Supporting Information for "Neural-network parameterization of subgrid momentum transport in the atmosphere"

Janni Yuval[1]and Paul A. O'Gorman[1]

[1]Department of Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Contents of this file

1. Text S1 to S2
2. Figures S1 to S9
3. Table S1

Here we describe the train and test datasets, training protocol, architecture of the neural networks and the implementation of the neural network (NN) parameterization in SAM (text S1), and the coarse-graining methods we use (text S2). We also add several figures to support our findings and a table showing the online performance of simulations we run (Figs. S1-S9 and Table S1).

Text S1. Training and implementation

The NN parameterization for energy and moisture variables is composed out of two different networks (Yuval et al., 2021). One NN (NN1) predicts the effect of subgrid vertical advection, sedimentation, microphysics and radiation on the moisture and energy variables, and a second NN (NN2) predicts the turbulent diffusivity and moisture and energy correction for surface fluxes. The NN parameterization for horizontal momentum variables (NN-MOM) is a separate NN in addition to NN1 and NN2, and its inputs and outputs are described in details in the manuscript. The training procedure for NN-MOM is overall similar to the training procedure done in YOH21 for training NN1 and NN2, but for completeness we repeat the description below including some small differences in the test dataset, number of epochs and learning rate. The training data for the NN momentum parameterization is obtained from 383.25 days of 3-hourly snapshots of model output taken from the hi-res simulation (overall 3066 snapshots). This data was split into train and test datasets, where the first 320.625 days (2565 snapshots) were used for training, and the last 62.625 days (501 snapshots) were used as a test dataset. For each 3-hourly snapshot that was used during training, we reduced the training data set size by randomly subsampling atmospheric columns at each latitude for each snapshot. When using a coarse-graining factor of x4 we randomly sub-sampled 15 (out of 144) atmospheric columns at each latitude and when using a coarse-graining factor of x8 we randomly sub-sampled 30 (out of 72) atmospheric columns at each latitude. This subsampling of the training data enables uploading all training data into the RAM during training. This results in training datasets size of $13,856,040$, where a sample is defined as an individual

atmospheric column for a given horizontal location and time step. To get better statistics for the calculation of the offline results, we did not subsample the test data when using a coarse-graining factor of x8, and the test data size is $6,492,960$ samples. When using a coarse-graining factor of x4, the test set was randomly sub-sampled (using 15 out of 144 longitudes at each latitude), which allowed us to easily to upload the whole test set to RAM, resulting in a test set of $2,705,400$ samples. This x4 test set was used to verify we do not overfit, and we do not present results or plots that rely on this test set.

The NN training is implemented in Python using PyTorch (Paszke et al., 2017). The weights and biases are optimized by the Adam optimizer (Kingma & Ba, 2014) combined with a cyclic learning rate (Smith, 2017). We use 1024 samples in each batch and train over 8000 batches before completing a full cycle in the learning rate. We use 10 epochs, where the first epoch is trained with a minimal learning rate of 0.0002 and a maximal learning rate of 0.002, and in the next five epochs both the minimal learning rate and maximal learning rate are reduced by 10% at each epoch. The last four epochs are trained after reducing both the minimum and maximum learning rates by a factor of 10 (giving a maximal learning rate of 0.000118). The NNs are stored as netcdf files, and then implemented in SAM using a Fortran module. The results presented in this work are for NNs with 128 nodes at each hidden layer and rectified linear unit activations (ReLu) except in the output layer where no activation function was used. NNs have five densely connected layers.

Prior to training, each input (feature) of the NN momentum parameterization and the outputs were standardized by removing the mean and rescaling to unit variance. To

standardize the subgrid momentum fluxes (and diffusivity) other than the surface fluxes, we calculated the mean and variance for standardization across 47 (15) vertical levels.

We do not present results for the NN momentum parameterization at a coarser grid spacing of 192km (corresponding to coarse graining factor of 16) since we find that the exact structure and width of the ITCZ at this grid spacing is sensitive to the structure of the NN parameterization of moisture and energy, which makes it difficult to choose a baseline simulation to compare against. This sensitivity is possibly related to results shown in a previous study that found that in an aquaplanet configuration with hemispherically symmetric SST the ITCZ structure is very sensitive to the exact parameterization (Möbis & Stevens, 2012).

Text S2. Coarse-graining on a collocated grid and on a staggered C-grid

As explained in the manuscript, we use two different coarse graining protocols: (a) coarse graining on a collocated grid and (b) coarse graining on a staggered C-grid. Coarse graining on the collocated grid is used for the offline results presented in the paper, while coarse graining on the staggered grid is needed for learning the parameterization that is actually used online in SAM which uses a staggered C-grid.

