Enviromental Data Science (2023), **4**:20 1–16

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

An Analysis of Deep Learning Parameterizations for Ocean Subgrid Eddy Forcing

Cem Gultekin[1], Adam Subel[1], Cheng Zhang[3], Matan Leibovich[1], Pavel Perezhogin[1], Alistair Adcroft[3], Carlos Fernandez-Granda[1,2] and Laure Zanna[1]

[1]Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA.
[2]Center for Data Science, New York University, New York, NY 10011, USA.
[3]Program in Atmospheric and Oceanic Sciences, Princeton University, Princeton, NJ 08542, USA.

Keywords: mesoscale eddies, deep learning parameterization, climate modeling

MSC Codes: *Primary* – CODE1; *Secondary* – CODE2, CODE3

Abstract
Due to computational constraints, climate simulations cannot resolve a range of small-scale physical processes, which have a significant impact on the large-scale evolution of the climate system. Parameterization is an approach to capture the effect of these processes, without resolving them explicitly. In recent years, data-driven parameterizations based on convolutional neural networks have obtained promising results. In this work, we provide an in-depth analysis of these parameterizations developed using data from ocean simulations. The parametrizations account for the effect of mesoscale eddies toward improving simulations of momentum, heat, and mass exchange in the ocean. Our results provide several insights into the properties of data-driven parameterizations based on neural networks. First, their performance can be substantially improved by increasing the geographic extent of the training data. Second, they learn nonlinear structure, since they are able to outperform a linear baseline. Third, they generalize robustly across different CO_2 forcings, but not necessarily across different ocean depths. Fourth, they exploit a relatively small region of their input to generate their output. Our results will guide the further development of ocean mesoscale eddy parameterizations, and multiscale modeling more generally.

Impact Statement

Climate simulations spanning decades or centuries cannot be run at a high spatial resolution due to computational constraints. Unfortunately, as a result, critical fine-scale physical processes such as mesoscale eddies are not captured by the simulations. Deep learning has emerged as a promising solution to incorporate the effect of these processes. In this work, we evaluate several properties of these models that are crucial for their deployment in climate simulations.

1. Introduction

Despite advances in hardware, climate simulations spanning decades or centuries are limited in their spatial resolution to tens of kilometers (Balaji, 2021). This resolution is insufficient to resolve mesoscale eddies that are crucial for the exchange of momentum, heat, and mass (Capet et al., 2008; Salmon, 1980) as well as for large-scale ocean circulation (Hallberg, 2013a; Keating et al., 2012; Waterman and Jayne, 2011). Parameterization is an approach to capture the effect of small-scale processes on the large-scale variables in climate models without resolving them explicitly. The main idea is to modify the governing equations of a low-resolution model by adding a forcing term called subgrid forcing. This term encapsulates the *missing physics*, representing the effect of the unresolved physical processes on the resolved variables in the climate model. The key challenge is that the subgrid forcing is a function of high-resolution quantities, but only low-resolution quantities are observed in the model. Traditional parameterization approaches are typically rooted in first-principle analysis of the climate-model physics (Bony et al., 2015; Randall et al., 2003; Schneider et al., 2017).

In recent years, data-driven parameterizations have been developed with promising results (Beucler et al., 2021; Bodner et al., 2023; Frezat et al., 2021; Guan et al., 2022; Guillaumin and Zanna, 2021; Perezhogin et al., 2024; Sane et al., 2023; Yuval and O'Gorman, 2020; Zanna and Bolton, 2020; Zhang et al., 2023). In the data-driven framework, a high-resolution simulation capable of resolving the physical processes of interest is utilized as ground truth. The resolution of the data is then artificially reduced via filtering and coarse-graining to enable the computation of "ground-truth" subgrid forcing. Machine learning (ML) algorithms are used to predict this forcing from the low-resolution data. In the case of mesoscale eddies, applying this framework to convolutional neural networks (CNNs) has achieved promising results (Guillaumin and Zanna, 2021; Zanna and Bolton, 2020).

Our goal in this work is to provide an in-depth study of CNNs for parameterization. We utilize the CM2.6 dataset as our source of high-resolution data. CM2.6 is a publicly-available advanced coupled climate model (Griffies, 2015) with an approximate resolution of 0.1°, making it well-suited for accurately representing mesoscale eddies (Hallberg, 2013b). The dataset includes surface and subsurface data for two levels of CO_2 in the atmosphere. Guillaumin and Zanna, 2021 proposed a CNN-parameterization for surface momentum fields in CM2.6 at 0.4° resolution. Here, we extend their approach to parameterize temperature, and build upon it to study the following key questions about CNN-based mesoscale-eddy parameterizations.

Does the geographic extent of the training dataset matter? In Guillaumin and Zanna (2021), the training domain was constrained to four relatively small regions of the Pacific and Atlantic Oceans. We show that extending the training dataset to encompass the entire global ocean surface leads to a substantial improvement in performance, particularly for temperature.

Do the properties of CNN parameterizations change at different resolutions? When training a CNN parameterization, an important consideration is how to choose the resolution of the target low-resolution climate model. We studied the performance of CNN-based parameterizations at resolutions of 0.4°, 0.8°, 1.2°, and 1.6°. Our results show to what extent reducing resolution decreases the skill of the parameterizations, and also demonstrate that their properties remain similar across the different resolutions.

Do CNN-parameterizations just *invert* the coarse-graining and filtering operator? As explained above, CNN-parameterizations estimate subgrid forcing from low-resolution data, obtained from a high-resolution model via filtering and coarse-graining. These filtering and coarse-graining operators are linear and can be partially inverted to estimate subgrid turbulence fields (and hence the subgrid forcing itself) (Langford and Moser, 1999). Examples of subgrid parameterizations based on partial filter inversion include the velocity gradient model (Chow et al., 2005; Clark et al., 1979), the scale-similarity model (Bardina et al., 1980; Meneveau and Katz, 2000) and the approximate deconvolution model (Stolz and Adams, 1999). An important question is whether CNN-parameterizations just learn to implement this partial inversion. We developed a baseline parameterization solely based on linear inversion to answer it. Our results indicate that CNNs outperform this baseline parameterization, suggesting that they are able to leverage physical structure and do not merely learn to perform inversion.

Do data-driven parameterizations generalize across CO_2 levels and ocean depths? Understanding the generalization properties of data-driven parameterizations is crucial for their deployment in realistic models. In particular, determining their behavior at different CO_2 levels is critical for long-term climate simulations and predictions. Our results show that a CNN-based parameterization trained at pre-industrial CO_2 levels generalizes robustly at significantly increased CO_2 levels. In addition, we investigate the generalization ability these models across different ocean depths, ranging from the surface down to 728 meters. In MOM-suite climate models, the ocean is represented as vertically stacked, irregularly spaced horizontal layers (Griffies, 2015), with substantially different dynamics. At the surface, wind, solar radiation, and atmospheric interactions dominate circulation, whereas at greater depths, dynamics depend more on temperature and salinity (Salmon, 1980). Our results show that surface-trained models have difficulties generalizing at greater depths, and, conversely, models trained beyond 55 meters do not generalize robustly to the surface.

What spatial extent do CNN-parameterizations require as input? The spatial extent needed to compute a parameterization at each grid point greatly influences its computational cost. In addition, it informs the degree of nonlocality needed to approximate the local subgrid forcing. In order to investigate its impact on CNN-parameterizations, we evaluated the performance of multiple CNNs with a range of input sizes. These experiments were complemented with a gradient-based sensitivity analysis. Our results indicate that a much smaller input size than the one used in previous models is sufficient to achieve strong performance.

The paper is organized into five sections, including the introduction. In Section 2, we describe the governing equations of the climate simulation, and explain how we generate the data to train the data-driven parameterizations. In Section 3, we define data-driven parameterizations based on deep learning, as well as a baseline parameterization based on linear inversion. In Section 4, we describe our experiments and present the results. Finally, in Section 5, we summarize our conclusions and discuss directions for future research.

2. Governing Equations and Data Coarsening

In this section we describe the governing equations of our climate model of interest, as well as the data-coarsening procedure used to generate the data needed to train data-driven parameterizations. Our *ground-truth* high-resolution climate model is CM2.6, a coupled model that includes various physical quantities describing oceanic evolution, such as momentum, temperature, salinity, and biogeochemical tracers (Griffies, 2015). The model resolution ($0.1°$) is sufficient to faithfully represent mesoscale eddies. In our experiments, we consider seven different ocean depths, including the surface, and two different CO_2 levels.

Our study focuses on the evolution of three variables: longitudinal and latitudinal momentum fields (u and v, respectively) and the temperature field (T). The evolution of these quantities is governed by the following equation:

$$\frac{\partial c^{\uparrow}}{\partial t} + \boldsymbol{u}^{\uparrow} \cdot \nabla c^{\uparrow} = F_c^{\uparrow}, \quad c \in \{u, v, T\} \tag{2.1}$$

We use the arrow symbol $^{\uparrow}$ to denote fine-grid variables. The vector $\boldsymbol{u}^{\uparrow}$ represents the components of longitudinal (u) and latitudinal (v) momentum on the fine grid. The gradient operator ∇ contains the spatial derivatives in these two directions using spherical coordinates. Note that we are omitting the discussion of the vertical momentum contribution to the equation, merely because its contribution to the subgrid terms diagnosed in the model is small compared to the horizontal terms. The momentum forcing terms $F_u^{\uparrow}$ and $F_v^{\uparrow}$ account for external sources of dynamics, such as frictional and pressure stresses, the Coriolis force, or transport between vertically adjacent stratified ocean layers (Griffies, 2015). The temperature forcing $F_T^{\uparrow}$ accounts for isopycnal and diapycnal mixing, as well as heat fluxes at the ocean surface.

