https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Deep learning of subgrid-scale parametrisations for short-term forecasting of sea-ice dynamics with a Maxwell-Elasto-Brittle rheology

Tobias Sebastian Finn[1], Charlotte Durand[1], Alban Farchi[1], Marc Bocquet[1], Yumeng Chen[2], Alberto Carrassi[2,3], and Véronique Dansereau[4]

[1]CEREA, École des Ponts and EDF R&D, Île-de-France, France
[2]Dept. of Meteorology and NCEO, University of Reading, Reading, United Kingdom
[3]Dept of Physics and Astronomy "Augusto Righi", University of Bologna, Italy
[4]Université Grenoble Alpes, CNRS, Grenoble INP, Laboratoire 3SR, Grenoble, France

Correspondence: Tobias Sebastian Finn (tobias.finn@enpc.fr)

Abstract. We introduce a scalable approach to parametrise the unresolved subgrid-scale of sea-ice dynamics with deep learning techniques. We apply this data-driven approach to a regional sea-ice model that accounts exclusively for dynamical processes with a Maxwell-Elasto-Brittle rheology. Our channel-like model setup is driven by a wave-like wind forcing, which generates examples of sharp transitions between unfractured and fully-fractured sea ice. Using a convolutional U-Net architecture, the parametrising neural network extracts multiscale and anisotropic features and, thus, includes important inductive biases needed for sea-ice dynamics. The neural network is trained to correct all nine model variables at the same time. With the initial and forecast state as input into the neural network, we cast the subgrid-scale parametrisation as model error correction, needed to correct unresolved model dynamics. We test the here-proposed approach in twin experiments, where forecasts of a low-resolution forecast model are corrected towards high-resolution truth states for a forecast lead time of about $10\ \mathrm{min}$. At this lead time, our approach reduces the forecast errors by more than $75\ \%$, averaged over all model variables. The neural network learns hereby a physically-explainable input-to-output relation. Furthermore, cycling the subgrid-scale parametrisation together with the geophysical model improves the short-term forecast up to one hour. We consequently show that neural networks can parametrise the subgrid-scale for sea-ice dynamics. We therefore see this study as an important first step towards hybrid modelling to forecast sea-ice dynamics on an hourly to daily timescale.

1 Introduction

Sea-ice models with Elasto-Brittle rheologies (e.g. Rampal et al., 2016) reproduce the dynamics of sea ice at an unprecedented resolution and accuracy (Rabatel et al., 2018; Bouchat et al., 2022; Boutin et al., 2022). These models represent the observed temporal and spatial scale-invariance of the sea-ice deformation and drift across multiple scales, up to the resolution of a single grid cell at the mesoscale (Dansereau et al., 2016; Rampal et al., 2019; Ólason et al., 2021). Elasto-Brittle rheologies parametrise the unresolved subgrid-scale processes associated with brittle fracturing through a progressive damage framework (Tang, 1997; Amitrano et al., 1999; Girard et al., 2011). Such framework connects the elastic modulus of the material at the

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

grid cell level to the degree of fracturing at the subgrid-scale. Comprised between 0, undamaged, and 1, completely damaged material, the fracturing is represented by the level of *damage*. When the internal stress locally exceeds a given damage criterion, the level of damage increases and the elastic modulus decreases, thereby reducing the local effective stress. Excessive stress is elastically redistributed throughout the material, causing overcritical stress elsewhere. Hence, the damage is highly localised and progressively propagated through the material, which also leads to a strong localisation of the deformation. The Maxwell-Elasto-Brittle rheology (Dansereau et al., 2016) adds to this framework the concept of an "apparent" viscosity. Coupled to the level of damage, the added viscosity allows accounting for the relaxation of stresses by all permanent deformations within a fractured sea-ice cover. Although models with such rheologies successfully simulate the observed scaling properties of sea-ice deformation, they locally underestimate very high convergence and shear rates in some instances (Ólason et al., 2022). Thus, some important, possibly subgrid-scale, processes are still unresolved at the mesoscale or unrepresented in Elasto-Brittle rheologies and their damage parametrisations.

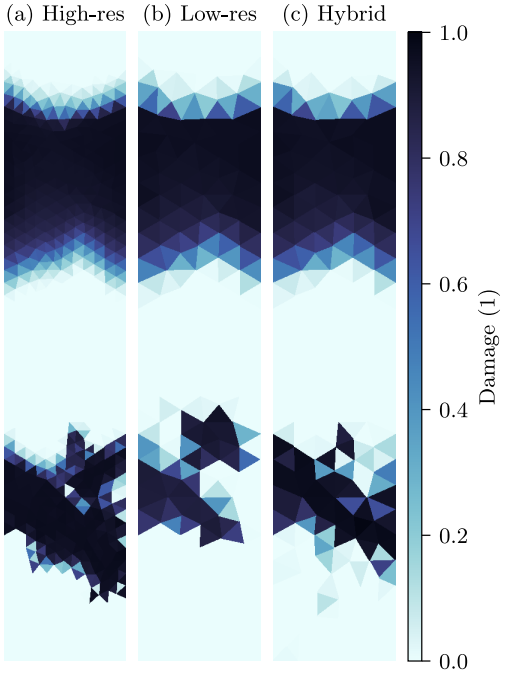

Figure 1. Snapshot of sea-ice damage after a one-hour forecast with the here-used sea-ice dynamics only-model. The initial conditions of the high-resolution truth (a, 4 km resolution) are projected into a low-resolution space with 8 km resolution. Started from these projected initial conditions, the low-resolution forecast (b) is unable to reproduce the dynamics of the truth. Running the low-resolution forecast in a hybrid mode (c) together with a subgrid-scale-parametrising neural network leads to a better representation of the sea-ice dynamics, which improves the forecast, tested up to one hour.

To exemplify the impact of these unresolved subgrid-scale processes on the sea-ice dynamics, and to see how deep learning can remedy these issues, we perform twin experiments with a sea-ice model that depicts exclusively the dynamics in a Maxwell-

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Elasto-Brittle rheology (Dansereau et al., 2016, 2017, 2021). By imposing a wave-like atmospheric wind forcing in a channel-like setup, we generate marginal ice zones with sharp transitions from undamaged sea ice to almost totally damaged sea ice. Starting two model simulations at different resolutions from the same, possibly projected, initial conditions can lead to different trajectories. Such different instances of sea-ice dynamics are caused by differently integrated processes. Consequently, the sea-ice damage can already significantly differ after one hour of simulation, as shown in Fig. 1. Here, the low-resolution simulation (b) misses the rapidly developed opening of sea ice in the high-resolution simulation (a). In this study, we introduce a baseline hybrid modelling approach to correct the missing processes with deep learning. By parametrising the subgrid-scale, the hybrid model better reproduces the temporal evolution of high-resolution simulations at the lower resolution (Fig. 1c).

Subgrid-scale parametrisations with machine learning have already been proved useful for other Earth system components (Brenowitz and Bretherton, 2018; Beucler et al., 2021; Irrgang et al., 2021). In the atmosphere, cloud processes can be learned from emulating super-parametrised or super-resolved models within a lower-resolution model (Gentine et al., 2018; Rasp et al., 2018; Seifert and Rasp, 2020). Additionally, machine learning can parametrise turbulent dynamics in the atmosphere (Beck and Kurz, 2021; Cheng et al., 2022) and in the ocean (Zanna and Bolton, 2020; Guillaumin and Zanna, 2021).

To predict the sea-ice concentration, purely data-driven surrogate models can replace geophysical models at daily (Liu et al., 2021) and seasonal forecast horizons (Andersson et al., 2021). Furthermore, multi-layer perceptrons can emulate granular simulations of ocean-sea-ice interactions, allowing to parametrise the effect of ocean waves onto the sea ice (Horvat and Roach, 2022). In this study, we take another point of view and show more generally that subgrid-scale processes for sea-ice dynamics can be parametrised with deep learning, correcting all forecast model variables at the same time.

The dynamics of sea-ice impose hereby new challenges for neural networks (NNs) that should parametrise the subgrid-scale:

- The marginal ice zone characterises a rapid spatial transition from open water to thick sea ice within a few kilometres (Horvat, 2021; Bennetts et al., 2022). As current sea-ice models represent the marginal ice zone in a band of few pixels, the zone can appear as a non-continuous jump function within the data. For such discrete-continuous mixture data distributions, NNs that simply learn to regress into the future tend to diffuse and blur the target (Ayzel et al., 2020; Ravuri et al., 2021), caused by their pixel-wise loss function. A correct representation of the marginal ice zone can thus induce problems within the training of the NN, resulting into a diffusion of the normally concentrated zone.

- In Elasto-Brittle models, the handling of the internal stress depends on the fragmentation of sea ice. This dependency also leads to different forecast error distributions for different fragmentation levels, even for variables only indirectly related to the stress, like the sea-ice thickness. Consequently, a NN that should correct the forecast error has to be trained across a range of fragmentation levels and should be able to output multimodal predictions in the best case.

- As sea ice is scale-invariant up to the kilometre-scale, fragmentation of sea-ice propagates from small, unresolved, scales to the larger, resolved, scales. Because the small scales are unresolved, the appearance of linear kinematic features seems to be stochastic from the resolved macro-scale point of view. Additionally, such features are inherently multifractal and propagate in an anisotropic medium (Wilchinsky and Feltham, 2006, 2011).

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Finally, the found subgrid-scale parametrisation approach should be scalable to a range of resolutions, from regional models used in this study to Arctic-wide models, like neXtSIM (Rampal et al., 2016; Ólason et al., 2022).

As a first step towards solving these challenges for NNs, we present the aforementioned twin experiments with a regional model. We create a training dataset with an ensemble of forcing parameters and initial cohesion, which describes the strength of the sea ice against stress. We simulate high-resolution truth trajectories at a $4\ \mathrm{km}$ resolution. Projected into a lower resolved space at a $8\ \mathrm{km}$ resolution, these truth trajectories are used as reference and initial condition for low-resolution forecasts. As the low-resolution forecast resolves fewer processes than the truth, the NN has to account for the unresolved subgrid-scale processes to correct model errors. In the following, we will thus use model error correction and subgrid-scale parametrisation interchangeably. We train the NN to correct forecasts for a lead time of $10\ \mathrm{min}$ and $8\ \mathrm{s}$ (a multiplier of our $16\ \mathrm{s}$ model time step), with the approach commonly known as the additive resolvent correction, which targets the difference between forecast and projected truth. When cycled during forecasting, the output of the NN is added to the forecast of the geophysical model; the model is then a hybrid machine-learning-physical model.

With such experiments, we have found a baseline deep learning architecture, based on the U-Net approach (Ronneberger et al., 2015) with applied tricks from the ConvNeXt architecture (Liu et al., 2022). As the model is discretised on a triangular grid, we introduce a projection step to a high-resolution Cartesian grid, where the aforementioned U-Net extract features. The extracted features are projected back into the original triangular space and then combined by linear functions, shared across all grid points, into an additive model error correction. The convolutional U-Net makes use of spatial correlations and extracts anisotropic features at multiple scales. Balanced with maximum likelihood during training, a single NN corrects all nine forecast model variables at the same time.

Our baseline architecture is only trained to correct the forecast for a lead time of $10\ \mathrm{min}$ and $8\ \mathrm{s}$. When the correction is cycled with the forecast model, uncorrected errors accumulate for longer forecast horizons, reducing the efficiency of our error correction. Nevertheless, we present first promising results for short-term forecasting (up to $60\ \mathrm{min}$), as showcased in Fig. 1c.

We briefly present the regional sea-ice model in Sect. 2. The different components of our deep-learning-based model error correction are introduced in Sect. 3. Our learning strategy and the specific features of the twin experiments are described in Sect. 4. Results are given in Sect. 5, summary and discussion in Sect. 6, and final, concise, conclusions in Sect. 7.

2 A regional sea-ice model with a Maxwell-Elasto-Brittle rheology

In the following paragraphs, we will briefly describe the most important properties of the regional sea-ice model used in this study. Appendix A presents the model parameters used in this study. For a more technical presentation of the model, we refer the reader to Dansereau et al. (2016, 2017).

Compared to Arctic and pan-Arctic sea-ice models, like neXtSIM (Rampal et al., 2016; Ólason et al., 2022), this model is a regional standalone model that accounts exclusively for dynamical processes. Like most sea-ice models, it is as well two-dimensional and based on a plane stress approximation. Nine variables constitute its prognostic state vector: sea-ice velocity in x- and y-direction, the three stress components, level of damage, cohesion, thickness, and concentration.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Atmospheric wind stress is the sole external mechanical forcing, whereas the ocean beneath the sea ice is assumed to be at rest. Given the small horizontal extent of our simulation domain (see Fig. 2), we also neglect the Coriolis force.

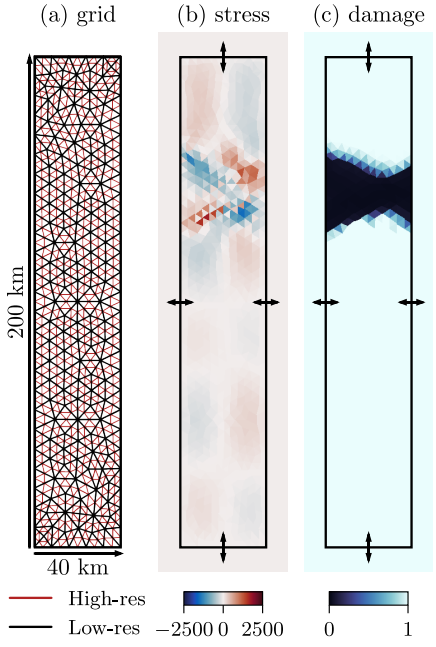

Figure 2. (a) The model domain with the high- (red) and low-resolution (black) grid; (b) a snapshot of the stress, σ_{xy} in Pa, where the arrows correspond to von Neumann boundary conditions on all four sides; (c) a snapshot of the damage, where the arrows correspond to an inflow of undamaged sea ice on all four sides. Both snapshots are taken at an arbitrary time and represent a commonly encountered case in our dataset.

The Maxwell-Elasto-Brittle rheology from Dansereau et al. (2016) specifies the constitutive law of the model. It combines elastic deformations, with an associated elastic modulus, and permanent deformations, with an associated apparent viscosity. The ratio of the viscosity to the elastic modulus defines the rate at which stresses are dissipated into permanent deformations. Both variables are coupled to the level of damage: deformations are strictly elastic over undamaged ice and completely irreversible over fully-damaged ice. The level of damage propagates in space and time due to damaging and healing. Ice is damaged, and thus the level of damage increases, when and where the stresses are overcritical according to a Mohr-Coloumb damage criteria (Dansereau et al., 2016). This mechanism parametrises the role of brittle failure processes from the subgrid-scale onto the mechanical weakening of ice at the mesoscale. Reducing the level of damage, ice is healed at a constant rate, which parameterises the effect of subgrid-scale refreezing of cracks onto the mechanical strengthening of the ice. By neglecting thermodynamical sources and sinks in the model, cohesion, thickness, and area are solely driven by advection and diffusion processes; the prognostic variable for the thickness is hereby the thickness of the ice-covered portion of a grid cell, defined as the ratio between thickness and area. For the prognostic sea-ice thickness and area, a simple volume-conserving scheme is introduced to represent the mechanical redistribution of the ice thickness associated with ridging (Dansereau et al., 2017).