In order to get coarse-grained variables on a collocated grid, each variable is coarse-grained slightly differently, depending on which grid it is found on in hi-res (Figure S1). Specifically, quantities that are found on the horizontal u grid (i.e., u, vertical fluxes of zonal momentum, zonal momentum surface fluxes) are coarse grained using (red circles

in Figure S1):

$$\begin{aligned}\overline{A}(i,j,k) \;=\; \frac{1}{N^2}[& \sum_{l=N(i-1)+2}^{l=Ni} \sum_{m=N(j-1)+1}^{m=Nj} A(l,m,k) \\ & +\frac{1}{2} \sum_{m=N(j-1)+1}^{m=Nj} A(N(i-1)+1,m,k) \\ & +\frac{1}{2} \sum_{m=N(j-1)+1}^{m=Nj} A(Ni+1,m,k)],\end{aligned} \tag{1}$$

where A is the high-resolution variable, $\overline{A}$ is the coarse-grained variable, N is the coarse graining factor, k is the index of the vertical level, and i, j (l, m) are the discrete indices of the longitudinal and latitudinal coordinates at coarse resolution (high resolution). Similarly quantities that are found on the horizontal v grid (i.e., v, vertical fluxes of meridional momentum, meridional momentum surface fluxes) are coarse grained using (red triangles in Figure S1):

$$\begin{aligned}\overline{A}(i,j,k) \;=\; \frac{1}{N^2}[& \sum_{l=N(i-1)+1}^{l=Ni} \sum_{m=N(j-1)+2}^{m=Nj} A(l,m,k) \\ & +\frac{1}{2} \sum_{l=N(i-1)+1}^{l=Ni} A(l,N(j-1)+1,k) \\ & +\frac{1}{2} \sum_{l=N(i-1)+1}^{l=Ni} A(l,Nj+1,k)],\end{aligned} \tag{2}$$

and quantities that are found on the horizontal w grid (e.g., w, H_{L}, q_{T}) are coarse grained using (red stars in Figure S1):

$$\overline{A}(i,j,k) = \frac{1}{N^2} \sum_{l=N(i-1)+1}^{l=Ni} \sum_{m=N(j-1)+1}^{m=Nj} A(l,m,k). \tag{3}$$

Likewise, in order to get coarse-grained variables on a staggered C-grid, each variable is coarse grained slightly differently, depending on which grid it is found on in hi-res (Figure S2). To obtain coarse-grained variables on a C-grid, the quantities that are found on the horizontal u grid are coarse-grained using (red circles in Figure S2):

$$\begin{aligned}\overline{A}(i,j,k) \;=\; \frac{1}{N^2}[& \sum_{l=N(i-1.5)+2}^{l=N(i-0.5)} \sum_{m=N(j-1)+1}^{m=Nj} A(l,m,k) \\ & +\frac{1}{2} \sum_{m=N(j-1)+1}^{m=Nj} A(N(i-1.5)+1,m,k) \\ & +\frac{1}{2} \sum_{m=N(j-1)+1}^{m=Nj} A(N(i-0.5)+1,m,k)],\end{aligned} \tag{4}$$

the quantities that are found on the horizontal v grid are coarse-grained using (red triangles in Figure S2):

$$\begin{aligned}\overline{A}(i,j,k) \;=\; \frac{1}{N^2}[& \sum_{l=N(i-1)+1}^{l=Ni} \sum_{m=N(j-1.5)+2}^{m=N(j-0.5)} A(l,m,k) \\ & +\frac{1}{2} \sum_{l=N(i-1)+1}^{l=Ni} A(l,N(j-1.5)+1,k) \\ & +\frac{1}{2} \sum_{l=N(i-1)+1}^{l=Ni} A(l,N(j-0.5)+1,k)],\end{aligned} \tag{5}$$

and the quantities that are found on the horizontal w grid are coarse-grained using equation 3 (red stars in Figure S2). Note that N is assumed to be even.

The two different coarse-graining protocols we use lead to a similar mean subgrid fluxes (not shown), but calculating the subgrid fluxes using a C-grid leads to a much larger variability in the fluxes because of the spatial staggering of the coarse-grained vertical

momentum fluxes and the resolved (coarse-grained) w, and this effect is particularly large in the extratropical jet regions (Figure S9).

We note that different choices of coarse graining protocols require modifications of some of the numerical schemes when calculating subgrid terms. For example, when using variables on a staggered C-grid, both resolved flux and the flux in hi-res due to vertical advection of zonal momentum are calculated as:

$$(u)_{\text{adv}}^{\text{subg}-\text{flux}}(i,j,k) = \frac{\rho_w(k)}{4}(w(i,j,k) + w(i-1,j,k))(u(i,j,k) + u(i,j,k-1)). \quad (6)$$

where ρ_w is the reference density profiles defined on the w grid, which are staggered vertically, and i, j, k are the discrete indices of the longitudinal, latitudinal and vertical coordinates, respectively. In contrast, when using a (horizontal) collocated grid, the resolved flux due to vertical advection of zonal momentum is calculated as:

$$(u)_{\text{adv}}^{\text{subg}-\text{flux}}(i,j,k) = \frac{\rho_w(k)}{2}w(i,j,k)(u(i,j,k) + u(i,j,k-1)), \quad (7)$$

while the hi-res flux is calculated using equation 6.

References

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

Möbis, B., & Stevens, B. (2012). Factors controlling the position of the Intertropical Convergence Zone on an aquaplanet. *Journal of Advances in Modeling Earth Systems*, *4*, M00A04.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic differentiation in pytorch. In *Neural information processing sys-*

tems workshop, 2017.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)* (pp. 464–472).

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021). Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letters*, *48*(6), e2020GL091363.