In climate modeling, it is common to employ a local Cartesian coordinate system that approximates the Earth's surface as a flat plane when dealing with small variations. This coordinate system allows for measurements in meters rather than angles. Therefore, we define $\nabla = (\partial_x, \partial_y)$ as the spatial gradient in meters along the longitudinal and latitudinal directions.

Our goal is to study the properties of data-driven parameterizations that estimate the subgrid forcing of low-resolution versions of CM2.6. To obtain these coarser datasets, we reduce the resolution of the CM2.6 data via a two-step procedure proposed in Guillaumin and Zanna (2021), consisting of filtering and coarse-graining. Filtering smooths the data on the original high-resolution fine grid.

We leverage two alternative filters for this purpose: Gaussian filtering and General-Circulation-Model (GCM) filtering (Loose et al., 2022).

We use a tilde ¯ to denote the low-resolution variables. The equations governing the low-resolution variables are derived from (2.1) by applying filtering on both sides.

The vertical advection term ($w\frac{\partial c}{\partial z}$) is significantly smaller than the horizontal terms, at all depths in this dataset. Therefore we reduce our analysis to the 2D system:

In order to achieve the same form as the high-resolution equation, the low-resolution advective term, $\overline{\boldsymbol{u}^{\uparrow}} \cdot \nabla \overline{c^{\uparrow}}$ is added to both sides and the terms are rearranged:

$$\overline{\left(\frac{\partial c^{\uparrow}}{\partial t} + \boldsymbol{u}^{\uparrow} \cdot \nabla c^{\uparrow}\right)} = \overline{F_c^{\uparrow}}, \tag{2.2a}$$

$$\frac{\partial \overline{c^{\uparrow}}}{\partial t} + \overline{\boldsymbol{u}^{\uparrow}} \cdot \nabla \overline{c^{\uparrow}} = \overline{\boldsymbol{u}^{\uparrow}} \cdot \nabla \overline{c^{\uparrow}} - \overline{\left(\boldsymbol{u}^{\uparrow} \cdot \nabla c^{\uparrow}\right)} + \overline{F_c^{\uparrow}} \tag{2.2b}$$

$$S_c^{\uparrow} \equiv \overline{\boldsymbol{u}^{\uparrow}} \cdot \nabla \overline{c^{\uparrow}} - \overline{\left(\boldsymbol{u}^{\uparrow} \cdot \nabla c^{\uparrow}\right)}, \quad c \in \{u, v, T\} \tag{2.2c}$$

The term $S_c^{\uparrow}$ is the *subgrid forcing* on the fine-grid. The low resolution data are mapped to a coarser grid via coarse-graining which reduces the grid dimensions by a coarse-graining factor κ. This yields the coarse-grid dataset. We use $\langle\rangle_\kappa$ to denote the coarse-graining operation. For example, the longitudinal momentum on the coarse grid equals

$$u = \left\langle \overline{u^{\uparrow}} \right\rangle_\kappa.$$

Crucially, the coarse-grained subgrid forcing S_c,

$$S_c = \left\langle \overline{\boldsymbol{u}^{\uparrow}} \cdot \nabla \overline{c^{\uparrow}} - \overline{\boldsymbol{u}^{\uparrow} \cdot \nabla c^{\uparrow}} \right\rangle_\kappa, \quad c = \{u, v, T\}, \tag{2.3}$$

depicted in Figure 1, depends on the fine-grained quantities. The key challenge of parameterization is to estimate this subgrid forcing from coarse-grained data, in order to *close* the corresponding coarse-grid equations.

3. Methodology

3.1. Data-Driven Parameterization Based on Deep Learning

In this section, we describe a data-driven framework for the estimation of the subgrid forcing defined in Section 2 using deep neural networks. To simplify the exposition, let $S_{t,x}$ indicate the subgrid forcing corresponding to one of our three variables of interest (longitudinal momentum, latitudinal momentum, or temperature) at a time t and location x. The goal is to estimate $S_{t,x}$ from the low-resolution momentum and temperature fields, denoted by $\mathbf{u}_{t,x}$ and $T_{t,x}$, surrounding that location at that time.

Following previous studies (Bolton and Zanna, 2019; Guillaumin and Zanna, 2021; Perezhogin et al., 2023; Ross et al., 2023), we used a deep convolutional neural network $\widehat{S}_\Theta$, where Θ represents the network parameters, to produce an estimate $\widehat{S}_\Theta(\mathbf{u}_{t,x}, T_{t,x})$ of the subgrid forcing $S_{t,x}$ given the low-resolution data $\mathbf{u}_{t,x}$ and $T_{t,x}$. In order to train and evaluate the neural network, we partitioned the CM2.6 data in time: the first 80% was used to create a training set $\mathcal{T}_{\text{train}}$ and the last 15% to create a test set $\mathcal{T}_{\text{test}}$, leaving a 5% buffer in between to minimize dependence between the sets due to autocorrelation. The training and test examples were generated from the high-resolution CM2.6 data: $\mathbf{u}_{t,x}$ and $T_{t,x}$ were obtained via filtering and coarse-graining of the high-resolution momentum and temperature maps, and the corresponding ground-truth subgrid forcing $S_{t,x}$ was computed following (2.3). These data correspond to a control simulation with pre-industrial atmospheric CO_2 levels.

In addition, a forced simulation with increased CO_2 levels was used to create another test set. The forced simulation experiences a one percent increase in CO_2 levels per year from the levels of

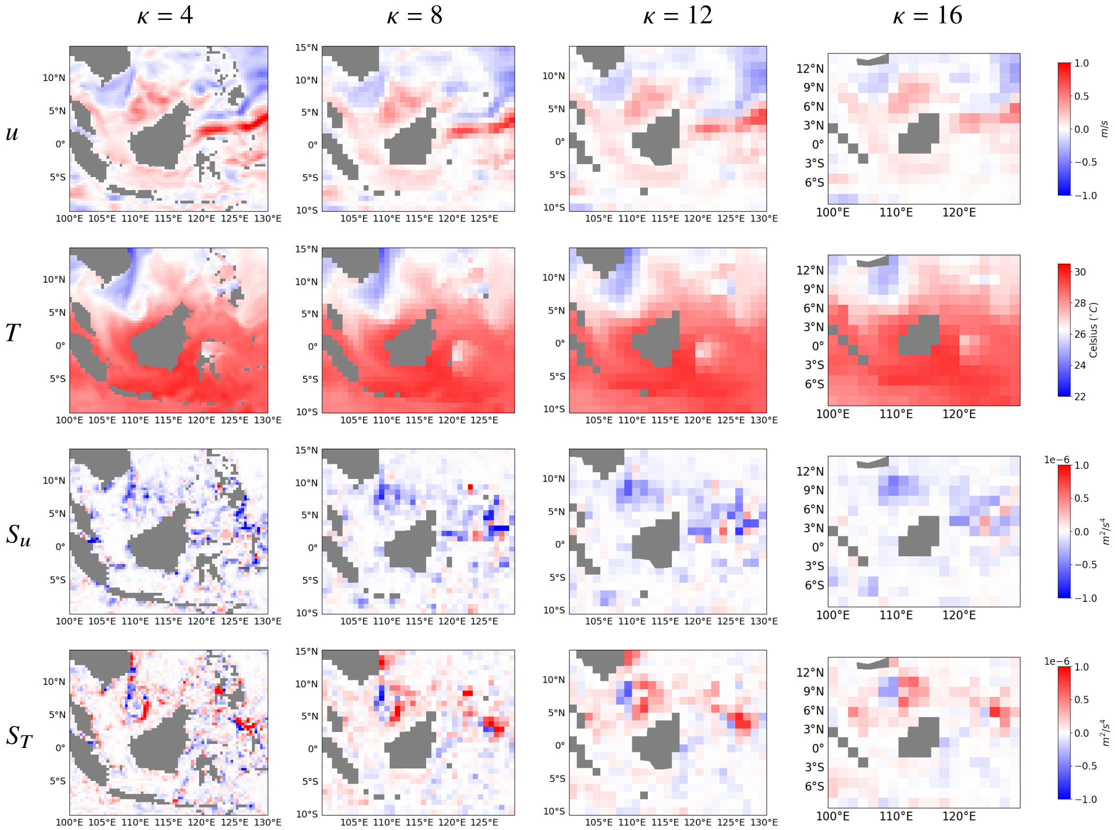

Figure 1. Examples of coarse-grid variables for different values of the coarsening factor κ along with their corresponding subgrid eddy forcing. The fine-grid data are extracted from the CM2.6 surface dataset, and filtered using GCM filtering. The rows represent longitudinal momentum (u), temperature (T), subgrid longitudinal momentum forcing (S_u), and subgrid temperature forcing (S_T) respectively.

the control simulation until it reaches double CO_2 levels after 70 years. The available data from the simulation corresponds to years 60–80. These data were processed in the same way as data from the control simulation to create the +1%CO_2 test dataset.

Additional training and test sets were created from subsurface data at different ocean depths. These data from the simulation at control CO_2 levels, are available in the form of 5-day averages over the same time period. They were processed in the same way as the surface data to generate corresponding training and test sets for each depth.

