Fourier analysis of the physics of transfer learning for data-driven subgrid-scale models of ocean turbulence

Moein Darman[1], Pedram Hassanzadeh[2], Laure Zanna[3], and Ashesh Chattopadhyay [1*]

[1]Department of Applied Mathematics, University of California, Santa Cruz, Santa Cruz

[2]Department of Geophysical Sciences, University of Chicago, Chicago

[3]Department of Atmospheric and Oceanic Sciences and Mathematics, New York University, New York

April 23, 2025

Abstract

Transfer learning (TL) is a powerful tool for enhancing the performance of neural networks (NNs) in applications such as weather and climate prediction and turbulence modeling. TL enables models to generalize to out-of-distribution data with minimal training data from the new system. In this study, we employ a 9-layer convolutional NN to predict the subgrid forcing in a two-layer ocean quasi-geostrophic system and examine which metrics best describe its performance and generalizability to unseen dynamical regimes. Fourier analysis of the NN kernels reveals that they learn low-pass, Gabor, and high-pass filters, regardless of whether the training data are isotropic or anisotropic. By analyzing the activation spectra, we identify why NNs fail to generalize without TL and how TL can overcome these limitations: the learned weights and biases from one dataset underestimate the out-of-distribution sample spectra as they pass through the network, leading to an underestimation of output spectra. By re-training only one layer with data from the target system, this underestimation is corrected, enabling the NN to produce predictions that match the target spectra. These findings are broadly applicable to data-driven parameterization of dynamical systems.

1 Introduction

Improving the accuracy of climate, weather, and ocean models requires enhancing their resolution. This necessitates more computational power, which is restricted by limitations. Practical models that run on low spatio-temporal resolution with high predictive accuracy require an accurate representation of subgrid-scale (SGS) processes that occur below the numerical model's grid size. Although these processes occur on such a small spatial and temporal scale that resolving them inside a model is computationally intractable, their effects on the large-scale dynamics is significant. SGS parameterization approximates these processes in models, allowing for more accurate simulations at lower computational costs. The accurate parameterization of SGS processes is essential for improving the accuracy of predictions of these nonlinear, multi-scale, and high-dimensional systems. Consequently, modeling SGS processes has been an active area of research for the past few decades [Bracco et al., 2024, 2025, Dipankar et al., 2015, Meneveau and Katz, 2000, Pressel et al., 2017, Sagaut et al., 2013, Sarlak et al., 2015, Schneider et al., 2017].

In the early stages of global climate modeling, Smagorinsky [1963] introduced a physics-based SGS model, aiming to parameterize the effects of SGS eddies through a scale-selective dissipative approach, marked by positive eddy viscosity and second-order diffusion. This model and its variants since then, have found applications across a wide array of fields, including weather and climate simulation, combustion, and magnetohydrodynamics, amongst others [Arakawa and Lamb, 1977, Fox, 2012, Fox-Kemper

---

and Menemenlis, 2008, Knaepen and Moin, 2004, Piomelli, 1999,?, Sagaut, 2005, Stevens et al., 2018, Tan et al., 2017]. Despite their utility in ensuring numerical stability for Large Eddy Simulations (LES), these purely diffusive models often fail to accurately represent inter-scale physical processes like energy transfers, notably omitting backscattering [Pope, 2000]. Backscattering – the transfer of energy from subgrid to resolved scales – is critical in various problems involving turbulent fluid flow and has prompted extensive research to incorporate it into physics-based SGS models [Carati et al., 1995, Domaradzki et al., 1987, Guan et al., 2024, Jansen and Held, 2014, Jansen et al., 2019, Kerr et al., 1996, Khani and Waite, 2016, Leslie and Quarini, 1979, Mason and Thomson, 1992,?, Shinde, 2020, Thuburn et al., 2013, Zhou, 1991]. Efforts have been made to refine these models for more accurate energy transfer depiction, such as the dynamic approach by Germano et al. [1991], Grooms [2023b] that allows for negative eddy viscosity to account for backscattering [Grooms, 2023a]. However, these advancements often come with trade-offs in numerical stability, highlighting the ongoing challenge of developing SGS models that precisely capture both forward and backscatter energy transfers. This gap underscores the limitations of physics-based models in SGS parameterization, paving the way for exploring data-driven approaches in subsequent research efforts.

The mathematical relationship between the large-scale dynamics of turbulent flows and the corresponding subgrid-scale forcing is generally nonlinear and poses a significant challenge for researchers. Therefore, neural networks (NNs), as universal approximators [Hornik et al., 1989], are attractive tools to establish such mapping and unveil more hidden knowledge from data, potentially providing better SGS models and even new insights into SGS physics. Nevertheless, some issues exist with relying solely on data-driven models to find the mapping between large-scale and small-scale processes. These models require a large training dataset that contains accurate SGS terms obtained from high-fidelity sources such as high-resolution observations or simulations. However, these sources are usually scarce, and the models only perform well on the dataset on which they were trained. They cannot adapt easily to new datasets, which reduces their effectiveness in predicting dynamics beyond their training dataset. On the other hand, the ability to generalize to newer dynamical regimes is crucial in both SGS modeling and the broader area of machine learning for the physical sciences. For example, NN-based SGS models must perform reliably across diverse climatic conditions to be effectively utilized in projections of global warming [Larraondo et al., 2019, Rasp et al., 2018] and its impact on extreme weather events, events that can only be estimated from the tails of the probability density function (PDF) of the variables. Additionally, NN-based SGS models are less interpretable, making it difficult to interpret the mappings they create, thereby challenging the trustworthiness and reliability of their predictions.

Given the fact that extrapolation to different climate conditions is an out-of-distribution generalization problem and is challenging for NNs [Krueger et al., 2020], transfer learning (TL) is a flexible and robust framework that enables this [Yosinski et al., 2014, Zhuang et al., 2021] and can help effective blending of disparate training sets. It involves building a new NN called *Transfer Learned* NN (TLNN) from *Base* NN (BNN), which can achieve a similar level of accuracy for a target system that may have different statistical and dynamical properties compared to the base system. This is accomplished by re-training a few layers from the BNN using a small amount of data (usually orders of magnitude less data than what was used to train the BNN) from the target system. The process can produce a TLNN with comparable out-of-sample accuracy for the target system as the BNN, despite using only a small amount of re-training data.

Numerous studies have been conducted on applying TL to improve NN generalizability for problems involving partial differential equations (PDEs). These investigations predominantly center around Physics-Informed Neural Networks (PINNs) [Chen et al., 2021, Desai et al., 2021, Gao et al., 2022, Goswami et al., 2020, Guo et al., 2022, Haghighat et al., 2021, Hanna et al., 2022, Li et al., 2021, Mattheakis et al., 2021, Xu et al., 2023], where models are fine-tuned or adapted via a physics-based loss function that is specific to the target PDE/ODE system. Subramanian et al. [2023] explores the efficacy of pre-trained machine learning models through TL across a wide range of physics problems. It reveals that fine-tuning pre-trained models reduces the need for extensive downstream training datasets, achieving desired accuracy levels even for out-of-distribution tasks. Desai et al. [2021] developed a framework utilizing TL with PINNs for one-shot inference across ODE and PDEs, demonstrating instant, highly accurate solutions for equations like first- and second-order linear ordinary equations, the Poisson equation and the time-dependent Schrödinger equation, without requiring comprehensive re-training of the network. Additionally, the application of TL to neural operators, defined

as operators that learn mappings between two functional spaces from a finite set of input-output pairs (representing coefficients, initial, or boundary conditions as inputs and the PDE solution function as outputs), has garnered significant interest among researchers [Goswami et al., 2022, Li et al., 2021, Xu et al., 2022]. Goswami et al. [2022] presents a TL framework utilizing a deep operator network (DeepONet) to solve nonlinear PDEs in complex geometry other changing dynamics. This framework efficiently addresses task heterogeneity and conditional shifts by fine-tuning particular layers, showcasing rapid learning capabilities despite significant variations between source and target domains. Xu et al. [2022] enhances the stability and long-time prediction accuracy of DeepONet for PDEs by utilizing TL. It involves sequentially updating DeepONets with minimal re-training to track evolution equations over different time frames, demonstrating improved accuracy and reduced training data requirements. The necessity of TL in PINNs underscores the vital importance of comprehending TL mechanisms. This understanding is crucial in enhancing the adaptation and efficacy of models across diverse physical systems, ensuring their broad applicability and performance.

In the context of using TL for improving parameterization generalizability, Chattopadhyay et al. [2020] also showed that TL can improve the generalization skill of data-driven parameterization when they move from one Lorenz 96 system to a more chaotic one. Subel et al. [2021] have shown that TL enables accurate/stable generalization to a flow with 10x higher Reynolds number (Re) for forced Burgers turbulence. Guan et al. [2022a] demonstrated that TL, through re-training the *Convolutional NN* (CNN) using a minimal subset of data from the new flow, achieves accurate and stable LES-CNN predictions for flows at $16\times$ higher Re, and supports higher spatio-temporal resolutions when necessary for achieving stability. Sun et al. [2023] has used NN-based emulators of the Whole Atmosphere Community Climate Model's (WACCM) physics-based gravity wave (GW) parameterizations as a test case. They showed that the accuracy of this NN-based parameterization of GW reduces for a warmer climate ($4\times CO2$). However, it is significantly improved by applying TL, using ≈ 1 % data from the warmer climate.

In order to develop parameterization models that are easier to interpret physically, studies aim to utilize ML methods to discover physical equations for the parameterization. This method utilizes a library of pre-defined physical terms and estimate coefficients such that the parameterization can be expressed as a combination of these coefficients and library terms. Zanna and Bolton [2020] built a fully data-driven, interpretable model employing relevance vector machines (RVM). They used a library of second-order velocity derivatives and their nonlinear combinations to develop a closed-form model for SGS momentum and buoyancy fluxes. While the model showed promising results in *a priori* (offline) evaluations, it revealed instability in *a posteriori* (online) tests when coupled with a low-resolution ocean solver. Jakhar et al. [2023] built on the work by Zanna and Bolton [2020] and used 2D forced homogeneous isotropic turbulence (2D-FHIT) and Rayleigh-Bénard convection (RBC) test cases to extend the analysis and showed that optimzing on regular mean squared error (MSE) using such pre-defined library terms would provably converge to the 2nd order term in the Taylor series of expansion of the filtering kernel.

Generally, most of these studies found that, NN-based parameterizations are more accurate offline and if stabilized in online mode, can lead to a more accurate coupled model. However, an important question remains: *what exactly is learned when a neural network is trained on a dataset without predefined assumptions?* It is crucial to identify the specific physical properties of the system that contribute to the model's effectiveness. Recent advancements have been made in understanding what physics learned through the training of CNNs. Subel et al. [2023] pioneered the analysis of physics learned from data when applying TL to CNNs for SGS modeling of 2D isotropic turbulence. They utilized a spectral analysis of kernel weights to elucidate how TL adapts to learn new filters, aligning with the spectral differences between base and target systems. Pahlavan et al. [2024a] aimed to establish a link between the kernels of NNs and the local and non-local dynamics involved in gravity wave propagation and dissipation, employing Fourier analysis of the CNN's kernels for this purpose.