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

The model equations are discretised in time using a first-order, Eulerian implicit scheme. Due to the coupling of the mechanical parameters to the level of damage, the constitutive law is non-linear, and a semi-implicit fixed point scheme is used to iteratively solve the momentum, the constitutive, and the damage equations. Within a model integration time step, these three fields are updated first. Cohesion, thickness, and area are updated secondly, using the already updated fields of sea-ice velocity and damage.

The equations are discretised in space by a discontinues Galerkin scheme. The velocity and forcing components are defined by linear, first-order, continuous finite elements. All other variables and derived quantities like deformation and advection are characterised by constant, zeroth-order, discontinuous elements. As the nodes are shared in the first-order elements, there are more grid points for all variables that are defined as zeroth-order elements than for the velocity and forcing components. The model is implemented in C++ and uses the *Rheolef* library (Version 6.7, Saramito, 2022).

Our virtual area spans $40\ \mathrm{km} \times 200\ \mathrm{km}$: a channel-like setup, which is nevertheless anisotropy-allowing. The model is based on a triangular grid with an average triangle size of $8\ \mathrm{km}$ for the low-resolution forecasts. The grid for the high-resolution truth trajectories is a refined version of the low-resolution with a spacing of $4\ \mathrm{km}$ (Fig. 2a).

If not otherwise stated, we initialise the simulations with undamaged sea ice, the velocity and stress components are set to zero, and the area and thickness to one. The cohesion is initialised with a random field, drawn from a uniform distribution between $5 \times 10^3\ \mathrm{Pa}$ and $1 \times 10^4\ \mathrm{Pa}$. We use von-Neumann boundary conditions on all four sides (Fig. 2, b), with an inflow of undamaged sea ice (Fig. 2, c) and a random cohesion, again between $5 \times 10^3\ \mathrm{Pa}$ and $1 \times 10^4\ \mathrm{Pa}$. The model configuration thus simulates a zoom into an (almost) undamaged region of sea ice, e.g. in the centre of the Arctic.

For the atmospheric wind forcing, we impose a wave-like velocity in y-direction and no velocity in x-direction. Because of the anisotropy, the sea ice can nevertheless move in x-direction. Depending on its length scale and amplitude, the wave-like forcing generates cases of rapid transitions between undamaged and fully-damaged sea ice. As spin-up for the high-resolution truth trajectories, the wind forcing is linearly increased over the course of the first simulation day. The parameters of the wind forcing are randomly drawn, as described in Sect. 4. The wind forcing is updated at each model integration time step ($8\,\mathrm{s}$ for the simulations at high resolution and $16\,\mathrm{s}$ for the simulations at low resolution).

3 A deep learning based subgrid-scale parametrisation

To represent spatial correlations and anisotropic features, we use convolutional neural networks (CNNs). We cast the learning of a subgrid-scale parametrisation as model error correction (Sect. 3.1), applied in a post-processing step, after the model forecast is generated. Since the Maxwell-Elasto-Brittle model is spatially discretised on a triangular space (cf. Sect. 2), we introduce a linear projection operator $\mathbf{P}$ (Sect. 3.2), interpolating from the triangular space to a Cartesian space that has a higher resolution, and where convolutions can easily be applied. After this projection, we apply a U-Net (Sect. 3.3) to extracts features in Cartesian space from the projected input fields. These features are then projected back into the triangular space with the pseudo-inverse of the projection operator $\mathbf{P}^\dagger$. There, linear functions combine pixel-wise (i.e., processing each element-defining grid point independently) the extracted features. Each linear function is shared across all grid points for each predicted

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

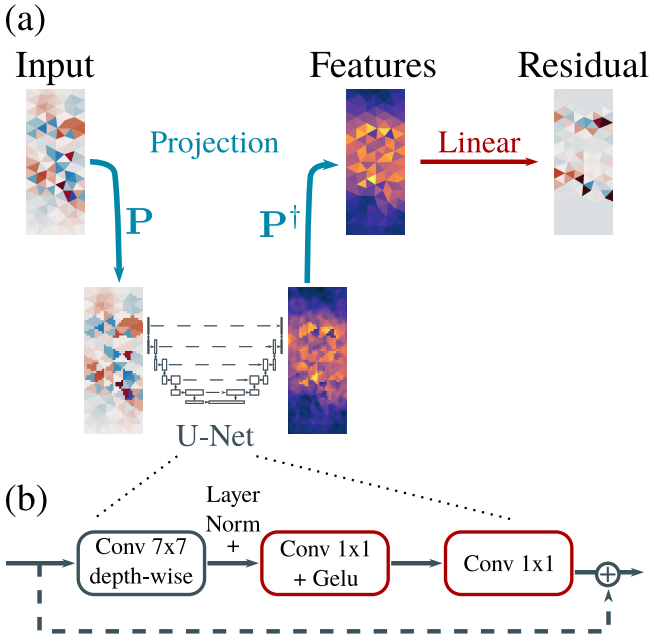

Figure 3. In our deep learning approach (a), the input fields are projected by a linear projection operator $\mathbf{P}$ from their triangular space into a Cartesian space that has a higher resolution, where the U-Net extracts features. Back-projected into triangular space by the pseudo-inverse of the linear projection operator $\mathbf{P}^\dagger$, these features are linearly combined to obtain the predicted residuals. The U-Net consists of multiple ConvNeXt-like blocks (b). A first convolutional layer extracts depth-wise (i.e. without mixing the channels) spatial features with a kernel size of 7×7. A 1×1 convolution with a Gaussian error linear unit (Gelu) activation function combines the layer-normalised channels. Together with a last 1×1 convolution, this path mixes the channels. Finally, a residual connection from the input of the block is added to the output of the block.

residual variable. The NN predicts the residuals for all nine forecast model variables at the same time with one shared U-Net. By sharing the U-Net across tasks, the NN has to learn patterns and features for error correction of all variables. To weight the nine different loss functions, we make use of a maximum likelihood approach (Sect. 3.4). This proposed pipeline (visualised in Fig. 3) can be seen as a baseline that enables a subgrid-scale parametrisation with deep learning for sea-ice dynamics, correcting all model variables at the same time.

3.1 Problem formulation

We cast the subgrid-scale parametrisation as forecast error correction, solvable with a NN in a post-processing step, after the forecast model has been applied. As input for the NN, we use the initial state $\boldsymbol{x}^{\text{in}}_{t-1}$ at time $t-1$ and the forecasted state $\boldsymbol{x}^{\text{f}}_t$, propagated from $t-1$ to t with the (non-corrected) geophysical model $\boldsymbol{x}^{\text{f}}_t = \mathcal{M}(\boldsymbol{x}^{\text{in}}_{t-1})$; to simplify the notation, time has been discretised, $t \in \mathbb{N}$. Based on the input fields, the NN $f(\boldsymbol{x}^{\text{in}}_{t-1}, \boldsymbol{x}^{\text{f}}_t, \boldsymbol{\phi})$ with its parameters $\boldsymbol{\phi}$ should predict the residual $\Delta\boldsymbol{x}_t = \boldsymbol{x}^{p}_t - \boldsymbol{x}^{\text{f}}_t$, defined as the difference between the projected truth $\boldsymbol{x}^{p}_t$ and the forecasted state. For cycling the error

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

correction together with the geophysical model, we can add the predicted residual to the forecast, resulting into the corrected forecast $\boldsymbol{x}_t^{\mathsf{c}}$, then used as subsequent initial state for the geophysical model. The equations for a complete forecast cycle from $t-1$ to t then read

$$\boldsymbol{x}_t^{\mathsf{f}} = \mathcal{M}(\boldsymbol{x}_{t-1}^{\mathsf{in}}),$$
$$\boldsymbol{x}_t^{\mathsf{c}} = \boldsymbol{x}_t^{\mathsf{f}} + f(\boldsymbol{x}_{t-1}^{\mathsf{in}}, \boldsymbol{x}_t^{\mathsf{f}}, \boldsymbol{\phi}).$$

When the correction is cycled with the forecast model, the initial state for the following cycle is simply the corrected state,

$$\boldsymbol{x}_t^{\mathsf{in}} = \boldsymbol{x}_t^{\mathsf{c}}.$$

3.2 The projection operator

For the Cartesian space, we chose a discretisation of 32×128 elements in the x- and y-directions, defined by constant, zeroth-order, Cartesian elements evenly distributed in the $40\ \mathrm{km} \times 200\ \mathrm{km}$ domain. As each Cartesian element has a resolution of $1.25\ \mathrm{km} \times 1.5625\ \mathrm{km}$, the Cartesian space has a higher resolution than the original triangular space ($\sim 8\ \mathrm{km}$). Using such a super-resolution mitigates the loss of information caused by the projection. Furthermore, the NN can learn interactions between variables on a smaller-scale than used in the model, which helps to parametrise the subgrid-scale, as we will see in Sect. 5.1.

As projection operator $\mathcal{P}$, we use Lagrange interpolation from the original triangular elements to the Cartesian ones. For the velocity and forcing components, defined as first-order elements, this interpolation corresponds to a (linear) Barycentric interpolation and to a nearest neighbour interpolation for all other variables, defined as zeroth-order elements; $\mathcal{P}$ thus reduces to a linear operator, hereafter written $\boldsymbol{P}$. Because of the higher resolution, there are multiple Cartesian elements per triangular element, and the inverse of the operator does not exist as the linear system is over-determined. Consequently, in order to define the backward projection from the Cartesian space to the triangular space, we use the Moore–Penrose pseudo-inverse $\boldsymbol{P}^\dagger$. Since the rank of $\boldsymbol{P}$ is by construction equal to the dimension of the triangular space, i.e. its column number, the pseudo-inverse is in our case equal to $\boldsymbol{P}^\dagger = (\boldsymbol{P}^\top \boldsymbol{P})^{-1} \boldsymbol{P}^\top$, where $\boldsymbol{P}^\top$ corresponds to the transposed operator. Note, for coarse Cartesian spaces, the mapping from Cartesian space to triangular space can be non-surjective, meaning that not all triangular elements are covered by at least one Cartesian element: the pseudo-inverse is in this case rank deficient.

In the case of zeroth-order discontinuous Galerkin elements, the projection operator assigns to each Cartesian element one triangular element. The back-projection operator then corresponds to an averaging of the Cartesian elements into their assigned triangular element. This averaging can be seen as a type of ensembling the information from smaller, normally unresolved, scales to larger, resolved scales. We have implemented this projection operator as a NN layer with fixed weights in PyTorch.

3.3 The U-Net feature extractor

We use CNNs in Cartesian space. The feature extractor should be able to extract multiscale features, and to represent rapid spatial transitions, which might occur only on finer scales. Consequently, we have selected a deep NN architecture with a U-like representation, a so-called U-Net (Ronneberger et al., 2015). The encoding part (on Fig. 3a, the left side of the U-Net)

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

extracts information on multiple scales (here on two), by cascading downsampling steps. The decoding part (on Fig. 3a, the right side of the U-Net) refines coarse-scale information up and combines them with information from finer scales, and outputs the extracted features. Consequently, the U-Net architecture can extract features at multiple scales, mapped onto the finest scale.

Our typical U-Net architecture consists of 3 different blocks: Residual blocks, mainly inspired by ConvNeXt blocks (Liu et al., 2022), a downsampling block, and an upsampling block. Our complete U-net architecture has in total approximately 1.2×10^6 trainable parameters and consists of five stages, cf. Table 1. The rectified linear unit (relu) in the output stage, $\boldsymbol{h}_{out} = \max(0, \boldsymbol{h}_{out-1})$, introduces a discontinuity into the features, which can help the NN to represent sharp transitions in the level of damage. The input fields projected into the Cartesian space are treated as input channels for the input stage and include nine state variables and one forcing field for both input time steps, resulting to in-total 20 input channels. The architecture is quite thick, with 128 output channels, to extract features for all model variables at the same time.

Table 1. Proposed baseline "U-NeXt" based on ConvNeXt-like blocks. "Down" and "Up" correspond to downsampling and upsampling blocks, respectively.

Stage	Operation	Params	n_{in}	n_{out}	n_x	n_y
Input	ConvNeXt	23 056	20	128	32	128
Down 1	Down	295 424	128	256	16	64
	ConvNeXt	145 152	256	256	16	64
	ConvNeXt	145 152	256	256	16	64
Bottleneck	ConvNeXt	145 152	256	256	16	64
Up 1	Up	295 552	256	128	32	128
	ConvNeXt	95 744	128	128	32	128
	ConvNeXt	39 808	128	128	32	128
Output	ConvNeXt	39 808	128	128	32	128
	relu	–	128	128	32	128

3.3.1 The ConvNeXt blocks

In our standard configuration, the processing blocks are mainly inspired by ConvNeXt blocks (Liu et al., 2022). The output $\boldsymbol{h}_l = f_l(\boldsymbol{h}_{l-1}) + g_l(\boldsymbol{h}_{l-1})$ of the l-th block is calculated based on the output of the previous block $\boldsymbol{h}_{l-1}$ by adding a residual connection $f_l(\boldsymbol{h}_{l-1})$ to a branch connection $g_l(\boldsymbol{h}_{l-1})$, as depicted in Fig. 3b.

The residual connection is an identity function $f_l(\boldsymbol{h}_{l-1}) = \boldsymbol{h}_{l-1}$ if the number of its output channels n_{out} equals the number of its input channels n_{in}. Otherwise, a convolution with a 1×1 kernel, called in the following 1×1 convolution, combines the n_{in} input channels to n_{out} output channels as a linear pixel-wise-shared function.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

In the branch connection, a single convolutional layer with a 7×7 kernel is applied depth-wise (i.e. on each input channel independently) to extract information about neighbouring pixels; before applying the convolution, the fields are zero padded by three pixels on all four sides, such that the output of the layer has the same size as the input. The output of this spatial extraction layer is normalised by layer normalisation (Ba et al., 2016) across all channels and grid points. Compared to batch normalisation (Szegedy et al., 2014), layer normalisation is independent of the number of samples per batch and performs on par in this type of block (Liu et al., 2022).