The parameters of the neural network were learned via minimization of a loss function, quantifying the approximation error over the training set. We consider two different loss functions. The first loss is the mean squared error (MSE), approximated by the residual sum of squared errors,

$$\mathrm{MSE}(\Theta) = \frac{1}{n_{\text{train}}} \sum_{t \in \mathcal{T}_{\text{train}}} \sum_{x \in \mathcal{X}} \left(\widehat{S}_{\Theta}(\mathbf{u}_{t,x}, T_{t,x}) - S_{t,x} \right)^2, \tag{3.1}$$

where $\mathcal{T}_{\text{train}}$, $\mathcal{X}$ and n_{train} denote the times, locations and number of data included in the training set. The second loss is the heteroscedastic Gaussian loss (HGL) proposed in Guillaumin and Zanna, 2021, which simultaneously estimates the conditional mean and conditional variance of the subgrid forcing by maximizing a Gaussian likelihood

$$\mathrm{HGL}(\Theta) = \sum_{t\in\mathcal{T}_{\mathrm{train}}} \sum_{x\in\mathcal{X}} \frac{\left(\widehat{S}_\Theta(\mathbf{u}_{t,x}, T_{t,x}) - S_{t,x}\right)^2}{2\widehat{V}_\Theta(\mathbf{u}_{t,x}, T_{t,x})} + \log(\widehat{V}_\Theta(\mathbf{u}_{t,x}, T_{t,x})). \tag{3.2}$$

$\widehat{V}_\Theta(\mathbf{u}_{t,x}, T_{t,x})$ is an estimate of the conditional variance of the subgrid forcing given the low-resolution data, which is also generated by a convolutional neural network. Both loss functions were minimized via batch-based stochastic gradient descent with respect to the network parameters. Additional details about the training procedure are reported in Supplementary Section A.

3.2. *Linear-Inversion Baseline Parameterization*

In this section we propose a procedure to estimate subgrid forcing based on partial inversion of the filtering and coarse-graining operations described in Section 2. Many subgrid parameterizations are based on an approximate inversion of the filtering operator, achieved for instance via popular inversion methods based on truncated Taylor series expansion and iterative deconvolution (Carati et al., 2001; Chow et al., 2005). The accuracy of these parameterizations depends on the number of iterations in the inversion procedure. Here we take a different approach, performing a complete (not iterative) inversion.

Let L represent a linear operator mapping a high-resolution variable defined on the fine grid to its filtered and coarse-grained counterpart. For example, the longitudinal momentum on the coarse grid equals

$$u = \left\langle \overline{u^\uparrow} \right\rangle_\kappa = Lu^\uparrow, \tag{3.3}$$

where $u^\uparrow$ is the high-resolution longitudinal momentum (following the notation of Section 2). The pseudoinverse $L^\dagger$ of the linear operator projects coarse-grid variables onto the fine grid,

$$c_{\mathrm{inv}} = L^\dagger c, \quad c = \{u, v, T\}, \tag{3.4}$$

$$\boldsymbol{u}_{\mathrm{inv}} = (u_{\mathrm{inv}}, v_{\mathrm{inv}}). \tag{3.5}$$

We define the linear-inversion parameterization as the estimate of the subgrid forcing obtained by plugging the fine-grid projection of the corresponding variable into equation (2.3):

$$\widehat{S}_{\mathrm{inv},c} = \left\langle \overline{\boldsymbol{u}_{\mathrm{inv}}} \cdot \nabla \overline{c_{\mathrm{inv}}} - \overline{\boldsymbol{u}_{\mathrm{inv}} \cdot \nabla c_{\mathrm{inv}}} \right\rangle_\kappa, \quad c = \{u, v, T\}. \tag{3.6}$$

Note that this parameterization completely ignores the high-frequency information suppressed by the coarse-graining operation. In addition, the filtering operation can only be inverted at frequencies where its Fourier transfer function (the set of eigenvalues that defines how the filter attenuates the Fourier harmonics (Grooms et al., 2021)) is nonzero (Stolz and Adams, 1999).

The pseudoinverse operator is computed according to the following formula: $L^\dagger = L^T(LL^T)^{-1}$, where the mapping L is represented by a sparse matrix and $(LL^T)^{-1}$ is approximated by a sparse matrix up to numerical precision, as explained in more detail in Supplementary Section B.

4. Experiments and Results

4.1. *Evaluation*

The parameterizations described in Section 3 were evaluated on the held-out test set, containing 10% of the CM2.6 data. Our main evaluation metric is the coefficient of determination R^2. Let $S_{t,x}$ denote the ground-truth subgrid forcing at time t and location x, and $\widehat{S}(\mathbf{u}_{t,x}, T_{t,x})$ the estimate computed from the low-resolution data $\mathbf{u}_{t,x}$ and $T_{t,x}$. The R^2 coefficient is defined as

$$R^2 = 1 - \frac{\sum_{t \in \mathcal{T}_{\text{test}}} \sum_{x \in \mathcal{X}} \left(\widehat{S}(\mathbf{u}_{t,x}, T_{t,x}) - S_{t,x} \right)^2}{\sum_{t \in \mathcal{T}_{\text{test}}} \sum_{x \in \mathcal{X}} S_{t,x}^2}, \tag{4.1}$$

where $\mathcal{T}_{\text{test}}$ contains the time indices corresponding to the test set and $\mathcal{X}$ determines the spatial extent over which we evaluate the model.

4.2. Geographic Extent of the Training Data

In order to study the effect of the geographic extent of the training data on CNN parameterizations, we train the same CNN models on two training sets with different geographic coverage:

- The **4-regions** models were trained on the four regions utilized in Guillaumin and Zanna (2021), which are depicted in Supplementary Figure C.
- The **global** models were trained on the full planet.

We trained versions of these models at different resolutions: 0.4°, 0.8°, 1.2°, and 1.6°, obtained by setting the data-coarsening factor κ equal to 4, 8, 16 and 32 respectively (see Section 2).

Figure 2 provides a visualization of the performance of both CNN models over the whole globe. The global CNN model trained on the whole planet consistently outperforms the 4-regions model for both momentum components and temperature. The difference in performance is especially pronounced at extreme northern and southern latitudes. The first row of Figure 4 reports the aggregated performance of both models for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16. The global model exhibits strong performance, surpassing the 4-regions model at all resolutions for both momentum and temperature. The difference in performance is particularly marked for temperature, where the 4-regions model performs poorly.

Figure 3 shows that the uncertainty-quantification capabilities of the CNN models are also improved by increasing the geographic extent of the training data. The figure shows the empirical distributions of the subgrid forcing corresponding to longitudinal momentum (top row) and temperature (bottom row) on the held-out test set, standardized using the conditional mean and conditional variance produced by CNNs trained using the heteroscedastic Gaussian loss (3.2). If these conditional estimates are accurate, the standardized distributions should have zero mean and unit variance. The global CNN (green) achieves this to a larger extent than the 4-regions CNN (red).

4.3. Comparison with Linear-Inversion Parameterization

The linear-inversion parameterization presented in Section 3.2 estimates the subgrid forcing via a partial linear *inversion* of the coarse-graining operator. It is important to note that this partial linear inversion is nontrivial, because the data-coarsening process is not translation invariant due to coastal boundaries.

Figure 2 shows the test performance of this baseline over the whole globe, and compares it to the CNN parameterizations. Close to the North Pole, its performance is superior, and it also outperforms the 4-regions CNN at extreme southern latitudes and some other regions, such as the South Pacific. The first row of Figure 4 compares the overall performance of the models at several resolutions. For momentum, the linear baseline is on par with the 4-regions CNN, but inferior to the global CNN. For temperature, it is on par with the global CNN, and superior to the 4-regions CNN. We conclude that partial linear inversion provides a strong baseline, and that the global CNN is able to learn high-resolution structure that is inaccessible via linear inversion.

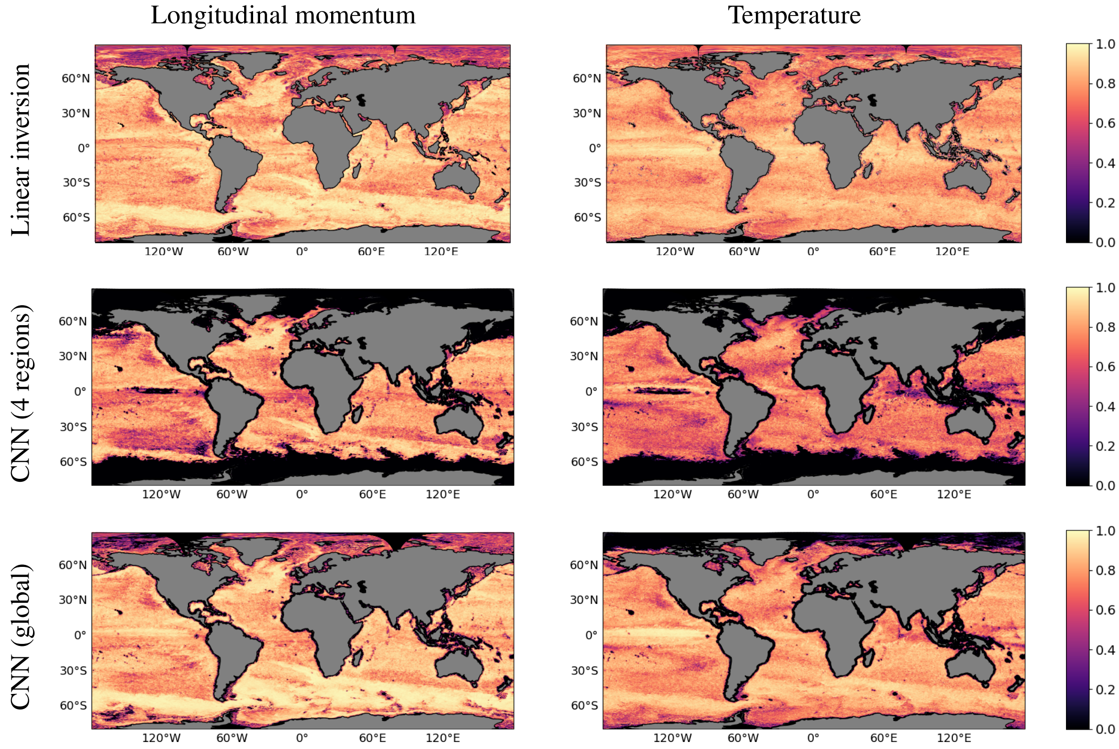

Figure 2. *Maps of the R^2 coefficient for longitudinal momentum (left column) and temperature (right column) parameterizations computed on a held-out test set. The rows correspond to the linear-inversion parameterization described in Section 3.2 (top) and the CNN parameterization described in Section 3.1 trained on 4 regions (center) and on the whole planet (bottom). The CNN models were trained by minimizing the MSE loss. The data were processed using General-Circulation-Model filtering and coarse-grained with a factor of 4. The global CNN model trained on the whole planet consistently outperforms the 4-regions model. It also outperforms the linear-inversion baseline, except in at latitudes close to the North Pole. The 4-regions model exhibits poor performance at extreme northern and southern latitudes.*

