2. Recently generative adversarial networks (GANs) that we benchmarked in [154] have outperformed current statistical and dynamical downscaling methods in accuracy, and runtime, respectively. Most GANs, however, suffer from mode collapse [286] and do not account for the ill-posed nature of the problem: one LR input corresponds to many HR scenarios [179]. Normalizing flows and diffusion-based methods account for this problem by modeling the full distribution and dramatically outperform GANs on superresolution tasks, but normalizing flows are expensive to train [229]. Here, I contribute to preliminary work in applying diffusion-based methods for downscaling data in the Earth sciences [193].

We picked the application of downscaling (Ant-)arctic surface meltwater. This is important because ice melting in Greenland and Antarctica has increasingly contributed to rising sea levels. Yet, the exact speed of melting, existence of abrupt tipping points, and in-detail links to climate change remain uncertain. Ice shelves essentially prevent the ice sheet from slipping into the ocean and better prediction of collapses is needed. Meltwater at the surface of ice shelves indicates ice shelf collapse through destabilizing ice shelves via fracturing and flexural processes [10] and is likely impacted by a warming climate [148]. Maps of meltwater have been created from in-situ and remote observations, but their low and irregular spatiotemporal resolution severely limits studies [149].

In particular, there does not exist daily high-resolution ($<$ 500m) maps of surface meltwater. We propose the first daily high-resolution surface meltwater maps by developing a deep learning-based downscaling method, called DailyMelt [193], that fuses observations and simulations of varying spatiotemporal resolution, as illustrated in Fig. 6-3. The created maps will improve understanding of the origin, transport, and controlling physical processes of surface meltwater. Moreover, they will act as a unified source to improve sea level rise and meltwater predictions in climate models.

To synthesize surface meltwater maps, we leverage observations from satellites (MODIS, Sen-1 SAR) which are high-resolution (500m, 10m), but have substantial temporal gaps due to repeat time and cloud coverage. We fuse them with simulations

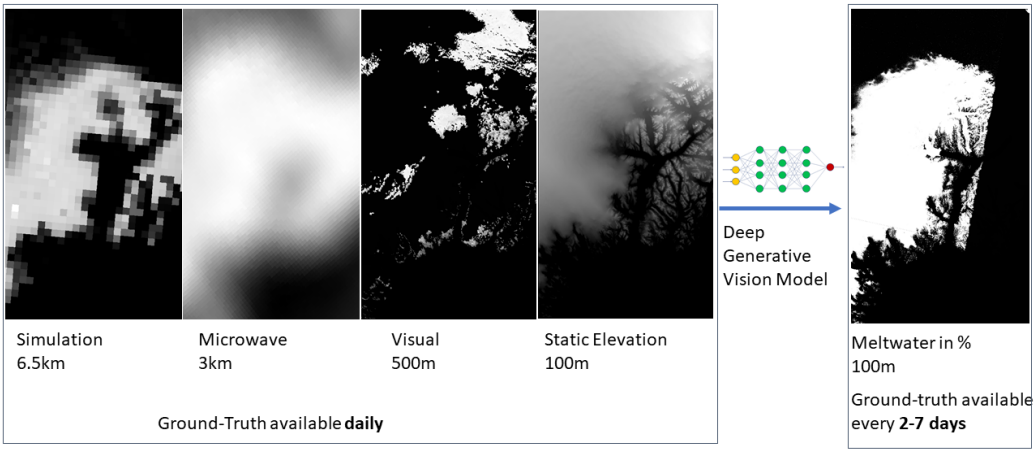


Figure 6-3: **Diffusion-based models** could increase the accuracy of spatial downscaling with deep generative vision models. I contributed to formulate the downscaling of (Ant-)arctic surface meltwater maps as a deep generative vision problem [193]. We have collected the depicted dataset over the Helheim Glacier, Greenland and will benchmark diffusion-based downscaling models on it.

(MAR) and passive microwave observations (MEaSURE) that are daily, but low-resolution (6km, 3.125km). In a significant remote sensing effort, we have downloaded, reprojected, and regridded all products into daily observations for our study area over Greenland's Helheim glacier as displayed in Fig. 6-3.

Within deep generative vision models, diffusion-based models promise sharp and probabilistic predictions [268]. We have implemented SRDiff [163] and tested it on spatially downscaling the wisosuper wind data [154]. As a baseline model, we will implement a statistical downscaling model that is a local hybrid physics-linear regression model [219]. In our planned benchmark, we expect a baseline UNet architecture that minimizes RMSE to create blurry maps and a generative adversarial network that minimizes adversarial loss to create sharp but deterministic maps. We have started with spatial downscaling and will later include temporal downscaling.

In summary, we hope to create the first daily high-resolution (500m) surface meltwater maps, introduce the first diffusion-based model for downscaling Earth sciences data, and create the first benchmark dataset for downscaling surface meltwater maps.

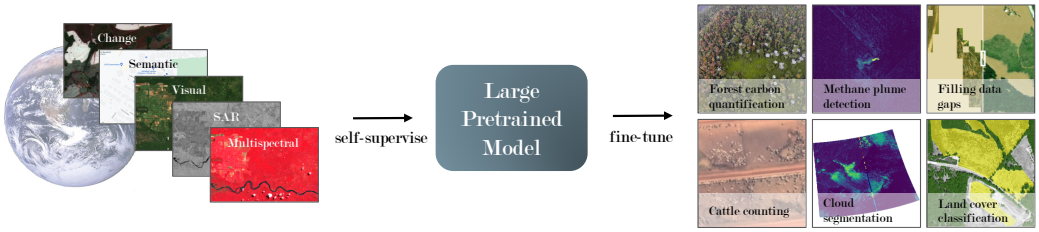


Figure 6-4: **Open-source foundation models in Earth monitoring** could reduce training data requirements and increase accuracy across many tasks.

6.2.2 Foundation models for reducing data requirements

Chapter 4 concluded that many deep learning applications in remote sensing and Earth monitoring are limited by the dataset size. Similarly, training climate emulators requires large-scale datasets from high-resolution simulations which are computationally very expensive to generate [319, 98]. There does not yet exist a solution to circumvent the cost of generating a high-resolution ground-truth dataset [129]. Various ideas exist in transfer learning, active learning, or reducing learning complexity in Chapter 3. Here, I contribute to a proposal of another way which is reducing data requirements via fine-tuning foundation models.

Recent advances in self-supervised learning have shown that pre-training large neural networks on large amounts of unlabeled data can lead to significant improvements in performance on downstream tasks [63]. These networks, which have been called foundation models [32], have transformed natural language processing (NLP) with BERT [77] and GPT-3 [46] and computer vision with CLIP [246] and DINO [55]. However, their applicability to remote sensing and tasks in Earth system modeling has been limited.

To address the gap in remote sensing, I have contributed to a benchmark for foundation models in remote sensing that comprises six classification and six segmentation tasks [156]. The Earth monitoring tasks were curated to be relevant for climate change and accompanied with easily accessible code and a robust evaluation protocol. I believe that open benchmarks like this will encourage the development of publicly available foundation models that can boost progress in Earth monitoring.

In climate emulation, only exploratory research in foundation models exists to-

day [213, 265]. This is likely due to the novel challenges that climate emulation poses to foundation models: datasets exceed petabytes, for example, CMIP6 is 20PB while the dataset in ChatGPT is only 570GB [46]. The plethora of variables, dynamics, temporal and spatial scales that could be modeled within Earth system modeling is hard to fathom and training one model that works across all tasks exceeds the scale of current foundation models by far. It is also unclear which self-supervised learning method, e.g., via masking, next-step predictions, constrastive learning, etc. will perform best on climate data.

6.2.3 Establishing long-term stability and physical consistency in ML-based emulators

Current spatiotemporal deep learning models, such as RNNs, CNN-LSTMs or transformers, forecast short - to medium term time-series (10s-100s of steps), which has enabled emulators in nowcasting [89] or weather modeling [236]. However, successful emulation in climate modeling requires statistics after millions of timesteps. Explicit integration of short time steps is the default, but likely not the best way to generate statistics of long-term time series. Further, successful methods will need to preserve physical-consistency [129]. Deep learning models that predict statistics of long-term spatiotemporal sequences are needed and could be developed by drawing parallels between deep learning, structure preserving integrators, stability theory, and multiscale physics [237]. Further, stability guarantees will need to be developed by working on equations of increasing complexity—from multiscale Lorenz96 over quasi-geostrophic turbulence to coupled climate simulations.

6.3 Establishing Cross-scientific Community and Collaborations

Climate change is a challenge across all disciplines and communities [128]. My research has focused on technical contributions in academia, but within this thesis had

to bridge fields in Earth system modeling, deep learning, numerical methods, and climate policy, which has only been possible through collaborations.

Machine learning and climate science research happens mostly in distinct communities. To incentivize collaboration we need journals, workshops, academic positions, grants, and large-scale research collaborations that focus on interdisciplinary contributions. As part of my PhD work I have contributed to bridging the communities by organizing the NSF Machine Learning for Polar Regions workshop and teaching at the Caltech Computer Vision for Ecology Summer Workshop [71]. Beyond workshops, there are two new journals in Environmental Data Science and AI for the Earth Systems, but ML conferences also need to be more receptive to interdisciplinary work. Beyond academica, collaborations between tech companies, such as Deepmind or NVIDIA, with governmental institutions seems fruitful to combine expertise in ML operations, server infrastructure, and research with expertise in the distribution of trustworthy weather and climate information to industry, politics, and the general public.

As climate change is a global challenge we need stronger incentives, agendas, and communities to work together.

Appendix A

Appendix: Multiscale Neural Operator

A.1 Choosing a loss function

Similar to problems in superresolution [179], there exist multiple realizations of the learned commutation error, $[\mathcal{G}*, \mathcal{N}](\bar{u})$, for a given ground-truth, $[\mathcal{G}*, \mathcal{N}](u)$. Using MSE will learn a smooth average and future work will explore adversarial losses [105] or an intersection between neural operators and normalizing flows [258] or diffusion-based models [285] to account for the stochasticity [324].

Using a physics-informed loss [249] would be another complementary extension to MNO. But, MNO does not require the large-scale solver to be autodifferentiable which significantly simplifies the implementation in large preexisting models in, e.g., climate. Any changes in the loss function should have this in mind.

Lastly, our implementation of MNO uses an a priori loss function and could likely be improved by implementing an a posteriori loss functions, i.e., a loss functions that propagates the loss over multiple time steps similar. Depending on the implementation an a posteriori loss function requires an autodifferentiable solver [95] or does not [37].

A.2 Quasi-geostrophic turbulence

We demonstrated multiscale neural operator on a high-dimensional system, specifically the one-layer quasi-geostrophic (QG) turbulence depicted in Fig. 3-1b. QG turbulence is a derivative of the Navier-Stokes equations and a good model for atmospheric turbulence at the equator. The equations are derived from the incompressible, i.e., $\nabla \cdot \boldsymbol{u} = 0$, Navier-Stokes equations by 1) taking the curl of velocity field, $w = \nabla \times \boldsymbol{u}$, and 2) assuming the beta-plane approximation, $f = f_0 + \beta y$, and hydrostatic, $\frac{\partial p}{\partial z} = -\rho g$, and geostrophic, $fv = \frac{1}{\rho}\frac{\partial p}{\partial x}$; $fu = -\frac{1}{\rho}\frac{\partial p}{\partial y}$, balances [196]. The resulting equations, called quasi-geostrophic turbulence, are given by:

$$\begin{aligned} \partial_t \omega + J(\psi, \omega) &= \nu \nabla^2 \omega - \mu\omega - \beta \partial_x \psi + F \\ \omega &= \nabla^2 \psi \end{aligned} \tag{A.1}$$

where ω is the vorticity, $\boldsymbol{u} = [u, v]^T = [-\partial_y \psi, \partial_x \psi]^T$ is the velocity vector, ψ is the streamfunction, $J(\psi, \omega) = \partial_x \psi \partial_y \omega - \partial_y \psi \partial_x \omega$ is the nonlinear Jacobian operator. Further, the parameters are the turbulent viscosity, ν, linear drag coefficient, μ, Rossby parameter, β, and varying source term, F. Vorticity can be computed with $\omega = \hat{\boldsymbol{z}} \cdot \nabla \times \boldsymbol{u} = \partial_x v - \partial_y u$

Filtering the equation with a kernel results in the parametrized large-scale equation given by [95]:

$$\partial_t \bar{\omega} + J(\bar{\psi}, \bar{\omega}) = \nu \nabla^2 \bar{\omega} - \mu\bar{\omega} - \beta \partial_x \bar{\psi} + \bar{F} + \underbrace{J(\bar{\psi}, \bar{\omega}) - \overline{J(\psi, \omega)}}_{\text{Parametrization: } h(\psi,\omega)} \tag{A.2}$$

where we used a cutoff kernel in spectral space: $G_\delta(k) = 0, \forall k > \pi \Delta \bar{x}^{-1}$. Cutting of high-frequencies likely generates the high-frequency patterns in the target parametrization, illustrated in Fig. 3-1b.

We then aim to approximate the subgrid-scale (SGS) parametrization, $\mathcal{K}_\theta \approx h$, with the neural operator.

The QG turbulence equation is solved with a pseudospectral solver in space and RK4 explicit time integration. We choose the parameters, $N_x = N_y = 512$, $\Delta t = 480$s,

$\mu = 1.25 \times 10^{-8}\text{s}^{-1}$, $\nu = 352\text{m}^2/\text{s}$, $\beta = 0$, and Reynolds number, $\text{Re} = 22 \times 10^4$. The variables are non-dimensionalized with $T_d = 1.2 \times 10^6 s$, i.e., $\Delta t_{\text{solver}} = \Delta t / T_d$ and $L_d = 504 \times 10^4 / \pi m$, i.e., $\Delta x_{\text{solver}} = 2\pi / N_x$. The reduced system is run with scale $\delta = 4$, such that $\bar{N}_x = \bar{N}_y = 128$. The forcing initiates turbulent mixing and simulates slowly varying wind stress according to the solution of, $F = C_f(t)[\cos(4y + \pi \sin(1.4t)) - \cos(4x + \pi \sin(1.5t))]$, and $0.5||F||_2 = 3$ with enstrophy injection rate, $C_F(t)$.