Afterwards, a convolution layer with a 1×1 kernel mixes the normalised channel information up. If not otherwise depicted, the output of this intermediate layer gets activated by a Gaussian error linear unit (Gelu, Hendrycks and Gimpel, 2020). The last 1×1 convolution linearly combines the activated channels into n_{out} channels. The output of this branch connection is scaled by learnable factors $\boldsymbol{\gamma}$, one for each output channel, and initialised with $\gamma_i = 1 \times 10^{-6}$. This type of scaling improves the convergence for deeper networks with residual layers (Bachlechner et al., 2020; De and Smith, 2020).

3.3.2 The down- and upsampling

For the downsampling operation, in the encoding part of the U-Net, we use a layer normalisation, followed by zero padding of one pixel on all four sides, and a convolution with a kernel size of 3×3 and stride of 2×2, similar to Liu et al. (2022). As this operation halves the data sizes in x- and y-direction, the number of channels is doubled in the convolution. By replacing max-pooling operations with a strided convolution, the downsampling operation becomes learnable (Springenberg et al., 2015).

For the upsampling operation, in the decoding part of the U-Net, we use a sequence of bilinear interpolation, which doubles the spatial resolution, layer normalisation, zero padding of one pixel on all four sides, and a convolution with a 3×3 kernel, which halves the number of channels. A bilinear interpolation followed by a convolution avoids unwanted checker-board effects (Odena et al., 2016), which can occur when using transposed convolutions for upsampling.

3.4 Learning via maximum likelihood

In our NN architecture, we want to predict a model error correction for all nine model variables at the same time, which causes nine different loss function terms, like nine different mean-squared errors or mean absolute errors (MAEs). As each of these variables has its own error magnitude, variability, and issues to correct, we have to weight the loss functions against each other with parameters λ_i, $\mathcal{L}_{total} = \sum_{i=1}^{9} \lambda_i \mathcal{L}_i$. To tune these parameters, we use a maximum likelihood approach, which relates the weighting parameters to the uncertainty of the nine different model variables (Cipolla et al., 2018).

In the maximum likelihood approach, a conditional probability distribution $p(\boldsymbol{\Delta x} \mid \boldsymbol{x}, \boldsymbol{\theta})$ parametrised by $\boldsymbol{\theta}$ is assumed to approximate the true, but unknown, data generating conditional probability distribution of the residuals $\boldsymbol{\Delta x}$ given the input $\boldsymbol{x}$ – note that for conciseness the initial state $\boldsymbol{x}^{\mathsf{in}}$ and the forecasted state $\boldsymbol{x}^{\mathsf{f}}$ have here been gathered in a single input vector $\boldsymbol{x}$. The parameters of this probability distribution are optimised such that the negative log-likelihood of the observed residuals $\boldsymbol{\Delta x}$ given the input $\boldsymbol{x}$ and parameters is minimised,

$$\boldsymbol{\theta}^{\star} \triangleq \underset{\boldsymbol{\theta}}{\operatorname{argmin}}[-\ln p(\boldsymbol{\Delta x} \mid \boldsymbol{x}, \boldsymbol{\theta})].$$

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

The log-likelihood factorises hereby as sum over multiple dimensions like the samples or variables.

We treat the output of our NN $f(\boldsymbol{x}, \boldsymbol{\phi})$ with its weights $\boldsymbol{\phi}$ as the median of a univariate approximated Laplace distribution. From the perspective of the NN, the negative log-likelihood is thus a weighted MAE loss function. As all data points are equally weighted, a Laplace distribution results into a more robust estimation against outliers than a Gaussian distribution. Contrary to the median predicted as field, we use a single scale parameter per variable, b_i, shared across all grid points. We optimise the nine scale parameters together with the NN by minimising the negative log-likelihood, averaged over B data pairs $(\boldsymbol{x}_j, \boldsymbol{\Delta x}_j)$. As we utilise a variant of stochastic gradient descent for optimisation, the data pairs are drawn from the training dataset $\mathcal{D}$ at each iteration. Before summing all nine loss terms up, we average the negative log-likelihood per-variable across all grid points (here simplified denoted as average across M grid points) as the velocity components have fewer data points than all other variables, caused by their spatial discretisation (c.f. Sect. 2),

$$\mathcal{L}_{total} = \frac{1}{BM} \sum_{i=1}^{9} \sum_{j=1}^{B} \sum_{k=1}^{M} \frac{1}{b_i} |\Delta x_{i,j,k} - f_{i,j,k}(\boldsymbol{x}, \boldsymbol{\phi})| + \ln(2b_i).$$

The factor in front of the absolute error, $\lambda_i = \frac{1}{b_i}$, is the weighting factor; the MAE can be recovered by setting $b_i = 1$ as constant. The additional term, $\ln(2b_i)$, origins from the normalisation of the Laplace distribution, is independent of the errors, and counteracts a too small b_i. This approach optimises b_i to match the expected MAE (Norton, 1984) in the training dataset and can be seen as an uncertainty estimate, e.g. recently used in Cipolla et al. (2018); Rybkin et al. (2020). Compared to using a fixed climatological value, this approach adaptively weights the loss, depending on the error of the NN for the different variables. This adaptive weighting marginally improves the training of the NN, as shown in Sect. C. Since we learn the scale parameters purely from data, this approach can be seen as type II maximum likelihood or empirical Bayes approach (Murphy, 2012)

4 Experimental setup

We train and test different NNs with twin experiments, using the sea-ice model described in Sect. 2. We compute trajectory simulations with a resolution of $4\ \mathrm{km}$ and an integration step of $8\ \mathrm{s}$, which stand as our truth for the twin experiments. Based on a coarsened $8\ \mathrm{km}$ version of the truth grid and an integration step of $16\ \mathrm{s}$, we simulate low-resolution forecasts. The NNs will learn to correct these low-resolution forecasts (hereafter simply called "forecasts") for a lead time of $10\ \mathrm{min}$ and $8\ \mathrm{s}$. In other words, the training target is the difference between the projected truth and the forecasts after an integration of $10\ \mathrm{min}$ and $8\ \mathrm{s}$, which equals 38 time steps at the low resolution.

We train the NN on an ensemble of 100 truth trajectories. The NN hyperparameters, like the depth of the network or the number of channels, are tuned against a distinct validation dataset with 20 trajectories. Finally, the scores are estimated using an independent test dataset with 50 trajectories.

All truth trajectories are initialised with a randomly chosen cohesion field; as the cohesion is one of the state variables, the cohesion is advected and diffused. The simulations also have different random external forcing parameters. These forcing parameters define the atmospheric wind speed $u_{\mathsf{a}}(x, y, t)$ depending on the spatial x- and y-position and the temporal t-position

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

by

$$u_{\mathsf{a}}(x,y,t) = A \cdot \sin\left[\frac{2\pi}{\lambda}(\phi + y + t \cdot \nu)\right] + u_0. \tag{1}$$

The model parameters values are given in Table 2: they have been chosen such that most trajectories have damaged sea ice in different regions of the simulation area.

Table 2. The random ensemble parameters and their distribution, $U(a,b)$ specifies a random variable drawn from a continuous uniform distribution with its two boundaries a and b. The cohesion is independently drawn for each grid point and ensemble member, whereas each ensemble member has one set of forcing parameters.

Description	Value
Cohesion C	$U(5 \times 10^3\ \mathrm{Pa}, 1 \times 10^4\ \mathrm{Pa})$
Amplitude A	$U(8\ \mathrm{m\,s^{-1}}, 20\ \mathrm{m\,s^{-1}})$
Wave length λ	$U(50\ \mathrm{km}, 200\ \mathrm{km})$
Phase ϕ	$U(-100\ \mathrm{km}, 100\ \mathrm{km})$
Advection ν	$U(-0.5\ \mathrm{m\,s^{-1}}, 0.5\ \mathrm{m\,s^{-1}})$
Base velocity u_0	$\max(20\ \mathrm{m\,s^{-1}} - A, U(0\ \mathrm{m\,s^{-1}}, 10\ \mathrm{m\,s^{-1}}))$

The truth trajectory simulations are run for three days of simulation time. The first day of simulation, where the forcing is linearly increased to its full strength after the day of simulation as in Dansereau et al. (2016), is treated as spin-up and omitted from the evaluation. Consequently, we use two days of truth trajectories to generate the dataset; the process of generating the dataset is depicted in Fig. 4. To make the truth comparable to the forecast simulations, the truth is projected to the $8\ \mathrm{km}$ resolution grid using a Lagrange interpolation, as in finite elements' discretisation. We generate initial conditions for the forecast trajectories by hourly slicing the projected truth. For these forecasts, the cohesion is one of the projected state variables. Since our only difference in the experiments should come from differently integrated processes, we keep the forcing parameters for the forecasts the same as for the truth. We run the forecasts for one hour of simulation time. After $10\ \mathrm{min}$ and $8\ \mathrm{s}$, we compute the residual (the difference between the projected truth and the forecasted state) and include it into our training dataset.

This dataset contains input-residual pairs $(\boldsymbol{x}_j, \boldsymbol{\Delta x}_j)$. The input for the NNs consists of 20 fields: nine fields for the initial conditions, nine fields for the forecasted state, one forcing velocity in y-direction at the initial time, and one forcing velocity at a lead time of $10\ \mathrm{min}$ and $8\ \mathrm{s}$. Our training target is the difference between the projected truth and the forecasted state at the same forecast lead time, and consists of nine fields. All the data is normalised by a global per-variable mean and standard deviation, both estimated from the training dataset.

The hourly slicing gives us 48 samples per truth trajectory. Consequently, in total, the training, validation, and test dataset have 4800, 960, and 2400 samples, respectively. If not otherwise specified, all NNs are trained for 1000 epochs with a batch size of 64 to minimise the Laplace negative log-likelihood (cf. Sect. 3.4) in the training data. Adam (Kingma and Ba, 2017)

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

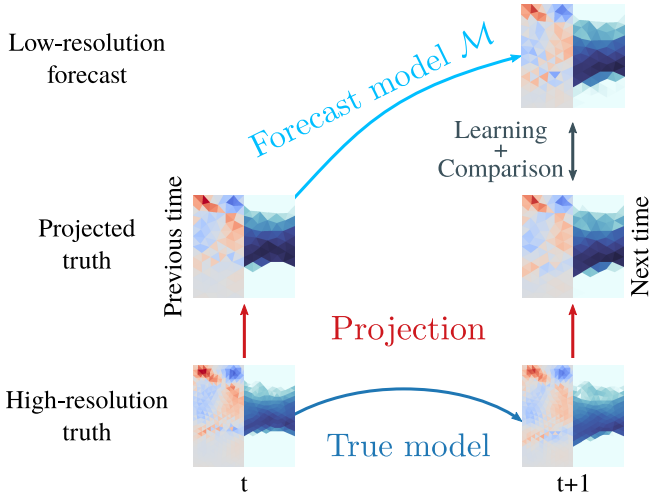

Figure 4. The truth state with a 4 km resolution is propagated from time t to time $t+1$ with the true high-resolution model; one discrete time step corresponds here to a lead time of 10 min and 8 s. The truth at time t is projected by Lagrange interpolation into low-resolution space (4 km resolution), acting as initial state for a forecast. A low-resolution forecast model $\mathcal{M}$ performs the forecast and propagates the state from time t to time $t+1$ in the low-resolution space. To learn a model error correction, the low-resolution forecast at time $t+1$ is compared to the truth at the same time, again projected into low-resolution space.

with an initial learning of $\gamma = 3 \times 10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, is used to optimise the NNs. We refrain of learning rate decay or early stopping, as such methods would make the experiments harder to compare.

All experiments are performed on the CNRS/IDRIS (French National Centre for Scientific Research) Jean Zay supercomputer, using a single NVIDIA Tesla V100 GPU per experiment. The NNs are implemented in PyTorch (Paszke et al., 2019) with PyTorch lightning (Falcon et al., 2020) and Hydra (Yadan, 2019). The code is publicly available under https://github.com/cerea-daml/hybrid_nn_meb_model.

5 Results

We propose a baseline architecture based on a U-Net, built out of ConvNeXt blocks (defined in Sect. 3.3), simply called U-NeXt. We have selected the parameters of the U-NeXt architecture after a randomised hyperparameter screening in the validation dataset with 200 different network configurations. We evaluate our trained NNs on the test dataset with the mean absolute error (MAE). To get comparable performances across the nine model variables, we normalise the errors by their expectation from the globally-averaged climatology in the training dataset. Note that this normalisation results into a constant weighting, differing from the adaptive weighting used during the training process, which depends on the training trajectory. In the case of the MAE, the errors are normalised by their expected MAE in the training dataset. Furthermore, this normalisation allows us to estimate the performance of the NNs with a single metric, averaged over all model variables. The NNs are trained

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

ten times with different random seeds ($s \in [0, 9]$), the results in Table 3–8 correspond to the averaged metric over the ten trained networks.

In the following, we discuss the results on the test dataset in Sect. 5.1, what we can learn about the model errors by analysing the sensitivity of the NN to its inputs in Sect. 5.2, and how we can combine the NN with the geophysical model for lead times up to one hour in Sect. 5.3.

5.1 Performance on the test dataset

In a first step, we evaluate the performance of our model error correction on the test dataset, without cycling the correction together with the geophysical model, Table 3. A comparison with other NN architectures, other loss functions, and other activation functions can be found in Appendix B, Appendix C, and Appendix D, respectively.

Table 3. Normalised MAE on the test dataset, averaged over ten NNs trained with different seeds. Reported are the errors for the velocity component in y-direction v, for the stress component σ_{yy}, the damage d, and the area A. The mean $\overline{\Sigma}$ is the error averaged over all nine model variables, including the non-shown ones. A score of one would correspond to the performance of the geophysical model forecast in the training dataset. A bold font indicates the afterwards used architecture and the best scores.

Name	v	σ_{yy}	d	A	$\overline{\Sigma}$
Persistence	2.37	0.60	0.29	0.37	0.79
Geophysical model	0.94	1.09	0.91	1.14	1.03
Hybrid model	**0.33**	**0.38**	**0.17**	**0.23**	**0.24**

The NN corrects the model forecasts across all variables, which results in an averaged gain of the hybrid model over $75\,\%$ compared to the geophysical model. For the stress, damage, and area, the persistence forecast performs better than the geophysical model for the lead time of $10\ \mathrm{min}$ and $8\ \mathrm{s}$, as the model forecast drifts towards the attractor of the geophysical model, as discussed in Sect. 5.3. Since the NN uses the initial conditions as input, the hybrid model surpasses the performance of persistence, even for variables where persistence is better than the geophysical model. In total, the NN consistently improves the forecast on the test dataset.

To apply CNNs to the raw data of our finite-elements-based sea-ice model, we project from triangular to Cartesian space, where the features are extracted. The number of elements in the Cartesian space determines its effective resolution and, thus, the finest scale on which the NN can extract features. To demonstrate the effect of different resolutions on the result, we perform three different experiments, where we change the grid size, while keeping the NN architecture the same (Table 4).