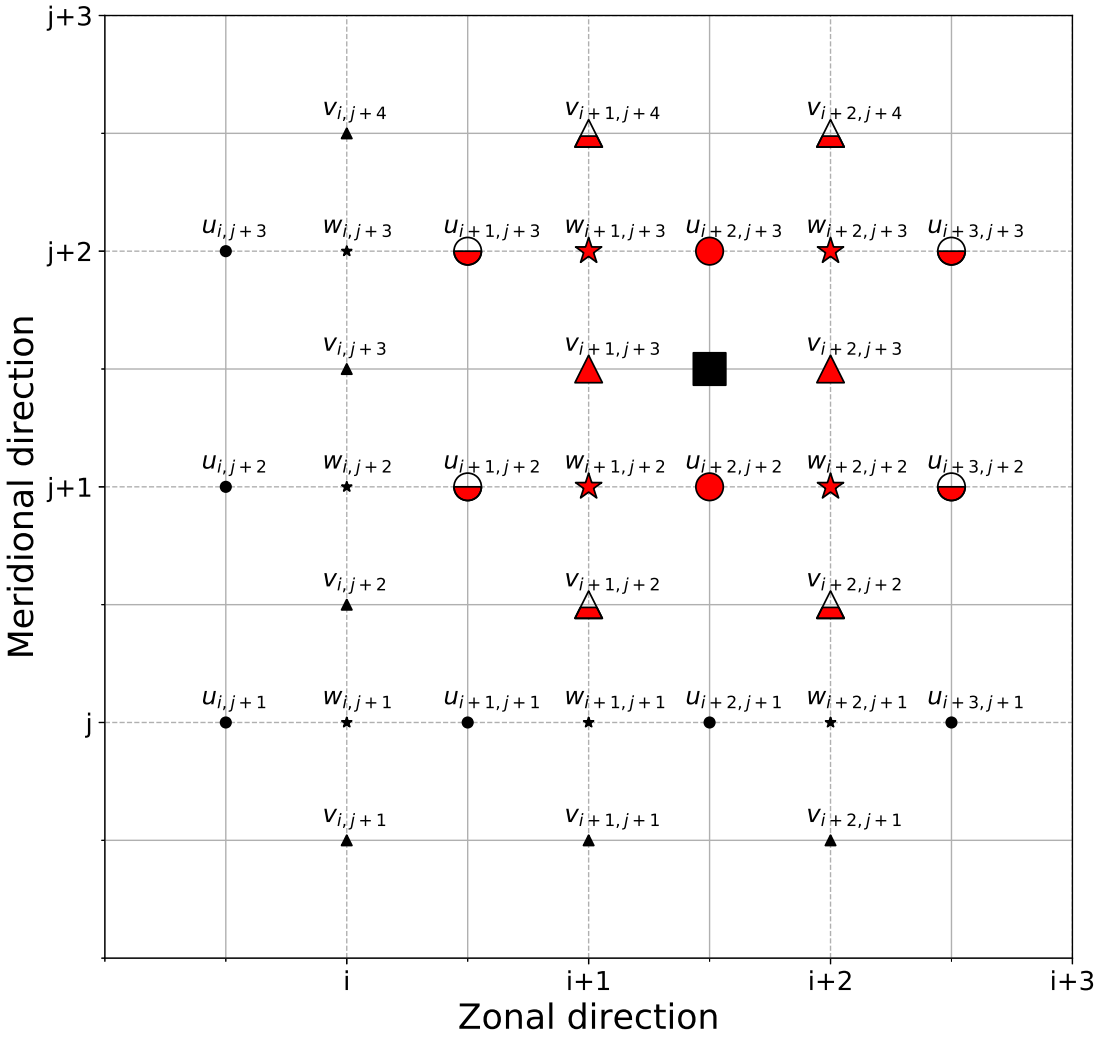

Figure S1. Illustration of the C-grid used in SAM and the coarse graining procedure done to achieve a **collocated grid** for the coarse-grained variables using a coarse-graining factor of 2 for simplicity. Circles, triangles and stars represent the high-resolution grids of u, v, and w, respectively. Red symbols show the grid points that are averaged for the coarse graining procedure that gives a point on the collocated coarse grid represented by the square. Symbols that are half-filled are weighted with a factor of 0.5 when coarse graining (see equations 1 and 2). Note that the square shows the collocated coarse-grained grid for u, v, and w.