Despite ongoing efforts to analyze the kernels, pinpointing the exact reasons behind the success of CNN parameterizations remains challenging. It is understood that training a CNN on a given dataset results in the adaptation of convolutional filters to that specific set of data. However, the underlying reasons why particular filters or combinations thereof are effective in parameterization remain unclear—understanding these mechanisms could reduce training costs and lessen dependence on large volumes of high-fidelity data by enabling more physics-informed initialization of model weights. This paper aims to explore the necessity of TL and its impact on the distribution of kernel weights

of CNNs used to parameterize the small-scale dynamics of an anisotropic canonical ocean model. The process of deciding *what-* and *how-to-learn* does not always align with the intended learning outcomes of the model. Therefore, a detailed examination of the kernels offers a pathway to better understand the relationship between the kernels and the physical properties of the system, shedding light on what is actually being learned as opposed to what was intended to be learned. Investigating TL and striving to comprehend its underlying mechanisms offers the dual advantage of fostering a model's generalizability while leveraging this insight to achieve efficient training with fewer samples and enhanced model explainability.

Expanding on the groundwork established by Subel et al. [2023], we incorporate the same analysis for SGS modeling in LES of oceanic flows. Our study broadens the scope to include highly anisotropic and isotropic flow conditions. This approach allows us to explore how CNNs' kernels adapt and learn across varied physical systems, thereby understanding their impact on system performance across various conditions. Our contributions are as follows:

1. We evaluate the CNN-based parameterizations, highlighting offline and online metrics that best assess their out-of-distribution generalization performance.

2. We explain why NNs fail to generalize by linking the spectral characteristics of different flows at different dynamical regimes to the Fourier spectra of activations.

3. We demonstrate that the learnt kernels act as Gabor, low-pass, and high-pass filters and show that their distribution adjusts based on the training data.

This paper is organized as follows. In Section 2, we introduce the methodology, including the governing equations of test cases (idealized two-layer quasi-geostrophic model), numerical solver setup, filtering and coarse-graining procedure for data, and CNN and TL employed. Section 3 presents the results. Discussion and summary are in Section 4.

2   Methods and Data

2.1   Framework Overview

The simulations in this study are facilitated by `pyqg` [Abernathey et al., 2022], a Python library designed for modeling the dynamical behavior of quasi-geostrophic (QG) systems using pseudo-spectral methods. QG systems serve as a suitable approximation for the complex equations governing motion in more realistic ocean models, especially in the limit of high stratification and rotation. They adeptly represent the formation and evolution of ocean mesoscale eddies. Additionally, QG systems offer better computational efficiency compared to ocean models or GCMs, making them vital for the broad scope of this study. We followed the same approach as [Ross et al., 2023]. The methodology detailed in 2.2 repeats their explanations. In section 2.3, the CNN setup is explained along with the method we use to analyze CNN in spectral space in section 2.4.

2.2   Data Collection and Preprocessing

2.2.1   Background: Idealized Two-Layer Quasi-Geostrophic Model

We utilize a two-layer quasi-geostrophic model provided by `pyqg`. The prognostic variable of this model is potential vorticity (PV), indicated as q_1 for the upper layer and q_2 for the lower layer:

$$q_m = \nabla^2 \psi_m + (-1)^m \frac{f_0^2}{g' H_m} \Delta \psi \quad \text{where} \quad m \in \{1, 2\}, \tag{1}$$

where ψ_m represents the streamfunction corresponding to the depth H_m, $\Delta \psi = (\psi_1 - \psi_2)$, and $\nabla = \langle \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \rangle$ denotes the horizontal gradient operator. The zonal and meridional velocities for each layer ($m \in \{1, 2\}$) can be derived from the streamfunction as $u_m = -\frac{\partial \psi_m}{\partial y}$ and $v_m = \frac{\partial \psi_m}{\partial x}$, respectively. The horizontal velocity vector is expressed as $\mathbf{u_m} = (u_m, v_m)$. We employ the beta-plane approximation, where the Coriolis acceleration f varies linearly with latitude (y), described by $f = f_0 + \beta y$. Here, g' represents the reduced gravity.

The prognostic equations, solved in the spectral space, are:

$$\frac{\partial \hat{q}_m}{\partial t} = -\hat{J}(\psi_m, q_m) - ik\beta_m \hat{\psi}_m - ikU\hat{q}_m + \delta_{m,2} r_{ek} \kappa^2 \hat{\psi}^2 + \hat{ssd}, \tag{2}$$

4

where $\frac{\partial}{\partial t}$ represents the Eulerian time derivative, $(\hat{\cdot})$ denotes the Fourier transform, and $\kappa = \sqrt{k^2 + l^2}$ is the radial wavenumber, with k and l as the zonal and meridional wavenumbers, respectively. The horizontal Jacobian is defined as $J(A, B) = A_x B_y - A_y B_x$. The mean PV gradient in each layer is given by $\beta_m = \beta + (-1)^{m+1} \frac{f_0^2}{g' H_m} \Delta U$, where $\Delta U = U_1 - U_2$ indicates a fixed mean zonal velocity shear between the two fluid layers. U_1 and U_2 are the mean zonal velocities at the upper and lower levels, respectively. The Dirac delta function $\delta_{m,2}$ signifies that the bottom drag, with coefficient r_{ek}, is applied solely to the second (bottom) layer. $\hat{q}$ and $\hat{\psi}$ are related to each other via

$$(\mathbf{M} - \kappa^2 \mathbf{I}) \cdot \begin{bmatrix} \hat{\psi}_1 \\ \hat{\psi}_2 \end{bmatrix} = \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \end{bmatrix}, \quad \text{where} \quad \mathbf{M} = \begin{bmatrix} -\frac{f_0^2}{g' H_1} & \frac{f_0^2}{g' H_1} \\ \frac{f_0^2}{g' H_2} & -\frac{f_0^2}{g' H_2} \end{bmatrix}, \tag{3}$$

where either q or ψ can independently identify the state of the system. We will define ssd below.

2.2.2 Numerical Solver Setup

The model is solved pseudospectrally [Fox and Orszag, 1973] by transforming the velocity field and PV to real space, calculating the Jacobian using real-space PV fluxes, and then transforming back to spectral space. The scale-selective dissipation (ssd), included as an additive term in Equation 2, is a highly scale selective operator that attenuates the largest $1/3$ of the spatial wavenumbers of all terms on the right-hand side of Equation 2. Specifically, the operator is an exponential filter, $F_c(\kappa)$, defined as:

$$F_c(\kappa^*) = \begin{cases} 1, & \kappa^* < \kappa_c \\ e^{-23.6(\kappa^* - \kappa_c)^4}, & \kappa^* \geq \kappa_c \end{cases} \tag{4}$$

where κ^* is the non-dimensional radial wavenumber and $\kappa_c = 0.65\pi$, is the cut-off wavenumber. After each time step, $\hat{q}_m(\kappa^*)$ values are multiplied by $F_c(\kappa^*)$. Similar to the $2/3$ dealiasing rule [Orszag, 1971], this filtering scheme reduces aliasing errors in the same range of scales while also providing the numerical dissipation necessary for stable simulations.

We configure the model with a doubly periodic square domain of size $L = 10^6$ m, featuring flat topography and a total depth of $H = H_1 + H_2$. The model includes a fixed mean zonal velocity shear, ΔU, with $U_2 = 0$. For our different cases, we use different deformation radii r_d, which is the characteristic scale for baroclinic instability and mesoscale turbulence, defined as $r_d^2 = \frac{g'}{f_0^2} \frac{H_1 H_2}{H}$.

We select the model's grid size, Δx, based on the deformation radius. To accurately resolve mesoscale eddies, $r_d/\Delta x$ must be greater than 2 [Hallberg, 2013]. For $r_d = 20,000$ m, a 256×256 grid with $\Delta x_{\text{hires}} = L/256 = 3906.25$ m yields $r_d/\Delta x_{\text{hires}} = 5.12$, ensuring mesoscale turbulence is well-resolved. Conversely, a 64×64 grid with $\Delta x_{\text{Lowres}} = L/64 = 15,625$ m results in $r_d/\Delta x_{\text{Lowres}} = 1.28$, which is insufficient to realistically simulate mesoscale eddies. In this lower-resolution setup, parameterization is necessary to account for the unresolved SGS processes. All simulations are conducted with a numerical timestep of $\Delta t = 1$ hour.

We consider four different cases categorized into two flow regimes to test our parameterization's generalization ability shown in Fig. 2: the *eddy configuration*, which leads to the formation of isotropically distributed eddies, and the *jet configuration*, which results in the formation of anisotropic jets. These configurations exemplify the two primary scaling regimes of meridional heat transport [Gallet and Ferrari, 2021].

2.2.3 Predicting Subgrid Forcing

We aim to create a mapping from low-resolution velocity (obtained by filtering and coarse-graining the high-resolution velocity fileds) to the subgrid forcing in potential vorticity, Π_{q_m}. First, we quantify the subgrid forcing by filtering and coarse-graining high-resolution simulations. This approach assumes the coarsened high-resolution data distribution is similar to low-resolution data for the same data-driven parameterizations to be applicable. We denote the filtering and coarse-graining operator as $(\bar{\circ})$. The prognostic equation, after filtering, can be expressed as:

$$\frac{\partial \hat{\bar{q}}_m}{\partial t} = -\hat{\bar{J}}(\overline{\psi}_m, \overline{q}_m) - ik\beta_m \hat{\bar{\psi}}_m - ikU_m \hat{\bar{q}}_m + \delta_{m,2} r_{ek} \kappa^2 \hat{\bar{\psi}}_2 + s\hat{s}d + \hat{\Pi}_{q_m}, \tag{5}$$

Let ∂_t^H and ∂_t^L represent the tendency operators for high- and low-resolution models, respectively. For any given high-resolution q, its subgrid forcing (encompassing nonlinear advection and numerical dissipation([Kent et al., 2016, Porta Mana and Zanna, 2014, Ross et al., 2023, Shevchenko and Berloff, 2021])) is defined as:

$$\Pi_{q_m} = \overline{\partial_t^H q_m} - \partial_t^L \bar{q}_m. \tag{6}$$

We calculate Π_q by initializing the high- and low-resolution models with q and $\bar{q}$, respectively, advancing both models one step with the same Δt, and then subtracting the low-resolution model's tendency from the filtered and coarse-grained tendency of the high-resolution model.

2.2.4 Filtering and Coarse-graining

For data generation, we first coarse-grain and then filter, which are commutative operations for elementwise spectral filtering. Coarse-graining involves reducing the simulation's resolution by a factor of K, specifically by truncating the spatial modes of $\hat{q}$ to retain only the first $1/K$ modes. Spectral filtering typically applies selective decay, diminishing the strength of the highest wavenumbers while retaining the low-wavenumber components after truncation. For the filtering, we apply the Gaussian filter to all remaining modes as follows:

$$\hat{\bar{q}}_k = \hat{q}_k * e^{-\kappa^2 (2\Delta x_{\text{Lowres}})^2 / 24} \tag{7}$$

This is a commonly used filter in SGS modeling [Guan et al., 2022a, Jakhar et al., 2023, Pope, 2000] where the filter width is chosen to be twice as large as the grid size of the coarse model [Lund, 1997].