The training loss, here the negative Laplace log-likelihood, measures how well a NN can be fitted towards the training dataset. Although its resolution is higher than the original resolution of $8\ \mathrm{km}$, the back-projection for the 8×32 grid is underdetermined, as the mapping is non-surjective, degrading the performance of the NN. Starting at the 16×64 grid, the Cartesian space covers all triangular grid points, and all NNs have a similar predictive power with similar training losses. Nevertheless, the MAE of the finest 32×128 grid is the lowest for all variables. As we keep the architecture the same for all resolutions, the higher the

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Table 4. Normalised MAE on the test dataset for different Cartesian grid sizes, x-direction $\times$ y-direction. The error components are estimated as in Table 3. The training loss is estimated as the expected Laplace negative log-likelihood, averaged over the training dataset, variables, and ten NNs trained from different seeds. A bold font indicates the selected grid size and the best scores.

Grid size	Loss	v	σ_{yy}	d	A	$\overline{\Sigma}$
8×32	-9.31	0.72	0.64	0.42	0.29	0.50
16×64	**-18.58**	0.43	0.41	0.18	0.26	0.28
32×128	-18.47	**0.33**	**0.38**	**0.17**	**0.23**	**0.24**

resolution, the smaller the receptive field of the NN. At the highest resolution, the NN is thus forced to extract more localised features. Such localised features seem to better represent the needed processes for the prediction of the residuals and for parametrising the subgrid-scale; this improvement by using a finer Cartesian space will be discussed more in detail in Sect. 6.

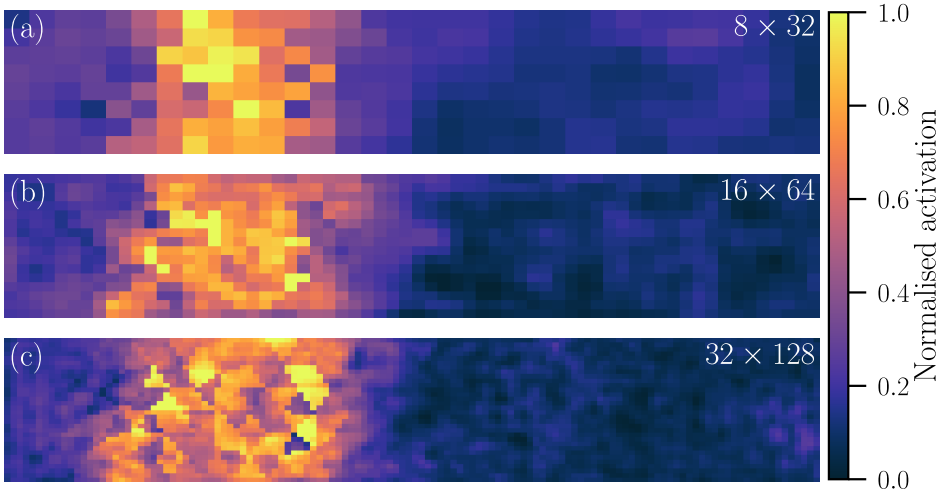

Figure 5. Normalised feature map in Cartesian space for grid sizes of (a) 8×32, (b) 16×64, and (c) 32×128. The feature map is estimated based on the same sample in the test dataset. The specific feature maps are selected such that the extracted features qualitatively match for all three resolutions. As the maps might have different order of magnitudes, they are normalised by their 99-th percentile for visualisation purpose, the colours are thus proportional to the activation.

In Fig. 5, we visualise a typical output of the U-Net before it gets projected back into triangular space and linearly combined. The used grid resolution impacts those feature maps; the higher the resolution, the sharper the feature map. Sharper features can better represent anisotropy and discrete processes in sea ice. For the highest resolution, Fig. 5c, the extracted features correspond to a mixture between localised features and a generally smoother background pattern. The localised features indicate an ability to parametrise the effect of the subgrid-scale onto the resolved scales. Moreover, as a consequence of the extraction

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

of more localised features for finer spaces, the NN also localises the background noise such that the field appears to be much noisier in the case of inactive zones, where the activation is low.

5.2 Sensitivity to the input

The inputs of the NN have a crucial impact on the performance of the model error correction. In the following, we evaluate the sensitivity of the NN with respect to its input variables. In a first step, we alter the input and measure the resulting performance of the NN with the normalised MAE (Table 5).

Table 5. Normalised MAE on the test dataset for different input sets. The error components are estimated as in Table 3. n_{in} corresponds to the number of input channels. Bold indicates the best scores.

Name	n_{in}	v	σ_{yy}	d	A	$\overline{\Sigma}$
Initial only	10	0.34	0.77	0.63	0.63	0.60
Forecast only	10	0.35	0.75	0.62	0.66	0.60
Both	20	0.33	0.38	0.17	0.23	0.24
W/o forcing	18	0.33	0.37	0.18	0.24	0.25
Difference only	10	0.30	0.37	0.15	0.19	0.23
+ initial state	20	**0.26**	**0.33**	0.15	**0.17**	**0.21**
+ forecasted state	20	**0.26**	**0.33**	**0.14**	**0.17**	**0.21**

Usually, only the initial conditions are used for a neural-network-based model error correction (Farchi et al., 2021a). As sea-ice dynamics is a first-order Markov system, the results are very similar when using only the initial state or only the forecast state as input. Using both improves the prediction by more than $58\ \%$. The NN learns to correct the residuals based on the difference between the forecast and initial state, representing the sea-ice dynamics. If only one state is used as input, the NN has to internally learn an emulator of the dynamics to correct residuals. Explicitly giving the difference to the NN instead of raw states improves the correction, although the number of predictors is halved. With the difference, the network can directly access the model dynamics. Adding the initial state or forecasted state to the difference further improves the correction; the network has then also access to absolute values.

In the second step, we analyse how the input variables influence the output of the NN. As we want to quantify the impact of the dynamics on the output, we base the analysis on the previous "Initial + Difference" experiment (Table 5). As global input variables importance metric, we use the permutation feature importance (Breiman, 2001): the NN is applied several times to the input variables, each time, another input variables is permuted across the samples. By permuting an input variable, its information is destroyed, and the output of the NN is changed. The possible change in the prediction error compared to the original, unchanged, output describes the importance of this input variable. Focussing on active regions, we measure the errors with the RMSE, estimated over the whole test dataset. The higher the RMSE for a permuted input variable, the higher the importance for this variable onto the errors, as summarised in Table 6.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Table 6. The permutation feature importance of the RMSE for the given output variable with respect to the input variable for "Initial + Difference" as input, estimated over the whole test dataset. The numbers show the multiplicatively RMSE increase of a specific output variable (y-axis) if a given input variable (x-axis) is permuted, a higher number corresponds to a higher feature importance. The colours are normalised by the highest feature importance for a given output variable and proportionally to the feature importance. The "Difference" variables specify the difference of the forecast state to the initial state as input into the neural network.

Output: $f(\mathbf{x})$	Initial: $\mathbf{x}_0$										Difference: $\Delta\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0$									
Input: $\mathbf{x}$	Forcing	u - Velocity	v - Velocity	σ_{xx}	σ_{xy}	σ_{yy}	Damage	Cohesion	Area	Thickness	Forcing	u - Velocity	v - Velocity	σ_{xx}	σ_{xy}	σ_{yy}	Damage	Cohesion	Area	Thickness
u - Velocity	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.3	1.1	1.0	5.7	1.1	1.0	1.0	1.0	1.0	1.0	1.1	1.2
v - Velocity	1.1	1.0	1.3	1.0	1.0	1.1	1.1	1.0	1.5	1.4	1.4	0.9	11.5	1.0	1.0	1.0	1.0	1.0	1.2	1.3
σ_{xx}	1.0	1.0	1.0	1.1	1.0	1.0	1.1	1.0	1.0	1.0	1.0	1.0	1.1	5.3	1.0	1.0	1.0	1.0	1.0	1.1
σ_{xy}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.2	1.0	4.5	1.0	1.0	1.0	1.0	1.0
σ_{yy}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.1	1.0	1.0	1.0	1.0	3.5	1.0	1.0	1.1	1.2
Damage	1.0	1.0	1.0	1.0	1.0	1.0	1.1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.1	3.6	1.0	1.0	1.0
Cohesion	1.0	1.0	1.3	1.0	1.0	1.0	1.0	2.2	1.2	1.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	5.7	1.1	1.3
Area	1.0	1.0	1.2	1.0	1.0	1.0	1.1	1.0	1.3	1.5	1.0	1.0	1.2	1.0	1.0	1.1	1.0	1.0	2.0	1.9
Thickness	1.0	1.0	1.2	1.0	1.0	1.0	1.1	1.0	1.4	1.8	1.0	1.0	1.2	1.0	1.0	1.1	1.0	1.0	1.7	2.4

Important / Unimportant

All model variables are highly sensitive to their own dynamics, showing their importance as predictors for the model error correction. Furthermore, the feature importance reflects the relations inherited by the model equations (cf. Sect. 2, Dansereau et al., 2017). For instance, caused by the dependence of the thickness upon the sea-ice area, they are linked together in the input-output relation. The wind forcing externally drives and influences the sea-ice velocity in y-direction, v. The v-velocity, however, advects and mixes the cohesion, area, and thickness. By modulating the momentum equation and mechanical parameters, respectively, the area and thickness influence the velocity and stress components. In total, the area and thickness are the most influential and most influenced variables in the whole dataset, as they represent the conditions of the sea ice. Therefore, the NN has learned meaningful relations between input and residuals.

As local measure, we move to the sensitivity $\frac{\partial f(\boldsymbol{x},\boldsymbol{\theta})}{\partial \boldsymbol{x}}$ of the NN output to its input fields (Simonyan et al., 2013), again for the "Initial + Difference" experiment. To showcase what the NN has learned in spatial meanings, we concentrate here on a single grid point in a single outputted sea-ice area field. To smooth the sensitivity and reduce its noise, we perturb the input to the NN 128 times with noise drawn from $\mathcal{N}(0, 0.1^2)$, and average the gradient over these noised versions (Smilkov et al.,

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

2017). The resulting saliency maps (Fig. 6) show which regions influence the selected grid point. The larger its amplitude, the more sensitive is the output to the pixel.

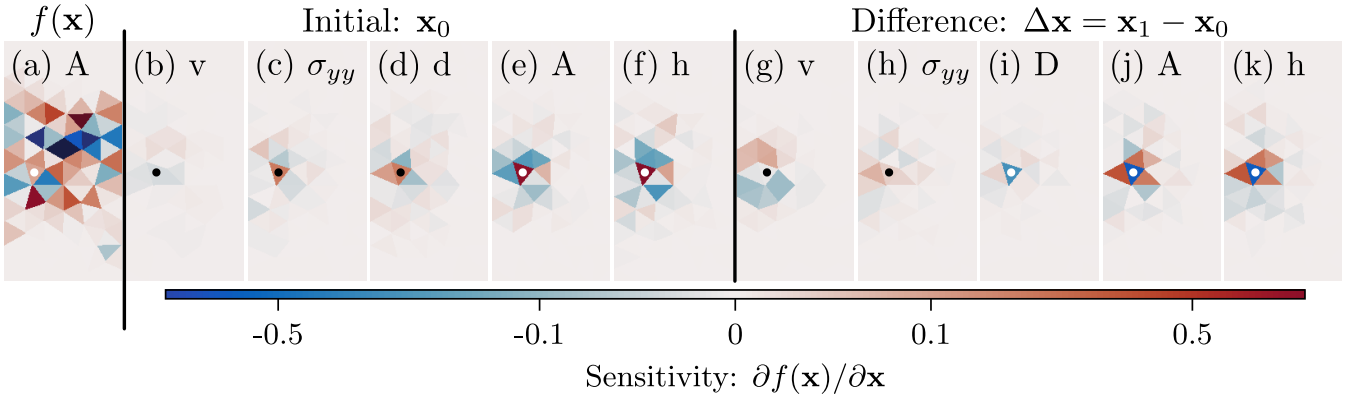

Figure 6. The sensitivity of the chosen grid point, indicated by either white or black dots depending on the background colour, on the residual prediction for the sea-ice area. To reduce the noise, the gradient is averaged around 128 noised input fields, perturbed by white noise with a standard deviation of 0.1. The following acronyms are used for the variables: v – velocity in y-direction; σ_{yy} – divergent stress in y-direction; d – damage; A – sea-ice area; h – sea-ice thickness.

In the selected active region, the prediction of the residual for the sea-ice area is especially sensitive to the thickness, either to its absolute values, Fig. 6f, or its dynamics, Fig.6k. This underlines the already mentioned relation between the sea-ice area and thickness, and confirms the global results of the permutation feature importance of Table **??**. As surrounding pixels are important for the sensitivity, spatial correlations in the input fields have to be correctly represented by the NN. Additionally, the sensitivity is directional dependent, Fig. 6g, and exhibits localised features, Fig. 6c and i. The NN can thus extract anisotropic and localised input-output relations to correct the model errors.

5.3 Cycled forecasting with error correction

After establishing the importance of the dynamics for the error correction, we cycle the error correction together with the low-resolution forecast model. As trained for a forecast horizon of 10 min and 8 s, we apply the NN to correct the forecast every 10 min and 8 s. Because the prognostic sea-ice thickness is represented as a ratio between the actual sea-ice thickness and the area, its error distribution can have very fat tails and can be non-well-behaved. Thus, we predict as output the actual sea-ice thickness, then, as post-processing step, translated into the prognostic sea-ice thickness. To reduce the amplification of small output errors of the NN over grid points with small sea-ice area, we limit the error correction of the prognostic sea-ice thickness to values between -1×10^{-3} and 1×10^{-3}. We additionally enforce physical bounds on all variables, by limiting the values to physical reasonable bounds after error correction. Related to optimal control theory in dynamical systems, we change the performance metric to be the RMSE. We evaluate the performance across all 2400 hourly time slices on the test dataset (Fig. 7 and Table 7).