To generate turbulent chaotic dynamics that are decoupled from the initial state, the simulation is initialized with some large-scale Fourier states and warmed up for 10000 iterations on the high-resolution grid. After warm-up we generate 80000 steps on the high-resolution (HR) grid – totaling 20000 low-resolution iterations. We store a coarse-grained version of the HR vorticity as training inputs and coarse-grained ground-truth parametrizations, $h(\psi, \omega) = J(\bar{\psi}, \bar{\omega}) - \overline{J(\psi, \omega)}$, as training targets leveraging the useful code from [95]. During training, the vorticity and parametrization are treated as 1-channel images. The integration is in spectral space, but we store data in real space – future work extends MNO to a (pseudo-)spectral solver with reduced mappings to real space.

We split the data into train, validation, and test set by sampling continuous snippets with length 20 iterations randomly without replacement. We found that testing extrapolation to future data or splitting the data into snippets that are not auto-correlated in time requires a significantly larger dataset which was computationally prohibitive for this study. Using our current approach the test data is within the training distribution yet individual samples are distinct from training samples.

A.3 Multiscale Lorenz96

A.3.1 Detailed equations on MNO for multiscale Lorenz96

The full set of multiscale Lorenz96 equations write:

$$\frac{\partial X_k}{\partial t} = \underbrace{X_{k-1}(X_{k+1}-X_{k-2})-X_k+F}_{\text{Large-scale Dyn.: } \frac{\partial \bar{X}_k}{\partial t}} \underbrace{-\frac{h_s c}{b}\sum_{j=0}^{J-1} Y_{j,k}(X_k)}_{\text{Parametrization: } h}, \tag{A.3}$$
$$\frac{\partial Y_{j,k}}{\partial t} = -cbY_{j+1,k}(Y_{j+2,k}-Y_{j-1,k})-cY_{j,k}+\frac{h_s c}{b}X_k.$$

With the multiscale framework from Section 3.3.1, we define:

$$u(x) = [X_0, Y_{0,0}, Y_{1,0}, ..., Y_{J,0}, X_1, Y_{0,1}, ... \\ , X_K, ..., Y_{J,K}]_x \ \forall x{\in}D_x{=}\{0, ..., K(J+1)\}$$
$$\mathcal{N}(u)(x) = \begin{cases} \frac{\partial X_k}{\partial t} & \text{if } x{=}k(J{+}1) \ \forall k{\in}\{0, \dots, K\} \\ \frac{\partial Y_{j,k}}{\partial t} & \text{otherwise,} \end{cases}$$
$$G(x, x') = \begin{cases} 1 \text{ if } x' = k(J+1) \ \forall k \in \{0, \dots, K\} \\ 0 \text{ otherwise,} \end{cases}$$

with the solution, u, operator, $\mathcal{N}$, and kernel, G.

A.3.2 Details and Interpretation

The equation contains K variables, $X_k \in \mathbb{R}$, and JK small-scale variables, $Y_{j,k} \in \mathbb{R}$ that represent large-scale or small-scale atmospheric dynamics such as the movement of storms or formation of clouds, respectively. At every time-step each large-scale variable, X_k, influences and is influenced by J small-scale variables, $Y_{0:J,k}$. The coupling could be interpreted as X_k causing static instability and $Y_{j,k}$ causing drag from turbulence or latent heat fluxes from cloud formation. The indexes k, j are both interpreted as latitude, while $k \in \{0, ..., K{-}1\}$ indexes boxes of latitude and $j \in \{0, ..., J{-}1\}$ indexes elements inside the box. Illustrated on a 1D Earth with a circumference of 360° that is discretized with $K = 36, J = 10$, one a spatial step in k, j would equal $10°, 1°$, respectively [175]; we choose $K = J = 4$. A time step with $\Delta t = 0.005$ would equal 36 minutes [175].

We choose a large forcing, $F > 10$, for which the equation becomes chaotic. The

last terms in each equation capture the interaction between small- and large-scale, $f_{x,k} = -\frac{hc}{b}\sum_{j=0}^{J} Y_{j,k}(X_k), f_y$. The scale interaction is defined by the parameters where $h = 0.5$ is the coupling strength between spatial scales (with no coupling if h would be zero), $b = 10$ is the relative magnitude, and $c = 8$ the evolution speed of $X - Y$. The linear, $-X_k$, and quadratic terms, X_*^2, model dissipative and advective (e.g., moving) dynamics, respectively.

The equation assumes perfect "scale separation" which means that small-scale variables of different grid boxes, k, are independent of each other at a given time step, $Y_{j_1,k_2}(t) \perp Y_{j_2,k_1}(t)\ \forall t, j_1, j_2, k_1 \neq k_2$. The separation of small- and large-scale variables can be along the same or different domain and the discretized variables would then be $y \in [0, \Delta x]$ or $y \in [y_0, y_{\text{end}}]$, respectively. The equation wraps around the full large- or small-scale domain by using periodic boundaries, $X_{-k}{:=}X_{K-k}$, $X_{K+k}{:=}X_k$, $Y_{-j,k}{:=}Y_{J-j,k}$, $Y_{J+j,k}{:=}Y_{j,k}$. Note that having periodic boundary conditions in the small-scale domain allows for superparametrization, i.e., independent simulation of the small-scale dynamics [52] and differs from the three-tier Lorenz96 where variables at the borders of the small-scale domain depend on small-scale variables of the neighboring k [299].

A.3.3 Simulation

The initial conditions are sampled uniformly from a set of integers, $X(t_0) \sim U(-5, -4, ..., 5, 6)$, as a mean-zero unit-variance Gaussian $Y(t_0) \sim \mathcal{N}(0, 1)$, and lower scale Gaussian $Z(t_0) \sim 0.05\mathcal{N}(0, 1)$. The train and test set contains 4k and 1k samples, respectively. Each sample is $T = 1$ model time unit (MTU) or 200 ($=T/\Delta t$) time-steps long, which corresponds to 5 Earth days ($= T/\Delta t * 36$min with $\Delta t = 0.005$) [175]. Hence, our results test the generalization towards different initial conditions, but not robustness to extrapolation or different choices of parameters, c, b, h, F. The sampling starts after $T = 10$. warm-up time. The dataset uses double precision.

We solve the equation by fourth order Runge-Kutta in time with step size $\Delta t = 0.005$, similar to [176]. For a PDE that is discretized with fixed time step, Δt, the ground-truth train and test data, $h_{x,0:K}(t)$, is constructed by integrating the coupled

large- and small-scale dynamics.

Note, that the neural operator only takes in the current state of the large-scale dynamics. Hence, , i.e., it uses the full large-scale spatial domain as input, which exploits spatial correlations and learns parametrizations that are independent of the large-scale spatial discretization.

Our method can be queried for infinite time-steps into the future as it does not use time as input.

We are incorporating the prior knowledge from physics by calculating the large-scale dynamics, $dX_{LS,0:K}$. Note that the small-scale physics do not need to be known. Hence, MNO could be applied to any fixed time-step dataset for which an approximate model is known.

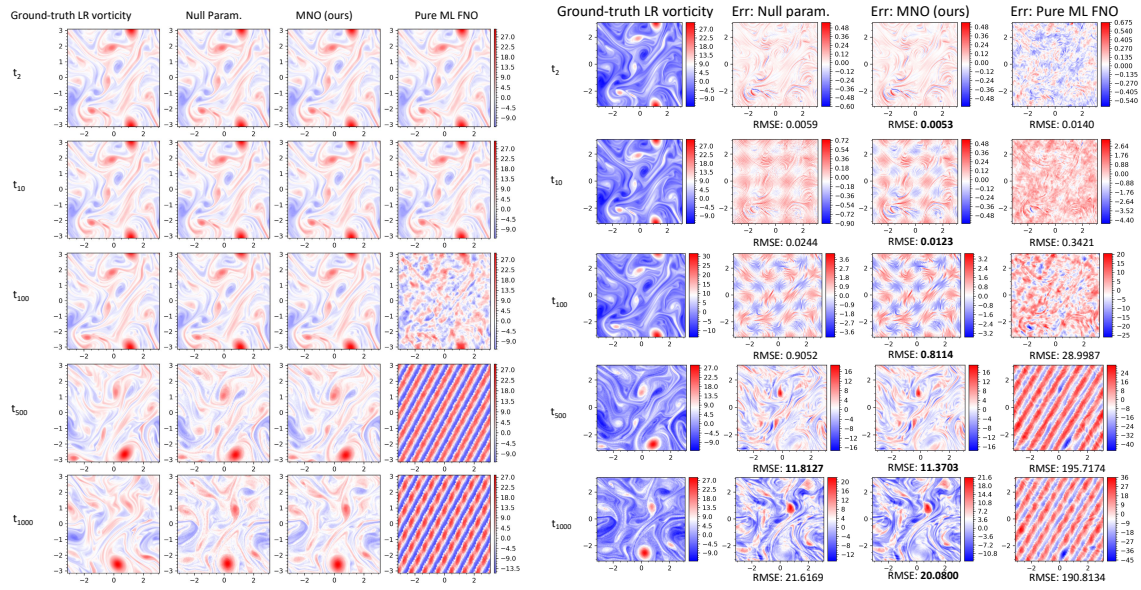
A.4 Appendix to Illustration of MNO via multiscale Lorenz96

The other large-scale (LS) and fine-scale (FS) terms are

$$\begin{aligned}
&\text{filtered FS dynamics, } \overline{\mathcal{N}(u)}(x) = \begin{cases} \frac{\partial X_k}{\partial t} & \text{if } x = k(J+1)\ \forall k \in \{0,\dots,K\} \\ 0 & \text{otherwise} \end{cases} \\
&\text{LS dynamics, } \mathcal{N}(\bar{u})(x) = \begin{cases} \frac{\partial \bar{X}_k}{\partial t} & \text{if } x = k(J+1)\ \forall k \in \{0,\dots,K\} \\ 0 & \text{otherwise} \end{cases} \\
&\text{with abbreviation, } \frac{\partial \bar{X}_k}{\partial t} := X_{k-1}(X_{k+1} - X_{k-2}) - X_k + F \\
&\text{LS state, } \bar{u}(x) = \mathcal{G} * u(x) = [X_0, 0, ..., 0, X_1, 0, ..., X_K]
\end{aligned} \tag{A.4}$$

A.5 Appendix to Results

A.5.1 Quasi-geostrophic turbulence



(a) Solution (b) Error

Figure A-1: **Autoregressive forecast of quasi-geostrophic turbulence over time. Left:** The hybrid models, MNO and Null Param., follow the coarse-grained high-resolution solution, titled Ground-truth LR vorticity, more closely than a pure machine learning (ML) model. **Right:** Calculating the error of predicted - ground-truth solution and RMSE shows that MNO is significantly more accurate than a pure ML model, but only slightly more accurate than a null parametrization.

Accuracy in autoregressive forecasts. We now compare the accuracy of the learned models during autoregressive forecasting. To do so, we use the last state of the training dataset as initial state and integrate all models autoregressively over time. Autoregressively meaning that each model updates the internal state with its' forecasted state.

Figure A-1a shows that an autoregressive forecast with the hybrid models, Null Param. and MNO, closely follow the ground-truth coarse-grained vorticity from a high-resolution simulation (left). In comparison, a pure machine learning (ML)-based model using Fourier neural operators diverges from the ground truth after ≈ 100 time steps. The difference between Null Param. and MNO is only marginal, as seen in the solution in Fig. A-1a and error in Fig. A-1b. Both, Null param. and MNO slightly lag behind the ground-truth solution in Fig. A-1a. If zoomed in, one can see that the Null Param. contains slightly more high-frequency speckle noise than MNO which could

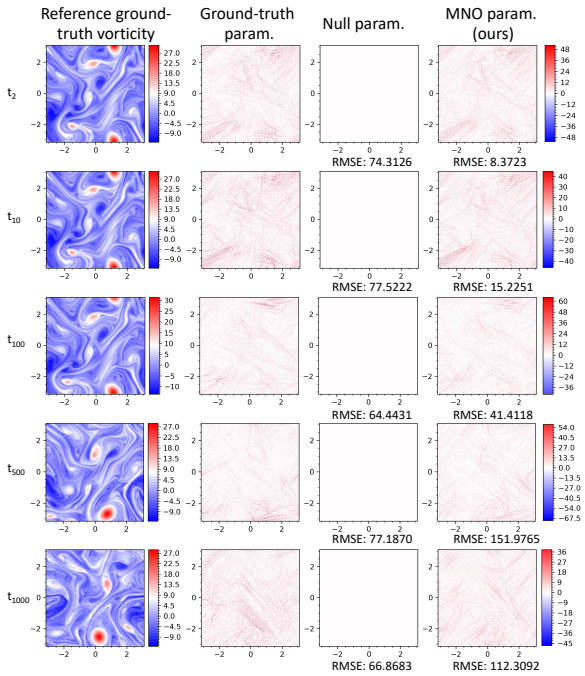


Figure A-2: **Parametrization accuracy over time.** In an autoregressive forecast, MNO predicts the parametrization more accurately than a baseline null parametrization until ≈ 100 time steps. After this, a null parametrization is more accurate.

indicate that learning the ground-truth parametrization with MNO can reconstruct missing high-frequency features.

Figure A-2 compares the ground-truth parametrizations (mid-left) and predicted parametrizations in MNO (right) in the autoregressive forecast. The MNO parametrization matches the ground-truth parametrization very closely for the first $\mathcal{O}(10)$ time steps: the learned parametrization visually captures the large-scale features of long lines and small-scale oscillatory peaks. The RMSE confirms quantitatively that the predicted parametrization has higher accuracy than a baseline null parametrization. The RMSE is calculated as the difference between predicted and ground-truth parametrizations. These parametrizations are calculated on the autoregressive forecasts of each vorticity simulation. After ≈ 100 time steps, the null parametrization is more accurate than the MNO-based parametrization. This is likely due to MNO's forecast drifting over time into a regime that was not seen during training. We believe that extending MNO to temporal architectures could alleviate these issues, for example, by using a pushforward [37] or in-situ loss function [95] or a model architecture

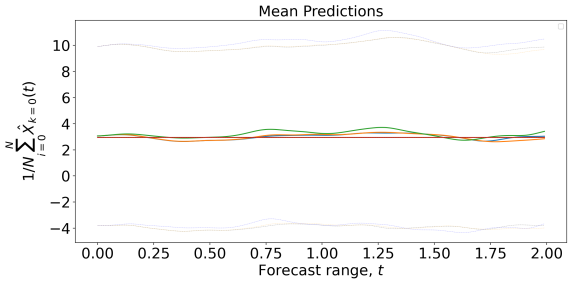


Figure A-3: **Multiscale Lorenz96 mean accuracy over time.** MNO (orange) forecasts the mean (solid) of the ground-truth DNS (blue) more accurately in comparison to ML-based parametrizations (green) and climatology (red). The standard deviations is plotted as dotted lines.

with temporal memory.