2.3 Convolutional Neural Network (CNN) and Transfer Learning (TL)

Building on our previous work [Guan et al., 2022a, 2023, Subel et al., 2023], we develop non-local data-driven SGS parameterizations for each case by training a CNN. The CNN takes input $\bar{\mathbf{u}}_{\mathbf{m}} = (\bar{u}_m(x,y), \bar{v}_m(x,y))$ and predicts $\Pi_{q_m}(x,y)$ as the output, where $m \in \{1,2\}$ represents the upper and lower levels. These CNNs consist of 9 sequential convolution layers, with 7 hidden layers each containing 64^2 kernels of size 5×5. The outputs of a convolutional layer, called activations, are denoted for channel j of layer ℓ as $g_\ell^j \in \mathbb{R}^{N_{\text{Lowres}} \times N_{\text{Lowres}}}$, and the activation equation is:

$$g_\ell^j(\mathbf{u}) = \sigma \left(\sum_\beta \left(W_\ell^{\beta, j} \circledast g_{\ell-1}^\beta(\mathbf{u}) \right) + b_\ell^j \right). \tag{8}$$

Note that $N_{\text{Lowres}} = 64$ for all cases. Here, $\circledast$ represents spatial convolution and $\sigma(\circ) = \max(0, \circ)$ is the ReLU activation function (absent in the linear output layer, $\ell = 9$). $W_\ell^{\beta, j} \in \mathbb{R}^{5 \times 5}$ is the weight matrix of a convolution kernel, and $b_\ell^j \in \mathbb{R}^{64 \times 64}$ is the regression bias, a constant matrix. For all layers, $\beta \in \{1, 2, \ldots, 64\}$ and $j \in \{1, 2, \ldots, 64\}$ except in the input layer ($\ell = 1$), where $\beta \in \{1, 2, 3, 4\}$, and in the output layer ($\ell = 9$), where $j \in \{1, 2\}$ as the output has two channels. The kernels' weights and biases are the trainable parameters of the NN, collectively referred to as $\theta \in \mathbb{R}^p$. Note that $g_{\text{in}} = g_0 = \bar{\mathbf{u}}_{\mathbf{m}}$ and $g_{\text{out}} = g_9 = \Pi_{q_m}$.

We refer to CNNs that are trained from scratch on each Case$_i$ and tested on Case$_j$ as BNN^{i,j}. All the trainable parameters θ are *randomly* initialized. BNNs are trained with 40,000 samples from each case. Training is conducted for 100 epochs with a starting learning rate of $l_r = 10^{-3}$ and a scheduler that reduces the learning rate by factor of 10 when metrics stop improving to avoid overfitting on each dataset. We used a batch size of 8 and the ADAM optimizer [Kingma and Ba, 2014]. The mean-squared error (MSE) was used as the loss function, and the model performance was evaluated using the metrics introduced in section 2.5.

For TL, we initialize the weights and biases of BNN^0 and retrain only the first hidden layer ($\ell = 2$) with different percentages of data from the target case, using the same setup for learning as the BNNs. The models utilizing transfer learning from Case$_i$ to Case$_j$ are referred to as TLNN^{i,j}. This approach allows us to use less data to achieve good performance on the target case. The schematic of the CNN used in parameterization, its architecture, and its input/output in physical space are shown in Fig. 1.
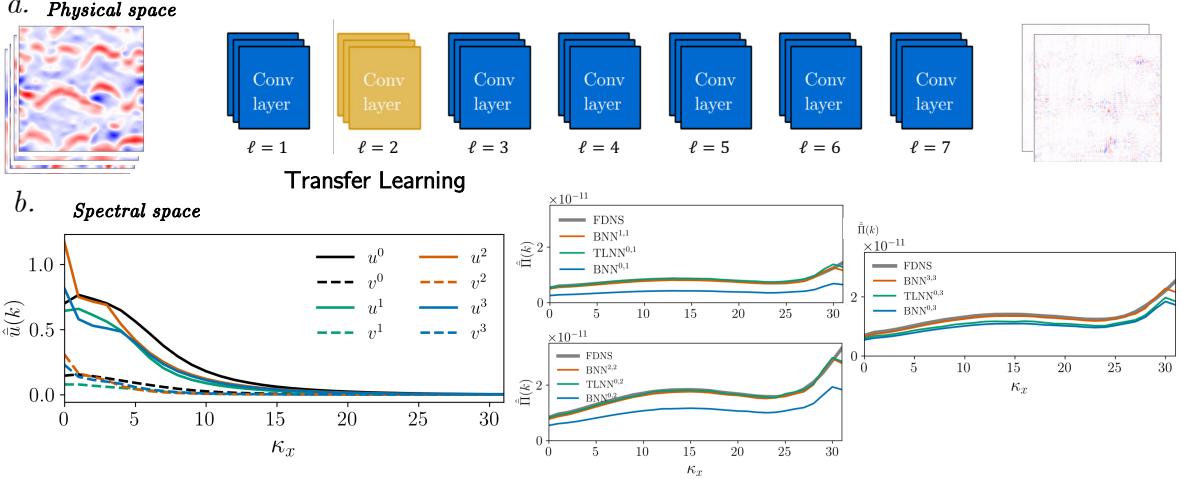
Figure 1: *Row a* displays the schematic of the CNN and inputs and outputs in physical space. Each TLNN is initialized with the weights of BNN^0, and only the first hidden layer ($\ell = 2$) is re-trained using a smaller percentage of data. The inputs are the meridional and zonal velocities of the upper and lower levels, and the output is the subgrid forcing for each level. *Row b* shows the inputs and outputs of the CNN in spectral space, with the spectrum meridionally averaged

2.4 Spectral Analysis of CNNs

The Fourier transform operator $\mathcal{F}$ is defined as

$$\left(\hat{\circ}\right) \equiv \mathcal{F}\left(\circ\right), \quad \mathcal{F}: \ \mathbb{R}^{64 \times 64} \longmapsto \mathbb{C}^{64 \times 64}. \tag{9}$$

To express convolution in the spectral domain, we first note that each kernel $W_\ell^{\beta,j} \in \mathbb{R}^{5 \times 5}$ can be extended to the full domain of the input by zero-padding, a common practice for faster training [Mathieu et al., 2013], resulting in $\widetilde{W}_\ell^{\beta,j} \in \mathbb{R}^{64 \times 64}$. Using the convolution theorem, we then obtain

$$W_\ell^{\beta,j} \circledast g_{\ell-1}^{\beta} = \mathcal{F}^{-1}\left(\widehat{\widetilde{W}}_\ell^{\beta,j} \odot \hat{g}_{\ell-1}^{\beta}\right), \tag{10}$$

where $\odot$ is element-wise multiplication.

Next, we define linear activation h_ℓ^j, which contains all the linear operations in Eq. (8):

$$h_\ell^j = \sum_\beta \left(W_\ell^{\beta,j} \circledast g_{\ell-1}^{\beta}\right) + b_\ell^j. \tag{11}$$

Even though Eq. (8) is nonlinear due to the ReLU function, its Fourier transform can still be derived analytically. By applying Eqs. (10) and (11) and utilizing the linearity property of the Fourier transform, we obtain

$$\hat{g}_\ell^j = \sum_\alpha \left(e^{-i(k_x x_\alpha + k_y y_\alpha)}\right) \circledast \hat{h}_\ell^j = \sum_\alpha \left(e^{-i(k_x x_\alpha + k_y y_\alpha)}\right) \circledast \left\{\sum_\beta \left(\widehat{\widetilde{W}}_\ell^{\beta,j} \odot \hat{g}_{\ell-1}^{\beta}\right) + \hat{b}_\ell^j\right\}, \tag{12}$$

where $(x_\alpha, y_\alpha) \in \left\{(x,y) \mid h_\ell^j(x,y) > 0\right\}$ and $i = \sqrt{-1}$. The sum over α arises from the ReLU function, involving grid points where $h_\ell^j > 0$. Note that $e^{-i(k_x x_\alpha + k_y y_\alpha)}$ represents the Fourier transform of the Heaviside function at (x_α, y_α). Also, since b_ℓ^j is a constant matrix, $\hat{b}_\ell^j$ is non-zero only at $k_x = k_y = 0$ (and is real).

Equation (12) indicates that the spectrum of $\hat{g}_\ell^j$ is influenced by the spectrum of $\hat{g}_{\ell-1}^j$, the spectra of the weights $\widehat{\widetilde{W}}_\ell^{\beta,j}$ (and constant biases $\hat{b}_\ell^j$), and the regions in the physical space where $h_\ell^j > 0$. With TL, updating the weights and biases changes both their spectra and the regions where $h_\ell^j > 0$.

To understand how TL enables the TLNN to capture the physics of different cases, we perform extensive analysis on how the weights $(\widetilde{\widehat{W}}_{\ell}^{\beta,j})$ and the spectrum of all layers $(\hat{g}_{\ell}^{\beta})$ change during TL. This allows us to explain how the distribution of kernels affects the shape of the spectrum and how these spectral changes propagate through the layers to the final layer where the loss function is applied. This process highlights the adaptation of weights to the physics and demonstrates how TL can adjust the weights to be consistent with the physics of the target system.

2.5 Offline metrics

We use several metrics to evaluate the model's *a priori* performance. The first metric is root mean square error, defined in Eq. (13), which assesses the optimality of the mapping between input and output. Here, N is the number of test samples, and $\|\cdot\|_2$ represents the L2 norm. This ensures the model prediction and true output are close in the L2 space.

$$\text{RMSE}_{\Pi_{q_m}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\bar{\Pi}_{\mathbf{q}_m i}^{\text{pred}} - \bar{\Pi}_{\mathbf{q}_m i}^{\text{true}}\right\|_2^2 \bigg/ \frac{1}{N}\sum_{i=1}^{N}\left\|\bar{\Pi}_{\mathbf{q}_m i}^{\text{true}}\right\|_2^2}. \tag{13}$$

The correlation coefficients (CC), averaged over N test samples, assess the CNN model's ability to capture the structure of the true data

$$CC_{\Pi_{q_m}} = \frac{\left\langle\left(\bar{\Pi}_{q_m}^{pred} - \langle\bar{\Pi}_{q_m}^{pred}\rangle\right)\left(\bar{\Pi}_{q_m}^{true} - \langle\bar{\Pi}_{q_m}^{true}\rangle\right)\right\rangle}{\sqrt{\left\langle\left(\bar{\Pi}_{q_m}^{pred} - \langle\bar{\Pi}_{q_m}^{pred}\rangle\right)^2\right\rangle}\sqrt{\left\langle\left(\bar{\Pi}_{q_m}^{true} - \langle\bar{\Pi}_{q_m}^{true}\rangle\right)^2\right\rangle}}, \tag{14}$$

where $\langle\cdot\rangle$ represents domain averaging.