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

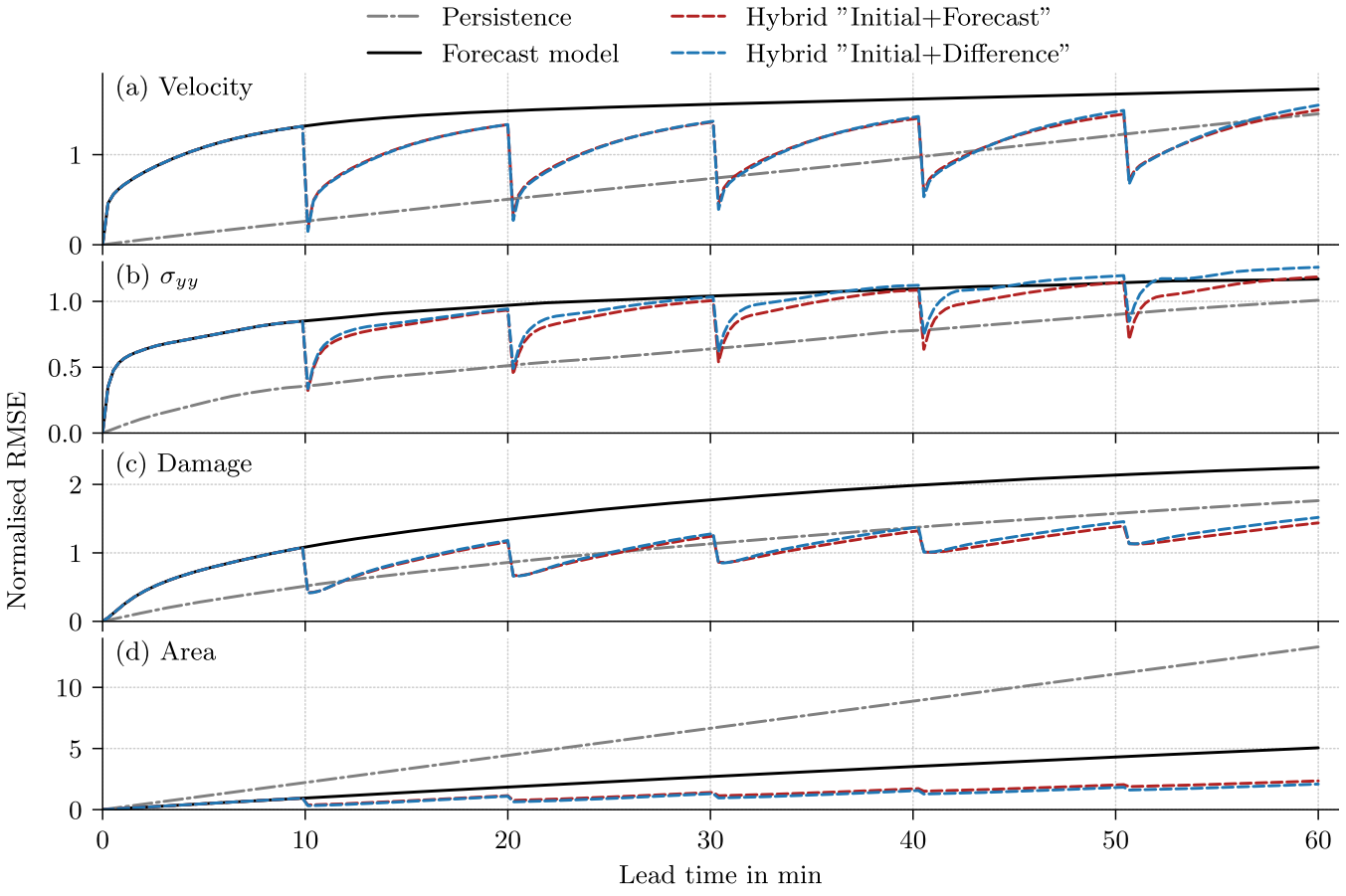

Figure 7. Normalised RMSE for (a) the velocity in y-direction, (b) the divergent stress in y-direction, (c) the damage, and (d) the sea-ice area as function of lead time on the test dataset, normalised by the expected RMSE on the training dataset for a lead time of 10 min and 8 s. In the hybrid models, the forecast is corrected every 10 min and 8 s, and the performance is averaged over all ten random seeds

Overall, the hybrid models surpass the performance of the original geophysical model (Fig. 7). The projected initial state lays not on the attractor of the forecast model, the forecast drifts towards the model attractor; a behaviour similar to what is typical in seasonal or decadal climate predictions initialised with observed data. This results in large deviations between geophysical forecast and projected truth. Consequently, the persistence forecast is better than the forecast model for the velocity, the stress, and the damage. And yet, updating the forecast model with NNs improves the forecasts at update time, even compared to persistence. Between two consecutive updates, when the model runs freely, the forecast drifts again towards the attractor, which leads to a decreased impact of the error correction. Nevertheless, the accumulated model error correction results into an improved forecast for a lead time of 60 min (Table 7), especially for the sea-ice area and damage.

By explicitly representing the dynamics as difference, the "Initial + Difference" experiment extracts more information from the dynamics than the "Initial + Forecast" experiment (see also Table 5). In some sense, the experiment is overfitted towards

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Table 7. Normalised RMSE on the test dataset for a lead time of 60 min. The last update in the hybrid models was at a lead time of 50 min and 40 s. The errors are normalised by the expected standard deviation on the training dataset. The symbolic representation of the variables has the same meaning as in Table 3.

Name	v	σ_{yy}	d	A	$\overline{\Sigma}$
Persistence	**1.13**	**0.81**	0.83	2.58	1.42
Geophysical model	1.34	0.93	1.06	0.98	1.06
Hybrid "Initial + Forecast"	1.16	0.95	**0.68**	0.46	**0.90**
Hybrid "Initial + Difference"	1.20	1.01	0.71	**0.41**	0.93

the use of the dynamics for the model error correction. Additionally, for the velocity, stress, and damage, the drift towards the attractor makes the training dataset less representative for the model forecast. As a consequence, the "Initial + Difference" experiment performs worse than the "Initial + Forecast" experiment in these variables, and averaged over all nine model variables.

Although the forecast error generally increases with lead time, the error reduction gets smaller with each update, especially for the sea-ice area. Since the NN correction is imperfect, the error during the next forecast cycle is an interplay between the initial condition error and the model error. The NN is trained with perfect initial conditions to correct the model error only. As the initial condition error increases with each update, the network corrects less and less forecast errors.

To show the effect of this error distribution shift, we analyse the centred spatial pattern correlation (Houghton et al., 2001, p. 733) between the updates and the residuals for the first and the fifth update (Table 8). By centring the covariances and variances, we neglected the influence of the amplitudes upon the performance of the NN. The correlations are independently estimated over space for each test sample and variable and averaged via a Fisher z-transformation (Fisher, 1915): the single correlations are transformed by the inverse hyperbolic tangent function. In transformed space, the values are approximately Gaussian distributed and averaged across samples. The average is transformed back by the hyperbolic tangent function. An averaged correlation of 1 would indicate a perfect pattern correlation.

Table 8. Centred pattern correlation on the test dataset between the updates and the residuals for the first update and fifth update. The symbols of the variables are the same as in Table 3.

Update	v	σ_{yy}	d	A	$\overline{\Sigma}$
First update	0.94	0.99	0.93	0.92	0.98
Fifth update	0.70	0.89	0.59	0.28	0.76

Since they are trained for this, the NN can predict the patterns of the residuals for the first update. At the fifth update, larger parts of the residual patterns are unpredictable for our trained networks. Caused by the drift towards the attractor, parts of the previous error correction are forgotten and get corrected again in the fifth update, especially for the divergent stress

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

components. As the sea-ice area has a longer memory for error corrections, its predicted patterns are almost unrelated to the residual patterns for the fifth update.

Our proposed parametrisation is deterministic and is designed to target the median value. On the resolved scale, sea-ice dynamics can look stochastically noised, with suddenly appearing strains and linear kinematic features, as discussed in the introduction. We show the effect of the seemingly stochastic behaviour in Fig. 8 with the temporal development of damage and total deformation for the truth, the forecast model without parametrisation, and the parametrised hybrid model.

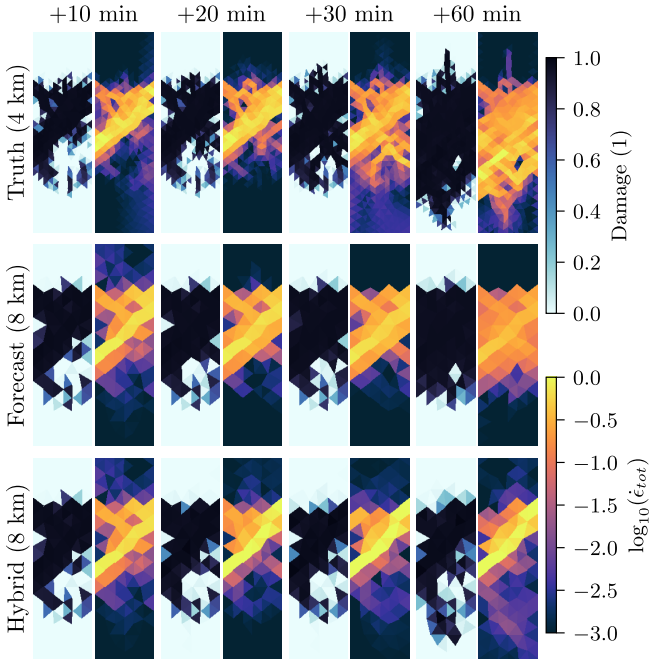

Figure 8. Snapshots of damage (left) and total deformation (right), showing their temporal evolution, in the high-resolution true model (top), the low-resolution forecast model (middle), and the low-resolution hybrid model (bottom).

The initial state exhibits damaged sea ice in the centre, corresponding to a diagonal main strain in the total deformation. In the high-resolution truth, the damaging process continues, leading to more widespread damaging of sea ice. Related to new strains, the damage is additionally extended towards the south. The low-resolution forecast model only diffuses the deformation without remaining main strain in the already damaged sea ice. As a result, the model misses the southward-extending strain and damaging process. Furthermore, the model extends the damage and deformation southwards, although the newly developed strain is weaker than in the truth. The parametrisation can represent widespread damaging of sea ice. However, the parametrisation misses the development of new strains and positions the main strain at the wrong place. This problem can especially occur on longer forecasting time scales, where the damage field is further developed compared to its initial state. Therefore, we see the need for parametrisations that can also reproduce the stochastic effects of subgrid-scales onto the resolved scales.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

6 Summary and discussion

We have introduced an approach to parametrise subgrid-scale dynamical processes in the sea ice based on deep learning techniques. Using twin experiments with a model of sea-ice dynamics that implements a Maxwell-Elasto-Brittle rheology, the NN learns to correct low-resolution forecasts towards high-resolution true trajectories for the forecast lead time of $10\ \mathrm{min}$ and $8\ \mathrm{s}$.

Our results show that NNs are able to correct forecast model errors related to the sea-ice dynamics and can thus parametrise the unresolved subgrid-scale processes as for other Earth system components. In addition, we are able to directly transfer recent improvements in deep learning, like ConvNeXt blocks (Liu et al., 2022), to ameliorate the representation of the subgrid-scale. We correct all model variables at the same time with one big NN, here with 1.5×10^6 parameters. We employ a maximum likelihood approach using a univariate Laplace log-likelihood, which corresponds to a mean absolute error reduction. By targeting a prediction of the median instead of the mean, the NN extracts more contrasted features, which improves the parametrisation compared to a Gaussian log-likelihood. Additionally, optimising globally-shared uncertainty parameters together with the NN balances the learning for all variables during training and results into an additional gain.

Our specific convolutional U-Net architecture improves the parametrisation compared to naively-stacked convolutional layers. The U-Net architecture extracts localised and anisotropic features on multiple scales. For feature extraction, we map the triangular input elements into a Cartesian space with a higher resolution to preserve correlations of the input data. Our results show hereby that higher resolved Cartesian spaces further improve the parametrisation; the network can then extract more information about the subgrid-scale. Mapped back into the original triangular space, the extracted features are linearly combined to predict the residual, which parametrises the effect of the subgrid-scale upon the resolved scales. Therefore, using such a mapping into Cartesian space, we can apply CNNs, which can efficiently scale to larger, Arctic-wide, models.

Our results suggest that the finer the Cartesian space resolution, the better the performance of the NN. This improvement could emerge from our type of twin experiments, where the main difference in the resolved processes is a result of different model resolutions. Consequently, extracting features at a higher resolution than the forecast model might be needed to represent the processes of the truth that has a higher resolution; the NN would act as an emulator for these processes. The resolution needed for the projection would be linked to the resolution of the truth. As processes have no discretised resolution in real-world, we would have difficulties to find the right resolution for the projection in such cases. Nevertheless, in the light of our results, this argument seems to be unlikely, as the performance of the finer 32×128 grid is higher than the 16×64 grid, although the latter one has already a higher resolution than the truth grid. Additionally, this argument cannot explain the increased training loss for the finer grid compared to its lowered errors on the test dataset.

The gain likely results from an inductive bias in the NN for Cartesian spaces with higher resolutions. We keep the NN architecture and its hyperparameters, like the size of the convolutional kernels, the same, independent of the resolution in the Cartesian space. Consequently, viewed from the original triangular space, the receptive field of the NN is reduced by increasing the resolution. The function space representable by such NN is more restricted, and, as the fitting power is reduced, the training loss increases again. The NN is biased towards more localised features. These localised features help the network

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

to represent previously unresolved processes and the underlying partial differential equation of the residuals better. This better representation improves the generalisation of the NN, resulting into lower errors on the test dataset. However, as this study is performed in very specific settings with twin experiments, it remains unknown to us if the projection into a space that has a higher resolution is advantageous for subgrid-scale parametrisations with machine learning in general.

The permutation feature importance as global feature importance and sensitivity maps as local importance help us to explain the learned NN by physical reasoning. The sensitivity map has additionally shown that the convolutional U-Net can extract anisotropic and localised features, depending on the relation between input and output. We see such an analysis as especially relevant for subgrid-scale parametrisations learned from observations, as the feature importance can be utilised to find the sources of model errors and guide model developments. Using such tools, we discover that the NN represents a physically explainable input-to-output mapping.

Cycling the NN correction together with the forecast model improves the forecasts up to one hour. Since the error correction is imperfect, the initial condition errors accumulate for longer forecast horizons. There, the targetted residuals in the training dataset are less representative, and also the correlation between the predicted error patterns and the true error patterns reduces. These issues would be solved in online training of the NNs (Rasp, 2020; Farchi et al., 2021a), which could be nevertheless too costly for real-world applications. Offline reinforcement learning additionally tackles similar issues (Levine et al., 2020; Lee et al., 2021; Prudencio et al., 2022) and can be thus a way to partially solve them.

Although the here-learned NNs can make continuous corrections, they represent only deterministic processes. As the evolution of sea ice propagates from the subgrid-scale to larger scales, unresolved processes can appear like stochastic noise from the resolved point of view. Consequently, the deterministic model error correction is unable to parametrise such stochastic-like processes, which can lead for example to wrongly-positioned strains and linear kinematic features. Generative deep learning (Tomczak, 2022) can offer a solution to such problems and could introduce a form of stochasticity into the subgrid-scale parametrisation, e.g. by meanings of generative adversarial networks (Goodfellow et al., 2014) or denoising diffusion models (Sohl-Dickstein et al., 2015). Such techniques can also be used to learn the loss metric, circumventing issues by defining a loss function for training.