4.4. Generalization Across CO_2 Levels and Depth

The second row of Figure 4 reports the overall performance of CNN models trained at pre-industrial CO_2 levels, when tested on a forced simulation with a 1% increase in CO_2 level per year, until the level doubles after 70 years (Griffies, 2015). Overall, we observe that the models preserve similar performance levels, indicating that they are able to generalize robustly across different CO_2 levels. Generalization for momentum is better than for temperature. This may be partially explained by the fact that the CO_2 increase produces a greater distribution shift for temperature than for momentum, as shown in Supplementary Figure 12.

In addition, we investigated the performance of CNN-parameterizations at different depths, utilizing the subsurface training and test sets described in Section 3.1. We trained separate CNN models on the whole planet at six different depths: 5 m, 55 m, 110 m, 181 m, 330 m, and 728 m. We tested these models and the surface model at all depths (including the surface). Figure 5 shows the results. The surface model generalizes well to 5 m and vice versa, but both models do not generalize robustly to greater depths. Conversely, models trained at depths 55 m and beyond do not generalize robustly to the surface or to 5 m, but do generalize to the remaining depths. This indicates that the models learn very different structure at the surface and near-surface, with respect to greater depths. The linear-inversion

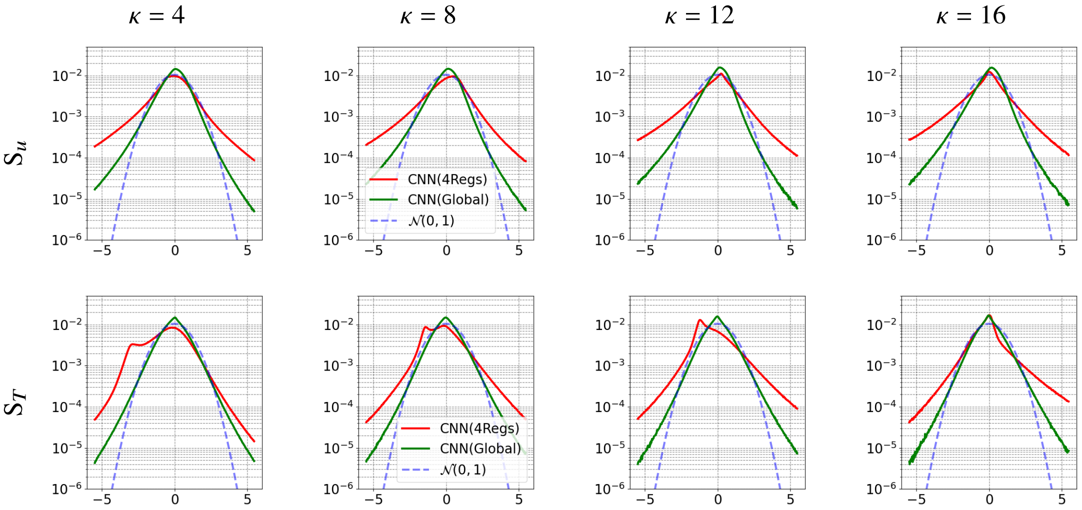

Figure 3. The graphs show the empirical distributions of the subgrid forcing corresponding to longitudinal momentum (top row) and temperature (bottom row) on the held-out test set, standardized using the conditional mean and conditional variance produced by CNNs trained using the heteroscedastic Gaussian loss (3.2)*. Results for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16 are shown. In all cases, the global CNN (green), trained on the entire globe, results in an empirical distribution that more closely resembles a Gaussian with unit variance (blue, dashed) than the 4-regions CNN (red), which indicates that the conditional-variance estimate is improved by increasing the geographic extent of the training data.*

parameterization, which is agnostic to physical structures, shows uniform performance at all depths. However, at most depths the linear-inversion model is less accurate than the CNN-parameterization trained at the same depths.

4.5. Input Stencil

A key consideration for the practical deployment of CNN-based parameterizations is their computational cost, in particular in the absence of GPUs (Zhang et al., 2023). This cost depends on the input stencil or field of view of the CNN, which is the spatial extent utilized by the model to produce an estimate at each location. To determine the influence of the input stencil on the performance of these models, we trained different models with the same number of parameters, but different input stencils. This was achieved by reducing the size of the convolutional filters in each layer, while simultaneously increasing the number of filters to preserve the overall number of parameters.

Figure 6 shows the test R^2 coefficient for the longitudinal momentum (left) and temperature (right) of CNN parameterizations with different input stencils. At all resolutions performance saturates when the stencil size reaches 7×7. This is a considerable reduction from the 21×21 input stencils of previous CNNs (Guillaumin and Zanna, 2021).

In order to test the hypothesis that CNN parameterizations primarily leverage information that closely surrounds the output location, we performed a gradient-based analysis. The gradient of neural-network outputs with respect to their input is widely used to interpret the functions learned by these models (Mohan et al., 2019; Ross et al., 2023; Simonyan et al., 2013). It quantifies the sensitivity of the network output to small changes in each input pixel. We measure the concentration of the gradient by computing the fraction of the sum of squared gradient amplitude that lies within a certain input stencil. Figure 7 shows the median concentration of the gradient energy of CNN parameterizations for

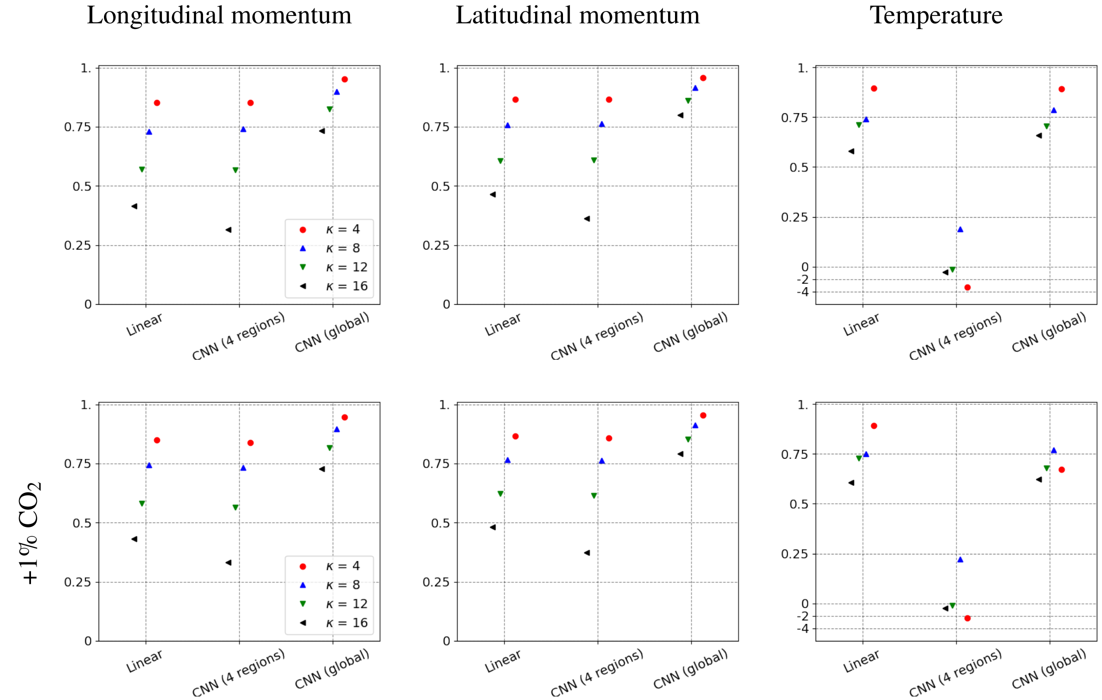

Figure 4. *The graphs in the first row show the test R^2 coefficient for longitudinal momentum (left), latitudinal momentum (center) and temperature (right) of the linear-inversion parameterization described in Section 3.2 and the CNN parameterization described in Section 3.1 trained on 4 regions and on the whole planet (global). Results for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16 are shown. The global model outperforms the 4-regions model and the linear-inversion baseline at all resolutions for both momentum components and temperature. The graphs in the second row show the same results for data generated at an increased CO$_2$ level. The CNN models were trained by minimizing the MSE loss and the data were processed using General-Circulation-Model filtering. Results for the heteroscedastic Gaussian loss and for Gaussian filtering are shown in Supplementary Figures 10 and 11, respectively.*

longitudinal momentum (left), latitudinal momentum (center) and temperature (right). At all resolutions, 90% or more gradient energy is concentrated within a stencil size of 7×7, which supports our hypothesis that the network primarily leverages a small input region to generate its output.