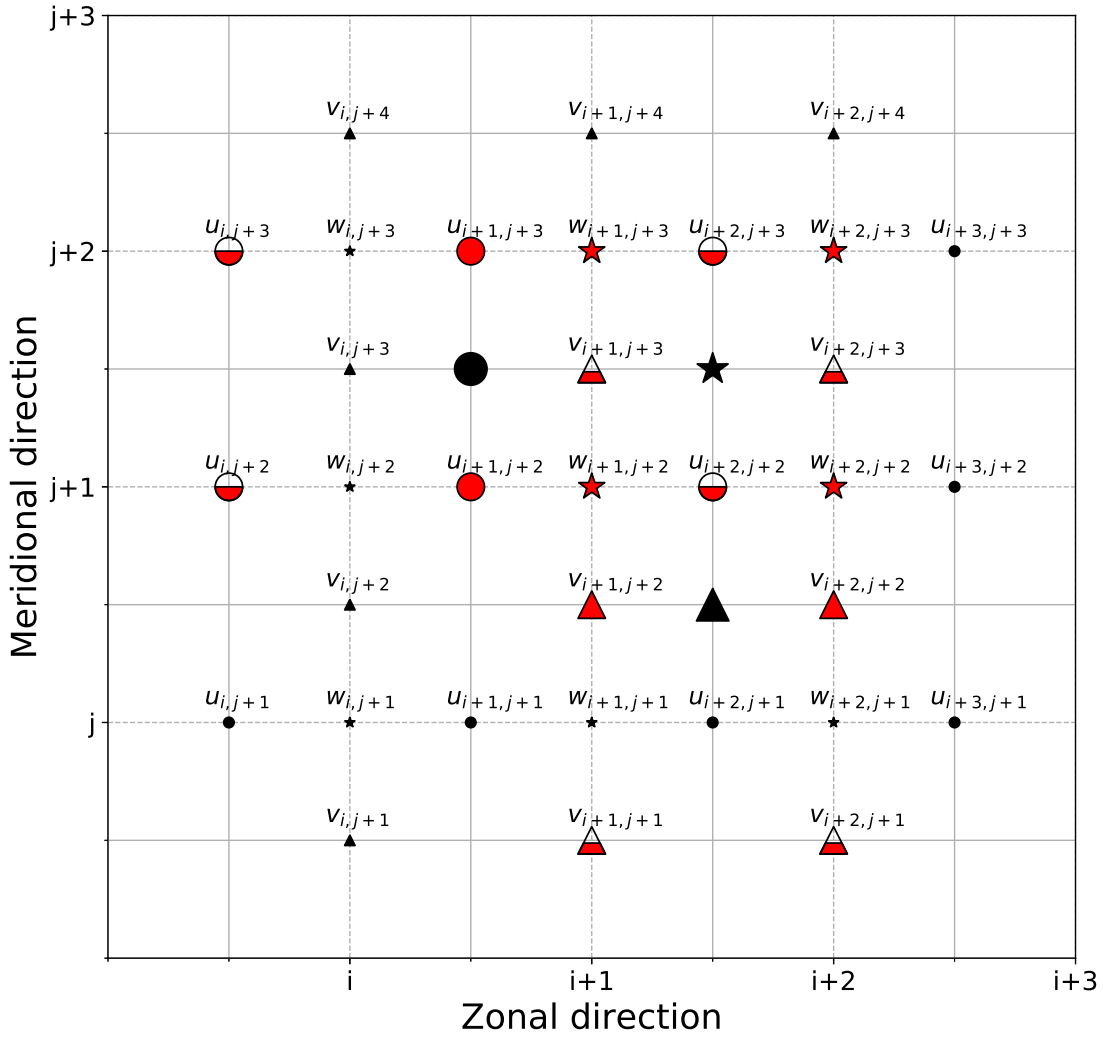

Figure S2. Illustration of the C-grid used in SAM and the coarse graining procedure done to achieve a **C-grid** for the coarse-grained variables using a coarse-graining factor of 2 for simplicity. Red and smaller black circles, triangles and stars represent the high-resolution grids of u, v, and w, respectively. Red symbols show grid points that are averaged for the coarse graining procedure that gives a coarse grid represented by the larger black circle, triangle and star (representing grid points in the coarse C-grid of u, v, and w, respectively). Symbols that are half-filled are weighted with a factor of 0.5 when coarse graining (see equations 4 and 5).

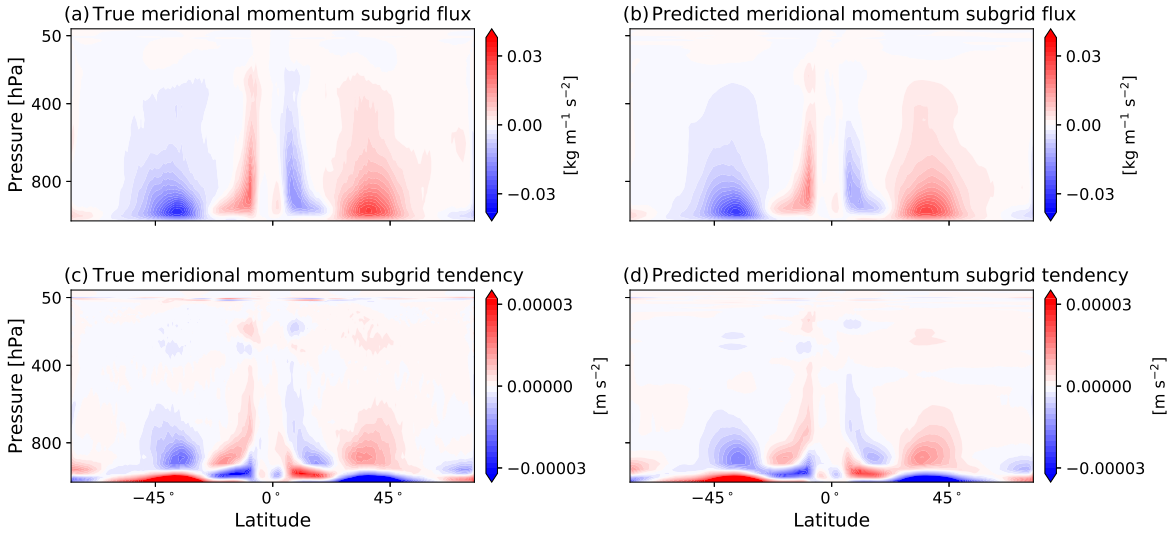

Figure S3. The time- and zonal-mean (a) true (calculated from hi-res) and (b) predicted (NN-MOM predictions) meridional momentum fluxes due to subgrid vertical advection and the associated time- and zonal-mean (c) true and (d) predicted meridional wind tendencies. Colors are saturated to highlight fluxes and tendencies outside of the boundary layer. The sponge layer is active above 50hPa. All quantities are calculated from 501 snapshots of the x8 test data set.