We evaluate models performance by calculating the RMSE between the predicted output spectrum and the true output spectrum

$$\text{Spectrum RMSE} = \frac{1}{N_{k_x}}\sum_{k_x}\left|\frac{\hat{\bar{\Pi}}_{q_m}^{\text{pred}}(k_x) - \hat{\bar{\Pi}}_{q_m}^{\text{true}}(k_x)}{\hat{\bar{\Pi}}_{q_m}^{\text{true}}(k_x)}\right|. \tag{15}$$

Spectrum RMSE is essential for ensuring that the model not only predicts accurate values but also preserves the spectral characteristics of the data.

3 Results

In this section, we present and discuss the results of our study. We start by examining the distinct physical and spectral characteristics of different cases. Subsequently, we focus on the generalization capabilities of BNN and the effectiveness of TL to enhance these capabilities both *a priori* and *a posteriori*. Furthermore, we emphasize the significance of spectral RMSE in assessing the model's generalization performance. Additionally, we examine the CNNs in spectral space to observe the impact of retraining one layer on subsequent hidden-layer activation spectra. This section also explains the role of weights as spectral filters and shows and quantifies how kernel distributions optimally adjust in response to changes in training data.
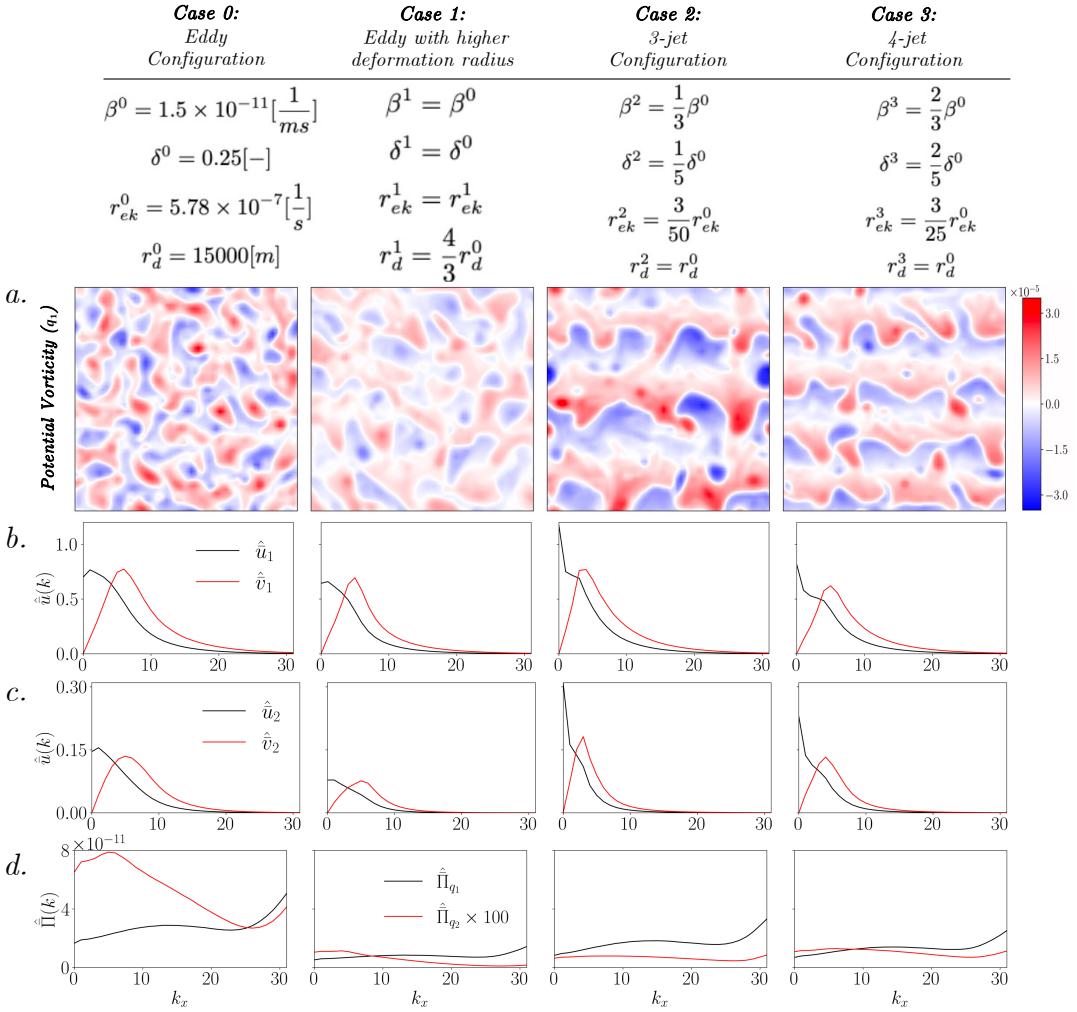
Case 0: Eddy Configuration	Case 1: Eddy with higher deformation radius	Case 2: 3-jet Configuration	Case 3: 4-jet Configuration
$\beta^0 = 1.5 \times 10^{-11}[\frac{1}{ms}]$	$\beta^1 = \beta^0$	$\beta^2 = \frac{1}{3}\beta^0$	$\beta^3 = \frac{2}{3}\beta^0$
$\delta^0 = 0.25[-]$	$\delta^1 = \delta^0$	$\delta^2 = \frac{1}{5}\delta^0$	$\delta^3 = \frac{2}{5}\delta^0$
$r_{ek}^0 = 5.78 \times 10^{-7}[\frac{1}{s}]$	$r_{ek}^1 = r_{ek}^1$	$r_{ek}^2 = \frac{3}{50}r_{ek}^0$	$r_{ek}^3 = \frac{3}{25}r_{ek}^0$
$r_d^0 = 15000[m]$	$r_d^1 = \frac{4}{3}r_d^0$	$r_d^2 = r_d^0$	$r_d^3 = r_d^0$

Figure 2: Comparative analysis of the base system and three target configurations using 10 years of simulation data. *Row a:* Snapshots of potential vorticity showcasing the spatial distribution of eddies in each system—eddies display a roughly isotropic structure, while jets exhibit more organized zonal alignment. *Row b:* Meridionally averaged spectra of velocity profiles at the upper level. *Row c:* Meridionally averaged spectra of velocity profiles at the lower level. *Row d:* Meridionally averaged spectra of subgrid forcing at both levels. In all spectral panels, k_x represents the zonal wavenumber, and spectra are averaged over the meridional direction

3.1 Overcoming BNN's Generalization Limits: How TL Bridges the Gap

Before explaining the physics behind TL, we need to quantify the improvement in parameterization performance due to TL. Here, we discuss the ways TL improves parameterization and metrics that can best assess the model's generalization. By examining Fig. 2 (a), we observe that adjusting the parameters of Eq. (2) leads to different physical characteristics. When examining the potential vorticity snapshots of the upper level (q_1) shown in Fig. 2 (a), isotropic eddies are formed for Case 0. The same system's behavior with larger eddies is observed by increasing the deformation radius (Case 1). Altering β, δ, and r_{ek} leads to creating highly anisotropic jets with varying numbers (Case 2 & 3). Different physical characteristics are reflected in different spectra, meaning that cases differ in both large and small scales. Figures 2 (b, c, and d) illustrate the distinct physics governing each system, resulting in differences in velocity and subgrid forcing spectra.

Figure 3 (a) shows that, when provided with sufficient data, the BNN^{i,i} performs best, within uncertainty, across all metrics. However, when tested on different cases, BNN^{0,i} struggles to generalize to other systems with different spectral properties, leading to sub-optimal performance. In particular, BNN^{0,i} fails to generalize across all metrics, except for the correlation coefficient in the upper layer,

9

which remains comparable to $BNN^{i,i}$. Furthermore, there is minimal change in the CC of the subgrid forcing at the upper level when more data is integrated into TL. However, the RMSE consistently decreases as more data is included in TL. Spectrum RMSE offers more valuable insights into the model's generalization capabilities. It consistently worsens at both upper and lower levels when the model is extrapolated and improves at both levels when TL is applied with 2% and 10% of target system data. These results demonstrate that TL enhances the generalization capability of BNN^0 using only a small fraction of the data needed to train a new BNN^i from scratch.

Figure 3 (b) shows the ratio of the output spectrum to the FDNS spectrum for each case. $BNN^{0,i}$ exhibits the largest gap relative to FDNS, while TL progressively closes this gap as more re-training data is used. This further confirms that TL helps align the spectral characteristics of the learned subgrid forcing with that of the true system.

As shown in Fig. 4, the TLNN can rectify mismatches in the kinetic energy (KE) spectra wherever there is room for improvement. Specifically, Fig. 4 (a) shows that $TLNN^{0,i}$ improves the KE spectra at high wavenumbers compared to $BNN^{0,i}$, with both parameterizations performing better than the simulation without any SGS parameterization. In contrast, Fig. 4 (c) shows that all parameterizations perform equally well in that case, and all surpass the no-parameterization baseline. Fig. 4 (e) demonstrates that $TLNN^{0,i}$ better captures the KE spectra at both low and high wavenumbers compared to $BNN^{0,i}$.

The scale-selective dissipation (ssd) introduced in Section 2.2.2 can obscure the positive effects of TL in improving online results. This is evident in the PDFs of q_1 in Fig. 4 (b, d, f), where even the baseline simulation without SGS parameterization captures the mean flow well. Nonetheless, TL proves beneficial wherever there is room for improvement. For example, Fig. 4 (d) shows that $TLNN^{0,i}$ improves the PDF tails in Case 2, highlighting its potential to fine-tune models efficiently for extreme events with limited re-training data across different dynamical regimes.

3.2 Why Does BNN Fail to Generalize? How Does TL Solve Generalization Issues?

Analyzing the CNN in the spectral space provides a clearer distinction between different physical systems [Pahlavan et al., 2024b, Subel et al., 2023]. When BNN^0 is applied to out-of-distribution inputs, it underestimates the channel-averaged, meridionally averaged activation spectrum relative to $BNN^{0,0}$, whose weights are tuned for in-distribution data. This underestimation begins in early layers and propagates through the network, ultimately resulting in a mismatch with the FDNS output spectrum.

Figure 5 illustrates this mechanism. *Row a* demonstrates that, when $BNN^{0,i}$ is tested with out-of-distribution data, underestimation of the spectral content is evident across all layers compared to $BNN^{0,0}$. This leads to a mismatch in the output spectrum relative to FDNS.

However, applying TL by re-training only the second hidden layer in $TLNN^{0,i}$ helps close this gap. As seen in *Rows b to d* of Fig. 5, this localized re-training causes an upshift in the spectrum at layer 2. While this adjustment may be subtle at $\ell = 2$, it propagates through subsequent layers, leading to an output spectrum that better matches the FDNS. The behavior of $TLNN^{0,i}$ becomes more similar to that of $BNN^{i,i}$, which serves as the ideal case for each target dataset.

Figure 5 thus highlights how targeted re-training of just one layer can realign the CNN's internal spectral response with the physics of the new system, enabling generalization across dynamical regimes.

3.3 Spectral Filters in Action: The Role of Weights in Generalization

The weights within a CNN act as spectral filters that influence how information at different spatial scales is processed through the network. To better understand what controls the spectral content of activations g_ℓ, we analyze the convolutional kernels in Fourier space—a powerful approach widely used in understanding the physics of turbulence.