5. Discussion and Outlook

This study revealed several insights regarding data-driven parameterizations for mesoscale eddies based on convolutional neural networks, which may be useful for the design and implementation of these models in other situations. First, we observed that the geographic extent represented in the training data has a significant influence on model performance. Second, we compared CNN parameterizations with a linear baseline, based on partial inversion of the coarse-graining operation typically used to generate training data for data-driven parameterizations. The CNN parameterizations mostly outperform the baseline, which suggests that the CNNs are able to learn nonlinear physical structure. Third, we evaluated the generalization ability of CNN parameterizations at different CO$_2$ levels and ocean depths, finding that they are able to generalize to increased CO$_2$ levels, but not from surface to subsurface depths beyond 50 m (or vice versa). It is possible that surface atmospheric forcing has a strong influence in the upper ocean, which is not present below the surface, creating a distribution shift that leads to a

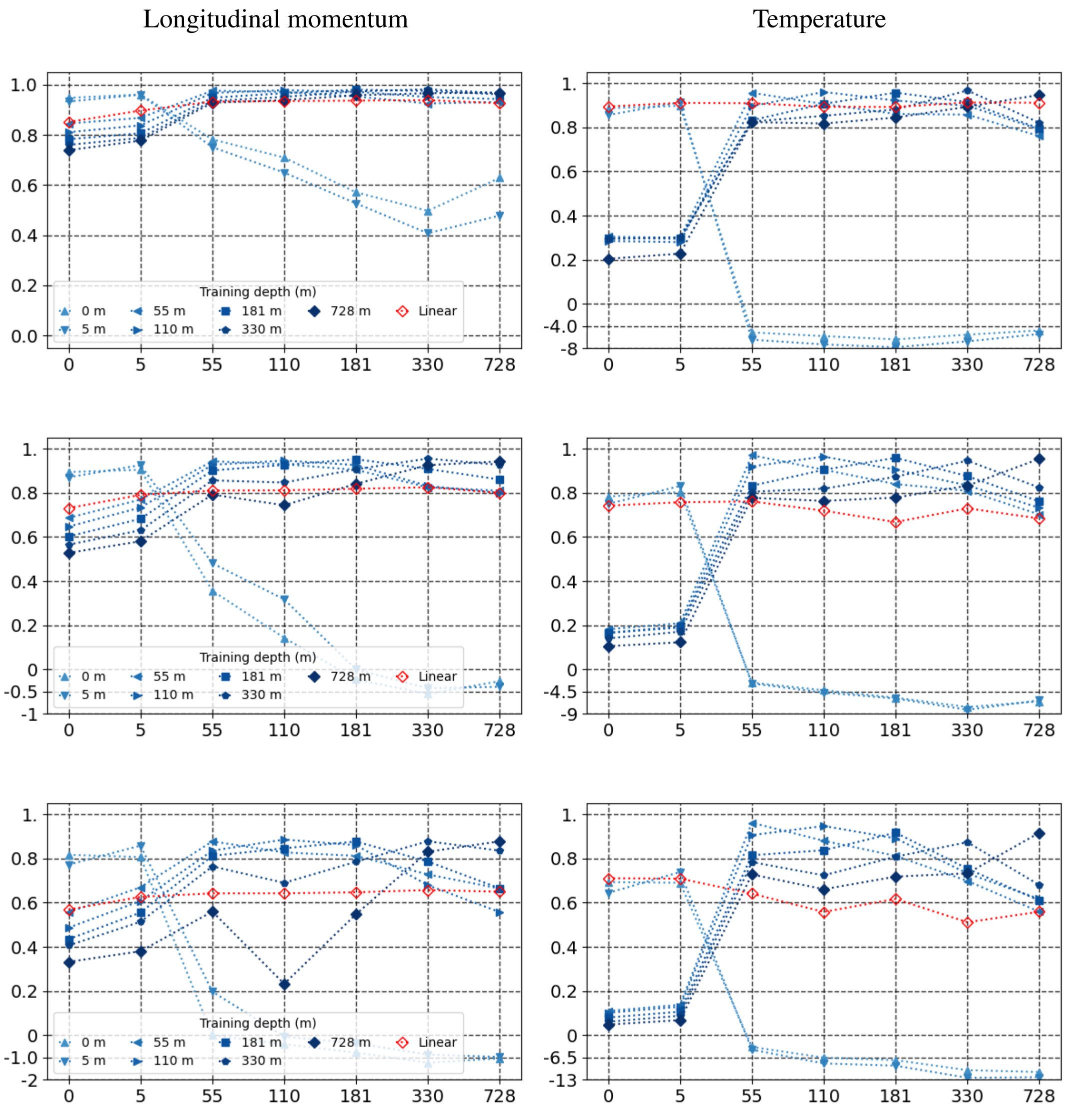

Figure 5. *The graphs show the test R^2 coefficient at different depths for the longitudinal momentum (left) and temperature (right) of the linear-inversion parameterization (in red) described in Section 3.2 and the CNN parameterization (in blue) described in Section 3.1 trained on the whole planet (global). The different markers indicate the depth at which the corresponding CNN model was trained. Models trained at shallow depth (surface and 5 m) do not generalize well to greater depth. Conversely, models trained at greater depth do not generalize well to the surface. The CNN models were trained by minimizing the MSE loss. The data were processed using General-Circulation-Model filtering and coarse-grained with a factors of 4 (top), 8 (center) and 12 (bottom).*

lack of generalization. Fourth, we determined that CNN parameterizations mainly exploit a relatively small surrounding region of their input to produce their outputs. These insights were consistent for both momentum and temperature parameterizations across different resolutions.

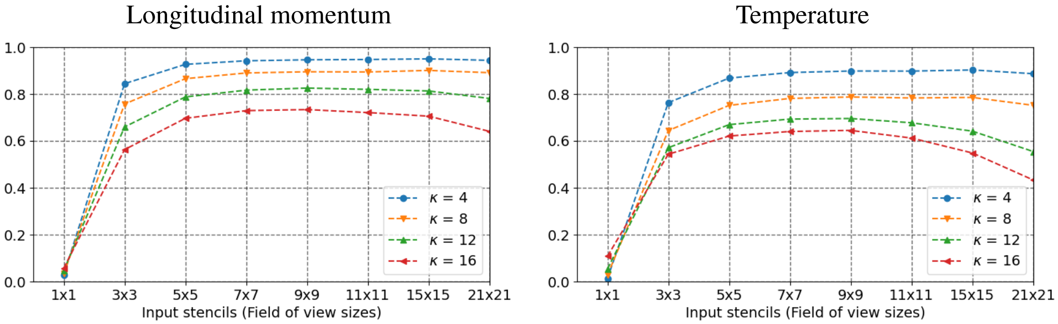

Figure 6. *The graphs plot the test R^2 coefficient for the longitudinal momentum (left) and temperature (right) of CNN parameterizations with different input stencils or fields of view (in pixels). Results for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16 are shown. The CNN models were trained by minimizing the MSE loss. The data were processed using General-Circulation-Model filtering. At all resolutions performance saturates when the stencil size reaches* 7×7.
.

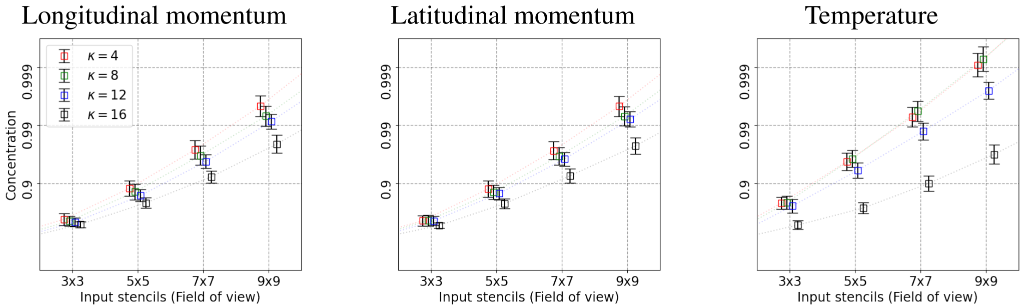

Figure 7. *The graphs plot the median concentration of the gradient energy of CNN parameterizations for longitudinal momentum (left), latitudinal momentum (center) and temperature (right), along with error bars indicating the 10% and 90% quantiles. The concentration is measured as the fraction of the sum of squared gradient amplitude within an input stencil or field of view of a certain size. Results for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16 are shown. The CNN models were trained by minimizing the MSE loss. The data were processed using General-Circulation-Model filtering. At all resolutions 90% or more gradient energy is concentrated within a stencil size of* 7×7.

Our study focused on offline metrics, which quantify to what extent parameterizations are able to approximate the missing subgrid forcing in coarse-grid climate simulations. A crucial direction for future research is to evaluate parameterizations in an online setting, once incorporated in realistic climate models (see (Perezhogin et al., 2024; Zhang et al., 2023) for recent progress in this direction).

Funding Statement. This project is supported by Schmidt Sciences, LLC. C.F.G was partially supported by NSF DMS Grant 2009752.

Competing Interests. The authors declare none.