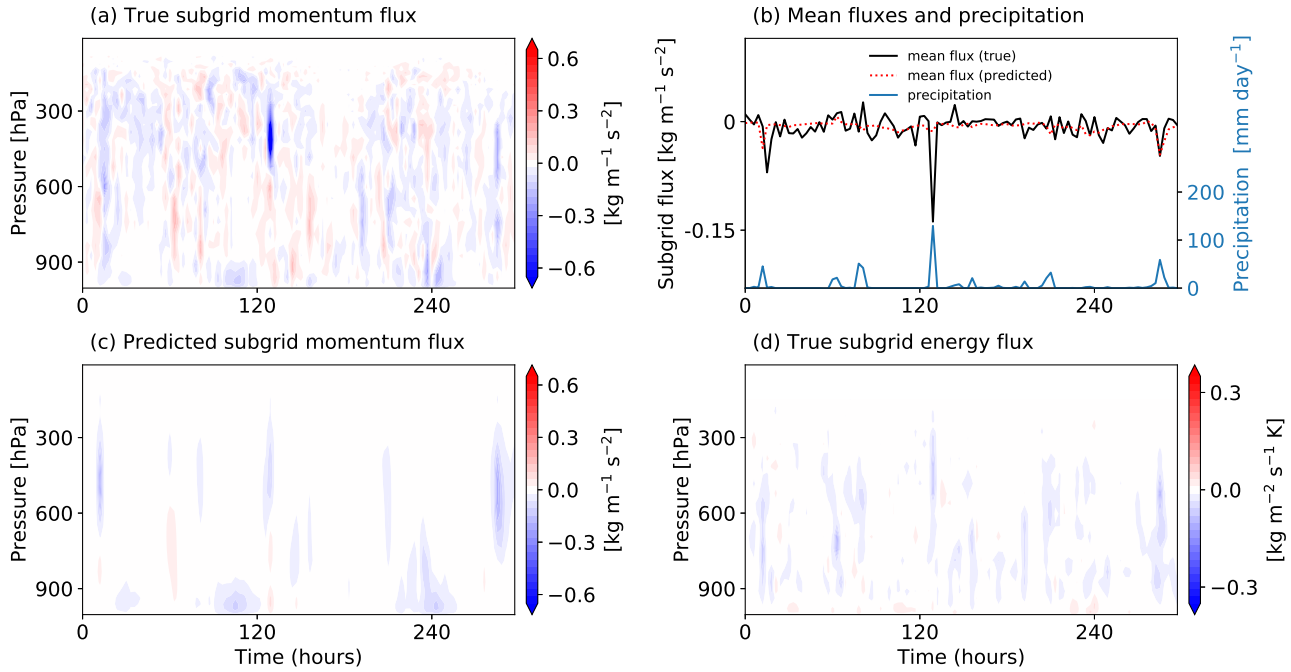

Figure S4. Time series of subgrid fluxes due to vertical advection for an extratropical column at latitude 35.3° and a coarse-graining factor of 8: (a) true zonal momentum flux, (b) vertical-mean true (black) and NN-MOM predicted (dotted red) zonal momentum flux, (c) NN-MOM predicted zonal momentum flux, (d) true subgrid energy flux rescaled by the specific heat capacity. Panel (c) also shows the surface precipitation (blue) as a function of time. Time zero is taken to be the beginning of the presented time series which occurs in the statistical-equilibrium phase of the hi-res simulation. The scale of the colorbars is chosen to be the same to Figure 2 in the main paper.