We examine all 64^2 kernels in the second convolutional layer. Since direct visualization is impractical, we apply the k-means clustering algorithm to identify representative cluster centers. As shown in Fig. 6, the cluster centers across all four cases consistently represent combinations of low-pass, Gabor, and high-pass filters. This behavior is largely independent of whether the training data are isotropic or anisotropic, and does not, on its own, distinguish between different physical systems. Nevertheless, the spectral magnitudes of the weights exhibit structured peaks at specific wavenumber pairs (k_x, k_y).

To analyze how the distribution of kernel weights adapts during transfer learning, we identify the wavenumber pair (k_x, k_y) where the absolute value of each Fourier-transformed, padded kernel $\left|\widehat{\widetilde{W}}_2^{\beta,j}\right|$ reaches its maximum. This gives a spectral "footprint" of each filter's dominant response. We focus specifically on layer 2—the only layer re-trained during TL—while keeping all other layers frozen (as discussed in Section 3.2). This setup allows us to isolate and quantify how the second layer adjusts when adapting to out-of-distribution samples.

Histograms of these dominant wavenumber pairs are shown in Fig. 7 (b–e) for both BNN^0 and $TLNN^{0,i}$. Most kernels behave as low-pass or high-pass filters, while Gabor filters appear less frequently and without a dominant orientation. The saturation in the center and corners of each histogram indicates a higher concentration of low-pass and high-pass filters.

To further analyze how these maxima change under TL, we divide the kernels into two categories:

1. Unchanged maxima locations: For filters whose dominant wavenumber remains the same after TL (Fig. 7 (f)), we compute the mean ratio of spectral amplitude at the maximum before and after re-training. As shown in Fig. 7 (g), this amplitude increases consistently, aligning with the upshift in the activation spectra observed in Fig. 5 (b–d).

2. Shifted maxima locations: For kernels whose dominant wavenumber changes after TL (Fig. 7 (h)), we compute the radial wavenumber $\kappa = \sqrt{k_x^2 + k_y^2}$ before and after TL to determine how these shifts affect the scale preference of the filters. Fig. 7 (i) shows that TL shifts many of these maxima toward lower wavenumbers, indicating a stronger focus on large-scale features. This behavior is consistent with the decaying nature of the activation spectra and supports the idea that adapting to new dynamics requires capturing the dominant large-scale structure.

4 Discussion

In Section 3, we introduced a non-intrusive approach to explain how kernels adapt during TL with minimal data. A detailed analysis of the weights in BNNs and TLNNs reveals that the learned kernels function as low-pass, Gabor, and high-pass filters, regardless of whether the training data are isotropic or anisotropic. This study presents the first comprehensive effort to relate the distribution of these spectral filters to the spectral characteristics of isotropic and anisotropic flows in the context of SGS modeling.

The methodology proposed here advances our understanding of what is learned during TL from physical data, particularly in high-dimensional systems. Rather than viewing TL purely as an optimization task, this study aims to uncover the underlying mechanisms that govern learning from limited data. For instance, our findings suggest that spectral bias [Chattopadhyay and Hassanzadeh, 2023, Chattopadhyay et al., 2024, Gray et al., 2024, Lupin-Jimenez et al., 2025] in NNs—where filters underrepresent high-wavenumber content—can be partially explained by how weights evolve during training.

While this analysis assumes that filters have a single global maximum, some kernels exhibit multiple significant local maxima that pass activation at different scales. Although this assumption simplifies the analysis, more detailed studies could provide a more precise understanding of what is learned during TL. Additionally, while clustering offers a starting point for explaining the learned kernels as spectral filters, more is needed to fully explain how these kernels process information at different scales. Furthermore, using radial wavenumber to analyze kernel spectra distribution may overlook directionality, necessitating further research to address this limitation. It is also important to note the role of numerical dissipation in the solver, which may obscure the improvements brought by TL.

Although our findings are likely specific to the test cases, network architecture, and SGS parameterization studied here, the analysis methods are broadly applicable. They can be extended to a wide range of base–target system combinations and applications, including data-driven forecasting and training set blending. The comprehensive analysis introduced here holds potential for many multi-scale dynamical systems applications.

In conclusion, we examined the performance of CNN-based parameterizations for quasi-geostrophic turbulence, focusing on various offline metrics to determine which best explains generalization performance. By analyzing CNNs in the spectral space, we uncovered why generalization fails in these systems, specifically by examining activation spectra and their propagation through subsequent layers.

To understand how TL can improve this sub-optimal performance, we investigated how kernel distributions adapt to the training dataset. Identifying the quantity and intensity of these filters before and after TL allowed us to explain how TL effectively bridges the generalization gap.

Data Availability Statement   The code used for the analysis in this project is available on GitHub at the following link: https://github.com/moeindarman77/TransferLearning-QG

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

Author Contributions   P.H., A.C., and L.Z. conceptualized the research. M.D. conducted the research. A.C. contributed to some of the early versions of the computational codes. A.C. and M.D. wrote the draft. All authors analyzed the results and edited the manuscript.

Competing Interests   The authors declare that they have no competing interests.

References

R. Abernathey, Rochanotes, A. Ross, M. Jansen, Ziwei Li, F. J. Poulin, N. C. Constantinou, Anirban Sinha, Dhruv Balwada, SalahKouhen, S. Jones, C. B. Rocha, C. L. P. Wolfe, Chuizheng Meng, H. Van Kemenade, J. Bourbeau, J. Penn, J. Busecke, M. Bueti, and , Tobias. pyqg/pyqg: v0.7.2, 2022. URL https://zenodo.org/record/6563667.

A. Arakawa and V. R. Lamb. *Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model*, page 173–265. Elsevier, 1977. doi: 10.1016/b978-0-12-460817-7.50009-4. URL http://dx.doi.org/10.1016/B978-0-12-460817-7.50009-4.

A. Bracco, J. Brajard, H. A. Dijkstra, P. Hassanzadeh, C. Lessig, and C. Monteleoni. Machine learning for the physics of climate. *Nature Reviews Physics*, 7(1):6–20, Nov. 2024. ISSN 2522-5820. doi: 10.1038/s42254-024-00776-3. URL http://dx.doi.org/10.1038/s42254-024-00776-3.

A. Bracco, J. Brajard, H. A. Dijkstra, P. Hassanzadeh, C. Lessig, and C. Monteleoni. Machine learning for the physics of climate. *Nature Reviews Physics*, 7(1):6–20, 2025.

D. Carati, S. Ghosal, and P. Moin. On the representation of backscatter in dynamic localization models. *Physics of Fluids*, 7(3):606–616, Mar. 1995. ISSN 1089-7666. doi: 10.1063/1.868585. URL http://dx.doi.org/10.1063/1.868585.

A. Chattopadhyay and P. Hassanzadeh. Long-term instabilities of deep learning-based digital twins of the climate system: The cause and a solution. *arXiv preprint arXiv:2304.07029*, 2023.

A. Chattopadhyay, A. Subel, and P. Hassanzadeh. Data-driven super-parameterization using deep learning: Experimentation with multiscale Lorenz 96 systems and transfer learning. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002084, 2020.

A. Chattopadhyay, M. Gray, T. Wu, A. B. Lowe, and R. He. Oceannet: A principled neural operator-based digital twin for regional oceans. *Scientific Reports*, 14(1):21181, 2024.

X. Chen, C. Gong, Q. Wan, L. Deng, Y. Wan, Y. Liu, B. Chen, and J. Liu. Transfer learning for deep neural network-based partial differential equations solving. *Advances in Aerodynamics*, 3(1), Dec. 2021. ISSN 2524-6992. doi: 10.1186/s42774-021-00094-7. URL http://dx.doi.org/10.1186/s42774-021-00094-7.

S. Desai, M. Mattheakis, H. Joy, P. Protopapas, and S. Roberts. One-shot transfer learning of physics-informed neural networks, 2021. URL https://arxiv.org/abs/2110.11286.

A. Dipankar, B. Stevens, R. Heinze, C. Moseley, G. Zängl, M. Giorgetta, and S. Brdar. Large eddy simulation using the general circulation model ¡scp¿icon¡/scp¿. *Journal of Advances in Modeling Earth Systems*, 7(3):963–986, July 2015. ISSN 1942-2466. doi: 10.1002/2015ms000431. URL http://dx.doi.org/10.1002/2015MS000431.

J. A. Domaradzki, R. W. Metcalfe, R. S. Rogallo, and J. J. Riley. Analysis of subgrid-scale eddy viscosity with use of results from direct numerical simulations. *Physical Review Letters*, 58(6): 547–550, Feb. 1987. ISSN 0031-9007. doi: 10.1103/physrevlett.58.547. URL http://dx.doi.org/10.1103/PhysRevLett.58.547.

D. G. Fox and S. A. Orszag. Pseudospectral approximation to two-dimensional turbulence. *Journal of Computational Physics*, 11(4):612–619, 1973.

R. O. Fox. Large-eddy-simulation tools for multiphase flows. *Annual Review of Fluid Mechanics*, 44(1):47–76, Jan. 2012. ISSN 1545-4479. doi: 10.1146/annurev-fluid-120710-101118. URL http://dx.doi.org/10.1146/annurev-fluid-120710-101118.

B. Fox-Kemper and D. Menemenlis. *Can large eddy simulation techniques improve mesoscale rich ocean models?*, page 319–337. American Geophysical Union, 2008. doi: 10.1029/177gm19. URL http://dx.doi.org/10.1029/177GM19.

B. Gallet and R. Ferrari. A quantitative scaling theory for meridional heat transport in planetary atmospheres and oceans. *AGU Advances*, 2(3):e2020AV000362, 2021.

Y. Gao, K. C. Cheung, and M. K. Ng. Svd-pinns: Transfer learning of physics-informed neural networks via singular value decomposition. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Dec. 2022. doi: 10.1109/ssci51031.2022.10022281. URL http://dx.doi.org/10.1109/SSCI51031.2022.10022281.

M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, July 1991. ISSN 0899-8213. doi: 10.1063/1.857955. URL http://dx.doi.org/10.1063/1.857955.

S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 106:102447, Apr. 2020. ISSN 0167-8442. doi: 10.1016/j.tafmec.2019.102447. URL http://dx.doi.org/10.1016/j.tafmec.2019.102447.

S. Goswami, K. Kontolati, M. D. Shields, and G. E. Karniadakis. Deep transfer operator learning for partial differential equations under conditional shift. *Nature Machine Intelligence*, 4(12):1155–1164, Dec. 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00569-2. URL http://dx.doi.org/10.1038/s42256-022-00569-2.

M. A. Gray, A. Chattopadhyay, T. Wu, A. Lowe, and R. He. Long-term prediction of the gulf stream meander using oceannet: a principled neural operator-based digital twin. *EGUsphere*, 2024:1–23, 2024.

I. Grooms. Backscatter in energetically-constrained leith parameterizations. *Ocean Modelling*, 186: 102265, Dec. 2023a. ISSN 1463-5003. doi: 10.1016/j.ocemod.2023.102265. URL http://dx.doi.org/10.1016/j.ocemod.2023.102265.