Data Availability Statement. The authors downloaded the simulation's ocean surface velocities from a Pangeo data catalog at https://raw.githubusercontent.com/pangeo-data/pangeo-datastore/master/intake-catalogs/ocean/GFDL_CM2.6.yaml made publicly available by the Geophysical Fluid Laboratory.

Ethical Standards. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

Author Contributions. Analysis and investigation: C.G.; Conceptualization: A.A., C.F.G., and L.Z.; Data curation: C.G., P.P., C.Z., and A.S.; Data visualization: C.G., Methodology: C.G., M.L., C.F.G., and L.Z.; Supervision: A.A., C.F.G., and L.Z.; Writing original draft: C.G., C.F.G., P.P., and L.Z.; Writing—reviewing and editing: C.G., P.P., C.Z., A.A., C.F.G., and L.Z. All authors approved the final submitted draft.

References

Balaji, V. (2021). Climbing down charney's ladder: Machine learning and the post-dennard era of computational climate science. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200085.

Bardina, J., Ferziger, J., & Reynolds, W. (1980). Improved subgrid-scale models for large-eddy simulation. *13th fluid and plasmadynamics conference*, 1357.

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, *126*(9), 098302.

Bodner, A., Balwada, D., & Zanna, L. (2023). A data-driven approach for parameterizing submesoscale vertical buoyancy fluxes in the ocean mixed layer. *arXiv preprint arXiv:2312.06972*.

Bolton, T., & Zanna, L. (2019). Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, *11*(1), 376–399.

Bony, S., Stevens, B., Frierson, D. M., Jakob, C., Kageyama, M., Pincus, R., Shepherd, T. G., Sherwood, S. C., Siebesma, A. P., Sobel, A. H., et al. (2015). Clouds, circulation and climate sensitivity. *Nature Geoscience*, *8*(4), 261–268.

Börm, S., Grasedyck, L., & Hackbusch, W. (2003). Introduction to hierarchical matrices with applications. *Engineering analysis with boundary elements*, *27*(5), 405–422.

Capet, X., Mcwilliams, J. C., Molemaker, M. J., & Shchepetkin, A. F. (2008). Mesoscale to submesoscale transition in the california current system. part i: Flow structure, eddy flux, and observational tests. *Journal of physical oceanography*, *38*(1), 29–43.

Carati, D., Winckelmans, G. S., & Jeanmart, H. (2001). On the modelling of the subgrid-scale and filtered-scale stress tensors in large-eddy simulation. *Journal of Fluid Mechanics*, *441*, 119–138.

Chow, F. K., Street, R. L., Xue, M., & Ferziger, J. H. (2005). Explicit filtering and reconstruction turbulence modeling for large-eddy simulation of neutral boundary layer flow. *Journal of the atmospheric sciences*, *62*(7), 2058–2077.

Clark, R. A., Ferziger, J. H., & Reynolds, W. C. (1979). Evaluation of subgrid-scale models using an accurately simulated turbulent flow. *Journal of fluid mechanics*, *91*(1), 1–16.

Frezat, H., Balarac, G., Le Sommer, J., Fablet, R., & Lguensat, R. (2021). Physical invariance in neural networks for subgrid-scale scalar flux modeling. *Physical Review Fluids*, *6*(2), 024607.

Griffies, S. (2015). A handbook for the gfdl cm2-o model suite.

Grooms, I., Loose, N., Abernathey, R., Steinberg, J., Bachman, S. D., Marques, G., Guillaumin, A. P., & Yankovsky, E. (2021). Diffusion-based smoothers for spatial filtering of gridded geophysical data. *Journal of Advances in Modeling Earth Systems*, *13*(9), e2021MS002552.

Guan, Y., Chattopadhyay, A., Subel, A., & Hassanzadeh, P. (2022). Stable a posteriori les of 2d turbulence using convolutional neural networks: Backscattering analysis and generalization to higher re via transfer learning. *Journal of Computational Physics*, *458*, 111090. https://doi.org/https://doi.org/10.1016/j.jcp.2022.111090

Guillaumin, A. P., & Zanna, L. (2021). Stochastic-deep learning parameterization of ocean momentum forcing. *Journal of Advances in Modeling Earth Systems*, *13*(9), e2021MS002534.

Hallberg, R. (2013a). Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects. *Ocean Modelling*, *72*, 92–103.

Hallberg, R. (2013b). Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects. *Ocean Modelling*, *72*, 92–103. https://doi.org/https://doi.org/10.1016/j.ocemod.2013.08.007

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

Keating, S. R., Majda, A. J., & Smith, K. S. (2012). New methods for estimating ocean eddy heat transport using satellite altimetry. *Monthly Weather Review*, *140*(5), 1703–1722.

Langford, J. A., & Moser, R. D. (1999). Optimal les formulations for isotropic turbulence. *Journal of fluid mechanics*, *398*, 321–346.

Loose, N., Abernathey, R., Grooms, I., Busecke, J., Guillaumin, A., Yankovsky, E., Marques, G., Steinberg, J., Ross, A. S., Khatri, H., Bachman, S., Zanna, L., & Martin, P. (2022). Gcm-filters: A python package for diffusion-based spatial filtering of gridded data. *Journal of Open Source Software*, *7*(70), 3947. https://doi.org/10.21105/joss.03947

Meneveau, C., & Katz, J. (2000). Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, *32*(1), 1–32.

Mohan, S., Kadkhodaie, Z., Simoncelli, E. P., & Fernandez-Granda, C. (2019). Robust and interpretable blind image denoising via bias-free convolutional neural networks. *International Conference on Learning Representations*.

Perezhogin, P., Zanna, L., & Fernandez-Granda, C. (2023). Generative data-driven approaches for stochastic subgrid parameterizations in an idealized ocean model. *Journal of Advances in Modeling Earth Systems*, *15*(10), e2023MS003681.

Perezhogin, P., Zhang, C., Adcroft, A., Fernandez-Granda, C., & Zanna, L. (2024). A stable implementation of a data-driven scale-aware mesoscale parameterization. *Journal of Advances in Modeling Earth Systems*, *16*(10), e2023MS004104.

Randall, D., Khairoutdinov, M., Arakawa, A., & Grabowski, W. (2003). Breaking the cloud parameterization deadlock. *Bulletin of the American Meteorological Society*, *84*(11), 1547–1564.

Ross, A., Li, Z., Perezhogin, P., Fernandez-Granda, C., & Zanna, L. (2023). Benchmarking of machine learning ocean subgrid parameterizations in an idealized model. *Journal of Advances in Modeling Earth Systems*, *15*(1), e2022MS003258.

Salmon, R. (1980). Baroclinic instability and geostrophic turbulence. *Geophysical & Astrophysical Fluid Dynamics*, *15*(1), 167–211.

Sane, A., Reichl, B. G., Adcroft, A., & Zanna, L. (2023). Parameterizing vertical mixing coefficients in the ocean surface boundary layer using neural networks. *Journal of Advances in Modeling Earth Systems*, *15*(10), e2023MS003890. https://doi.org/https://doi.org/10.1029/2023MS003890

Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, K. G., Schär, C., & Siebesma, A. P. (2017). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*(1), 3–5.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Stolz, S., & Adams, N. A. (1999). An approximate deconvolution procedure for large-eddy simulation. *Physics of Fluids*, *11*(7), 1699–1701.

Waterman, S., & Jayne, S. R. (2011). Eddy-mean flow interactions in the along-stream development of a western boundary current jet: An idealized model study. *Journal of Physical Oceanography*, *41*(4), 682–707.

Yuval, J., & O'Gorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature communications*, *11*(1), 1–10.

Zanna, L., & Bolton, T. (2020). Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, *47*(17), e2020GL088376.

Zhang, C., Perezhogin, P., Gultekin, C., Adcroft, A., Fernandez-Granda, C., & Zanna, L. (2023). Implementation and evaluation of a machine learned mesoscale eddy parameterization into a numerical ocean circulation model. *Journal of Advances in Modeling Earth Systems*, *15*(10), e2023MS003697.

	layer-1	layer-2	layer-3	layer-4	layer-5	layer-6	layer-7	layer-8
$\kappa = 4$								
width	(2,3)x128	128x64	64x 32	32x 32	32x32	32x32	32x32	32x(4,6)
kernel size	5x5	5x5	3x3	3x3	3x3	3x3	3x3	3x3
$\kappa = 8$								
width	(2,3)x208	208x104	104x52	52x52	52x52	52x52	52x52	52x(4,6)
kernel size	3x3	3x3	2x2	2x2	2x2	2x2	2x2	2x2
$\kappa =$ 12,16								
width	(2,3)x279	279x140	140x70	70x70	70x70	70x70	70x70	70x(4,6)
kernel size	2x2	2x2	2x2	2x2	2x2	2x2	1x1	1x1

Table 1. *Structure of the convolutional layers of the CNN architectures used for each coarse-graining factor κ.*

A. Neural-Network Implementation Details

Following (Guillaumin and Zanna, 2021) the convolutional neural networks used in this study consist of eight convolutional layers interleaved with ReLU nonlinearities. Table 1 provides a detailed description of the convolutional layers used for the different coarse-graining factors. We incorporate a batch-normalization layer (Ioffe and Szegedy, 2015) after each intermediate layer. For the models trained with the heteroscedastic Gaussian, the output layer has multiple channels, associated with the conditional mean and conditional variance estimate. The latter estimate is generated in the form of the inverse conditional standard deviation, which is constrained to be nonnegative by a softplus activation function, as in (Guillaumin and Zanna, 2021).