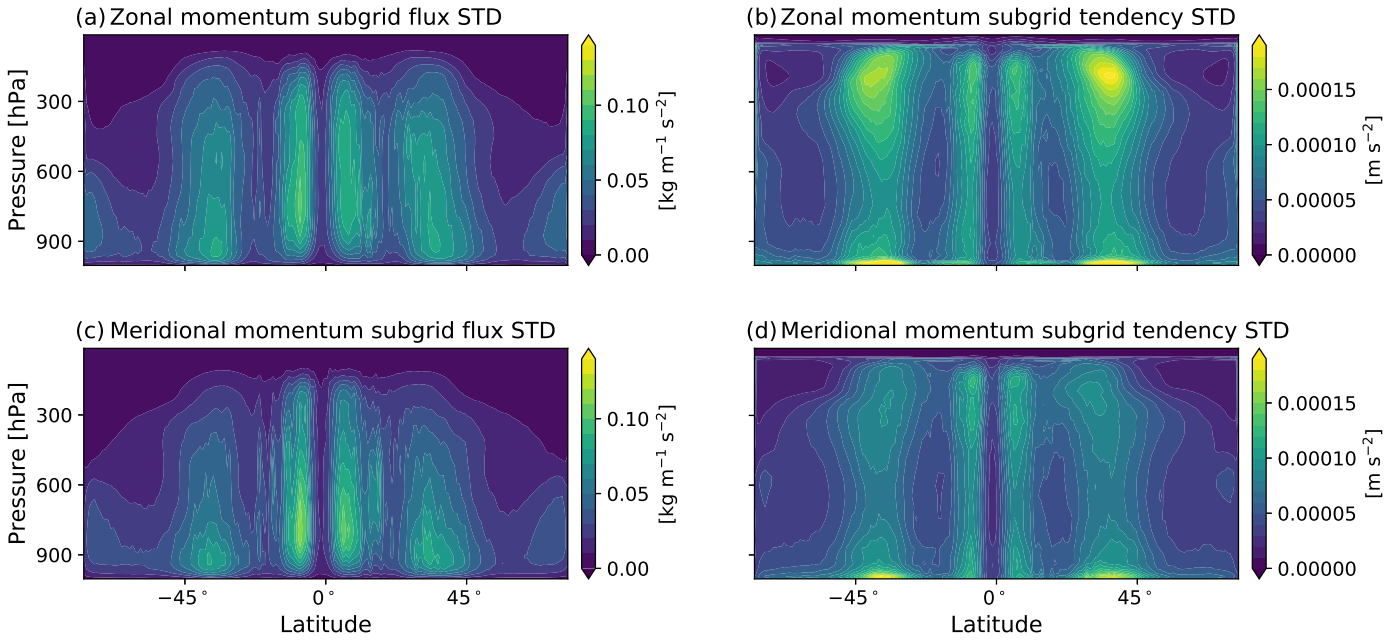

Figure S5. The standard deviation of subgrid terms due to vertical advection as a function of pressure and latitude when using coarse graining on a **collocated grid** for (a) zonal momentum flux, (b) zonal wind tendency, (c) meridional momentum flux, and (d) meridional wind tendency, as calculated from 501 snapshots of x8 test data set.

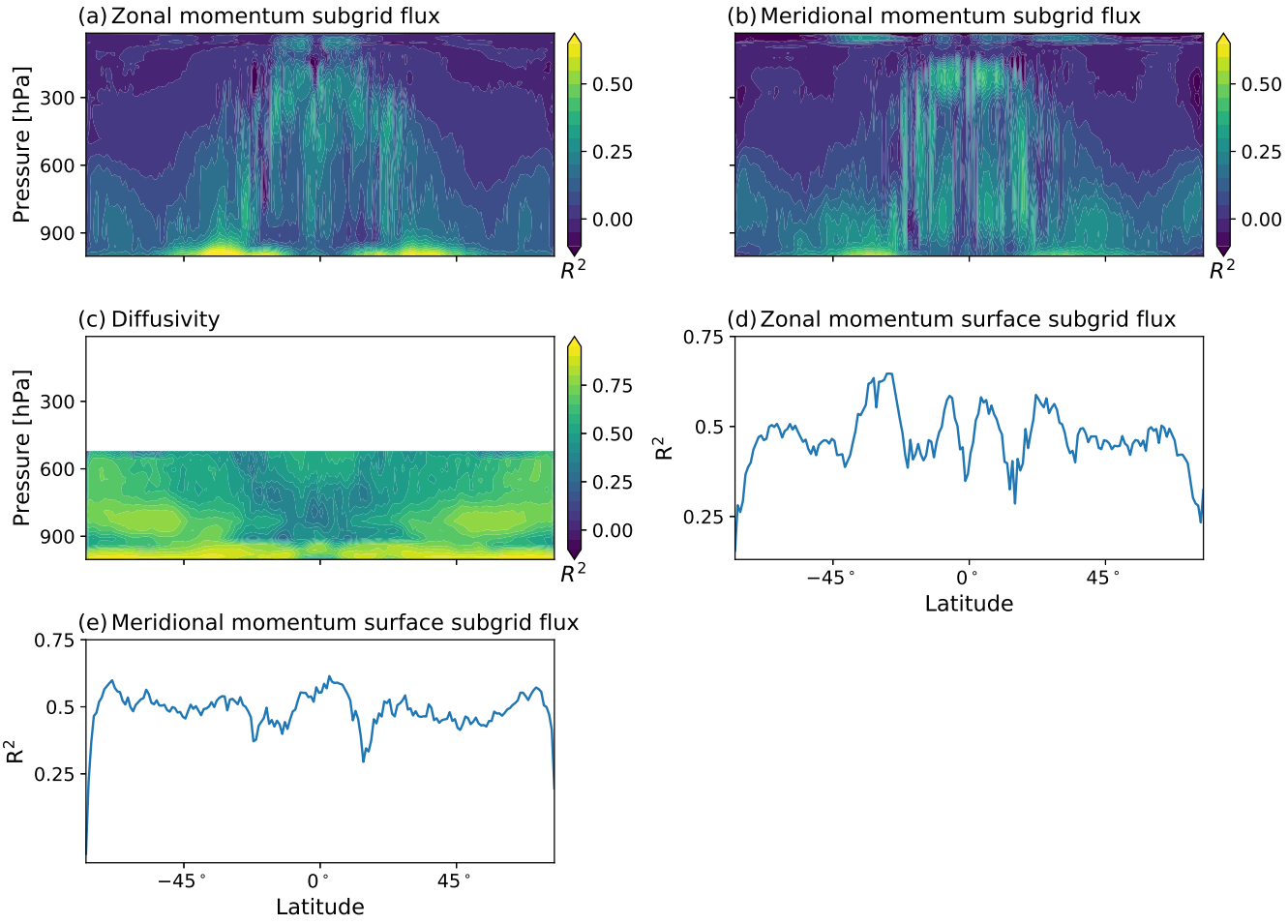

Figure S6. The coefficient of determination (R^2) for offline performance of NN-MOM for the: (a) subgrid zonal momentum fluxes due to vertical advecton (b) subgrid meridional momentum fluxes due to vertical advecton and (c) turbulent diffusivity as a function of pressure and latitude, and of (d) subgrid surface zonal momentum flux and (e) subgrid surface meridional momentum flux as a function of latitude. The coefficient of determination is calculated from 501 snapshots of x8 test data set.