Because of missing subgrid-scale processes in the low-resolution forecast model, the projected truth states are far off the low-resolution attractor. Consequently, when the forecast model is run freely, it drifts toward its own attractor, resulting into large deviations from the forecasted states to the projected truth states. This difficult forecast setting is indeed quite realistic, as models miss also in reality subgrid-scale processes (Bouchat et al., 2022; Ólason et al., 2022), such that empirical free-drift or even persistence forecasts are difficult to beat with forecast models (Schweiger and Zhang, 2015; Korosov et al., 2022). As the attractor of the forecast models does not match the attractor of the observations or the projected truth state, also finding the best state on the model attractor would not necessarily lead to an improved forecast (e.g. Stockdale, 1997; Carrassi et al., 2014). The only way is therefore to improve the forecast model, thereby changing its attractor.

A subgrid-scale parametrisation can be generally seen as a kind of forcing. Here, we use a resolvent correction, where we correct the forecast model with NNs at integrated time steps; the parametrisation is like Euler integrated in time. Our results show that casting the subgrid-scale parametrisation as post-processing is needed to correct the sea-ice dynamics with such

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

a resolvent correction. Then, the NN has access to the dynamics of the model and can correct tendencies related to the drift towards the wrong attractor, at least at update time. One strategy can be thus to increase the update frequency or to distribute the correction over an update window, similarly to incremental analysis update in data assimilation (Bloom et al., 1996). Another strategy is to use tendency corrections (Bocquet et al., 2019; Farchi et al., 2021a), where the parametrising NN is directly incorporated as an external forcing term into the model equation. As the tendency correction is included into the model itself, it also changes and possibly corrects the attractor. The adjoint model of the forecasting model is typically unavailable for large-scale sea-ice models, but would be needed to train such a tendency correction (Farchi et al., 2021a). To remedy such needs, one could also rely on training the NN as a resolvent correction for a specific forecast horizon and scaling the correction to a tendency correction, constant over the specified forecast horizon.

Ideally, subgrid-scale parametrisations should be learned by incorporating observations into the forecast. Combining machine learning with data assimilation enables such learning from sparsely-distributed observations (Bocquet et al., 2019, 2020; Brajard et al., 2020, 2021; Farchi et al., 2021b; Geer, 2021). This combination could be the next step towards the goal of using observations to learn data-driven parametrisations for sea-ice models.

7 Conclusions

Based on our results for twin experiments with a sea-ice dynamics-only model in a channel-setup, we conclude the following:

- Deep learning can correct forecast errors and can thus parametrise unresolved subgrid-scale processes related to the sea-ice dynamics. For its trained forecast horizon, the neural network can reduce the forecast errors by more than 75 %, averaged over all model variables. This error correction makes the forecast better than persistence for all model variables at update time.

- A single neural network can parametrise processes related to all model variables at the same time. The needed weighting parameters can be hereby spatially-shared and learned with a maximum likelihood approach. A Laplace likelihood improves the extracted features compared to a Gaussian likelihood and is better suited to parametrise the sea-ice dynamics.

- Convolutional neural networks with a U-Net architecture can represent important inductive biases for sea-ice dynamics by extracting localised and anisotropic features from multiple scales. Mapping the input data into a Cartesian space that has a higher resolution than the original space, such scalable convolutional neural networks can be applied for feature extraction in sea-ice models defined on a triangular or unstructured grid. The finer Cartesian space keeps correlations in the input data intact and enables the network to extract better features related to subgrid-scale processes.

- Because forecast errors in the sea-ice dynamics are likely linked to errors of the forecast model attractor, we have to cast the model error correction as a post-processing step and to input into the neural network the initial and forecasted state. This way, the neural network has access to the model dynamics and can correct them. Consequently, the dynamics of the forecast model variables are the most important predictors in a forecast error correction for sea-ice dynamics.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

- Although only trained for correction at the first update step, cycling the error correction together with the forecast model improves the forecast, tested up to one hour. The accumulation of uncorrected errors results into a distribution shift in the forecast errors, making the error correction less efficient for longer forecast horizons. These stochastic effects can result in wrongly positioned strains and damaging processes for a deterministic error correction. Online training or techniques borrowed from offline reinforcement learning would be needed to remedy this distribution shift.

- The deterministic hybrid model leads to a better damage and total deformation representation than the forecast model without parametrisation. Nevertheless, the unresolved subgrid-scale in the sea-ice dynamics can have seemingly stochastic effects on the resolved scales. To properly parametrise such effects, we would need generative neural networks.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Appendix A: The parameters of the regional sea-ice model

Our regional sea-ice model is almost the same as that presented in Dansereau et al. (2017). Since our chosen parameters differ a bit from the parameters used in Dansereau et al. (2016, 2017), we give them in Table A1.

Table A1. The parameters for the sea-ice dynamics-only model (Dansereau et al., 2016, 2017) used in this study.

Parameter		Values
Poison s ratio	ν	0.3
Internal friction coefficient	μ	0.7
Ice density	ρ	$900\ \mathrm{kg\,m^{-3}}$
Velocity of the elastic shear in ice	c	$500\ \mathrm{m\,s^{-1}}$
Undamaged elastic modulus	E_0	5.85×10^8 Pa
Undamaged apparent viscosity	η_0	5.85×10^{15} Pa s
Undamaged relaxation time	λ_0	1×10^7 s
Damage exponent	α	4
Characteristic time for damage	t_d	16 s
Characteristic time for healing	t_h	5×10^5 s
Average grid resolution	Δx	4 km (high-res)
		8 km (low-res)
Integration time step	Δt	8 s (high-res)
		16 s (low-res)
Air drag coefficient	$C_{\mathrm{d_a}}$	1.5×10^{-3}
Air density	ρ_a	$1.3\ \mathrm{kg\,m^{-3}}$
Water drag coefficient	$C_{\mathrm{d_w}}$	5.5×10^{-3}
Water density	ρ_w	$1 \times 10^3\ \mathrm{kg\,m^{-3}}$

The characteristic time in the damaging process is chosen to be no source of forecast error.

Appendix B: Screening of architectures

As our NN architecture accommodates multiple decisions, we will here explore how they influence the results on the test dataset. We show what would happen if we would use other CNN architectures (Table B1). Detailed NN configurations can be found in Appendix B1. We have selected the parameters of the U-Net architecture after a randomised hyperparameter screening in the validation dataset with 200 different network configurations per architecture, like for the U-NeXt architecture.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Table B1. Normalised MAE on the test dataset for different NN architectures, averaged over ten NNs trained with different seeds. Reported are the errors for the velocity component in y-direction v, for the stress component σ_{yy}, the damage d, and the area A. The mean $\overline{\Sigma}$ is the error averaged over all nine model variables, including the non-shown ones. A score of one would correspond to the performance of the raw forecast in the training dataset. Bold indicate the afterwards used architecture and the best scores.

Name	Params ($\times 10^6$)	v	σ_{yy}	d	A	$\overline{\Sigma}$
Persistence	-	2.37	0.60	0.29	0.37	0.79
Raw forecast	-	0.94	1.09	0.91	1.14	1.03
Bias-corrected forecast	-	0.94	1.09	0.90	1.14	1.02
Conv ($\times 1$)	0.05	0.61	0.46	0.31	0.35	0.36
Conv ($\times 5$)	0.29	0.35	1.00	0.24	0.33	0.35
U-Net	3.7	**0.33**	0.41	0.24	0.35	0.28
U-NeXt	1.2	**0.33**	**0.38**	**0.17**	**0.23**	**0.24**

The simplest approach to correct the model forecast is to estimate a global bias, one for each variable, in the training dataset and to correct the forecast by this constant. As the low-resolution forecast has almost no systematic bias compared to the projected truth, this bias correction has only a negligible effect on the results.

As a next level of complexity, we introduce a shallow CNN architecture with one layer as feature extractor, called "Conv ($\times 1$)". Using dilation in the convolutional kernel, this layer can extract shallow multiscale spatial information for each grid point. This shallow architecture with only around 5×10^4 parameters constantly improves the forecast by around $65\ \%$ in average. Introducing a hierarchy of five convolutional layers in the "Conv ($\times 5$)" architecture increases the number of parameters to 2.9×10^5. However, the averaged metric is only marginally better than for the shallow CNN. Its multiscale capacity is limited, and the NN cannot scale with the depth of the network to extract more information. Caused by their limited capacity, the NNs have to focus on some variables, creating an imbalance between variables, which harms the performance for other variables, like the stress in the case of "Conv ($\times 5$)". A shallow network with a single convolutional layer can be nevertheless an option to obtain a small and fast NN for a subgrid-scale parametrisation.

An approach to extract multiscale information is to use a U-Net architecture that extracts and combines information from different levels of coarsened resolution. To make the approaches comparable, we use almost the same configuration as specified in Sect. 3.3 and Table 1, except that we replace the ConvNeXt blocks by simple convolutional layers with a kernel size of 3×3, followed by batch normalisation (Szegedy et al., 2014), and a Gaussian error linear unit activation function. Using such a U-Net decreases the forecast errors by more than $20\ \%$ compared to the basic CNN. Although the improvement is for some variables only small compared to the simpler networks, the U-Net improves the balance between different variables. Consequently, the metric for the U-Net is always better than for persistence, showing its capacity and potential to extract multiscale information.

Replacing the classical convolutional layers with ConvNeXt blocks as described in Sect. 3.3 gives an additional improvement in the performance of the NNs. Although the number of parameters for the U-NeXt is only a third of the number for the U-Net,

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

the ConvNeXt blocks reduce the forecast errors by additional $14\ \%$. The blocks reduce hereby especially the errors in the damage and area. The ConvNeXt blocks are able to extract more information from existing data than convolutional layers. Because the U-NeXt is the best performing method also in the validation dataset, we will continue to use this architecture for the rest of the manuscript.

B1 Neural network configurations

By mapping from triangular space into high resolution Cartesian space, several Cartesian elements are caught in one triangular element. Consequently, a simple convolutional layer would have problems to extract information across multiple scales. To circumvent such problems, we apply in the case of the naively-stacked convolutional layers two convolutional layers at the same time – one local filter with a 3×3 kernel, and one larger-scale filter with 3×3 kernel and a dilation of 6×7, such that the filter sees the next triangular element – we call such a layer "MultiConv". Using zero padding, we keep the output of the layers the same. The output of both convolutional layers is averaged to get a single output. As usual for CNNs, we use batch normalisation instead of layer normalisation. We keep Gelu as activation function, as for the ConvNeXt blocks, except for the last layer, where we use relu. The "Conv ($\times 1$)" uses a single block (Table B2), whereas the "Conv ($\times 5$) stacks five blocks (Table B3) with increasing number of feature channels.

Table B2. "Conv ($\times 1$)" based on a single multiscale convolutional layer.

Operation	Params	n_{in}	n_{out}	n_x	n_y
MultiConv	46 208	20	128	32	128
Batch normalisation	256	128	128	32	128
relu	-	128	128	32	128

Our baseline "U-Net" (Ronneberger et al., 2015) has classical convolutional blocks instead of ConvNeXt blocks. Like in the case for the U-NeXt, we have optimised the hyperparameters for this U-Net with a random hyperparameter sweep over 200 different network configuration. Similarly to the U-Next (c.f. Table 1), the here-used configuration is based on one level of depth, where the fields are downsampled in the encoder and upsampled in the decoder part. Our convolutional blocks have one convolutional layer with a 3×3 kernel with zero padding, batch normalisation, and Gelu as activation layer. For the *downsampling* in the encoder, we sequentially use: one convolutional layer with a 3×3 kernel, stride of 2×2, and zero padding; batch normalisation, and Gelu as activation layer. For the *upsampling* in the decoder, we sequentially use: bilinear interpolation; one convolutional layer with a 3×3 kernel, stride of 2×2, and zero padding; batch normalisation, and Gelu as activation layer. The encoder and decoder are connected via a bottleneck and a shortcut connection at the non-scaled level. Because we use several convolutional layers which extract spatial information and mix the channel at the same time, the network has much more parameters than the U-NeXt (Table B4).

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Table B3. "Conv ($\times 5$)" based on five stages with multiscale convolutional layer.

Stage	Operation	Params	n_{in}	n_{out}	n_x	n_y
Stage 1	MultiConv	11 552	20	32	32	128
	Batch normalisation	64	32	32	32	128
	Gelu	-	32	32	32	128
Stage 2	MultiConv	18 464	32	32	32	128
	Batch normalisation	64	32	32	32	128
	Gelu	-	32	32	32	128
Stage 3	MultiConv	36 928	32	64	32	128
	Batch normalisation	128	64	64	32	128
	Gelu	-	64	64	32	128
Stage 4	MultiConv	73 792	64	64	32	128
	Batch normalisation	128	64	64	32	128
	Gelu	-	64	64	32	128
Stage 5	MultiConv	147 584	64	128	32	128
	Batch normalisation	256	128	128	32	128
	relu	-	128	128	32	128

Table B4. "U-Net" with normal convolutional blocks.

Stage	Operation	Params	n_{in}	n_{out}	n_x	n_y
Input	Conv	23 296	20	128	32	128
Down 1	Downsampling	295 424	128	256	16	64
	Conv	590 336	256	256	16	64
	Conv	590 336	256	256	16	64
	Conv	590 336	256	256	16	64
Bottleneck	Conv	590 336	256	256	16	64
Up 1	Upsampling	295 168	256	128	32	128
	Conv	295 168	128	128	32	128
	Conv	147 712	128	128	32	128
	Conv	147 712	128	128	32	128
Output	Conv (w/o Gelu)	147 712	128	128	32	128
	relu	–	128	128	32	128

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Appendix C: Results for the loss function

Optimising the Laplace log-likelihood corresponds to minimising the MAE, whereas an optimisation of a Gaussian log-likelihood minimises the mean-squared error. Thus, we report the averaged root-mean-squared error (RMSE) and MAE over all variables to measure the influence of the loss function on the performance of the NNs (Table C1). As the RMSE and MAE are normalised by their climatological values in the training dataset, the weighting between the model variables is fixed to their climatological values, favouring fitting networks with fixed climatological weighting.

Table C1. The average RMSE and MAE, normalised by their expected climatology, on the test dataset for different training loss functions. Bold indicates the selected loss function and the best scores.

Name	RMSE	MAE
Gaussian (fixed)	0.34	0.30
Gaussian (trained)	0.33	0.29
Laplace (fixed)	0.33	0.25
Laplace (trained)	**0.32**	**0.24**

Compared to a Gaussian log-likelihood with trainable variance parameters, the Laplace log-likelihood as loss function improves not only the MAE by around $17\,\%$, but also the RMSE by around $3\,\%$. Despite the fixed weighting, fitting the uncertainty parameters together with the NN marginally improves these metrics in both cases. Using adaptive uncertainty parameters modulates the gradient during training, and the optimisation benefits from this adaption, resulting into the shown error decrease.