A.5.2 Multiscale Lorenz96

Accuracy Figure A-3 shows that the predicted mean of MNO (solid orange) closely follows the ground-truth (solid blue). The ML-based parametrization (solid green) follows the ground-truth only for a few time steps (until $\sim t = 0.125$). The climatology (solid red) depicts the average prediction in the training dataset. None of the models accurately represents the standard deviation (blue dotted).

Details of the runtime comparison The runtimes have been calculated by choosing the best of 1-100k runs depending on grid size on a single-threaded Intel Xeon Gold 6248 CPU@2.50GHz with 164Gb RAM. We time a one step update which, for DNS, is the calculation of Eq. (3.7) and for MNO the calculation of Eq. (3.8), i.e., the sum of a large-scale step and a pass through the neural operator.

In Fig. 3-3, the runtime of MNO and DNS plateaus at low-resolution ($K < 2^9$), because runtime measurement is dominated by operations whose cost is independent of spatial resolution. DNS plateaus at a lower runtime, because MNO contains several fixed-cost matrix transformations. The runtime of DNS has a slight discontinuity at $K \approx 2^9$ due to extending from cache to RAM memory.

A.5.3 Model configuration

Multiscale Lorenz96: MNO For multiscale Lorenz96 we use the number of channels, $n_v = 64$, number of retained modes, $k_{\max} = 3$, number of Fourier layers, $n_d = 3$, and no batch norm layer. The time-series modeling task uses a history of only one time step to learn chaotic dynamics [167]. We are using ADAM optimizer with learning rate, $\lambda = 0.001$, number of epochs, $n_e = 2$ [146]. MNO uses float32 precision. Training took $1:50$min on a single core Intel i7-7500U CPU@2.70GHz.

Multiscale Lorenz96: ML-based parametrization The ML-based parametrizations uses a fully-connected neural network (FCNN) with $n_{\text{layers}} = 2$ and $n_{\text{units}} = 32$ units. The model contains a residual connection from the input to the output layer. The model is optimized with Adam [146] with learning rate 0.01, $\beta = (0.9, 0.999)$, $\epsilon = 1 * 10^{-8}$, trained for $20 n_{\text{epochs}} = 20$.

Multiscale Lorenz96: Traditional parametrization The traditional parametrization uses least-squares to find the best linear fit. Inference is conducted with $\hat{y} = Ax$. This parametrization is local, which in this case, means that the weight matrix is shared across every grid point and the in- and outputs are one-dimensional, $x, \hat{y} \in \mathbb{R}$. The weight matrix, A, is computed on the training set with $A = (X^T X)^{-1} X^T Y$, where X and Y are the concatenation of input large-scale features and target parametrizations, respectively.

Quasi-geostrophic turbulence: MNO To model quasi-geostrophic turbulence with MNO, we chose the number of channels, $n_v = 20$, number of retained modes, $k_{\max} = 64$, number of Fourier layers, $n_d = 3$, and no batch norm layer. We are using ADAM optimizer with initial learning rate, $\lambda = 0.01$, batch size of 20 snippets that each contain 20 time steps. The learning rate is decayed by a factor of $\gamma = 0.5$ every 15 steps. We train in parallel on 40 Intel Xeon Gold 6248 CPUs with at 2.50GHz for $n_{\text{epochs}} = 19$ which takes $\approx 6hrs$. These are the hyperparameters with the best validation performance in random grid search over the ranges: $n_v \in \{5, 10, 20, 32\}$,

$k_{\max} \in \{12, 32, 64\}$, $n_d \in \{1, 3, 4\}$, $\lambda \in \{0.1, 0.01, 0.001\}$, batch size $\in \{20, 56\}$, $\lambda \in \{0.1, 0.01, 0.001\}$, $\gamma \in \{0.1, 0.5\}$, and learning rate step size $\in \{10, 15, 20\}$. All runs used the maximum number of epochs, $n_e = 100$, and stopped when the validation loss increased. For data augmentation, we tried using horizontal and vertical flip augmentations, but found that they reduced the train and validation performance. This indicates that we are operating in a regime of limited data and more data could increase the performance.

Quasi-geostrophic turbulence: Null Param. The null parametrization predicts the global mean over all parametrizations in the training dataset. In this case the mean parametrization is zero and we call this model Null Param.

Quasi-geostrophic turbulence: Pure ML FNO. The pure machine learning-based model trains a 2D Fourier neural operator [166] to autoregressively forecast the coarse-grained simulation: $\bar{\omega}_{\text{FNO}}(t + \Delta t) = \bar{\omega}_{\text{FNO}}(t)$. The model uses the same dataset of a coarse-grained high-resolution simulation and the same train/val/test data split than the other QG turbulence models. In comparison to the other models the ground-truth prediction target of the pure ML model is the coarse-grained high-resolution simulation at the next time step and not the parametrization. The pure ML FNO uses the same hyperparameters as the FNO that learns the parametrizations in MNO, except the pure ML model is stopped after $n_e = 16$ epochs. We tested minor variations of the hyperparameters, but only found decreasing performance. We lacked the computational resources for a full hyperparameter sweep over the same hyperparameter ranges as MNO.

Model configuration: Other parametrizations Comparing FNO vs. CNN-based parametrizations [11] would be interesting to, e.g., analyse the effects of spectral transform vs. locality biases, respectively. FNO could be more accurate due to the periodic features in our problems, but we do not claim that FNO is more accurate than a CNN-based parametrization. For this study, the comparison was out of scope and we note that the accuracy gains in Fig. 3-4 stem, instead, from a change from

local to global receptive fields and in Fig. 3-7 from a pure ML to hybrid ML-physics architecture. We also do not compare against methods that require an autodifferentiable solver. Models with soft or hard constraints, e.g., PINNs [249] or DC3 [81], are complementary to MNO.

A.5.4 Evaluation

The RMSE over time for multiscale Lorenz96 is calculated as:

$$\text{RMSE}(t) = \frac{1}{K}\sum_{k=0}^{K}\sqrt{(\frac{1}{N}\sum_{i=0}^{N}(\hat{X}_{k,i}(t) - X_{k,i}(t))^2)}. \tag{A.5}$$

A.6 Appendix to Limitations

A.6.1 Resolution-variable

We have stated that MNO is resolution-variable. We refer to a model architecture as resolution-variable if it can be trained and tested on grids of different resolutions without retraining the model or significant sacrifices in accuracy. MNO is resolution-variable because the underlying neural operator, here a Fourier neural operator (FNO), is resolution-variable. For FNO to be resolution-variable, it assumes that the in- and outputs are represented on equispaced grids which are grids with uniform spacing in between every neighboring cell [166]. Further, FNO assumes that all necessary information to reconstruct the in- and output is represented at the given spatial resolutions [166].

To understand resolution-variability in detail, we analyze the Fourier layer, specifically, the operation $\mathcal{F}^{-1}(R_\phi \cdot \mathcal{F}\underline{v}_i)$ from Eq. (A.6). All other transformations in FNO are shared transformations that are the same for every grid point, thus resolution-independent. The Fourier transform, $\mathcal{F}$, and inverse Fourier transform, $\mathcal{F}^{-1}$, are implemented via `torch.fft` and `torch.ifft`, respectively which currently assume an equispaced grid. Replacing this transform with a transform on non-equispaced grids [86] would lift the assumption of equispaced grids.

After the Fourier transform, the spectral signal is multiplied with a matrix, R_ϕ. In the current application this matrix multiplication will also set all Fourier frequencies beyond the cut-off frequency, $k_{\max}$, to zero. As a result, the number of retained spectral modes, $k_{\max}$, is less or equal to the number of spatial grid points, $\bar{N}$, during training: $k_{\max} <= \bar{N}_{\text{train}}$. The number of retained spectral modes, $k_{\max}$, is fixed after training.

If the spatial resolution during test is finer than the training resolution, $\bar{N}_{\text{test}} > \bar{N}_{\text{train}}$, the architecture will cut-off all frequencies beyond, $k_{\max}$. If the first $k_{\max}$ frequencies of the test signal are the same, whether it is represented on the higher, $\bar{N}_{\text{test}}$, or lower, $\bar{N}_{\text{train}}$, resolution grid, FNO will be resolution-invariant. Here, we assume that the first $k_{\max}$ frequencies will be approximately the same and thus call FNO resolution-variable. If the test resolution is lower than the training resolution, $\bar{N}_{\text{test}} < \bar{N}_{\text{train}}$, we fill in zeroes as higher resolution frequencies.

A.6.2 Complex initial and boundary conditions

We have evaluated MNO on equations with periodic boundary conditions. As a result, using Fourier transforms was a natural choice. Realistic Earth System Models, however, account for complex boundary conditions such as coastlines, bathymetry, an Arctic cycle cut-off, etc. Thus, extensions of FNO to more complex boundary conditions, for example, via incorporating Chebyshev polynomials in the vertical direction or spherical hermonics are very interesting.

We have also evaluated MNO only on deterministic initial conditions (ICs). In realistic weather and climate modeling, however, ICs are often partially known and uncertain. The uncertainty in ICs can be accounted for via combining physics-informed ML models with data assimilation or ensemble runs [142]. Further, MNO was only evaluted on IC profiles that naturally occur in chaotic multiscale Lorenz96 and QG-turbulence simulations. In practice, ICs can show inversions, sharp gradients, and other features that can be investigated in more detail.

A.7 Fourier Neural Operator

Our neural operator for learning subgrid parametrizations is based on Fourier neural operators [166]. Intuitively, the neural operator learns a parameter-to-solution mapping by learning a global convolution kernel. In detail, it learns the operator to transforms the current large-scale state, $\underline{X}(x_{0:K}, t) \in \mathbb{R}^{K \times d_X}$ to the subgrid parametrization, $\hat{\underline{f}}_x(x_{0:K}, t) := \underline{X}_{0:K} \in \mathbb{R}^{K \times d_X}$ with number of grid points, K, and input dimensionality, d_X, according to the following equations:

$$\begin{aligned} \underline{v}_0 &= \underline{X}_{0:K} P^T + 1^{K \times 1} b_P \\ \underline{v}_{i+1} &= \sigma\left(\underline{v}_i W^T + \int_{D_x} \kappa_\phi(x, x') v_i(x') dx'\right) \\ &\approx \sigma\left(\underline{v}_i W^T + 1^{n_v \times 1} b_W + \mathcal{F}^{-1}(R_\phi \cdot \mathcal{F}\underline{v}_i)\right) \\ \hat{f}_{x,0:K} &= \underline{v}_{n_d} Q^T + 1^{K \times 1} b_Q \end{aligned} \tag{A.6}$$

First, MNO lifts the input via a linear transform with matrix, $P \in \mathbb{R}^{n_v \times d_X}$, bias, $b_P \in \mathbb{R}^{1 \times n_v}$, vector of ones, $1^{K \times 1}$, and number of channels, n_v. The linear transform is local in space, i.e., the same transform is applied to each grid point.

Second, multiple nonlinear "Fourier layers" are applied to the encoded/lifted state. The encoded/lifted state's, $\underline{v}_i \in \mathbb{R}^{K \times n_v}$, spatial dimension is transformed into the Fourier domain via a fast Fourier transform. We implement the FFT as a multiplication with the pre-built forward and inverse Type-I DST matrix, $\mathcal{F} \in \mathbb{C}^{k_{\max} \times K}$ and $\mathcal{F}^{-1} \in \mathbb{C}^{K \times k_{\max}}$, respectively, returning the vector, $\mathcal{F}\underline{v}_i \in \mathbb{C}^{k_{\max} \times n_v}$.

The dynamics are learned via convoluting the encoded state with a weight matrix. In Fourier space, convolution is a multiplication, hence each frequency is multiplied with a complex weight matrix across the channels, such that $R \in \mathbb{C}^{k_{\max} \times n_v \times n_v}$. In parallel to the convolution with R, the encoded state is multiplied with the linear transform, $W \in \mathbb{R}^{n_v \times n_v}$, and bias, $b_W \in \mathbb{R}^{1 \times n_v}$. From a representation learning-perspective, the Fourier decomposition as a fast and interpretable feature extraction method that extracts smooth, periodic, and global features. The linear transform can be interpreted as residual term concisely capturing nonlinear residuals.

So far, we have only applied linear transformations. To introduce nonlinearities, we apply a nonlinear activation function, σ, at the end of each Fourier layer. While the non-smoothness of the activation function ReLu, $\sigma(z) = \max(0, z)$, could introduce unwanted discontinuities in the solution, we choose it resulted in more accurate models than smoother activation functions such as tanh or sigmoid.

Finally, the transformed state, v_{n_d}, is projected back onto solution space via another linear transform, $Q \in \mathbb{R}^{d_X \times n_v}$, and bias, b_Q.

The values of all trainable parameters, P, R, W, Q, b_*, are found by using a nonlinear optimization algorithm, such as stochastic gradient descent or, here, Adam [146]. We have used MSE between the predicted, $\hat{f}_x$, and ground-truth, f_x, subgrid parametrizations as loss. The neural operator is implemented in pytorch, but does not require an autodifferentiable PDE solver to generate training data. During implementation, we used the DFT which assumes a uniformly spaced grids, but can be exchanged with non-uniform DFTs (NUDFT) to transform non-uniform grids [86].