During training, we use a variable learning rate scheduler called `ReduceLROnPlateau`, available in `pytorch`. Starting from an initial learning rate (equal to 0.01), the scheduler checks if the validation error has decreased at the end of each epoch. If this is not the case, over a certain number of epochs (equal to 3), the learning rate is decreased. The process continues until the learning rate reaches below a certain threshold (equal to 10^{-7}). The training mini-batches contain four frames for a coarse-graining factor of 4, as in (Guillaumin and Zanna, 2021). At larger factors, the minibatch sizes are increased to preserve the number of grid points.

In Section 4.5 we report results for different CNNs with the same number of parameters, but different input stencils. This was achieved by reducing the size of the convolutional filters in each layer, while simultaneously increasing the number of filters to preserve the overall number of parameters. This was achieved by first shrinking the kernel sizes to achieve a certain stencil size, and then increasing the number of channels.

B. Linear Baseline Details

Section 3.2 proposes a procedure to estimate subgrid forcing based on partial inversion of the filtering and coarse-graining operations described in Section 2. The procedure involves computing the pseudoinverse of the linear operator L that maps the high-resolution variables defined on the fine grid to their filtered and coarse-grained counterparts.

We first reshape two-dimensional arrays of velocities or temperature into one-dimensional vectors and exclude the land points. This allows to represent operator L with a sparse matrix. The pseudoinverse is given by the formula $L^{\dagger} = L^T(LL^T)^{-1}$, where the inverse matrix $((LL^T)^{-1})$ is computed via hierarchical matrix factorization following Börm et al. (2003). One factorization step divides any matrix M into 4

	lyr-1	lyr-2	lyr-3	lyr-4	lyr-5	lyr-6	lyr-7	lyr-8	stencil
width	128	64	32	32	32	32	32	32	
ker. size	5	5	3	3	3	3	3	3	21x21
width	186	93	46	46	46	46	46	46	
ker. size	3	3	3	3	3	3	2	2	15x15
width	208	104	52	52	52	52	52	52	
ker. size	3	3	2	2	2	2	2	2	11x11
width	271	136	68	68	68	68	68	68	
ker. size	2	2	2	2	2	2	2	2	9x9
width	279	140	70	70	70	70	70	70	
ker. size	2	2	2	2	2	2	1	1	7x7
width	296	148	74	74	74	74	74	74	
ker. size	2	2	2	2	1	1	1	1	5x5
width	327	164	82	82	82	82	82	82	
ker. size	2	2	1	1	1	1	1	1	3x3
width	528	264	132	132	132	132	132	132	
ker. size	1	1	1	1	1	1	1	1	1x1

Table 2. *Across the study, we use adapt the original 8 layer CNN architecture to various stencil sizes. The stencil size refers to the total input window size of the CNNs. Each row corresponds to a CNN model. Each convolutional layer has a number of output channels (width) and filter size (ker. size). All kernels are square shaped and the total input sizes (stencil) are provided in the last column..*

blocks as follows:

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \tag{B.1}$$

where each block is half the size of the original matrix. The block matrix can be formally inverted as follows:

$$M^{-1} = \begin{pmatrix} (M/D)^{-1} & -(M/D)^{-1}BD^{-1} \\ -D^{-1}C(M/D)^{-1} & D^{-1} + D^{-1}C(M/D)^{-1}BD^{-1} \end{pmatrix}, \tag{B.2}$$

where $M/D = A - BD^{-1}C$ is the Schur complement. The procedure is applied recursively to the matrix LL^T until the individual matrices can be directly inverted. An essential ingredient of our approach is the approximation of the inverse matrix ($(LL^T)^{-1}$) by a sparse matrix. Sparsity of the inverse matrix is achieved by neglecting matrix elements that are below the numerical precision during each factorization step.

C. Additional Figures

Figure C shows the geographic locations used to train the 4-regions CNN model, following (Guillaumin and Zanna, 2021). Figure 9 shows maps of the test R^2 coefficient for temperature of the CNN parameterization trained on the whole globe by minimizing the heteroscedastic Gaussian loss (left) and MSE (right). Figures 10 and 11 show analogous results to Figure 4, but for the heteroscedastic Gaussian loss and for Gaussian filtering, respectively. Figure 12 shows the distribution shift in the subgrid forcing corresponding to momentum and temperature, between the control simulation and the simulation with increased CO_2 levels.

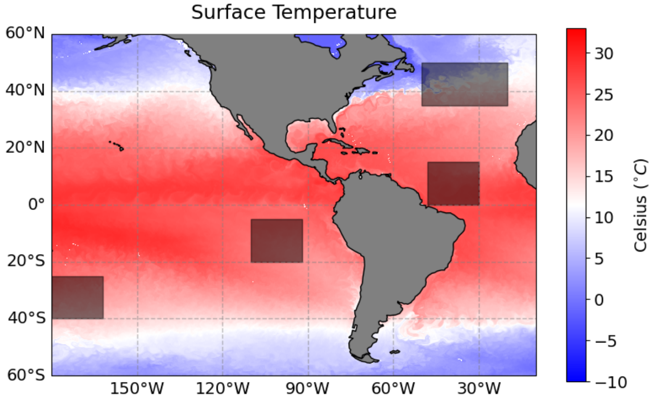

Figure 8. *The shaded regions indicate the geographic locations used to train the 4-regions CNN model following (Guillaumin and Zanna, 2021).*

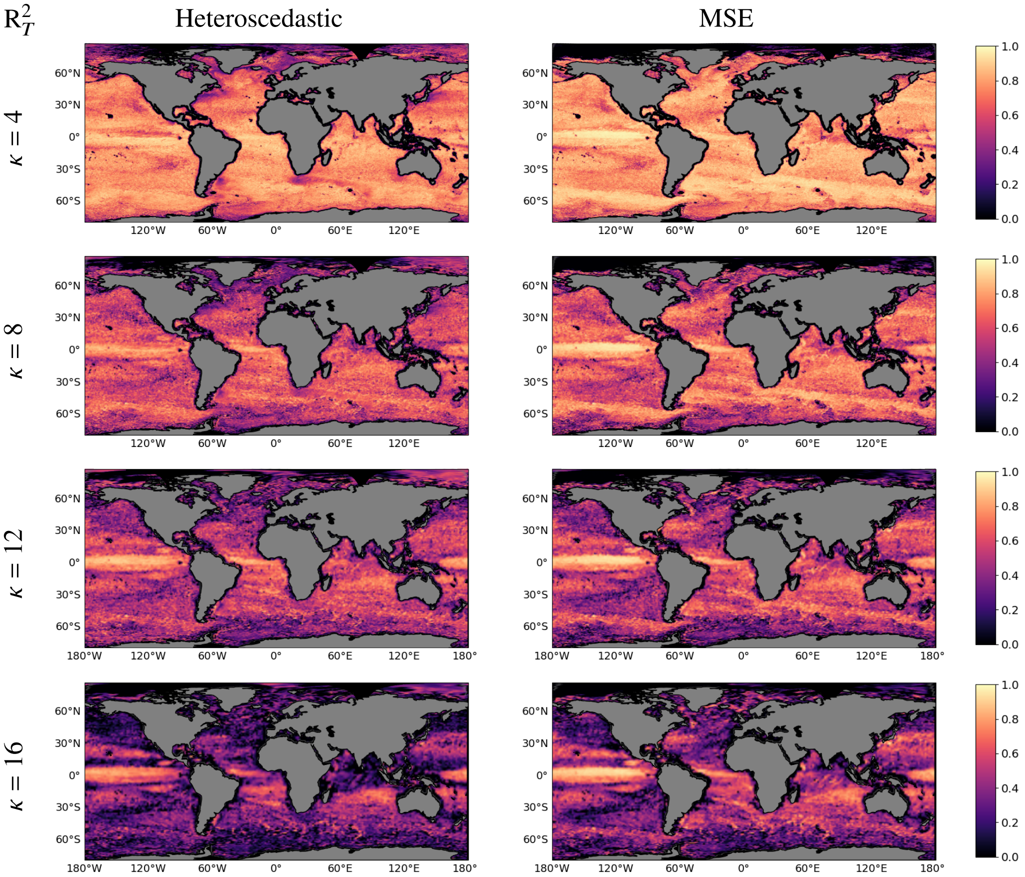

Figure 9. *Maps of the test R^2 coefficient for temperature of the CNN parameterization trained on the whole globe by minimizing the heteroscedastic Gaussian loss (left) and MSE (right). The data were processed using General-Circulation-Model filtering at coarse-grained at different factors, indicated by the value of κ in each row.*