The loss function influences the output of the NN and the learned features before they are linearly combined to the output (Fig. C1). In the learned features, a Laplace log-likelihood increases the contrast between highly-activated, active, regions and passive regions in the background with a low activation value, Fig. C1, (a) and (b). Here, we define the contrast of a feature map as the ratio between its spatially-averaged standard deviation σ to its spatially-averaged mean value μ. For the Laplace log-likelihood, the median contrast (1.15) is higher than for the Gaussian log-likelihood (0.93), as can be seen in Fig. C1, (c). The distribution for the Laplace log-likelihood is additionally more balanced, meaning that less extreme values appear on both ends. We attribute these differences to the different behaviour of the loss function (Hodson, 2022). As the Gaussian log-likelihood is more sensitive to larger errors in the training dataset, the NN has to learn specialised feature maps for these cases. The Laplace log-likelihood leads to a higher contrast in the feature maps and to more balanced feature maps. Based on its increased contrast, we hypothesise that the Laplace log-likelihood results into better linearly separable feature maps. On their basis, the linear functions can more easily combine the features to predict sharper and more localised residuals, improving the performance of the NN.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

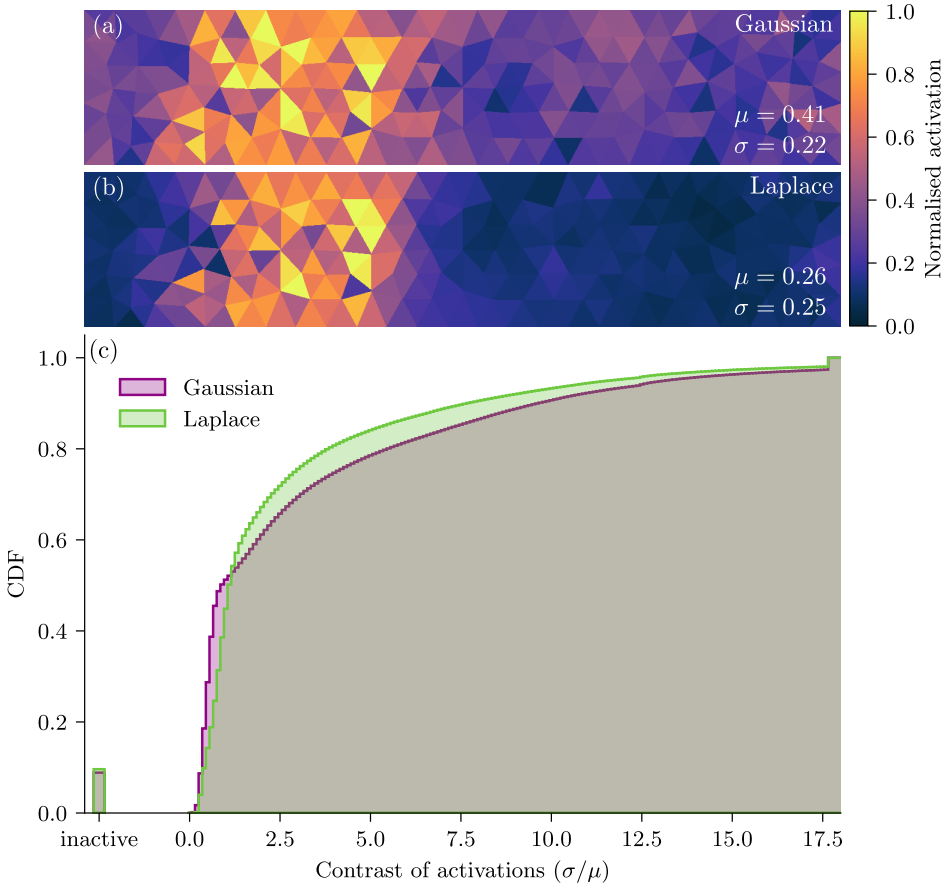

Figure C1. Two typical feature maps for a NN trained with either (a) a Gaussian log-likelihood or (b) a Laplace log-likelihood. For visualisation purpose, the feature maps are again normalised by their 99-th percentile, as in Fig. 5. The contrast (c) is estimated as the spatial standard deviation σ divided by the spatial mean μ of each feature map, extracted from the test dataset. The number of inactivate features maps (=constant zero) is normalised by the same value as the CDF. A higher contrast indicates better linear separable features.

Appendix D: Results for the activation functions

Another decision that we took in our architecture is to use the Gaussian error linear unit (Gelu) in the blocks and the rectified linear unit activation (relu) function as activation of the features, before they are projected back into triangular space and linearly combined. The Gelu activation function is recommended for use in a ConvNeXt block (Liu et al., 2022), but its performance seems to us to be on par with the relu activation function. Whereas the Gelu activation function is a smooth function inspired by dropout Hendrycks and Gimpel (2020), relu is a non-smooth function, which induces sparsity in the feature maps.

As similarly found in Liu et al. (2022), replacing the Gelu activation function with a relu activation function in the ConvNeXt blocks leads to almost the same results on the test dataset (Table D1). Furthermore, also the activation function for the extracted features at the end of the feature extractor has only a small influence. Even using no activation function at this position degrades

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Table D1. Normalised MAE on the test dataset for different activation functions in the ConvNeXt blocks and as feature activation (w/o: no activation function). The error components are estimated as in Table 3 and the same acronyms are used. Bold indicates the selected activation functions and the best scores.

Activation	v	σ_{yy}	d	A	$\overline{\Sigma}$
relu & relu	**0.33**	**0.37**	0.17	0.24	**0.24**
Gelu & Gelu	**0.33**	0.39	0.17	0.23	**0.24**
Gelu & w/o	0.34	0.42	**0.16**	**0.22**	0.25
Gelu & relu	**0.33**	0.38	0.17	0.23	**0.24**

the mean performance by $4\ \%$, for some variables like the damage or area using no activation function leads to the best results. Because the Gelu activation function is state-of-the-art in many deep learning tasks and recommended Liu et al. (2022), we use the Gelu within the ConvNeXt blocks.

In the following, we show feature maps for different activation functions at the feature output of the U-Net as qualitative measure (Fig. D1). Using no activation function (Fig. D1, a), extracts continuous features. These features roughly follow a Cauchy distribution around 0 without enforcing sparsity (Fig. D1, d, the black contour line is the Cauchy distribution fitted via maximum likelihood). Caused by its weighting with the Gaussian error function, Gelu squashes the negative values of the activation values together, leading to a peak of the values around zero. Nevertheless, only few values are truly zero, as GELU do not set values explicitly to zero. In contrast, using relu enforces sparsity, and the NN can extract localised and "patchified" features (Fig. D1, c). Consequently, the relu activation generates many deactivated pixels. Although the performance of the relu activation is the same as for the Gelu activation, we hypothesize that sparse features can improve the representation of subgrid-scale processes, as sea ice, especially at the marginal ice zone, has a discrete character. Therefore, we use for our experiments the Gelu activation function as activation in the ConvNeXt blocks and the relu activation function as feature activation, before the features are linearly combined.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

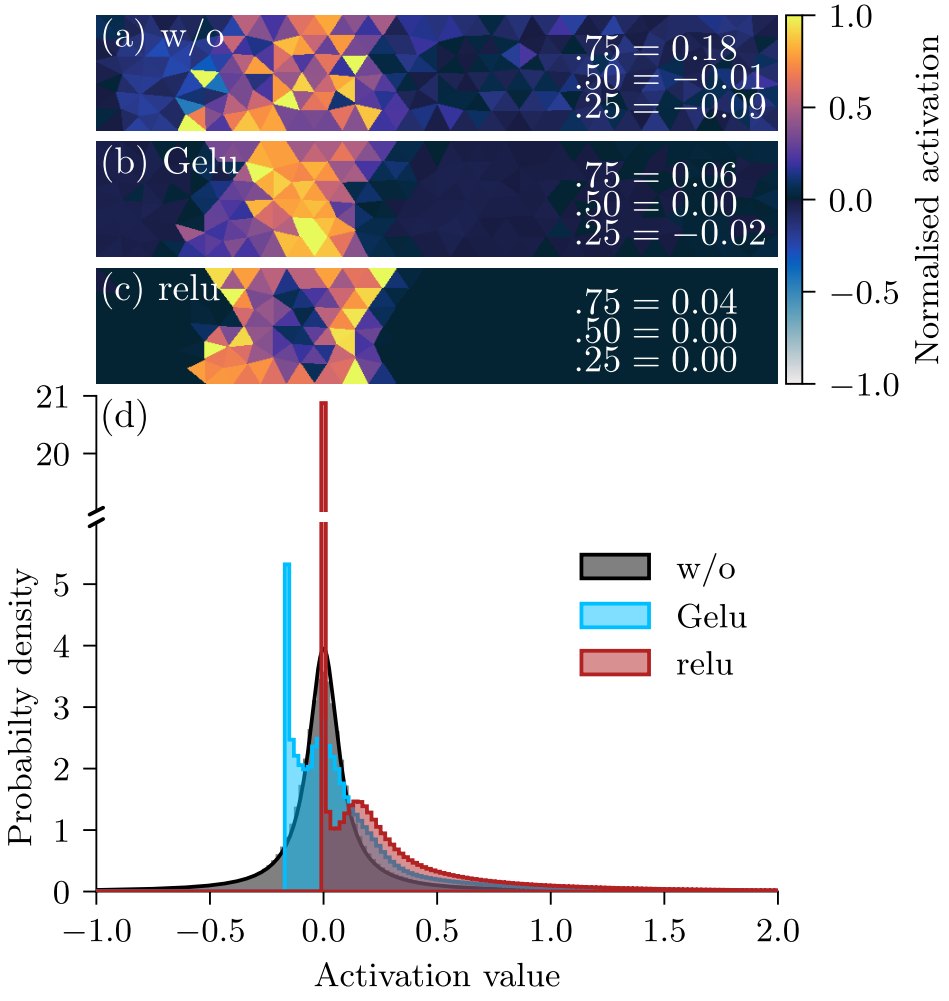

Figure D1. Snapshot of typical feature maps for (a) no feature activation (w/o), (b) the Gaussian error linear unit (Gelu), or (c) the rectified linear unit (relu). For visualisation purpose, the feature maps are normalised by their 99-th percentile. The numbers indicate the percentiles of the normalised feature maps. The histogram (d) represents the unnormalised feature activation values over the whole test dataset. As the histogram for the relu activation function have a large spike at 0 in (d), the y-axis is broken.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Code and data availability. The authors will provide access to the data and weights of the neural networks upon request. The source code for the experiments and the neural networks is publicly available under https://github.com/cerea-daml/hybrid_nn_meb_model. The regional sea-ice model source code will be made available upon request.

Author contributions. MB and AC initialised the scientific questions. TSF, CD, AF, and MB refined the scientific questions and prepared an analysis strategy. TSF, YC, and VD advanced the codebase of the regional sea-ice model. TSF performed the experiments. TSF, CD, AF, and MB analysed and discussed the results. TSF wrote the manuscript with CD, AF, MB, YC, AC, and VD reviewing.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors would like to thank the other members of the project SASIP for delightful discussions along the track. Furthermore, the authors acknowledge the support of the project SASIP (grant no. 353) funded by Schmidt Futures – a philanthropic initiative that seeks to improve societal outcomes through the development of emerging science and technologies. This work was additionally granted access to the HPC resources of IDRIS under the allocation 2021-AD011013069 made by GENCI.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

References

Amitrano, D., Grasso, J.-R., and Hantz, D.: From Diffuse to Localised Damage through Elastic Interaction, Geophysical Research Letters, 26, 2109–2112, https://doi.org/10.1029/1999GL900388, 1999.

Andersson, T. R., Hosking, J. S., Pérez-Ortiz, M., Paige, B., Elliott, A., Russell, C., Law, S., Jones, D. C., Wilkinson, J., Phillips, T., Byrne, J., Tietsche, S., Sarojini, B. B., Blanchard-Wrigglesworth, E., Aksenov, Y., Downie, R., and Shuckburgh, E.: Seasonal Arctic Sea Ice Forecasting with Probabilistic Deep Learning, Nature Communications, 12, 5124, https://doi.org/10.1038/s41467-021-25257-4, 2021.

Ayzel, G., Scheffer, T., and Heistermann, M.: RainNet v1.0: A Convolutional Neural Network for Radar-Based Precipitation Nowcasting, Geoscientific Model Development, 13, 2631–2644, https://doi.org/10.5194/gmd-13-2631-2020, 2020.

Ba, J. L., Kiros, J. R., and Hinton, G. E.: Layer Normalization, arXiv:1607.06450 [cs, stat], 2016.

Bachlechner, T., Majumder, B. P., Mao, H. H., Cottrell, G. W., and McAuley, J.: ReZero Is All You Need: Fast Convergence at Large Depth, https://doi.org/10.48550/arXiv.2003.04887, 2020.

Beck, A. and Kurz, M.: A Perspective on Machine Learning Methods in Turbulence Modeling, GAMM-Mitteilungen, 44, e202100 002, https://doi.org/10.1002/gamm.202100002, 2021.

Bennetts, L. G., Bitz, C. M., Feltham, D. L., Kohout, A. L., and Meylan, M. H.: Theory, Modelling and Observations of Marginal Ice Zone Dynamics: Multidisciplinary Perspectives and Outlooks, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 380, 20210 265, https://doi.org/10.1098/rsta.2021.0265, 2022.

Beucler, T., Ebert-Uphoff, I., Rasp, S., Pritchard, M., and Gentine, P.: Machine Learning for Clouds and Climate (Invited Chapter for the AGU Geophysical Monograph Series "Clouds and Climate"), https://doi.org/10.1002/essoar.10506925.1, 2021.

Bloom, S. C., Takacs, L. L., da Silva, A. M., and Ledvina, D.: Data Assimilation Using Incremental Analysis Updates, Monthly Weather Review, 124, 1256–1271, https://doi.org/10.1175/1520-0493(1996)124<1256:DAUIAU>2.0.CO;2, 1996.

Bocquet, M., Brajard, J., Carrassi, A., and Bertino, L.: Data Assimilation as a Learning Tool to Infer Ordinary Differential Equation Representations of Dynamical Models, Nonlinear Processes in Geophysics, 26, 143–162, https://doi.org/10.5194/npg-26-143-2019, 2019.

Bocquet, M., Brajard, J., Carrassi, A., and Bertino, L.: Bayesian Inference of Chaotic Dynamics by Merging Data Assimilation, Machine Learning and Expectation-Maximization, Foundations of Data Science, 2, 55, https://doi.org/10.3934/fods.2020004, 2020.

Bouchat, A., Hutter, N., Chanut, J., Dupont, F., Dukhovskoy, D., Garric, G., Lee, Y. J., Lemieux, J.-F., Lique, C., Losch, M., Maslowski, W., Myers, P. G., Ólason, E., Rampal, P., Rasmussen, T., Talandier, C., Tremblay, B., and Wang, Q.: Sea Ice Rheology Experiment (SIREx): 1. Scaling and Statistical Properties of Sea-Ice Deformation Fields, Journal of Geophysical Research: Oceans, 127, e2021JC017 667, https://doi.org/10.1029/2021JC017667, 2022.