A.8 Neural networks vs. neural operators

Most work in physics-informed machine learning relies on fully-connected neural networks (FCNNs) or convolutional neural networks [138]. FCNNs however are mappings between finite-dimensional spaces and learn mappings for single equation instances rather than learning the PDE solver. In our case FCNNs only learn mappings on fixed spatial grids. We leverage the recently formulated neural operators to extend the formulation to arbitrary grids. The key distinction is that the FCNN learns a parameter-dependent set of weights, Φ_{a_y}, that has to be retrained for every new parameter setting. The neural operator is a learned function mapping with parameter-variable weights, Θ, that takes parameter settings as input and returns a function over the spatial domain, $G_\Theta(a_y)$. In comparison, the forcing term is approximated by an FCNN as $\hat{f}_{x,\Phi}(x_k; a_y) = g_{\Phi_{a_y}}(x_k)$ and by a neural operator as $\hat{f}_{x,\Theta}(x_k; a_y) = G_\Theta(a_y)(x_k)$. The

mappings are given by:

$$\begin{aligned} &\text{FCNN: } g_{\Phi_{a_y}} : \ D_x \to \mathbb{R}^{d_X}, \\ &\text{NO: } G_\Theta : \ H_{a_y}(D_x; \mathbb{R}^{d_{a_y}}) \to H_X(D_x; \mathbb{R}^{d_X}). \end{aligned} \tag{A.7}$$

H_{a_y} is a function space (Banach) of PDE parameter functions, a_y, that map the spatial domain, D_y, onto d_{a_y} dimensional parameters, such as ICs, BCs, parameters, or forcing terms. H_X is the function space of residuals that map the spatial domain, D_x, onto the space of d_X-dimensional residuals, $\mathbb{R}^{d_X}$.

Appendix B

Appendix: Satellite Imagery from the Future

B.1 Dataset

B.1.1 Pre- and post-flood imagery

Post-flood images that display standing water are challenging to acquire due to cloud-cover, time of standing flood, satellite revisit rate, and cost of high-resolution imagery. To the extent of the authors' knowledge, xBD [112] is the best publicly available data-source for preprocessed high-resolution imagery of pre- and post-flood images. More open-source, high-resolution, pre- and post-disaster images can be found in unprocessed format on DigitalGlobe's Open Data repository [80].

- Data Overview: 3284 flood-related RGB image pairs from seven flood events at 1024×1024px of ∼0.5m/px resolution of which 30% display a standing flood (∼1370).

- The dataset contains imagery of hurricanes (Harvey, Florence, Michael, Matthew in the U.S. East or Golf Coast), spring floods (2019 in Midwest U.S.), a tsunami (in Indonesia), and the monsoon (in Nepal).

- Our evaluation test set is composed of 216 images: 108 images of each hurricane

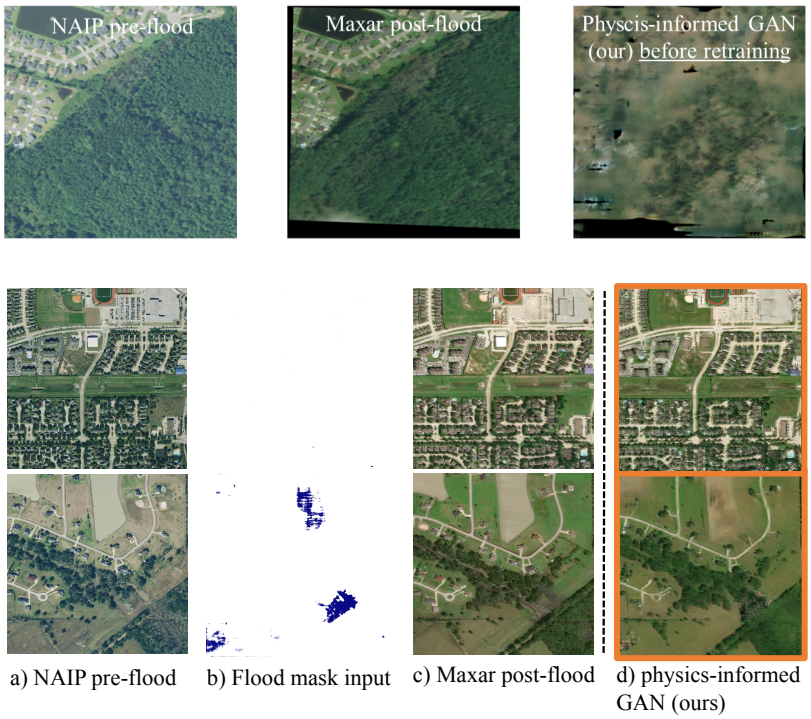


Figure B-1: **Generalization across remote sensing instruments.** Our first model was trained on satellite imagery from Maxar. The model is very sensitive to changes in the inputs: when we change the inputs from Maxar to NAIP aerial images the model generates unintelligible imagery (top-row, right). This is likely due to the changes in resolution, color calibration, atmospheric noise, and more. To overcome this issue and train a model that generalizes across remote sensing instruments, we have compiled an additional dataset of 6500 image-pairs with $1.0m/px$ input aerial NAIP imagery and $0.5m/px$ output Maxar satellite imagery (a, c in two bottom rows). While the generated imagery in d) is not production ready it does suggest that a retrained model can learn to translate images across different remote sensing instruments.

Harvey and Florence. The test set excludes imagery from hurricane Michael or Matthew, because the majority of tiles does not display standing flood.

- We did not use digital elevation maps (DEMs), because the information of low-resolution DEMs is contained in the storm surge model and high-resolution DEMs for the full U.S. East Coast were not publicly available.

B.1.2 Pre- and post-reforestation imagery

We went through all of VERA registered ARR (Afforestation, Reforestation, and Revegetation) carbon projects, and downloaded the shapefiles (.kmz) when they were

available. We used Google Earth Pro [source] to first confirm that historical imagery was available, meaning that high-resolution imagery was available over the years for the specific regions from the shapefiles and had no clouds. Then we visually verified that the reforestation did happen. These conditions were considered satisfied if we had imagery of bare land where we could see trees planted and grown over the years. In that case, we considered "pre-reforestation" as the earliest available imagery before we could see that trees were planted. For instance, if on the high-resolution imagery, we saw that trees were planted in 2010, then we would go back in time to the previously available imagery which could be 2009 or 2005, depending on the regions. For the "post-reforestation" imagery, we selected the most recent imagery available without clouds, so ideally it would be 2022. Our dataset timestamps go from 2005 to 2022 with some in-between years when those years were not available or not good quality (too many clouds, overlapping rasters from different years, etc).

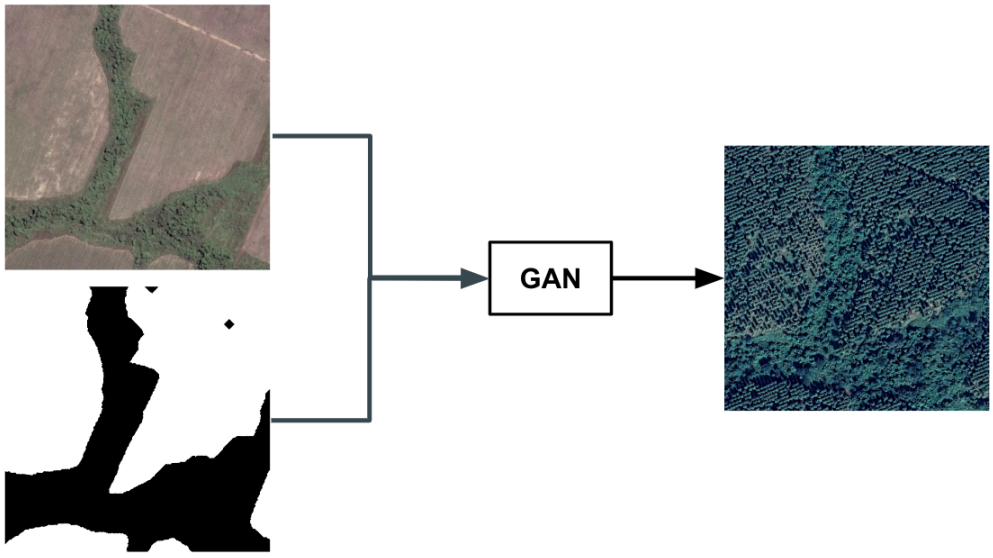


Figure B-2: **Model Architecture for Reforestation.** Our model leverages the semantic image synthesis model, Pix2pixHD, and combines a pre-reforestation satellite image with a reforestation map to generate post-reforestation satellite imagery.

Ultimately, the selected high-resolution images of before and after reforestation were exported from Google Earth Pro (Map data: Google, Maxar Technologies, CNS/Airbus) with a resolution of 4800x4800 and an eye altitude of 1500 meters.

Figure B-3: **Generated post-flooding imagery of** 64 **randomly chosen tiles** of hurricanes Harvey and Florence test set.

The image of the pre-reforestation with the visual of the shapefile layer was also exported to be then used as the binary reforested area mask. The image with the visual of the shapefile was converted to a binary mask using an image processing method that removed all non-white pixels.

The 1024x1024 tiles were generated from the 4800x4800 exported images. The binary reforestation area masks were stacked on top of the RGB images. Only tiles that had pixels belonging to a reforested area were saved.

B.1.3 Pre- and post-melt Arctic sea ice imagery

To visualize melting Arctic sea ice we created 19445 image-triplets of pre-melt image, post-melt image, and post-melt segmentation mask, as displayed in Fig. B-5. We found $\sim 20k$ matching pre- and post-melt images by finding matching pairs across 27172 Winter images in 1st Oct. 2019 - 1st May 2020 and 32433 Summer images in 1st Jun. - 31st Aug. 2020 within the study area in Fig. B-6a. We downloaded cloud-free Sentinel-2 MSI Level 1-C visual imagery at 10m/px resolution in tiles of 1024x1024px that matched the criteria. Our ice segmentation model creates binary ice

Figure B-4: **Dalle-2.** We prompted the diffusion-based model, DALL·E 2, to generate ´A realistic and physically-consistent satellite image of Houston, TX being flooded.´ The visualized urban scenery is noisy and lacks the detail of buildings (zoom in). The blur is likely due to the complexity of generating the high-frequency and structured information that characterizes satellite images. This further indicates the continued challenge that generating satellite imagery, instead of drawings or first person images, is still posing to deep generative vision models.

segmentation masks (1=ice) by thresholding grayscaled images into white and non-white (intensity < 255) areas, `x = (1 if x==1 else 0)`. We created the post-melt segmentation masks by applying the segmentation model on the post-melt imagery. After creating segmentation masks, we rejected all image pairs that only contain ocean or land(mask=0). Figure B-6b shows the final locations of all image pairs.

B.2 Flood Segmentation Model

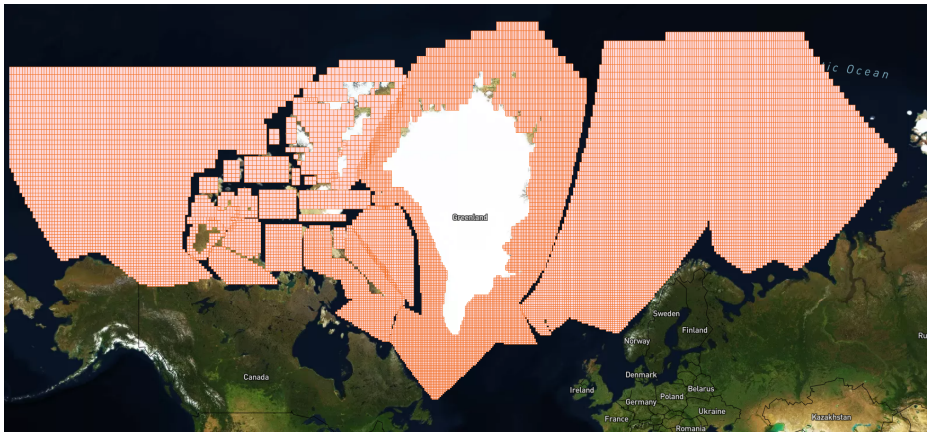
We evaluate the physical-consistency of the generated flood imagery by measuring intersection over union (IoU). Specifically, we measure the IoU of a flood mask that is derived from the generated image via a flood segmentation model with the ground-truth flood mask that was used as input. Here, we provide additional methodology and results for the flood segmentation model.

We trained two independent flood segmentation models – one for the main exper-

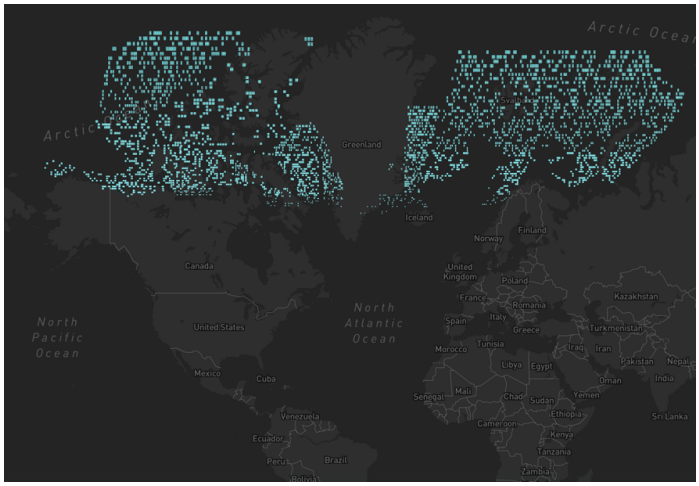
(a) Data Samples

Figure B-5: We allow the extension of *Earth Intelligence Engine* to visualize melting Arctic sea ice by publishing an according dataset.

iments on 109 labelled images in the Maxar → Maxar dataset in Section 4.3.2 and another on the 260 labeled images in the NAIP → Maxar dataset in Section 4.3.4. Our implementation is a pix2pix model [130] where we added on an IoU loss. For both datasets, we used a UNet with 120 trainable layers to predict 1-channel segmentation masks from 3-channel images of size 1024x1024. The loss function is a weighted sum (1:1:5) of a vanilla GAN loss, L1 loss, and negative IoU wrt. the ground-truth segmentation mask. We trained the model for 80 epochs with batch size, 8, and learning rate, 0.0002, on 4 GPUs. For the Maxar → Maxar dataset, we then fix the first 100 layers and fine-tune the last 20 layers network for 80 epochs using only L1 and IoU loss (0:1:3) and otherwise the same hyperparameters. For the NAIP → Maxar dataset, we did not fine-tune the network. We experimented with decreasing the number of epochs and removing the GAN loss to train a UNet on L1+IoU, which both individually decreased performance. Otherwise, we are using the default parameters in https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix. As a train-test split we used a 5-fold cross validation to maximize usage of our limited data, i.e., we split the dataset into 5 equally sized partitions and train the model with considered hyperparameters 5-times on 4 partitions while holding out 1 partition. After finding the



(a) Study Area



(b) Final data distribution

Figure B-6: We compiled a dataset of $\sim 20k$ image-pairs of the Arctic to visualize melting Arctic sea ice.

model with the best hyperparameters, we train it on the full labeled dataset and use it to infer masks on the rest of the unlabeled data. The model for Maxar $\rightarrow$ Maxar was trained similarly.