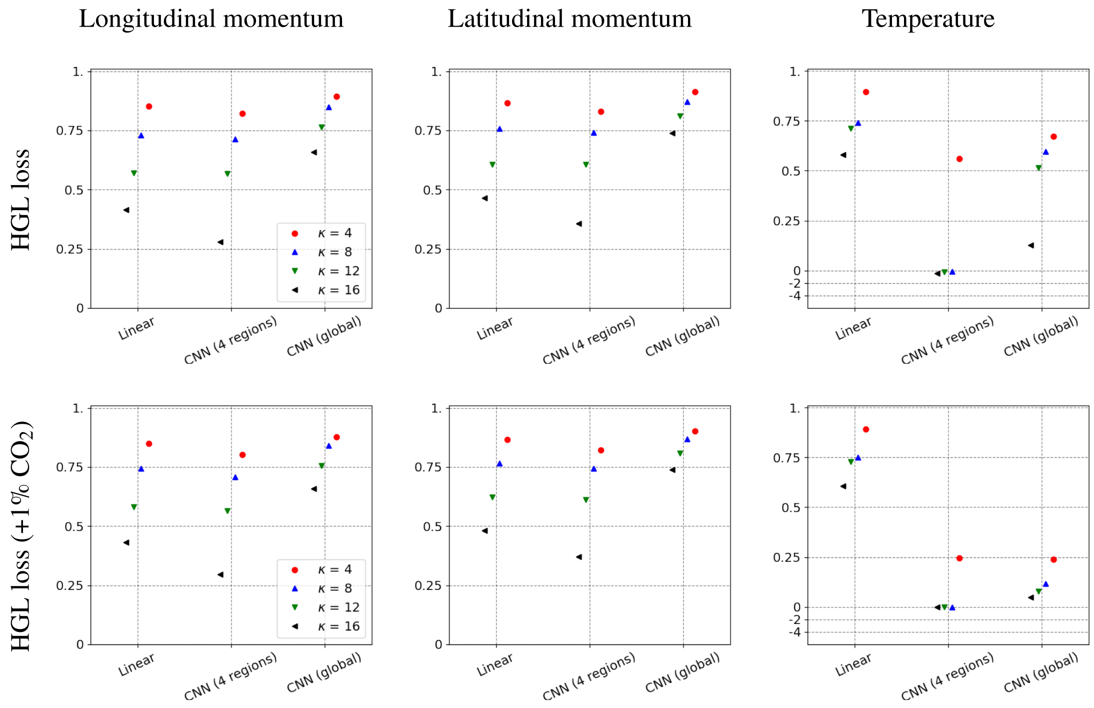

Figure 10. *The graphs in the first row show the test R^2 coefficient for longitudinal momentum (left), latitudinal momentum (center) and temperature (right) of the linear-inversion parameterization described in Section 3.2 and the CNN parameterization described in Section 3.1 trained on 4 regions and on the whole planet (global). Results for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16 are shown. The graphs in the second row show the same results for data generated at an increased CO$_2$ level. The CNN models were trained by minimizing the heteroscedastic Gaussian loss and the data were processed using General-Circulation-Model filtering.*

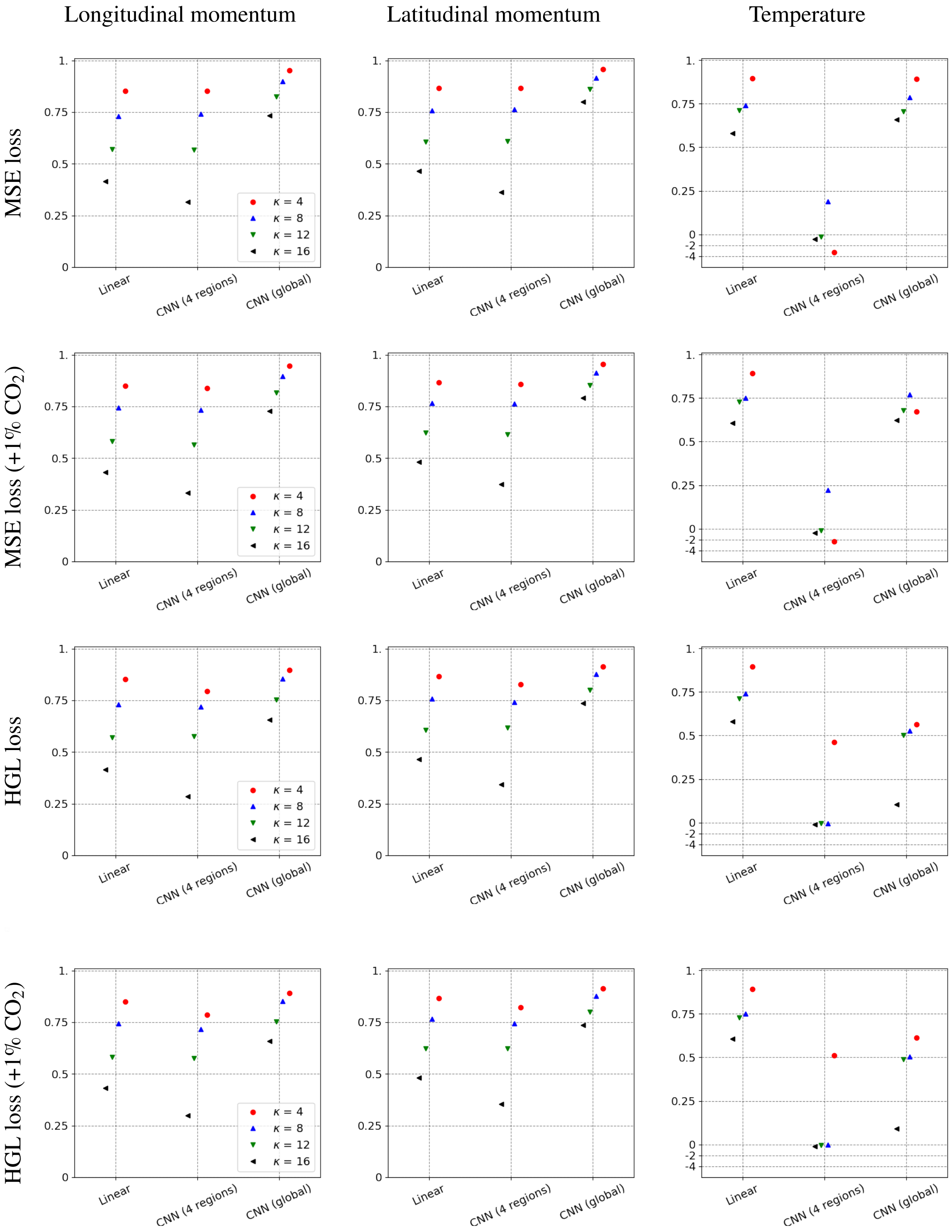

Figure 11. *Test R^2 coefficient for longitudinal momentum (left), latitudinal momentum (center) and temperature (right) of the linear-inversion parameterization described in Section 3.2 and the CNN parameterization described in Section 3.1 trained on 4 regions and on the whole planet (global). Results for four different resolutions, corresponding to coarse-graining factors of 4, 8, 12 and 16 are shown. The CNN models were trained by minimizing the MSE loss (rows 1 and 2) and the Gaussian heteroscedastic loss (rows 3 and 4). Row 2 and 4 show results for the test set with increased CO$_2$ level. The data were processed using Gaussian filtering.*

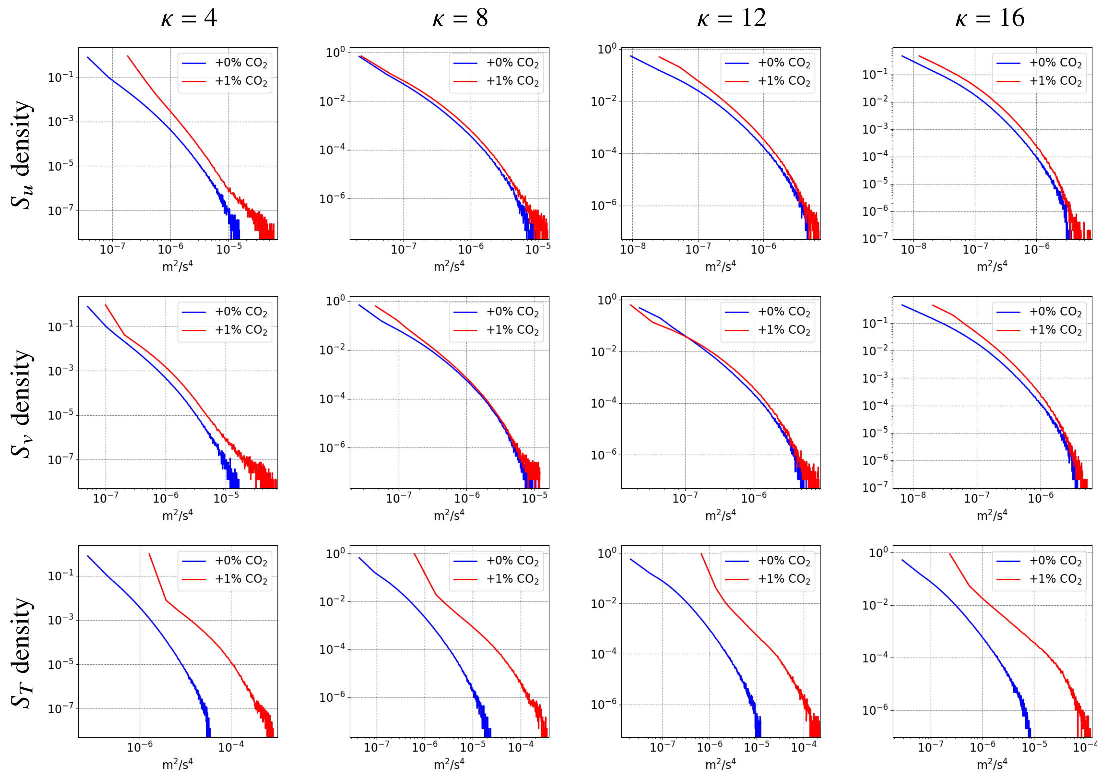

Figure 12. *Distribution of the subgrid forcing for longitudinal momentum (top), latitudinal momentum (center) and temperature (bottom) for different coarse-graining factors (indicated by the value of κ above each column), computed from data at pre-industrial CO$_2$ levels (blue) and at significantly increased CO$_2$ levels (red). The data were processed using General-Circulation-Model filtering. We observe a clear distribution shift for the temperature data.*