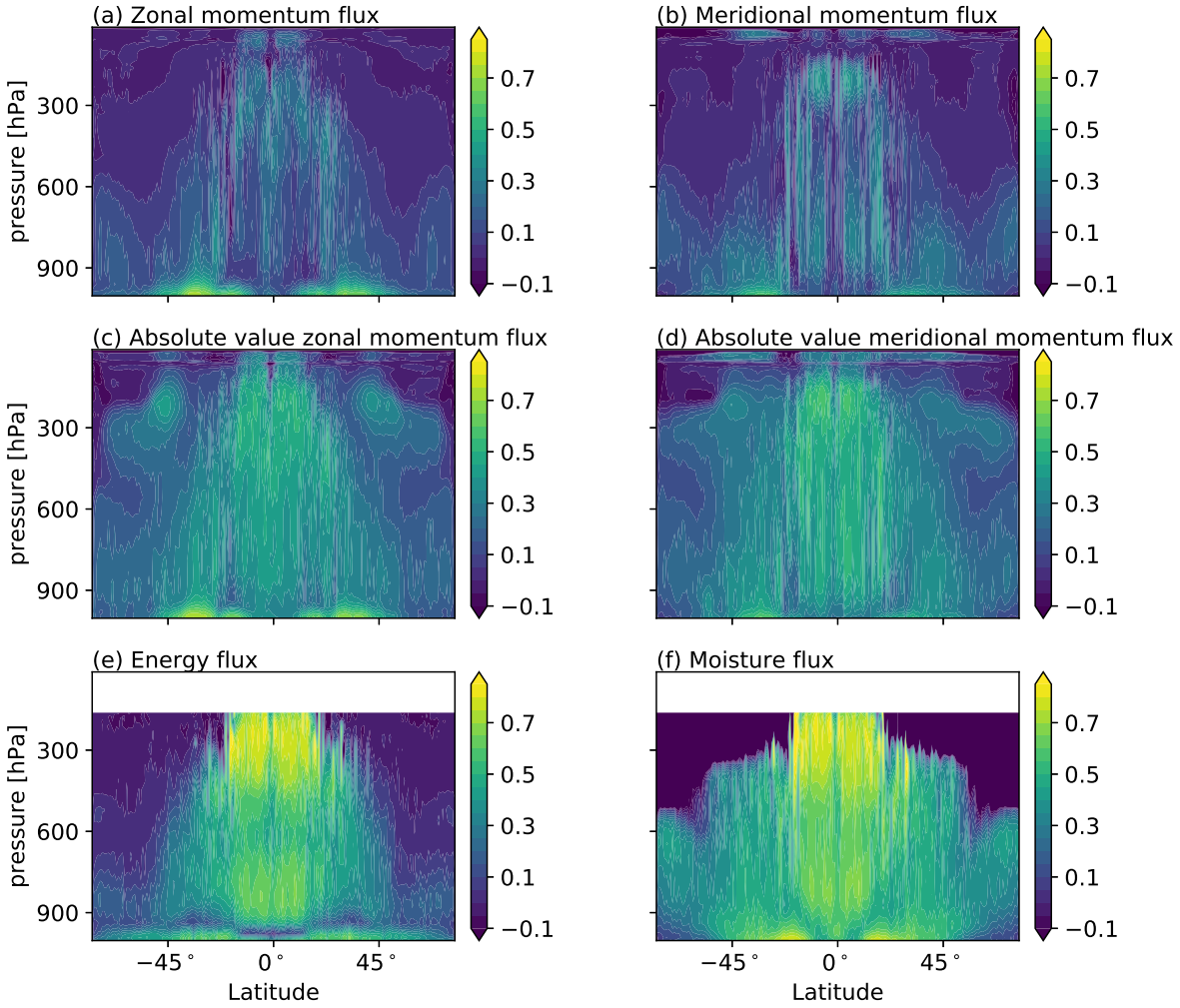

Figure S7. Comparing offline performance (defined as the coefficient of determination) at x8 of NN-MOM that predicts the subgrid fluxes of (a) zonal and (b) meridional momentum, of an NN that was trained to predict the absolute value of subgrid fluxes (NN-MOM-ABS) of (c) zonal and (d) meridional momentum, and of the NN that predicts the subgrid fluxes of (e) energy (H_{L}) and (f) non-precipitating water mixing ratio. The NN parameterization that predicts the subgrid energy and moisture fluxes is described in detail in YOH21.