The model on the Maxar $\rightarrow$ Maxar dataset has IoU 0.343 and on the NAIP $\rightarrow$ Maxar dataset has IoU 0.23. Many images in our dataset display very little flooding and, hence, achieving high IoU scores is very difficult. For a better understanding that the quality of our current segmentation model should be sufficient, we are plotting 20 randomly selected samples from the NAIP $\rightarrow$ Maxar test set in Fig. B-7.

The segmentation model could be improved, e.g., by considering more novel segmentation architectures, such as, PSPNet, PAN, or DeepLabv3+. Due to the limited

amount of data, we expect semi-supervised learning to impact the performance more than architectural choices. Specifically, future work could pretrain a network on a reconstruction loss or on other remote sensing tasks and datasets. Unfortunately, during development of this study there did not exist an accessible backbone that was pretrained on visual satellite imagery. We explored using ImageNet backbones, but they classified "background" for every image, which led us to believe that mid-to-late-stage layers have little to no signal and only early layers would be useful.

B.3 Experiments

B.3.1 Data augmentation.

To visualize floods, we applied standard data augmentation, here rotation, random cropping, hue, and contrast variation, and state-of-the art augmentation - here elastic transformations [279]. Furthermore, spectral normalization [203] was used to stabilize the training of the discriminator. A relativistic loss function has been implemented to stabilize adversarial training. We also experimented with training pix2pixHD on LPIPS loss. Quantitative evaluation of these experiments, however, showed that they did not have significant impact on the performance and, ultimately, the results in the paper have been generated by the pytorch implementation of pix2pixHD [316] extended to 4-channel inputs.

To visualize reforestation, we used downscale (to 0.8 scale), h- and v-flip, and colorjitter (brightness=0.4, contrast=0.2, saturation=0, hue=0) augmentations with $p = 0.67$ from the albumentations library [49]. The model hyperparameters are chosen to equal the pytorch implementation in [316].

B.3.2 Pre-training LPIPS on satellite imagery.

The standard LPIPS did not clearly distinguish in between the handcrafted baseline and the physics-informed GAN, contrasting the opinion of a human evaluator. This is most likely because LPIPS currently leverages a neural network that was trained

on object classification from ImageNet. The neural network might not be capable to extract meaningful high-level features to compare the similarity of satellite images. In preliminary tests the ImageNet-pretrained network would classify all satellite imagery as background image, indicating that the network did not learn features to distinguish satellite images from each other. Future work, will use LPIPS with a network trained to have satellite imagery specific features, e.g., Tile2Vec or a land-use segmentation [259] model.

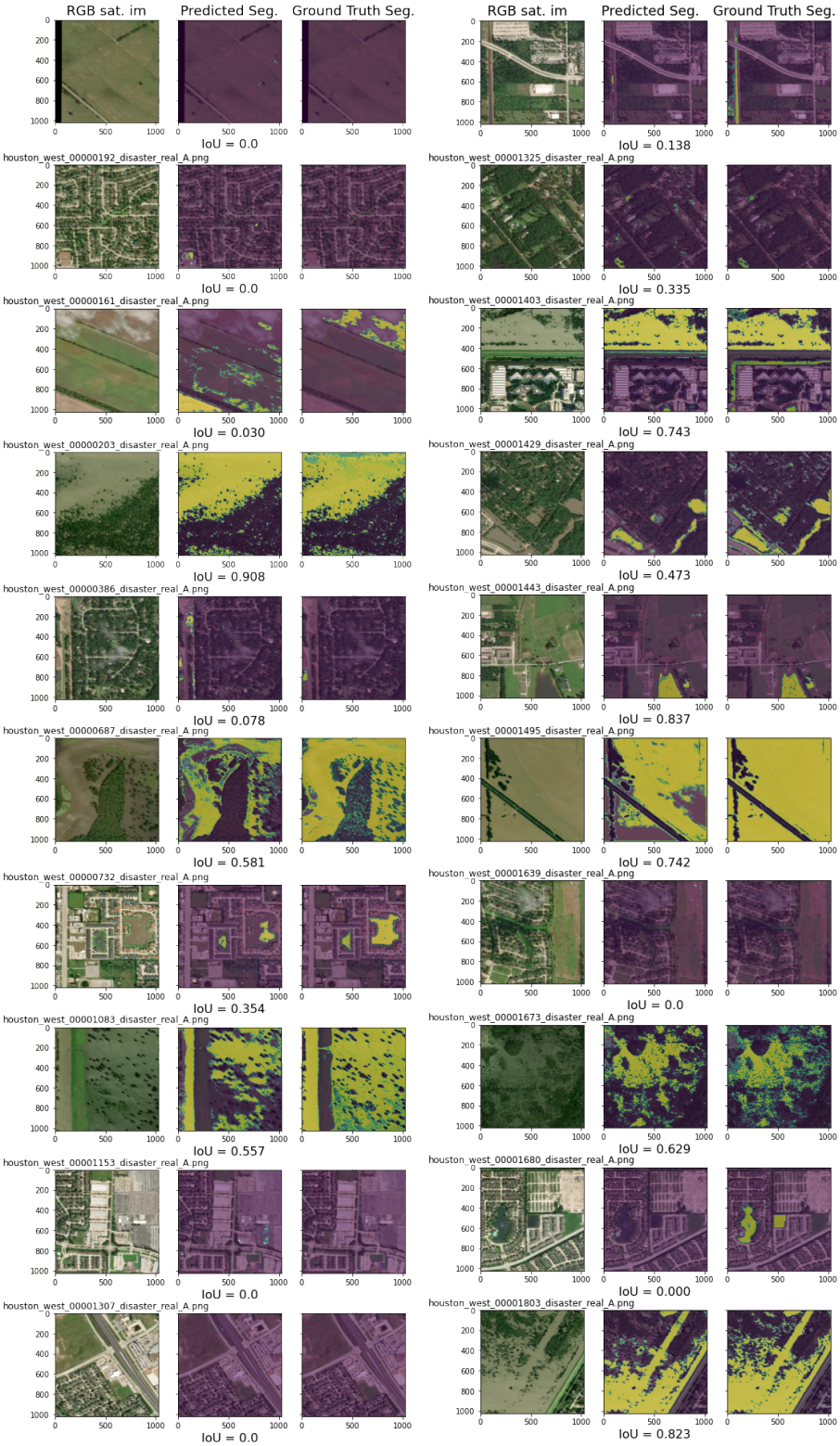


Figure B-7: Generated segmentation masks for 20 randomly chosen tiles of NAIP → Maxar test dataset (zoom in). The mean IoU across the dataset is 0.23.

Bibliography

[1] Sistema nacional de informacao geografica (snig), national infrastructure for geographical information, 2006. URL http://snig.igeo.pt/. last accessed 01/21.

[2] The cmip6 landscape. *Nature Climate Change*, 9:727, 2019.

[3] G. Althor, J. E. M. Watson, and R. A. Fuller. Global mismatch between greenhouse gas emissions and the burden of climate change. *Scientific Reports*, 6, 2016.

[4] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[5] A. Anandkumar, K. Azizzadenesheli, K. Bhattacharya, N. Kovachki, Z. Li, B. Liu, and A. Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

[6] K. Anderson, B. Ryan, W. Sonntag, A. Kavvada, and L. Friedl. Earth observation in service of the 2030 agenda for sustainable development. *Geo-spatial Information Science*, 20(2):77–96, 2017.

[7] N. Audebert, B. Le Saux, and S. Lefevre. Generative adversarial networks for realistic synthesis of hyperspectral samples. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 4359–4362, 2018.

[8] S. Bai, V. Koltun, and J. Z. Kolter. Multiscale Deep Equilibrium Models. 33, 2020.

[9] S. Bamberg and G. Möser. Twenty years after hines, hungerford, and tomera: A new meta-analysis of psycho-social determinants of pro-environmental behaviour. *Journal of Environmental Psychology*, 27(1):14–25, 2007. ISSN 0272-4944.

[10] A. F. Banwell, D. R. MacAyeal, and O. V. Sergienko. Breakup of the larsen b ice shelf triggered by chain reaction drainage of supraglacial lakes. *Geophysical Research Letters*, 40(22):5872–5876, 2013.

[11] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31), 2019.

[12] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82 – 115, 2020.

[13] R. Basri, D. W. Jacobs, Y. Kasten, and S. Kritchman. The convergence rate of neural networks for learned functions of different frequencies. *CoRR*, abs/1906.00425, 2019.

[14] P. D. Bates, M. S. Horritt, and T. J. Fewtrell. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1):33–45, 2010.

[15] P. Battaglia, J. B. C. Hamrick, V. Bapst, A. Sanchez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. Allen, C. Nash, V. J. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv*, 2018. URL https://arxiv.org/pdf/1806.01261.pdf.

[16] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13, 2022.

[17] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020. ISSN 0027-8424.

[18] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.*, 18(1): 5595–5637, jan 2017. ISSN 1532-4435.

[19] J. Behler. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.*, 13, 2011.

[20] P. M. Behler J. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys Rev Lett.*, 14, 2007.

[21] A. Bennett and B. Nijssen. Deep learned process parameterizations provide better representations of turbulent heat fluxes in hydrologic models. *Earth and Space Science Open Archive*, page 20, 2020.

[22] T. Beucler, S. Rasp, M. Pritchard, and P. Gentine. Achieving Conservation of Energy in Neural Network Emulators for Climate Modeling. jun 2019.

[23] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.*, 126:098302, Mar 2021.

[24] T. Beucler, M. S. Pritchard, J. Yuval, A. Gupta, L. Peng, S. Rasp, F. Ahmed, P. A. O'Gorman, J. D. Neelin, N. J. Lutsko, and P. Gentine. Climate-invariant machine learning. *CoRR*, 2021.

[25] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart. Model reduction and neural networks for parametric pdes, 2020.

[26] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart. Model Reduction And Neural Networks For Parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021.

[27] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive flow control with limited sensor data and online learning, 2020.

[28] S. Bire, B. Lütjens, D. Newman, and C. Hill. Oceanfourcast: Emulating ocean models with transformers for adjoint-based data assimilation. *EGU General Assembly 2023*, pages EGU23–10810, 2023.

[29] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[30] S. S. Blakseth, A. Rasheed, T. Kvamsdal, and O. San. Deep neural network enabled corrective source term approach to hybrid analysis and modeling. *Neural Networks*, 146:181–199, 2022.

[31] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proceedings of the Combustion Institute*, 38(2):2617–2625, 2021.

[32] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[33] M. Bonavita and P. Laloyaux. Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, 12(12), 2020.

[34] A. Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[35] W. Bounliphone, E. Belilovsky, M. B. Blaschko, I. Antonoglou, and A. Gretton. A test of relative similarity for model selection in generative models, 2016.

[36] J. Boyd. *Chebyshev and Fourier Spectral Methods: Second Revised Edition*. Dover Books on Mathematics. Dover Publications, 2013.

[37] J. Brandstetter, D. E. Worrall, and M. Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations (ICLR)*, 2022.

[38] J. Brandstetter, D. E. Worrall, and M. Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

[39] N. D. Brenowitz and C. S. Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.

[40] N. D. Brenowitz, T. Beucler, M. Pritchard, and C. S. Bretherton. Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, 77(12):4357 – 4375, 2020.

[41] N. D. Brenowitz, T. Beucler, M. Pritchard, and C. S. Bretherton. Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, 77(12):4357 – 4375, 2020.

[42] C. S. Bretherton, B. Henn, A. Kwa, N. D. Brenowitz, O. Watt-Meyer, J. McGibbon, W. A. Perkins, S. K. Clark, and L. Harris. Correcting coarse-grid weather and climate models by machine learning from global storm-resolving simulations. *Journal of Advances in Modeling Earth Systems*, 14(2), 2022.

[43] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial (2nd Ed.)*. Society for Industrial and Applied Mathematics, USA, 2000. ISBN 0898714621.

[44] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[45] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, 2021.

[46] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[47] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[48] K. J. Burns, G. M. Vasil, J. S. Oishi, D. Lecoanet, and B. P. Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2), Apr. 2020.

[49] A. Buslaev, A. Parinov, E. Khvadchenya, V. I. Iglovikov, and A. A. Kalinin. Albumentations: fast and flexible image augmentations. *arXiv e-prints*, 2018.

[50] S. R. Cachay, E. Erickson, A. F. C. Bucker, E. Pokropek, W. Potosnak, S. Bire, S. Osei, and B. Lütjens. The world as a graph: Improving el niño forecasts with graph neural networks, 2020.

[51] S. R. Cachay, V. Ramesh, J. N. S. Cole, H. Barker, and D. Rolnick. ClimART: A benchmark dataset for emulating atmospheric radiative transfer in weather and climate models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[52] J. Campin, C. Hill, H. Jones, and J. Marshall. Super-parameterization in ocean modeling: Application to deep convection. *Ocean Modelling*, 36:90–101, 2011.

[53] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu. Towards understanding the spectral bias of deep learning. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2205–2211. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[54] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91, Dec 2019.

[55] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[56] Centre for Research on the Epidemiology of Disasters (CRED) and UN Office for Disaster Risk Reduction UNISDR. The human cost of weather-related disasters 1995-2015, 2015.

[57] S. Chakraborty, S. Adhikari, and R. Ganguli. The role of surrogate models in the development of digital twins of dynamic systems. *Applied Mathematical Modelling*, 90:662–681, 2021.

[58] S. Chan and A. H. Elsheikh. Parametric generation of conditional geological realizations using generative neural networks, 2019.

[59] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.

[60] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, and K. Kashinath. Deep spatial transformers for autoregressive data-driven forecasting of geophysical turbulence. In *Proceedings of the 10th International Conference on Climate Informatics*, CI2020, page 106–112, New York, NY, USA, 2020. Association for Computing Machinery.

[61] A. Chattopadhyay, A. Subel, and P. Hassanzadeh. Data-driven super-parameterization using deep learning: Experimentation with multiscale lorenz 96 systems and transfer learning. *Journal of Advances in Modeling Earth Systems*, 12(11), 2020.

[62] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[63] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[64] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc., 2018.

[65] W. Chen and F. Ahmed. Padgan: Learning to generate high-quality novel designs. *Journal of Mechanical Design*, 143(3).