I. Grooms. Backscatter in energetically-constrained leith parameterizations. *Ocean Modelling*, 186: 102265, 2023b.

Y. Guan, A. Chattopadhyay, A. Subel, and P. Hassanzadeh. 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, 2022a.

Y. Guan, A. Subel, A. Chattopadhyay, and P. Hassanzadeh. Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate LES. *Physica D: Nonlinear Phenomena*, 443:133568, 2023.

Y. Guan, P. Hassanzadeh, T. Schneider, O. Dunbar, D. Z. Huang, J. Wu, and I. Lopez-Gomez. Online learning of eddy-viscosity and backscattering closures for geophysical turbulence using ensemble kalman inversion, 2024. URL https://arxiv.org/abs/2409.04985.

H. Guo, X. Zhuang, P. Chen, N. Alajlan, and T. Rabczuk. Analysis of three-dimensional potential problems in non-homogeneous media with physics-informed deep collocation method using material transfer learning and sensitivity analysis. *Engineering with Computers*, 38(6):5423–5444, Mar. 2022. ISSN 1435-5663. doi: 10.1007/s00366-022-01633-6. URL http://dx.doi.org/10.1007/s00366-022-01633-6.

E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379:113741, June 2021. ISSN 0045-7825. doi: 10.1016/j.cma.2021.113741. URL http://dx.doi.org/10.1016/j.cma.2021.113741.

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

J. M. Hanna, J. V. Aguado, S. Comas-Cardona, R. Askri, and D. Borzacchiello. Residual-based adaptivity for two-phase flow simulation in porous media using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 396:115100, June 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2022.115100. URL http://dx.doi.org/10.1016/j.cma.2022.115100.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, Jan. 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL http://dx.doi.org/10.1016/0893-6080(89)90020-8.

K. Jakhar, Y. Guan, R. Mojgani, A. Chattopadhyay, and P. Hassanzadeh. Learning closed-form equations for subgrid-scale closures from high-fidelity data: Promises and challenges, 2023. URL https://arxiv.org/abs/2306.05014.

M. F. Jansen and I. M. Held. Parameterizing subgrid-scale eddy effects using energetically consistent backscatter. *Ocean Modelling*, 80:36–48, Aug. 2014. ISSN 1463-5003. doi: 10.1016/j.ocemod.2014.06.002. URL http://dx.doi.org/10.1016/j.ocemod.2014.06.002.

M. F. Jansen, A. Adcroft, S. Khani, and H. Kong. Toward an energetically consistent, resolution aware parameterization of ocean mesoscale eddies. *Journal of Advances in Modeling Earth Systems*, 11(8):2844–2860, Aug. 2019. ISSN 1942-2466. doi: 10.1029/2019ms001750. URL http://dx.doi.org/10.1029/2019MS001750.

J. Kent, C. Jablonowski, J. Thuburn, and N. Wood. An energy-conserving restoration scheme for the shallow-water equations. *Quarterly Journal of the Royal Meteorological Society*, 142(695):1100–1110, Jan. 2016. ISSN 1477-870X. doi: 10.1002/qj.2713. URL http://dx.doi.org/10.1002/qj.2713.

R. M. Kerr, J. A. Domaradzki, and G. Barbier. Small-scale properties of nonlinear interactions and subgrid-scale energy transfer in isotropic turbulence. *Physics of Fluids*, 8(1):197–208, Jan. 1996. ISSN 1089-7666. doi: 10.1063/1.868827. URL http://dx.doi.org/10.1063/1.868827.

S. Khani and M. L. Waite. Backscatter in stratified turbulence. *European Journal of Mechanics - B/Fluids*, 60:1–12, Nov. 2016. ISSN 0997-7546. doi: 10.1016/j.euromechflu.2016.06.012. URL http://dx.doi.org/10.1016/j.euromechflu.2016.06.012.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

B. Knaepen and P. Moin. Large-eddy simulation of conductive flows at low magnetic reynolds number. *Physics of Fluids*, 16(5):1255–1261, May 2004. ISSN 1089-7666. doi: 10.1063/1.1651484. URL http://dx.doi.org/10.1063/1.1651484.

D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. L. Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex), 2020. URL https://arxiv.org/abs/2003.00688.

P. R. Larraondo, L. J. Renzullo, I. Inza, and J. A. Lozano. A data-driven approach to precipitation parameterizations using convolutional encoder-decoder neural networks, 2019. URL https://arxiv.org/abs/1903.10274.

D. C. Leslie and G. L. Quarini. The application of turbulence theory to the formulation of subgrid modelling procedures. *Journal of Fluid Mechanics*, 91(01):65, Mar. 1979. ISSN 1469-7645. doi: 10.1017/s0022112079000045. URL http://dx.doi.org/10.1017/S0022112079000045.

Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar. Physics-informed neural operator for learning partial differential equations, 2021. URL https://arxiv.org/abs/2111.03794.

T. Lund. On the use of discrete filters for large eddy simulation. *Annual Research Briefs*, pages 83–95, 1997.

L. Lupin-Jimenez, M. Darman, S. Hazarika, T. Wu, M. Gray, R. He, A. Wong, and A. Chattopadhyay. Simultaneous emulation and downscaling with physically-consistent deep learning-based regional ocean emulators. *arXiv preprint arXiv:2501.05058*, 2025.

P. J. Mason and D. J. Thomson. Stochastic backscatter in large-eddy simulations of boundary layers. *Journal of Fluid Mechanics*, 242:51–78, Sept. 1992. ISSN 1469-7645. doi: 10.1017/s0022112092002271. URL http://dx.doi.org/10.1017/S0022112092002271.

M. Mathieu, M. Henaff, and Y. LeCun. Fast training of convolutional networks through ffts. *arXiv preprint arXiv:1312.5851*, 2013.

M. Mattheakis, H. Joy, and P. Protopapas. Unsupervised reservoir computing for solving ordinary differential equations, 2021. URL https://arxiv.org/abs/2108.11417.

C. Meneveau and J. Katz. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1):1–32, Jan. 2000. ISSN 1545-4479. doi: 10.1146/annurev.fluid.32.1.1. URL http://dx.doi.org/10.1146/annurev.fluid.32.1.1.

S. A. Orszag. On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components. *Journal of Atmospheric Sciences*, 28(6):1074–1074, 1971.

H. A. Pahlavan, P. Hassanzadeh, and M. J. Alexander. Explainable offline-online training of neural networks for parameterizations: A 1d gravity wave-qbo testbed in the small-data regime. *Geophysical Research Letters*, 51(2), Jan. 2024a. ISSN 1944-8007. doi: 10.1029/2023gl106324. URL http://dx.doi.org/10.1029/2023GL106324.

H. A. Pahlavan, P. Hassanzadeh, and M. J. Alexander. Explainable offline-online training of neural networks for parameterizations: A 1d gravity wave-qbo testbed in the small-data regime. *Geophysical Research Letters*, 51(2):e2023GL106324, 2024b.

U. Piomelli. Large-eddy simulation: achievements and challenges. *Progress in Aerospace Sciences*, 35(4):335–362, May 1999. ISSN 0376-0421. doi: 10.1016/s0376-0421(98)00014-1. URL http://dx.doi.org/10.1016/S0376-0421(98)00014-1.

S. Pope. *Turbulent flows*. Cambridge university press, 2000.

P. Porta Mana and L. Zanna. Toward a stochastic parameterization of ocean mesoscale eddies. *Ocean Modelling*, 79:1–20, July 2014. ISSN 1463-5003. doi: 10.1016/j.ocemod.2014.04.002. URL http://dx.doi.org/10.1016/j.ocemod.2014.04.002.

K. G. Pressel, S. Mishra, T. Schneider, C. M. Kaul, and Z. Tan. Numerics and subgrid-scale modeling in large eddy simulations of stratocumulus clouds. *Journal of Advances in Modeling Earth Systems*, 9(2):1342–1365, June 2017. ISSN 1942-2466. doi: 10.1002/2016ms000778. URL http://dx.doi.org/10.1002/2016MS000778.

S. Rasp, M. S. Pritchard, and P. Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, Sept. 2018. ISSN 1091-6490. doi: 10.1073/pnas.1810286115. URL http://dx.doi.org/10.1073/pnas.1810286115.

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

P. Sagaut. *Large eddy simulation for incompressible flows: an introduction.* Springer Science & Business Media, 2005.

P. Sagaut, M. Terracol, and S. Deck. *Multiscale and multiresolution approaches in turbulence-LES, DES and Hybrid RANS/LES Methods: Applications and Guidelines.* World Scientific, 2013.

H. Sarlak, C. Meneveau, and J. Sørensen. Role of subgrid-scale modeling in large eddy simulation of wind turbine wake interactions. *Renewable Energy*, 77:386–399, May 2015. ISSN 0960-1481. doi: 10.1016/j.renene.2014.12.036. URL http://dx.doi.org/10.1016/j.renene.2014.12.036.

T. Schneider, S. Lan, A. Stuart, and J. Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44(24), Dec. 2017. ISSN 1944-8007. doi: 10.1002/2017gl076101. URL http://dx.doi.org/10.1002/2017GL076101.

I. Shevchenko and P. Berloff. On a minimum set of equations for parameterisations in comprehensive ocean circulation models. *Ocean Modelling*, 168:101913, Dec. 2021. ISSN 1463-5003. doi: 10.1016/j.ocemod.2021.101913. URL http://dx.doi.org/10.1016/j.ocemod.2021.101913.

V. Shinde. Proper orthogonal decomposition assisted subfilter-scale model of turbulence for large eddy simulation. *Physical Review Fluids*, 5(1), Jan. 2020. ISSN 2469-990X. doi: 10.1103/physrevfluids.5.014605. URL http://dx.doi.org/10.1103/PhysRevFluids.5.014605.

Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment*. *Monthly Weather Review*, 91(3):99–164, Mar. 1963. ISSN 1520-0493. doi: 10.1175/1520-0493(1963)091⟨0099:gcewtp⟩2.3.co;2. URL http://dx.doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

R. J. Stevens, L. A. Martínez-Tossas, and C. Meneveau. Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments. *Renewable Energy*, 116:470–478, Feb. 2018. ISSN 0960-1481. doi: 10.1016/j.renene.2017.08.072. URL http://dx.doi.org/10.1016/j.renene.2017.08.072.

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

A. Subel, Y. Guan, A. Chattopadhyay, and P. Hassanzadeh. Explaining the physics of transfer learning in data-driven turbulence modeling. *PNAS Nexus*, page pgad015, 2023.

S. Subramanian, P. Harrington, K. Keutzer, W. Bhimji, D. Morozov, M. Mahoney, and A. Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. 2023. doi: 10.48550/ARXIV.2306.00258. URL https://arxiv.org/abs/2306.00258.

Y. Q. Sun, H. A. Pahlavan, A. Chattopadhyay, P. Hassanzadeh, S. W. Lubis, M. J. Alexander, E. Gerber, A. Sheshadri, and Y. Guan. Data imbalance, uncertainty quantification, and generalization via transfer learning in data-driven parameterizations: Lessons from the emulation of gravity wave momentum transport in waccm, 2023. URL https://arxiv.org/abs/2311.17078.