Boutin, G., Williams, T., Horvat, C., and Brodeau, L.: Modelling the Arctic Wave-Affected Marginal Ice Zone: A Comparison with ICESat-2 Observations, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 380, 20210 262, https://doi.org/10.1098/rsta.2021.0262, 2022.

Brajard, J., Carrassi, A., Bocquet, M., and Bertino, L.: Combining Data Assimilation and Machine Learning to Emulate a Dynamical Model from Sparse and Noisy Observations: A Case Study with the Lorenz 96 Model, Journal of Computational Science, 44, 101 171, https://doi.org/10.1016/j.jocs.2020.101171, 2020.

Brajard, J., Carrassi, A., Bocquet, M., and Bertino, L.: Combining Data Assimilation and Machine Learning to Infer Unresolved Scale Parametrization, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379, 20200 086, https://doi.org/10.1098/rsta.2020.0086, 2021.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Breiman, L.: Statistical Modeling: The Two Cultures (with Comments and a Rejoinder by the Author), Statistical Science, 16, 199–231, https://doi.org/10.1214/ss/1009213726, 2001.

Brenowitz, N. D. and Bretherton, C. S.: Prognostic Validation of a Neural Network Unified Physics Parameterization, Geophysical Research Letters, 45, 6289–6298, https://doi.org/10.1029/2018GL078510, 2018.

Carrassi, A., Weber, R. J. T., Guemas, V., Doblas-Reyes, F. J., Asif, M., and Volpi, D.: Full-Field and Anomaly Initialization Using a Low-Order Climate Model: A Comparison and Proposals for Advanced Formulations, Nonlinear Processes in Geophysics, 21, 521–537, https://doi.org/10.5194/npg-21-521-2014, 2014.

Cheng, Y., Giometto, M. G., Kauffmann, P., Lin, L., Cao, C., Zupnick, C., Li, H., Li, Q., Huang, Y., Abernathey, R., and Gentine, P.: Deep Learning for Subgrid-Scale Turbulence Modeling in Large-Eddy Simulations of the Convective Atmospheric Boundary Layer, Journal of Advances in Modeling Earth Systems, 14, e2021MS002 847, https://doi.org/10.1029/2021MS002847, 2022.

Cipolla, R., Gal, Y., and Kendall, A.: Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7482–7491, IEEE, Salt Lake City, UT, USA, https://doi.org/10.1109/CVPR.2018.00781, 2018.

Dansereau, V., Weiss, J., Saramito, P., and Lattes, P.: A Maxwell Elasto-Brittle Rheology for Sea Ice Modelling, The Cryosphere, 10, 1339–1359, https://doi.org/10.5194/tc-10-1339-2016, 2016.

Dansereau, V., Weiss, J., Saramito, P., Lattes, P., and Coche, E.: Ice Bridges and Ridges in the Maxwell-EB Sea Ice Rheology, The Cryosphere, 11, 2033–2058, https://doi.org/10.5194/tc-11-2033-2017, 2017.

Dansereau, V., Weiss, J., and Saramito, P.: A Continuum Viscous-Elastic-Brittle, Finite Element DG Model for the Fracture and Drift of Sea Ice, in: Challenges and Innovations in Geomechanics, edited by Barla, M., Di Donna, A., and Sterpi, D., Lecture Notes in Civil Engineering, pp. 125–139, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-030-64514-4_8, 2021.

De, S. and Smith, S. L.: Batch Normalization Biases Residual Blocks Towards the Identity Function in Deep Networks, https://doi.org/10.48550/arXiv.2002.10444, 2020.

Falcon, W., Borovec, J., Wälchli, A., Eggert, N., Schock, J., Jordan, J., Skafte, N., Ir1dXD, Bereznyuk, V., Harris, E., Murrell, T., Yu, P., Præsius, S., Addair, T., Zhong, J., Lipin, D., Uchida, S., Bapat, S., Schröter, H., Dayma, B., Karnachev, A., Kulkarni, A., Komatsu, S., Martin.B, SCHIRATTI, J.-B., Mary, H., Byrne, D., Eyzaguirre, C., cinjon, and Bakhtin, A.: PyTorchLightning: 0.7.6 Release, Zenodo, https://doi.org/10.5281/zenodo.3828935, 2020.

Farchi, A., Bocquet, M., Laloyaux, P., Bonavita, M., and Malartic, Q.: A Comparison of Combined Data Assimilation and Machine Learning Methods for Offline and Online Model Error Correction, Journal of Computational Science, 55, 101 468, https://doi.org/10.1016/j.jocs.2021.101468, 2021a.

Farchi, A., Laloyaux, P., Bonavita, M., and Bocquet, M.: Using Machine Learning to Correct Model Error in Data Assimilation and Forecast Applications, Quarterly Journal of the Royal Meteorological Society, 147, 3067–3084, https://doi.org/10.1002/qj.4116, 2021b.

Fisher, R. A.: Frequency Distribution of the Values of the Correlation Coefficient in Samples from an Indefinitely Large Population, Biometrika, 10, 507–521, https://doi.org/10.2307/2331838, 1915.

Geer, A. J.: Learning Earth System Models from Observations: Machine Learning or Data Assimilation?, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379, 20200 089, https://doi.org/10.1098/rsta.2020.0089, 2021.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G.: Could Machine Learning Break the Convection Parameterization Deadlock?, Geophysical Research Letters, 45, 5742–5751, https://doi.org/10.1029/2018GL078202, 2018.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Girard, L., Bouillon, S., Weiss, J., Amitrano, D., Fichefet, T., and Legat, V.: A New Modeling Framework for Sea-Ice Mechanics Based on Elasto-Brittle Rheology, Annals of Glaciology, 52, 123–132, https://doi.org/10.3189/172756411795931499, 2011.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: Generative Adversarial Nets, in: Advances in Neural Information Processing Systems 27, edited by Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., pp. 2672–2680, Curran Associates, Inc., 2014.

Guillaumin, A. P. and Zanna, L.: Stochastic-Deep Learning Parameterization of Ocean Momentum Forcing, Journal of Advances in Modeling Earth Systems, 13, e2021MS002 534, https://doi.org/10.1029/2021MS002534, 2021.

Hendrycks, D. and Gimpel, K.: Gaussian Error Linear Units (GELUs), https://doi.org/10.48550/arXiv.1606.08415, 2020.

Hodson, T. O.: Root-Mean-Square Error (RMSE) or Mean Absolute Error (MAE): When to Use Them or Not, Geoscientific Model Development, 15, 5481–5487, https://doi.org/10.5194/gmd-15-5481-2022, 2022.

Horvat, C.: Marginal Ice Zone Fraction Benchmarks Sea Ice and Climate Model Skill, Nature Communications, 12, 2221, https://doi.org/10.1038/s41467-021-22004-7, 2021.

Horvat, C. and Roach, L. A.: WIFF1.0: A Hybrid Machine-Learning-Based Parameterization of Wave-Induced Sea Ice Floe Fracture, Geoscientific Model Development, 15, 803–814, https://doi.org/10.5194/gmd-15-803-2022, 2022.

Houghton, J. T., Ding, Y., Griggs, D. J., Noguer, M., van der Linden, P. J., Dai, X., Maskell, K., and Johnson, C.: Climate Change 2001: The Scientific Basis: Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge university press, 2001.

Irrgang, C., Boers, N., Sonnewald, M., Barnes, E. A., Kadow, C., Staneva, J., and Saynisch-Wagner, J.: Towards Neural Earth System Modelling by Integrating Artificial Intelligence in Earth System Science, Nature Machine Intelligence, 3, 667–674, https://doi.org/10.1038/s42256-021-00374-3, 2021.

Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, arXiv:1412.6980 [cs], 2017.

Korosov, A., Rampal, P., Ying, Y., Ólason, E., and Williams, T.: Towards Improving Short-Term Sea Ice Predictability Using Deformation Observations, The Cryosphere Discussions, pp. 1–20, https://doi.org/10.5194/tc-2022-46, 2022.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J.: Addressing Distribution Shift in Online Reinforcement Learning with Offline Datasets, 2021.

Levine, S., Kumar, A., Tucker, G., and Fu, J.: Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems, https://doi.org/10.48550/arXiv.2005.01643, 2020.

Liu, Q., Zhang, R., Wang, Y., Yan, H., and Hong, M.: Daily Prediction of the Arctic Sea Ice Concentration Using Reanalysis Data Based on a Convolutional LSTM Network, Journal of Marine Science and Engineering, 9, 330, https://doi.org/10.3390/jmse9030330, 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S.: A ConvNet for the 2020s, 2022.

Murphy, K. P.: Machine Learning: A Probabilistic Perspective, Adaptive Computation and Machine Learning Series, MIT Press, Cambridge, MA, 2012.

Norton, R. M.: The Double Exponential Distribution: Using Calculus to Find a Maximum Likelihood Estimator, The American Statistician, 38, 135–136, https://doi.org/10.2307/2683252, 1984.

Odena, A., Dumoulin, V., and Olah, C.: Deconvolution and Checkerboard Artifacts, Distill, 1, e3, https://doi.org/10.23915/distill.00003, 2016.

Ólason, E., Rampal, P., and Dansereau, V.: On the Statistical Properties of Sea-Ice Lead Fraction and Heat Fluxes in the Arctic, The Cryosphere, 15, 1053–1064, https://doi.org/10.5194/tc-15-1053-2021, 2021.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Ólason, E., Boutin, G., Korosov, A., Rampal, P., Williams, T., Kimmritz, M., Dansereau, V., and Samaké, A.: A New Brittle Rheology and Numerical Framework for Large-Scale Sea-Ice Models, Journal of Advances in Modeling Earth Systems, 14, e2021MS002 685, https://doi.org/10.1029/2021MS002685, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems 32, edited by Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., and Garnett, R., pp. 8024–8035, Curran Associates, Inc., 2019.

Prudencio, R. F., Maximo, M. R. O. A., and Colombini, E. L.: A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems, https://doi.org/10.48550/arXiv.2203.01387, 2022.

Rabatel, M., Rampal, P., Carrassi, A., Bertino, L., and Jones, C. K. R. T.: Impact of Rheology on Probabilistic Forecasts of Sea Ice Trajectories: Application for Search and Rescue Operations in the Arctic, The Cryosphere, 12, 935–953, https://doi.org/10.5194/tc-12-935-2018, 2018.

Rampal, P., Bouillon, S., Ólason, E., and Morlighem, M.: neXtSIM: A New Lagrangian Sea Ice Model, The Cryosphere, 10, 1055–1073, https://doi.org/10.5194/tc-10-1055-2016, 2016.

Rampal, P., Dansereau, V., Olason, E., Bouillon, S., Williams, T., Korosov, A., and Samaké, A.: On the Multi-Fractal Scaling Properties of Sea Ice Deformation, The Cryosphere, 13, 2457–2474, https://doi.org/10.5194/tc-13-2457-2019, 2019.

Rasp, S.: 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, 2185–2196, https://doi.org/10.5194/gmd-13-2185-2020, 2020.

Rasp, S., Pritchard, M. S., and Gentine, P.: Deep Learning to Represent Subgrid Processes in Climate Models, Proceedings of the National Academy of Sciences, 115, 9684–9689, https://doi.org/10.1073/pnas.1810286115, 2018.

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., Prudden, R., Mandhane, A., Clark, A., Brock, A., Simonyan, K., Hadsell, R., Robinson, N., Clancy, E., Arribas, A., and Mohamed, S.: Skilful Precipitation Nowcasting Using Deep Generative Models of Radar, Nature, 597, 672–677, https://doi.org/10.1038/s41586-021-03854-z, 2021.

Ronneberger, O., Fischer, P., and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597 [cs], 2015.

Rybkin, O., Daniilidis, K., and Levine, S.: Simple and Effective VAE Training with Calibrated Decoders, arXiv:2006.13202 [cs, eess, stat], 2020.

Saramito, P.: Efficient C++ Finite Element Computing with Rheolef, p. 279, 2022.

Schweiger, A. J. and Zhang, J.: Accuracy of Short-Term Sea Ice Drift Forecasts Using a Coupled Ice-Ocean Model, Journal of Geophysical Research: Oceans, 120, 7827–7841, https://doi.org/10.1002/2015JC011273, 2015.

Seifert, A. and Rasp, S.: Potential and Limitations of Machine Learning for Modeling Warm-Rain Cloud Microphysical Processes, Journal of Advances in Modeling Earth Systems, 12, e2020MS002 301, https://doi.org/10.1029/2020MS002301, 2020.

Simonyan, K., Vedaldi, A., and Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, https://doi.org/10.48550/arXiv.1312.6034, 2013.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M.: SmoothGrad: Removing Noise by Adding Noise, https://doi.org/10.48550/arXiv.1706.03825, 2017.

https://doi.org/10.5194/egusphere-2022-1342
Preprint. Discussion started: 2 January 2023

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S.: Deep Unsupervised Learning Using Nonequilibrium Thermodynamics, arXiv:1503.03585 [cond-mat, q-bio, stat], 2015.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M.: Striving for Simplicity: The All Convolutional Net, https://doi.org/10.48550/arXiv.1412.6806, 2015.

Stockdale, T. N.: Coupled Ocean–Atmosphere Forecasts in the Presence of Climate Drift, Monthly Weather Review, 125, 809–818, https://doi.org/10.1175/1520-0493(1997)125<0809:COAFIT>2.0.CO;2, 1997.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.: Going Deeper with Convolutions, arXiv:1409.4842 [cs], 2014.

Tang, C.: Numerical Simulation of Progressive Rock Failure and Associated Seismicity, International Journal of Rock Mechanics and Mining Sciences, 34, 249–261, https://doi.org/10.1016/S0148-9062(96)00039-3, 1997.

Tomczak, J. M.: Deep Generative Modeling, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-030-93158-2, 2022.

Wilchinsky, A. V. and Feltham, D. L.: Modelling the Rheology of Sea Ice as a Collection of Diamond-Shaped Floes, Journal of Non-Newtonian Fluid Mechanics, 138, 22–32, https://doi.org/10.1016/j.jnnfm.2006.05.001, 2006.

Wilchinsky, A. V. and Feltham, D. L.: Modeling Coulombic Failure of Sea Ice with Leads, Journal of Geophysical Research: Oceans, 116, https://doi.org/10.1029/2011JC007071, 2011.

Yadan, O.: Hydra - A Framework for Elegantly Configuring Complex Applications, Github, 2019.

Zanna, L. and Bolton, T.: Data-Driven Equation Discovery of Ocean Mesoscale Closures, Geophysical Research Letters, 47, e2020GL088 376, https://doi.org/10.1029/2020GL088376, 2020.