[66] City of Boston. Climate ready boston final report, December 2016.

[67] Climate Central. Sea level rise, predicted sea level rise impacts on major cities from global warming up to 4c, 2018. URL https://earthtime.org/stories/sea_level_rise.

[68] Climate Central. Coastal risk screening tool, 2023. URL https://coastal.climatecentral.org/. last accessed 04/23.

[69] S. J. Cohen, S. Sheppard, A. Shaw, D. Flanders, S. Burch, B. Taylor, D. Hutchinson, A. Cannon, S. Hamilton, B. Burton, and J. Carmichael. Downscaling and visioning of mountain snow packs and other climate change implications in North Vancouver, British Columbia. *Mitigation and Adaptation Strategies for Global Change planning, Visualization, Water management*, 17 (1):25–49, jan 2012.

[70] T. S. Cohen and M. Welling. Steerable cnns. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France*, 2017.

[71] E. Cole, S. Stathatos, B. Lütjens, T. Sharma, J. Kay, J. Parham, B. Kellenberger, and S. Beery. Teaching computer vision for ecology, 2023.

[72] A. Costa Nogueira, J. L. de Sousa Almeida, G. Auger, and C. D. Watson. Reduced order modeling of dynamical systems using artificial neural networks applied to water circulation. In H. Jagode, H. Anzt, G. Juckeland, and H. Ltaief, editors, *High Performance Computing*, pages 116–136, Cham, 2020. Springer International Publishing.

[73] S. L. Cox, A. J. Lopez, A. C. Watson, N. W. Grue, and J. E. Leisch. Renewable energy data, analysis, and decisions: A guide for practitioners. 3 2018. doi: 10.2172/1427970. URL https://www.osti.gov/biblio/1427970.

[74] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, Jul 2022.

[75] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[76] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[77] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[78] G. Di Baldassarre, G. Schumann, and P. Bates. Near real time satellite imagery to support and verify timely flood modelling. *Hydrological Processes*, 23(5):799–803, 2009.

[79] Y. B. Dibike and P. Coulibaly. Temporal neural networks for downscaling climate variability and extremes. *Neural Networks*, 19(2):135–144, 2006. ISSN 0893-6080. Earth Sciences and Environmental Applications of Computational Intelligence.

[80] DigitalGlobe. Open data for disaster response, 2020. URL https://www.digitalglobe.com/ecosystem/open-data.

[81] P. L. Donti, D. Rolnick, and J. Z. Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations (ICLR)*, 2021.

[82] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems 29*, pages 658–666. Curran Associates, Inc., 2016.

[83] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[84] P. D. Dueben and P. Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009, 2018.

[85] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51(1):357–377, 2019.

[86] A. Dutt and V. Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing*, 14(6):1368–1393, 1993.

[87] Earth System Grid Federation (ESGF). World climate research program cmip6, 2023. URL https://esgf-node.llnl.gov/search/cmip6/. last accessed 04/23.

[88] D. A. Edmonds, R. L. Caldwell, E. S. Brondizio, and S. M. Siani. Coastal flooding will disproportionately impact people on river deltas. *Nature communications*, 11(1):1–8, 2020.

[89] L. Espeholt, S. Agrawal, C. Sønderby, M. Kumar, J. Heek, C. Bromberg, C. Gazen, R. Carver, M. Andrychowicz, J. Hickey, A. Bell, and N. Kalchbrenner. Deep learning for twelve hour precipitation forecasts. *Nature Communications*, 13(1):5145, Sep 2022.

[90] Z. I. Espinosa, A. Sheshadri, G. R. Cain, E. P. Gerber, and K. J. DallaSanta. Machine learning gravity wave parameterization generalizes to capture the qbo and response to increased co2. *Geophysical Research Letters*, 49(8), 2022.

[91] S. Farlow. *Partial Differential Equations for Scientists and Engineers*. Dover books on advanced mathematics. Dover Publications, 1993.

[92] T. Fiddaman, L. Siegel, E. Sawin, A. Jones, and J. Sterman. C-roads simulator reference guide, 2018.

[93] First Street Foundation. First street foundation flood model technical methodology document, 2020.

[94] L. Freese. Energy policy impacts on air quality, climate change, and equity at the national and global scale. *Doctoral Thesis*, 2023.

[95] H. Frezat, J. L. Sommer, R. Fablet, G. Balarac, and R. Lguensat. A posteriori learning for quasi-geostrophic turbulence parametrization, 2022.

[96] A. Frühstück, I. Alhashim, and P. Wonka. Tilegan: Synthesis of large-scale non-homogeneous textures. *ACM Trans. Graph.*, 38(4), July 2019.

[97] F. Fuchs, D. Worrall, V. Fischer, and M. Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

[98] O. Fuhrer, T. Chadha, T. Hoefler, G. Kwasniewski, X. Lapillonne, D. Leutwyler, D. Lüthi, C. Osuna, C. Schär, T. C. Schulthess, and H. Vogt. Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 gpus with cosmo 5.0. *Geosci. Model Dev.*, 11:1665 – 1681, 2018.

[99] O. Fuks and H. A. Tchelepi. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 1(1), 2020.

[100] K. Fukunaga and W. Koontz. Application of the karhunen-loève expansion to feature selection and ordering. *IEEE Transactions on Computers*, C-19(4): 311–318, 1970.

[101] A. R. Ganguly, E. A. Kodra, A. Agrawal, A. Banerjee, S. Boriah, S. Chatterjee, S. Chatterjee, A. Choudhary, D. Das, J. Faghmous, P. Ganguli, S. Ghosh, K. Hayhoe, C. Hays, W. Hendrix, Q. Fu, J. Kawale, D. Kumar, V. Kumar, W. Liao, S. Liess, R. Mawalagedara, V. Mithal, R. Oglesby, K. Salvi, P. K. Snyder, K. Steinhaeuser, D. Wang, and D. Wuebbles. Toward enhanced understanding and projections of climate extremes using physics-guided data mining techniques. *Nonlinear Processes in Geophysics*, 21(4):777–795, 2014.

[102] L. A. Garcia and A. Shigidi. Using neural networks for parameter estimation in ground water. *Journal of Hydrology*, 318(1):215–231, 2006.

[103] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[104] P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45(11):5742–5751, 2018.

[105] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[106] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[107] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT Press, 2016.

[108] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 15379–15389. Curran Associates, Inc., 2019.

[109] B. Groenke, L. Madaus, and C. Monteleoni. Climalign: Unsupervised statistical downscaling of climate variables via normalizing flows. In *Proceedings of the*

10th International Conference on Climate Informatics, CI2020, page 60–66, New York, NY, USA, 2020. Association for Computing Machinery.

[110] G. Gupta, X. Xiao, and P. Bogdan. Multiwavelet-based operator learning for differential equations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[111] G. Gupta, X. Xiao, and P. Bogdan. Multiwavelet-based operator learning for differential equations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24048–24062. Curran Associates, Inc., 2021.

[112] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston. Creating xBD: A Dataset for Assessing Building Damage from Satellite Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[113] Y.-G. Ham, J.-H. Kim, and J.-J. Luo. Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572, Sep 2019.

[114] F. Hamilton, A. L. Lloyd, and K. B. Flores. Hybrid modeling and prediction of dynamical systems. *PLOS Computational Biology*, 13(7):1–20, 07 2017.

[115] W. L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

[116] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, and K.-R. Müller. Assessment and validation of machine learning methods for predicting molecular atomization energies. *Journal of Chemical Theory and Computation*, 9(8):3404–3419, 2013.

[117] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. High-resolution global maps of 21st-century forest cover change. *Science*, 342(6160): 850–853, 2013. ISSN 0036-8075.

[118] P. Harder, V. Ramesh, A. Hernandez-Garcia, Q. Yang, P. Sattigeri, D. Szwarcman, C. Watson, and D. Rolnick. Physics-constrained deep learning for climate downscaling, 2022.

[119] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu. Liquid time-constant networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9): 7657–7666, May 2021.

[120] Z. Hausfather and P. Forster. Analysis: Do cop26 promises keep global warming below 2c?, November 2021. URL https://www.carbonbrief.org/analysis-do-cop26-promises-keep-global-warming-below-2c/. last accessed 04/23.

[121] N. Herger, B. M. Sanderson, and R. Knutti. Improved pattern scaling approaches for the use in climate impact studies. *Geophysical Research Letters*, 42(9):3486–3494, 2015.

[122] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. de Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J.-N. Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[123] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[124] T. Hoar and D. Nychka. Statistical downscaling of the community climate system model (ccsm) monthly temperature and precipitation projections, April 2008.

[125] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080.

[126] P. D. Howe, M. Mildenberger, J. R. Marlon, and A. Leiserowitz. Geographic variation in opinions on climate change at state and local scales in the usa. *Nature Climate Change*, 5(6):596–603, Jun 2015.

[127] IPCC. Global warming of 1.5c. an ipcc special report on the impacts of global warming of 1.5c above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty, 2018.

[128] IPCC. Ipcc 2022: Summary for policymakers, 2022.

[129] C. Irrgang, N. Boers, M. Sonnewald, E. A. Barnes, C. Kadow, J. Staneva, and J. Saynisch-Wagner. Towards neural earth system modelling by integrating artificial intelligence in earth system science. *Nature Machine Intelligence*, 3 (8):667–674, Aug 2021.

[130] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[131] C. P. Jelesnianski, J. Chen, and W. A. Shaffer. Slosh: Sea, lake, and overland surges from hurricanes. *NOAA Technical Report NWS 48, National Oceanic and Atmospheric Administration, U. S. Department of Commerce*, page 71, 1992. (Scanning courtesy of NOAA's NOS's Coastal Service's Center).

[132] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, and V. Kumar. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *SIAM International Conference on Data Mining, SDM 2019*, SIAM International Conference on Data Mining, SDM 2019, pages 558–566. Society for Industrial and Applied Mathematics Publications, 2019.

[133] X. Jia, J. Willard, A. Karpatne, J. S. Read, J. A. Zwart, M. Steinbach, and V. Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Trans. Data Sci.*, 2 (3), 2021.

[134] P. Jiang, N. Meinert, H. Jordão, C. Weisser, S. Holgate, A. Lavin, B. Lütjens, D. Newman, H. Wainwright, C. Walker, and P. Barnard. Digital Twin Earth – Coasts: Developing a fast and physics-informed surrogate model for coastal floods via neural operators. *2021 NeurIPS Workshop on Machine Learning for the Physical Sciences (ML4PS)*, 2021.

[135] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132, 12 2020.

[136] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596 (7873):583–589, Aug 2021.

[137] J. N. Kani and A. H. Elsheikh. DR-RNN: A deep residual recurrent neural network for model reduction. *CoRR*, abs/1709.00939, 2017.

[138] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021.

[139] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar. Theory-guided data science: A

new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[140] A. Karpatne, W. Watkins, J. Read, and V. Kumar. Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. *arXiv e-prints*, page arXiv:1710.11431, Oct. 2017.

[141] A. Karpatne, I. Ebert-Uphoff, S. Ravela, H. A. Babaie, and V. Kumar. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, Aug 2019.

[142] K. Kashinath, M. Mustafa, A. Albert, J.-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, and n. Prabhat. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194), 2021.

[143] R. Keisler. Forecasting global weather with graph neural networks, 2022.

[144] M. Khairoutdinov, D. Randall, and C. DeMott. Simulations of the atmospheric general circulation using a cloud-resolving model as a superparameterization of physical processes. *Journal of the Atmospheric Sciences*, 62(7 I):2136–2154, jul 2005.

[145] Y. Khoo and L. Ying. Switchnet: A neural network model for forward and inverse scattering problems. *SIAM Journal on Scientific Computing*, 41(5), 2019.

[146] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[147] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[148] J. Kingslake, J. C. Ely, I. Das, and R. E. Bell. Widespread movement of meltwater onto and across antarctic ice shelves. *Nature*, 544(7650):349–352, Apr 2017.

[149] J. Kingslake, A. Banwell, R. Bell, I. Das, R. DeConto, and J. Lenaerts. Report on antarctic surface hydrology workshop, ldeo, 2018, 2019.

[150] E. Kirezci, I. R. Young, R. Ranasinghe, S. Muis, R. J. Nicholls, D. Lincke, and J. Hinkel. Projections of global-scale extreme sea levels and resulting episodic coastal flooding over the 21st century. *Scientific reports*, 10(1):1–12, 2020.

[151] K. Klemmer. Improving neural networks for geospatial applications with geographic context embeddings, 2022.

[152] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer. Machine learning accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

[153] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. Neural operator learning maps between function spaces. *CoRR*, abs/2108.08481, 2021.

[154] R. Kurinchi-Vendhan, B. Lütjens, R. Gupta, L. Werner, and D. Newman. Wisosuper: Benchmarking super-resolution methods on wind and solar data. *Conference on Neural Information Processing Systems (NeurIPS) Workshop on Tackling Climate Change with Machine Learning (CCML)*, 2021.

[155] R. Kwok. Can climate change games boost public understanding?, April 2019.

[156] A. Lacoste, E. D. Sherwin, H. Kerner, H. Alemohammad, B. Lütjens, J. Irvin, D. Dao, A. Chang, M. Gunturkun, A. Drouin, P. Rodriguez, and D. Vazquez. Toward foundation models for earth monitoring: Proposal for a climate change benchmark, 2021.

[157] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, 2019.

[158] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687.

[159] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020. ISSN 0021-9991.

[160] T. M. Lenton, J. Rockström, O. Gaffney, S. Rahmstorf, K. Richardson, W. Steffen, and H. J. Schellnhuber. Climate tipping points — too risky to bet against. November 2019.

[161] T. Lesort, M. Seurin, X. Li, N. Díaz-Rodríguez, and D. Filliat. Deep unsupervised state representation learning with robotic priors: a robustness analysis. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[162] M. E. Levine and A. M. Stuart. A framework for machine learning of model error in dynamical systems, 2021.

[163] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, and Y. Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. ISSN 0925-2312.

[164] Y. Li, Q. Yu, M. Tan, J. Mei, P. Tang, W. Shen, A. Yuille, and cihang xie. Shape-texture debiased neural network training. In *International Conference on Learning Representations*, 2021.

[165] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, A. Stuart, K. Bhattacharya, and A. Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766. Curran Associates, Inc., 2020.

[166] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *ICML*, 2021.