Z. Tan, T. Schneider, J. Teixeira, and K. G. Pressel. Large-eddy simulation of subtropical cloud-topped boundary layers: 2. cloud response to climate change. *Journal of Advances in Modeling Earth Systems*, 9(1):19–38, Jan. 2017. ISSN 1942-2466. doi: 10.1002/2016ms000804. URL http://dx.doi.org/10.1002/2016MS000804.

J. Thuburn, J. Kent, and N. Wood. Cascades, backscatter and conservation in numerical models of two-dimensional turbulence. *Quarterly Journal of the Royal Meteorological Society*, 140(679):626–638, June 2013. ISSN 1477-870X. doi: 10.1002/qj.2166. URL http://dx.doi.org/10.1002/qj.2166.

C. Xu, B. T. Cao, Y. Yuan, and G. Meschke. Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios. *Computer Methods in Applied Mechanics and Engineering*, 405:115852, Feb. 2023. ISSN 0045-7825. doi: 10.1016/j.cma.2022.115852. URL http://dx.doi.org/10.1016/j.cma.2022.115852.

W. Xu, Y. Lu, and L. Wang. Transfer learning enhanced deeponet for long-time prediction of evolution equations, 2022. URL https://arxiv.org/abs/2212.04663.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? 2014. doi: 10.48550/ARXIV.1411.1792. URL https://arxiv.org/abs/1411.1792.

L. Zanna and T. Bolton. Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, 47(17), Aug. 2020. ISSN 1944-8007. doi: 10.1029/2020gl088376. URL http://dx.doi.org/10.1029/2020GL088376.

Y. Zhou. Eddy damping, backscatter, and subgrid stresses in subgrid modeling of turbulence. *Physical Review A*, 43(12):7049–7052, June 1991. ISSN 1094-1622. doi: 10.1103/physreva.43.7049. URL http://dx.doi.org/10.1103/PhysRevA.43.7049.

F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, Jan. 2021. ISSN 1558-2256. doi: 10.1109/jproc.2020.3004555. URL http://dx.doi.org/10.1109/JPROC.2020.3004555.

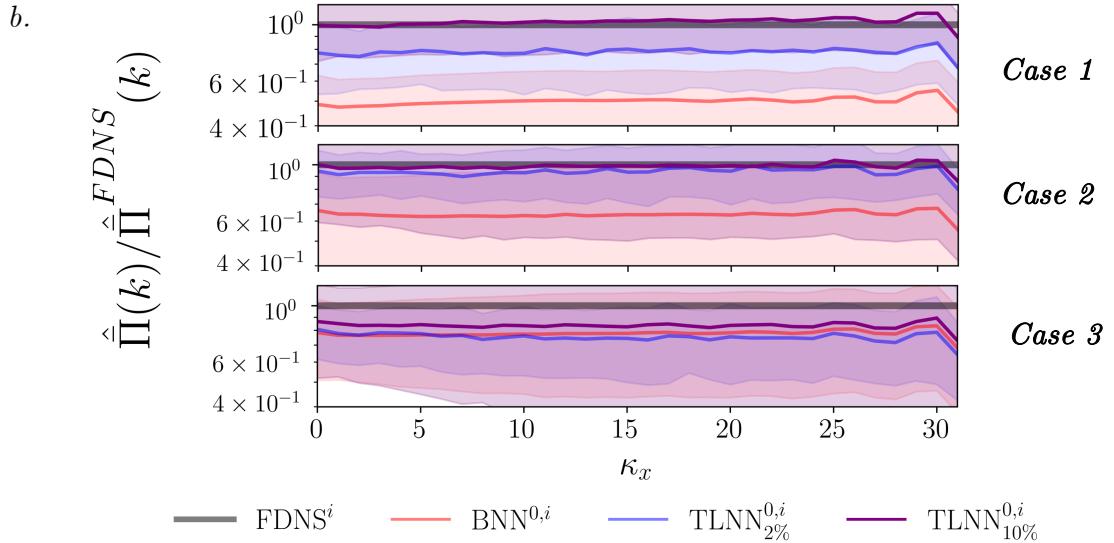| a.	CC ($\times 10^2$)		RMSE ($\times 10^2$)		Spectrum RMSE	
	Π_{q1}	Π_{q2}	Π_{q1}	Π_{q2}	$\Pi_{q1} \times 10^{-12}$	$\Pi_{q2} \times 10^{-14}$
BNN trained from random initialization						
$\text{BNN}^{1,1}$	92.9±0.8	99.1±0.2	0.58±0.03	0.22±0.02	0.77 ± 0.21	0.23 ± 0.09
$\text{BNN}^{2,2}$	93.2±1.6	92.3±1.7	0.58±0.06	0.61±0.08	2.03 ± 0.67	0.83 ± 0.33
$\text{BNN}^{3,3}$	94.5±0.9	96.0±0.7	0.52±0.04	0.44±0.04	1.26 ± 0.33	0.76 ± 0.30
BNN tested on target						
$\text{BNN}^{0,1}$	93.0±0.8	99.1±0.2	0.88±0.03	0.73±0.02	4.13 ± 0.77	2.93 ± 0.65
$\text{BNN}^{0,2}$	92.9±2.3	63.4±3.7	0.75±0.09	1.36±0.02	6.63 ± 1.28	5.29 ± 1.02
$\text{BNN}^{0,3}$	94.3±0.9	83.5±2.1	0.59±0.06	1.12±0.03	3.21 ± 0.57	6.68 ± 1.50
2% TL						
$\text{TLNN}^{0,1}$	93.0±0.7	99.3±0.1	0.58±0.03	0.19±0.01	0.93 ± 0.23	0.15 ± 0.04
$\text{TLNN}^{0,2}$	94.0±0.7	89.1±1.4	0.53±0.03	0.71±0.04	1.65 ± 0.33	1.07 ± 0.23
$\text{TLNN}^{0,3}$	93.8±0.9	93.1±0.7	0.54±0.04	0.57±0.03	1.47 ± 0.30	1.06 ± 0.30
10% TL						
$\text{TLNN}^{0,1}$	93.5±0.6	99.3±0.2	0.56±0.03	0.19±0.02	0.73 ± 0.17	0.15 ± 0.06
$\text{TLNN}^{0,2}$	93.8±1.1	89.5±1.6	0.55±0.04	0.70±0.05	2.10 ± 0.69	1.04 ± 0.34
$\text{TLNN}^{0,3}$	94.1±0.9	93.8±0.9	0.53±0.04	0.54±0.04	1.36 ± 0.29	1.03 ± 0.36