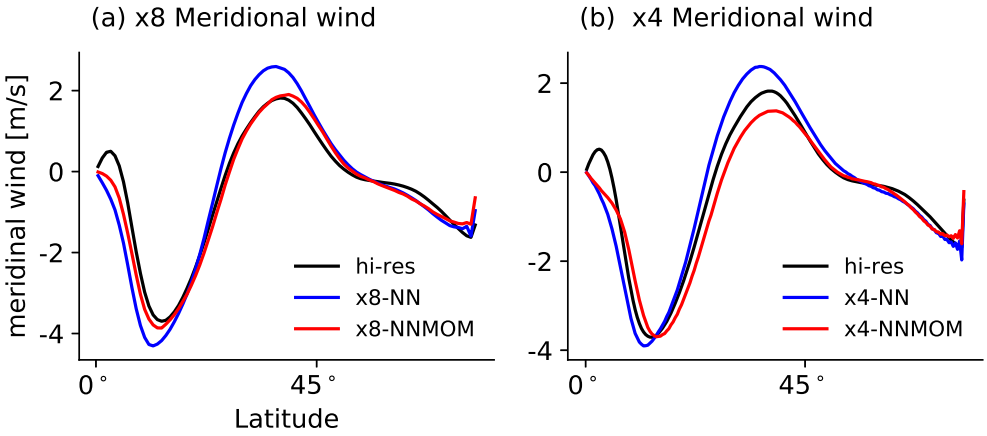

Figure S8. The zonal- and time-mean meridional surface wind as a function of latitude for simulations with coarser grids by factors of (a) 8 and (b) 4 compared to hi-res. Results are shown for simulations with NN parameterization for thermodynamic and moisture variables (blue), simulations with NN parameterization for thermodynamic, moisture and momentum variables (red) and hires (black; coarsened to x8 and x4, respectively).

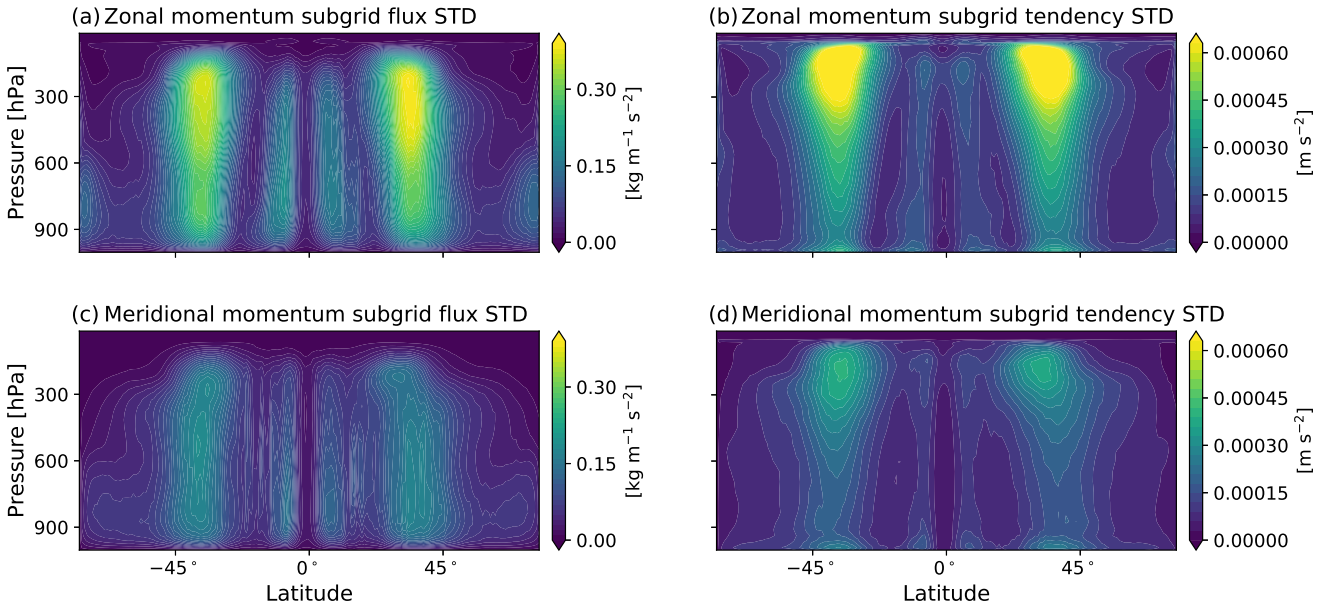

Figure S9. The standard deviation of subgrid terms due to vertical advection as a function of pressure and latitude when using coarse graining on a **staggered C-grid** for (a) zonal momentum flux, (b) zonal wind tendency, (c) meridional momentum flux, and (d) meridional wind tendency, as calculated from 501 snapshots of x8 test data set.

	RMSE relative to hi-res					
	x8	x8-NN	x8-NNMOM	x4	x4-NN	x4-NNMOM
Zonal wind [m s^{-1}]	4.51	3.21	2.71	4.48	2.70	2.85
Meridional wind [m s^{-1}]	0.63	0.25	0.17	0.66	0.21	0.23
Vertical wind [cm s^{-1}]	0.383	0.079	0.057	0.402	0.067	0.123
EKE [m^2 s^{-2}]	42.89	34.91	34.70	41.70	28.22	34.23
Precipitation [mm day^{-1}]	4.69	0.71	0.44	4.80	0.39	1.13
Streamfunction [kg m^{-1} s^{-1} $\times 10^2$]	9.69	2.17	1.36	9.87	2.06	3.20

Table S1. Online performance as measured by the root mean square error (RMSE) of the time- and zonal-mean zonal wind, meridional wind, vertical wind, eddy kinetic energy, precipitation, and the mass streamfunction. The RMSE is calculated relative to hi-res for the coarse-resolution simulations with no ML parameterization (x4, x8), the simulations with the NN parameterization only for thermodynamic and moisture variables (x4-NN, x8-NN) and for the simulations with NN parameterization for thermodynamic, moisture and horizontal momentum variables (x4-NNMOM, x8-NNMOM). The mass streamfunction is defined here as $1/g \int_p^{p_s} [v] dp$, where g is the gravitational acceleration, p_s is the pressure surface and $[v]$ is the zonal- and time-mean meridional velocity. The eddy kinetic energy is defined with respect to the zonal and time mean.