[167] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. Markov neural operators for learning chaotic systems. *CoRR*, abs/2106.06898, 2021.

[168] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140, 2017.

[169] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[170] B. Liu, N. Kovachki, Z. Li, K. Azizzadenesheli, A. Anandkumar, A. M. Stuart, and K. Bhattacharya. A learning-based multiscale method and its application to inelastic impact problems. *Journal of the Mechanics and Physics of Solids*, 158:104668, 2022.

[171] Z. Liu, B. Wang, Q. Meng, W. Chen, M. Tegmark, and T.-Y. Liu. Machine-learning nonconservative dynamics for new-physics detection. *Phys. Rev. E*, 104:055302, Nov 2021.

[172] Y. Long, X. She, and S. Mukhopadhyay. Hybridnet: Integrating model-based and data-driven learning to predict evolution of dynamical systems. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 551–560. PMLR, 29–31 Oct 2018.

[173] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-net: Learning PDEs from data. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3208–3216. PMLR, 10–15 Jul 2018.

[174] Z. Long, Y. Lu, and B. Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399: 108925, 2019.

[175] E. Lorenz. Predictability - a problem partly solved. In T. Palmer and R. Hagedorn, editors, *Predictability of Weather and Climate*. Cambridge University Press, Cambridge, 2006.

[176] E. N. Lorenz and K. A. Emanuel. Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, 55(3):399 – 414, 1998.

[177] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

[178] R. A. Luettich, J. J. Westerink, and N. Scheffner. ADCIRC: an advanced three-dimensional circulation model for shelves coasts and estuaries, report 1: theory and methodology of ADCIRC-2DDI and ADCIRC-3DL. In *Dredging Research Program Technical Report*, page 137. DRP-92-6, U.S. Army Engineers Waterways Experiment Station, 1992.

[179] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020.

[180] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/c8067ad1937f728f51288b3eb986afaa-Paper.pdf.

[181] B. Lusch, J. Kutz, and S. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.*, 9, 2018.

[182] B. Lutjens and D. Newman. Physically-Consistent Generative Adversarial Networks for Visualizing Climate Impacts. In *AGU Fall Meeting Abstracts*, volume 2021, pages IN33B–07, Dec. 2021.

[183] B. Lütjens, M. Everett, and J. P. How. Safe Reinforcement Learning with Model Uncertainty Estimates. *International Conference on Robotics and Automation (ICRA), 2018*, 2018.

[184] B. Lütjens, M. Everett, and J. P. How. Certified Adversarial Robustness for Deep Reinforcement Learning. *2019 Conference on Robot Learning (CoRL)*, 2019.

[185] B. Lütjens*, B. Leshchinskiy*, C. Requena-Mesa*, F. Chishtie*, N. Díaz-Rodriguez*, O. Boulais*, A. Piña, D. Newman, A. Lavin, Y. Gal, and C. Raïssi. Physics-informed GANs for Coastal Flood Visualization. *arXiv e-prints*, 2020. * equal contribution.

[186] B. Lütjens, C. H. Crawford, M. Veillette, and D. Newman. Pce-pinns: Physics-informed neural networks for uncertainty propagation in ocean modeling. *International Conference on Learning Representations (ICLR) Workshop on AI for Modeling Oceans and Climate Change*, May 2021.

[187] B. Lütjens, C. H. Crawford, M. Veillette, and D. Newman. Spectral PINNs: Fast uncertainty propagation with physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS) Workshop on The Symbiosis of Deep Learning and Differential Equations (DLDE)*, 2021.

[188] B. Lütjens*, B. Leshchinskiy*, C. Requena-Mesa*, F. Chishtie*, N. Díaz-Rodríguez*, O. Boulais*, A. Sankaranarayanan*, A. P. na, Y. Gal, C. Raïssi, A. Lavin, and D. Newman. Physically-consistent generative adversarial networks for coastal flood visualization. *ICLR Workshop on AI: Modeling Oceans and Climate Change; in journal sumibssion*, 2021. * equal contribution.

[189] B. Lutjens, C. Watson, C. Crawford, and D. Newman. Matryoshka Neural Operators: Learning Fast PDE solvers with Limited Training Data. In *AGU Fall Meeting Abstracts*, pages A15E–1672, Dec. 2021.

[190] M. Lutter, C. Ritter, and J. Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[191] B. Lütjens, L. Liebenwein, and K. Kramer. Machine learning-based estimation of forest carbon stocks to increase transparency of forest preservation efforts. In *NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning*, 2019.

[192] B. Lütjens, C. H. Crawford, C. D. Watson, M. Veillette, C. Hill, and D. Newman. Multiscale neural operator: Creating fast pde surrogates by learning resolution-independent subgrid parametrizations, 2022.

[193] B. Lütjens, P. Alexander, R. Antwerpen, G. Cervone, M. Kearney, B. Luo, D. Newman, and M. Tedesco. Dailymelt: Diffusion-based models for spatiotemporal downscaling of (ant-)arctic surface meltwater maps. *EGU General Assembly 2023*, pages EGU23–4044, 2023.

[194] G. Madec and N. S. Team. *NEMO ocean engine*, January 2016. version 3.6 stable.

[195] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[196] A. Majda and X. Wang. *Nonlinear Dynamics and Statistical Theories for Basic Geophysical Flows*. Cambridge University Press, 2006.

[197] J. Marlon, L. Neyens, M. Jefferson, P. Howe, M. Mildenberger, and A. Leiserowitz. Yale climate opinion maps 2021, 2021. URL https://climatecommunication.yale.edu/visualizations-data/ycom-us/.

[198] Maxar DigitalGlobe. Open data program, hurricane harvey, 8/31/2017, tileid: 105001000b95e100. URL https://www.maxar.com/open-data/.

[199] M. C. McGraw and E. A. Barnes. Memory matters: A case for granger causality in climate variability studies. *Journal of Climate*, 31(8):3289 – 3300, 2018.

[200] K. McGuffie and A. Henderson-Sellers. *A Climate Modeling Primer, Third Edition*. John Wiley and Sons, Ltd., Jan. 2005.

[201] D. I. A. McKay, A. Staal, J. F. Abrams, R. Winkelmann, B. Sakschewski, S. Loriani, I. Fetzer, S. E. Cornell, J. Rockström, and T. M. Lenton. Exceeding 1.5c global warming could trigger multiple climate tipping points. *Science*, 377 (6611), 2022.

[202] T. M. Mitchell. The need for biases in learning generalizations, 1980.

[203] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *2018 International Conference on Learning Representations (ICLR)*, 2018.

[204] A. Mohan, D. Daniel, M. Chertkov, and D. Livescu. Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence, 2019.

[205] A. T. Mohan, N. Lubbers, D. Livescu, and M. Chertkov. Embedding hard physical constraints in neural network coarse-graining of 3d turbulence. *ICLR Workshop on AI for Earth Sciences*, 2020.

[206] T. Mohandoss, A. Kulkarni, D. Northrup, E. Mwebaze, and H. Alemohammad. Generating synthetic multispectral satellite imagery from sentinel-2. *NeurIPS 2020 Workshop on AI for Earth Sciences*, 2020.

[207] S. C. Moser. Reflections on climate change communication research and practice in the second decade of the 21st century: what more is there to say? *WIREs Climate Change*, 7(3):345–369, 2016.

[208] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[209] NASA/Goddard Space Flight Center Scientific Visualization Studio. Annual arctic sea ice minimum 1979-2020 with area graph, 2020. URL https://climate.nasa.gov/climate_resources/155/video-annual-arctic-sea-ice-minimum-1979-2020-with-area-graph/. last accessed March 2021.

[210] National Geodetic Survey. Emergency response imagery. URL https://storms.ngs.noaa.gov/. last accessed 04/23.

[211] NCAR. Community earth system model 2 (cesm2), 2023.

[212] T. Nguyen, J. Han, and D.-C. Park. Satellite image classification using convolutional learning. *AIP Conference Proceedings*, 1558(1):2237–2240, 2013.

[213] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover. Climax: A foundation model for weather and climate, 2023.

[214] NOAA. Noaa sea level rise viewer, 2020. URL https://coast.noaa.gov/slr/.

[215] NOAA. Climate data online, 2023. URL https://www.ncei.noaa.gov/cdo-web/. last accessed 04/23.

[216] NOAA. Sea level rise viewer, 2023. URL https://coast.noaa.gov/slr/. last accessed 04/23.

[217] NOAA National Centers for Environmental Information (NCEI). U.S. Billion-Dollar Weather and Climate Disasters (2023), 2023. URL https://www.ncdc.noaa.gov/billions/.

[218] NOAA National Weather Service National Hurrican Center Storm Surge Prediction Unit. National Storm Surge Hazard Maps, Texas to Maine, Category 5, 2020. URL https://noaa.maps.arcgis.com/apps/MapSeries/index.html?appid=d9ed7904dbec441a9c4dd7b277935fad&entry=1.

[219] B. Noël, W. J. van de Berg, H. Machguth, S. Lhermitte, I. Howat, X. Fettweis, and M. R. van den Broeke. A daily, 1km resolution data set of downscaled greenland ice sheet surface mass balance (1958–2015). *The Cryosphere*, 10(5): 2361–2377, 2016.

[220] A. C. Nogueira Jr., F. C. T. Carvalho, J. L. S. Almeida, A. Codas, E. Bentivegna, and C. D. Watson. Reservoir computing in reduced order modeling for chaotic dynamical systems. In H. Jagode, H. Anzt, H. Ltaief, and P. Luszczek, editors, *High Performance Computing*, pages 56–72, Cham, 2021. Springer International Publishing.

[221] P. A. O'Gorman and J. G. Dwyer. Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, 10(10):2548–2563, 2018.

[222] M. Ojala. Hope and climate change: the importance of hope for environmental engagement among young people. *Environmental Education Research*, 18(5): 625–642, 2012.

[223] P. J. Olver. *Symmetry Groups of Differential Equations*, pages 77–185. Springer New York, New York, NY, 1986.

[224] S. O'Neill. Engaging with climate change imagery, 10 2017.

[225] S. O'Neill and S. Nicholson-Cole. "fear won't do it": Promoting positive engagement with climate change through visual and iconic representations. *Science Communication*, 30(3):355–379, 2009. doi: 10.1177/1075547008329201.

[226] S. P. *Large Eddy Simulation for Incompressible Flows: An Introduction*. Scientific Computation. Springer-Verlag Berlin Heidelberg, 3 edition, 2006.

[227] T. Palmer, B. Stevens, and P. Bauer. We need an international center for climate modeling, 2019. URL https://blogs.scientificamerican.com/observations/we-need-an-international-center-for-climate-modeling/. last accessed 04/13/20.

[228] L. Pan, E. M. Powell, K. Latychev, J. X. Mitrovica, J. R. Creveling, N. Gomez, M. J. Hoggard, and P. U. Clark. Rapid postglacial rebound amplifies global sea level rise following west antarctic ice sheet collapse. *Science Advances*, 7(18), 2021.

[229] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(1), jan 2021. ISSN 1532-4435.

[230] E. J. Parish and K. Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016. ISSN 0021-9991.

[231] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[232] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR, 10–15 Jul 2018.

[233] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[234] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120, Jan 2018.

[235] J. Pathak, M. Mustafa, K. Kashinath, E. Motheau, T. Kurth, and M. Day. Using machine learning to augment coarse-grid computational fluid dynamics simulations, 2020.

[236] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. Feb. 2022.

[237] G. Pavliotis and A. Stuart. *Multiscale Methods Averaging and Homogenization*, volume 53 of *Texts in Applied Mathematics*. Springer-Verlag New York, 1 edition, 2008. ISBN 978-0-387-73829-1.

[238] G. C. Y. Peng, M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl. Multiscale modeling meets machine learning: What can we learn? *Archives of Computational Methods in Engineering*, 28:1017–1037, 2021.

[239] A. Prakash, K. E. Jansen, and J. A. Evans. Invariant Data-Driven Subgrid Stress Modeling in the Strain-Rate Eigenframe for Large Eddy Simulation. *arXiv e-prints*, June 2021.

[240] M. J. Purvis, P. D. Bates, and C. M. Hayes. A probabilistic methodology to estimate future coastal flood risk due to sea level rise. *Coastal Engineering*, 55 (12):1062–1073, 2008. ISSN 0378-3839.

[241] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. Lift and learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406, 2020.

[242] Z. Qian, K. Kacprzyk, and M. van der Schaar. D-CODE: Discovering closed-form ODEs from observed trajectories. In *International Conference on Learning Representations*, 2022.

[243] A. Quarteroni and G. Rozza. Reduced order methods for modeling and computational reduction, 2014.

[244] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. Ramadhan. Universal differential equations for scientific machine learning. *ArXiv*, abs/2001.04385, 2020.

[245] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[246] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[247] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks, 2019. URL https://openreview.net/forum?id=r1gR2sC9FX.

[248] M. Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018.

[249] 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*, 378: 686 – 707, 2019.

[250] S. Rasp. Lorenz 96 is too easy! machine learning research needs a more realistic toy model, July 2019. URL https://raspstephan.github.io/blog/lorenz-96-is-too-easy/. last accessed May 2022.

[251] S. Rasp. Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and lorenz 96 case study (v1.0). *Geoscientific Model Development*, 13(5):2185–2196, 2020.

[252] S. Rasp, M. S. Pritchard, and P. Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

[253] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey. Weatherbench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11), 2020.

[254] D. Ravì, A. B. Szczotka, S. P. Pereira, and T. Vercauteren. Adversarial training with cycle consistency for unsupervised super-resolution in endomicroscopy. *Medical image analysis*, 53:123–131, 2019.

[255] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195 – 204, 2019.

[256] G. Reiersen, D. Dao, B. Lütjens, K. Klemmer, K. Amara, A. Steinegger, C. Zhang, and X. Zhu. Reforestree: A dataset for estimating tropical forest carbon stock with deep learning and aerial imagery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12119–12125, Jun. 2022.

[257] C. Requena-Mesa, M. Reichstein, M. Mahecha, B. Kraft, and J. Denzler. Predicting landscapes from environmental conditions using generative networks. In *German Conference on Pattern Recognition*, pages 203–217. Springer, 2019.

[258] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1530–1538. JMLR.org, 2015.