Figure 3: *A priori* evaluation of CNN parameterization. *Panel a*: CC, RMSE, and spectrum RMSE for upper and lower levels, comparing $\text{BNN}^{i,i}$, $\text{BNN}^{0,i}$, and $\text{TLNN}^{0,i}$ with different re-training data percentages across three target cases. *Panel b*: Ratio of output spectrum to FDNS spectrum for each case
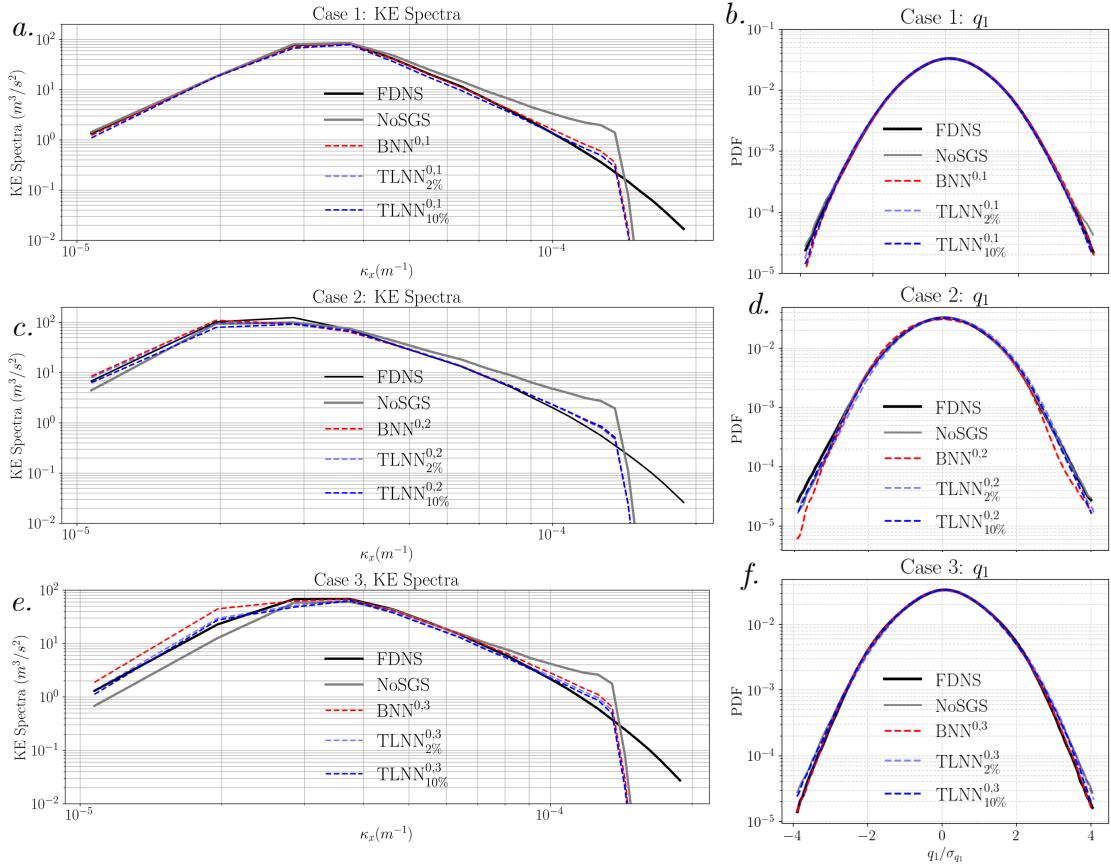
Figure 4: *A posteriori* evaluation of CNN parameterization. *Panels a, c, and e*: Kinetic energy spectra from 10-year simulations using $BNN^{0,i}$ and $TLNN^{0,i}$ across different cases. *Panels b, d, and f*: PDFs of potential vorticity at the upper level for the same simulations
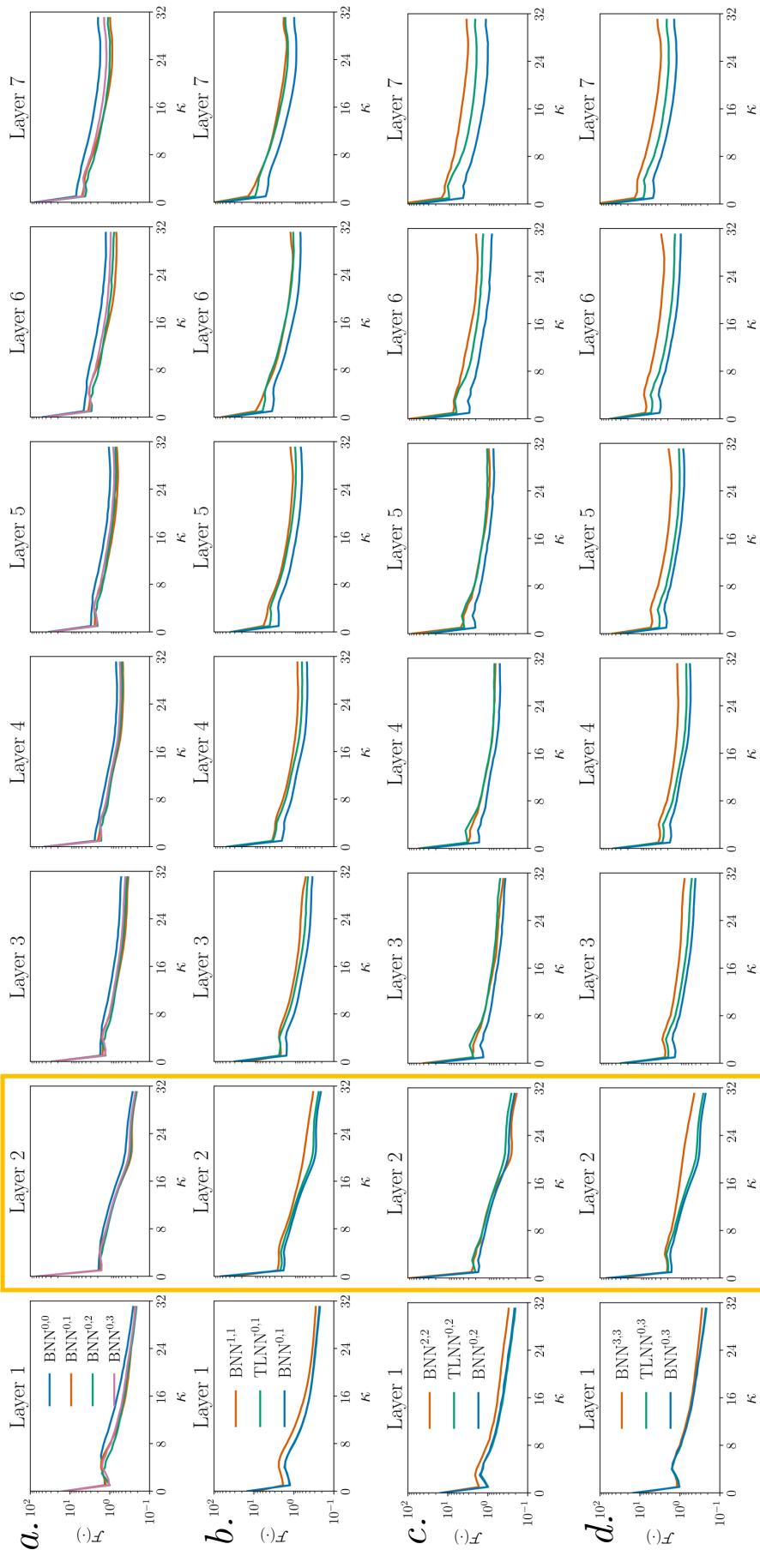
Figure 5: Channel-averaged, meridionally averaged spectra of hidden layer activations across different networks. *Rows a to d* show spectra of hidden layers for different models and cases.
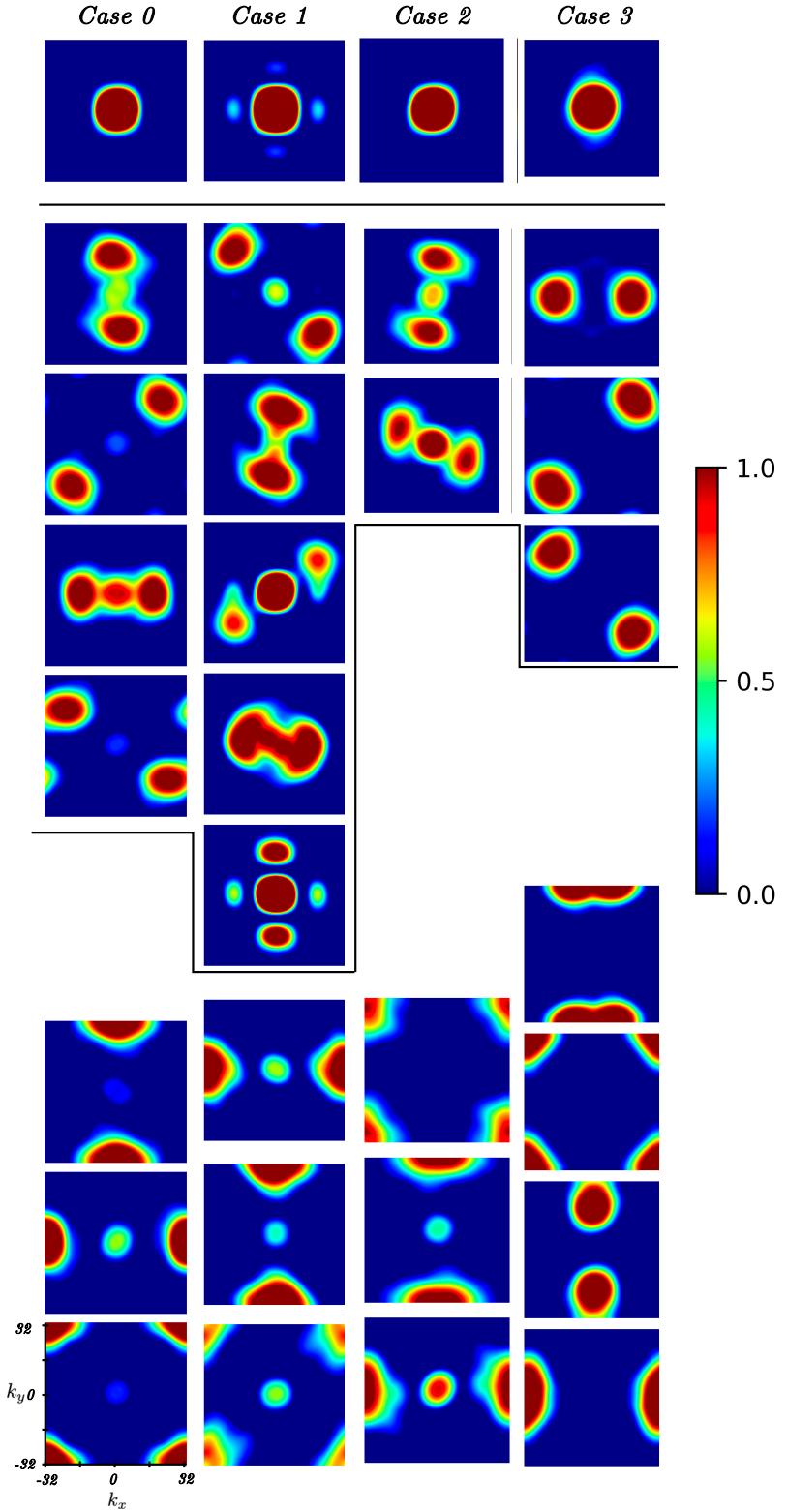
Figure 6: Cluster centers of filter spectra obtained by applying the k-means algorithm to the 64^2 padded weight matrices $\left|\widehat{\widetilde{W}}_\ell^{\beta,j}\right|$ from layer 2 of BNN^0 and TLNN^i. The number of cluster centers varies in each case as we increase the number of cluster centers until qualitatively similar patterns are observed
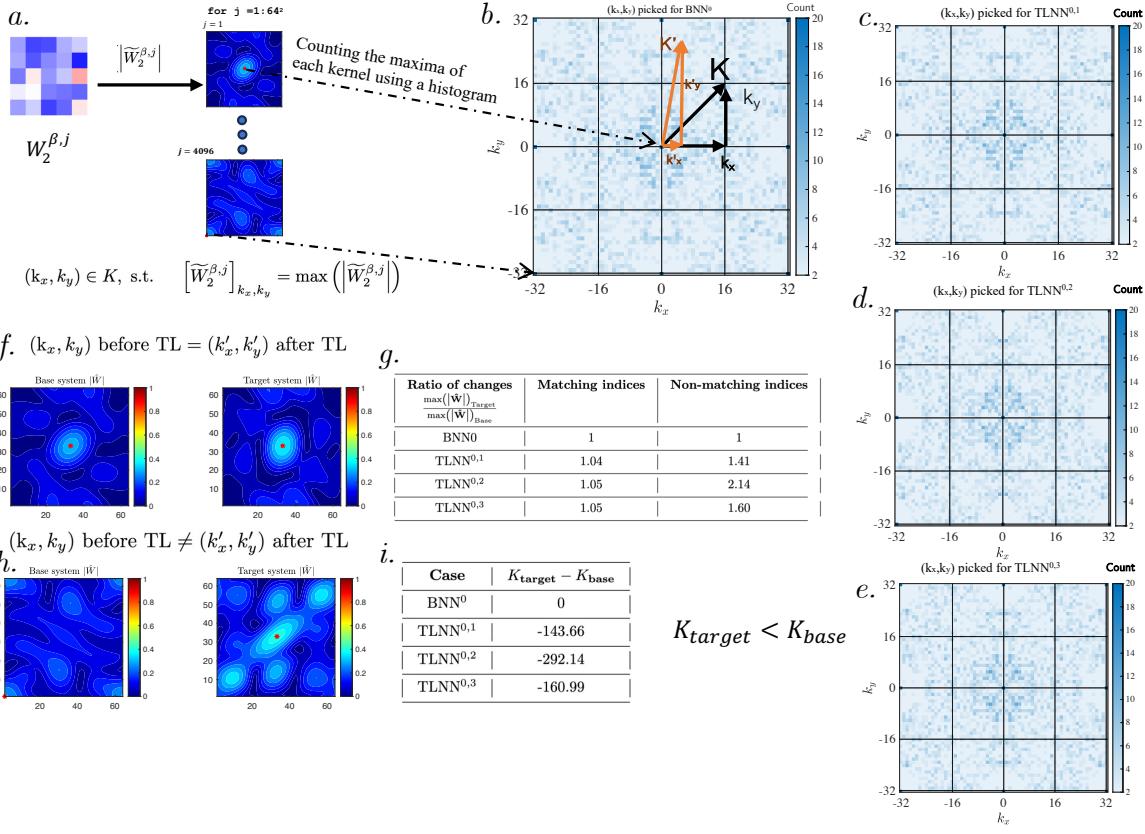
Figure 7: Schematic and analysis of kernel changes in spectral space. *Panel a*: Identification of the 64^2 wavenumber pairs corresponding to the maxima of the padded and Fourier-transformed layer-2 kernels $\left|\widehat{\widetilde{W}}_\ell^{\beta,j}\right|$. *Panels b–e*: Histograms of these wavenumber pairs for BNN^0 and TLNN^{0,i}. *Panel f*: Wavenumber pairs whose maxima locations do not change after TL. *Panel g*: Change in the mean amplitude ratio at the maxima. *Panel h*: Wavenumber pairs whose maxima locations shift after TL. *Panel i*: Comparison of change in radial wavenumber between base and target systems