[259] C. Robinson, L. Hou, K. Malkin, R. Soobitsky, J. Czawlytko, B. Dilkina, and N. Jojic. Large scale high-resolution land cover mapping with multi-resolution data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[260] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. S. Luccioni, T. Maharaj, E. D. Sherwin, S. K. Mukkavilli, K. P. Kording, C. P. Gomes, A. Y. Ng, D. Hassabis, J. C. Platt, F. Creutzig, J. Chayes, and Y. Bengio. Tackling climate change with machine learning. *ACM Comput. Surv.*, 55(2), feb 2022. ISSN 0360-0300.

[261] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[262] J. N. Rooney-Varga, J. D. Sterman, E. Fracassi, T. Franck, F. Kapmeier, V. Kurker, E. Johnston, A. P. Jones, and K. Rath. Combining role-play with interactive simulation to motivate informed climate action: Evidence from the world climate simulation. *PLoS ONE*, 13, August 2018.

[263] J. N. Rooney-Varga, F. Kapmeier, J. D. Sterman, A. P. Jones, M. Putko, and K. Rath. The climate action simulation. *Simulation & Gaming*, 51(2):114–140, 2020. URL https://en-roads.climateinteractive.org/.

[264] J. N. Rooney-Varga, M. Hensel, C. McCarthy, K. McNeal, N. Norfles, K. Rath, A. H. Schnell, and J. D. Sterman. Building consensus for ambitious climate action through the world climate simulation. *Earth's Future*, 9(12), 2021.

[265] I. Rugina, R. Dangovski, M. Veillette, P. Khorrami, B. Cheung, O. Simek, and M. Soljacic. Meta-learning and self-supervised pretraining for real world image translation, 2022.

[266] S. Rühling Cachay, E. Erickson, A. F. C. Bucker, E. Pokropek, W. Potosnak, S. Osei, and B. Lütjens. Graph Deep Learning for Long Range Forecasting. In *EGU General Assembly Conference Abstracts*, pages EGU21–9141, Apr. 2021.

[267] J. Runkle, K. Kunkel, R. Frankson, D. Easterling, A. DeGaetano, B. Stewart, W. Sweet, and J. Spaccio. Massachusetts state climate summary 2022, 2022.

[268] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi. Palette: Image-to-image diffusion models. *CoRR*, abs/2111.05826, 2021.

[269] D. Sailor, T. Hu, X. Li, and J. Rosen. A neural network approach to local downscaling of gcm output for assessing wind power implications of climate change. *Renewable Energy*, 19(3):359–378, 2000. ISSN 0960-1481.

[270] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[271] S. Santamaria, D. Dao, B. Lütjens, and C. Zhang. Truebranch: Metric learning-based verification of forest conservation projects. In *ICLR 2020 Workshop on Tackling Climate Change with Machine Learning*, 2020.

[272] C.-F. Schleussner, T. K. Lissner, E. M. Fischer, J. Wohland, M. Perrette, A. Golly, J. Rogelj, K. Childers, J. Schewe, K. Frieler, M. Mengel, W. Hare, and M. Schaeffer. Differential climate impacts for policy-relevant limits to global warming: the case of 1.5 °c and 2 °c. *Earth System Dynamics*, 7(2):327–351, 2016.

[273] V. Schmidt, A. Luccioni, S. K. Mukkavilli, N. Balasooriya, K. Sankaran, J. Chayes, and Y. Bengio. Visualizing the consequences of climate change using cycle-consistent adversarial networks. *International Conference on Learning Representations (ICLR) Workshop on Tackling Climate Change with AI*, 2019.

[274] T. Schneider, J. Teixeira, C. S. Bretherton, F. Brient, K. G. Pressel, C. Schär, and A. P. Siebesma. Climate goals and computing the future of clouds, jan 2017.

[275] E. Schönfeld, V. Sushko, D. Zhang, J. Gall, B. Schiele, and A. Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning Representations*, 2021.

[276] S. R. Sheppard. Landscape visualisation and climate change: the potential for influencing perceptions and behaviour. *Environmental Science and Policy*, 8 (6):637–654, 2005. ISSN 1462-9011. Mitigation and Adaptation Strategies for Climate Change.

[277] S. R. Sheppard. *Visualizing Climate Change A Guide to Visual Communication of Climate Change and Developing Local Solutions*. Taylor and Francis Group, March 2012.

[278] Y. Shin, J. Darbon, and G. Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. *Communications in Computational Physics*, 28(5):2042–2074, 2020.

[279] P. Y. Simard, D. Steinkraus, J. C. Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

[280] P. Singh and N. Komodakis. Cloud-gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1772–1775, 2018.

[281] J. Sirignano, J. F. MacArt, and J. B. Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, 423, 2020.

[282] P. Slovic, M. L. Finucane, E. Peters, and D. G. MacGregor. Risk as analysis and risk as feelings: Some thoughts about affect, reason, risk, and rationality. *Risk Analysis*, 24(2):311–322, 2004.

[283] R. C. Smith. Uncertainty quantification: Theory, implementation, and applications. In *Computational science and engineering*, page 382. SIAM, 2013.

[284] Snagglebit. Sloth nesting dolls - hand painted modern russian matryoshka doll set, 2022. URL https://www.etsy.com/ie/listing/690540535/sloth-nesting-dolls-hand-painted-modern. last accessed 01/22.

[285] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[286] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[287] K. Stachenfeld, D. B. Fielding, D. Kochkov, M. Cranmer, T. Pfaff, J. Godwin, C. Cui, S. Ho, P. Battaglia, and A. Sanchez-Gonzalez. Learned simulators for turbulence. In *International Conference on Learning Representations*, 2022.

[288] K. Stengel, A. Glaws, D. Hettinger, and R. N. King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.

[289] J. N. K. Steven L. Brunton. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, February 2019.

[290] B. A. Storer, M. Buzzicotti, H. Khatri, S. M. Griffies, and H. Aluie. Global energy spectrum of the general oceanic circulation. *Nature Communications*, 13(1):5314, Sep 2022. ISSN 2041-1723.

[291] B. Strauss. Surging seas: Sea level rise analysis, 2015. URL https://sealevel.climatecentral.org/.

[292] B. H. Strauss, S. A. Kulp, D. J. Rasmussen, and A. Levermann. Unprecedented threats to cities from multi-century sea level rise. *Environmental Research Letters*, 16(11):114015, oct 2021.

[293] A. Subel, A. Chattopadhyay, Y. Guan, and P. Hassanzadeh. Data-driven subgrid-scale modeling of forced burgers turbulence using deep learning with generalization to higher reynolds numbers via transfer learning. *Physics of Fluids*, 33(3):031702, 2021.

[294] C. Sweeney and P. Lynch. Adaptive post-processing of short-term wind forecasts for energy applications, 2011.

[295] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[296] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler. Long range arena: A benchmark for efficient transformers, 2020.

[297] E. F. Thomas, C. McGarty, and K. I. Mavor. Transforming "apathy into movement":the role of prosocial emotions in motivating action for social change. *Personality and Social Psychology Review*, 13(4):310–333, 2009.

[298] N. Thomas, T. E. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018.

[299] T. Thornes, P. Duben, and T. Palmer. On the use of scale-dependent precision in earth system modelling. *Quarterly Journal of the Royal Meteorological Society*, 143(703):897–908, 2017.

[300] J. Thuemmel, M. Karlbauer, S. Otte, C. Zarfl, G. Martius, N. Ludwig, T. Scholten, U. Friedrich, V. Wulfmeyer, B. Goswami, and M. V. Butz. Inductive biases in deep learning models for weather prediction, 2023.

[301] A. E. R. Tilmann Gneiting. Weather forecasting with ensemble methods. *Science*, 310:248–249, 2005.

[302] B. A. Toms, E. A. Barnes, and I. Ebert-Uphoff. Physically interpretable neural networks for the geosciences: Applications to earth system variability. *Journal of Advances in Modeling Earth Systems*, 12(9), 2020.

[303] C. Tong, L. Yanran, P. J. Athul, B. Yoshua, and L. Wenjie. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[304] K. E. Trenberth, J. T. Fasullo, and T. G. Shepherd. Attribution of climate extreme events. *Nature Climate Change*, 5(8):725–730, Aug 2015.

[305] K. Um, R. Brand, Y. Fei, P. Holl, and N. Thuerey. Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *Advances in Neural Information Processing Systems*, 2020.

[306] United Nations Climate Change. Paris agreement - status of ratification, 2016. URL https://unfccc.int/process/the-paris-agreement/status-of-ratification.

[307] U.S. Energy Information Administration. Annual energy outlook 2021, 2021.

[308] USDA-FSA-APFO Aerial Photography Field Office. National Geospatial Data Asset National Agriculture Imagery Program (NAIP) Imagery, 2019. URL http://gis.apfo.usda.gov/arcgis/rest/services/NAIP.

[309] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R. Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1663–1672, New York, NY, USA, 2017. Association for Computing Machinery.

[310] M. Veillette, S. Samsi, and C. Mattioli. Sevir : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22009–22019. Curran Associates, Inc., 2020.

[311] M. S. Veillette, E. P. Hassey, C. J. Mattioli, H. Iskenderian, and P. M. Lamey. Creating synthetic radar imagery using convolutional neural networks. *Journal of Atmospheric and Oceanic Technology*, 35(12):2323–2338, 2018.

[312] P. Wang, J. Yuval, and P. A. O'Gorman. Non-local parameterization of atmospheric subgrid processes with neural networks, 2022.

[313] S. Wang, A. Corner, D. Chapman, and E. Markowitz. Public engagement with climate imagery in a changing digital landscape. *WIREs Climate Change*, 9(2): e509, 2018.

[314] S. Wang, H. Wang, and P. Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science Advances*, 7(40), 2021.

[315] S. Wang, H. Wang, and P. Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021. ISSN 0045-7825.

[316] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans.

In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 8798–8807, 2018. URL https://github.com/NVIDIA/pix2pixHD.

[317] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[318] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[319] D. Watson-Parris, Y. Rao, D. Olivie, O. Seland, P. Nowack, G. Camps-Vall, P. Stier, S. Bouabid, M. Dewey, E. Fons, J. Gonzalez, P. Harder, K. Jeggle, J. Lenhardt, P. Manshausen, M. Novitasari, L. Ricard, and C. Roesch. Climatebench v1.0: A benchmark for data-driven climate projections. *Journal of Advances in Modeling Earth Systems*, 14(10), 2022.

[320] M. J. Webb, A. P. Lock, C. S. Bretherton, S. Bony, J. N. Cole, A. Idelkadi, S. M. Kang, T. Koshiro, H. Kawai, T. Ogura, R. Roehrig, Y. Shin, T. Mauritsen, S. C. Sherwood, J. Vial, M. Watanabe, M. D. Woelfle, and M. Zhao. The impact of parametrized convection on cloud feedback. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373(2054), nov 2015.

[321] T. Weber, A. Corotan, B. Hutchinson, B. Kravitz, and R. Link. Technical note: Deep learning for creating surrogate models of precipitation in earth system models. *Atmospheric Chemistry and Physics*, 20(4):2303–2317, 2020.

[322] M. Wehner, P. Gleckler, and J. Lee. Characterization of long period return values of extreme daily temperature and precipitation in the cmip6 models: Part 1, model evaluation. *Weather and Climate Extremes*, 30:100283, 2020. ISSN 2212-0947.

[323] P. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. doi: 10.1109/5.58337.

[324] D. S. Wilks. Effects of stochastic parametrizations in the lorenz '96 system. *Quarterly Journal of the Royal Meteorological Society*, 131(606):389–407, 2005.

[325] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Comput. Surv.*, jan 2022.

[326] S. William. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Elsevier, 1991.

[327] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1432–1467, 2015.

[328] P. with Code. Semantic segmentation on cityscapes test. URL https://paperswithcode.com/sota/semantic-segmentation-on-cityscapes. last accessed 04/23.

[329] J. Woetzel, D. Pinner, H. Samandari, H. Engel, M. Krishnan, B. Boland, and C. Powis. Climate risk and response: Physical hazards and socioeconomic impacts, January 2020.

[330] J.-L. Wu, H. Xiao, and E. Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids*, 3:074602, Jul 2018.

[331] J.-L. Wu, K. Kashinath, A. Albert, D. Chirila, Prabhat, and H. Xiao. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics*, 406, 2020.

[332] Y. Xie, E. Franz, M. Chu, and N. Thuerey. Tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.*, 37(4), jul 2018.

[333] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.

[334] Z. Xu, Y. Han, C.-Y. Tam, Z.-L. Yang, and C. Fu. Bias-corrected cmip6 global dataset for dynamical downscaling of the historical and future climate (1979–2100). *Scientific Data*, 8(1):293, Nov 2021. ISSN 2052-4463.

[335] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLOS Computational Biology*, 16(11):1–19, 11 2020.

[336] Y. Yin, V. L. Guen, J. Dona, E. de Bézenac, I. Ayed, N. Thome, and P. Gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021 (12):124012, dec 2021.

[337] J. Yuval and P. A. O'Gorman. Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, 11, 2020.

[338] J. Yuval, P. A. O'Gorman, and C. N. Hill. Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letter*, 48:e2020GL091363, 2021.

[339] Y. Zeng, J.-L. Wu, and H. Xiao. Enforcing imprecise constraints on generative adversarial networks for emulating physical systems. *Communications in Computational Physics*, 30(3):635–665, 2021.

[340] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397:108850, 2019.

[341] L. Zhang, J. Han, H. Wang, R. Car, and W. E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120, Apr 2018.

[342] L. Zhang, G. Wang, and G. B. Giannakis. Real-time power system state estimation and forecasting via deep unrolled neural networks. *IEEE Transactions on Signal Processing*, 67(15):4069–4077, 2019.

[343] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[344] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81, 2020.

[345] S. Zhou, A. Luccioni, G. Cosne, M. S. Bernstein, and Y. Bengio. Establishing an evaluation metric to quantify climate change image realism, 2019.

[346] S. Zhou, A. Luccioni, G. Cosne, M. S. Bernstein, and Y. Bengio. Establishing an evaluation metric to quantify climate change image realism. *Machine Learning: Science and Technology*, 1(2):025005, 2020.

[347] X.-H. Zhou, J. Han, and H. Xiao. Learning nonlocal constitutive models with neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113927, 2021. ISSN 0045-7825.

[348] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[349] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NeurIPS) 30*, pages 465–476. 2017. URL https://github.com/junyanz/BicycleGAN.

[350] X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5:8–36, 12 2017.