

A comparison of data-driven approaches to build low-dimensional ocean models

Niraj Agarwal¹, Dmitri Kondrashov², Peter Dueben³,
Eugene Ryzhov¹, Pavel Berloff¹

¹ Imperial College London

² University of California, Los Angeles

³ European Centre for Medium-Range Weather Forecasts (ECMWF)

Contents

- Aim
- Dataset
- Models
 - Linear Regression
 - Stochastic
 - Deep learning
- Assessment metrics
- Results
- Conclusions and possible extensions

Aim:

- Build low-dimensional, low-cost, statistical ocean models -- capable of reproducing the essential statistical characteristics of the oceans and provide correct forecasts on both short and long timescales.

Such emulators will be beneficial in, e.g., ocean-atmosphere coupled systems, where the ocean component takes more than 50% of the running time.

- We consider the most relevant statistical methods and diagnose their accuracy, stability, and computational cost.
- Inter-compare them to identify the best approach for robust and realistic upper ocean emulation.

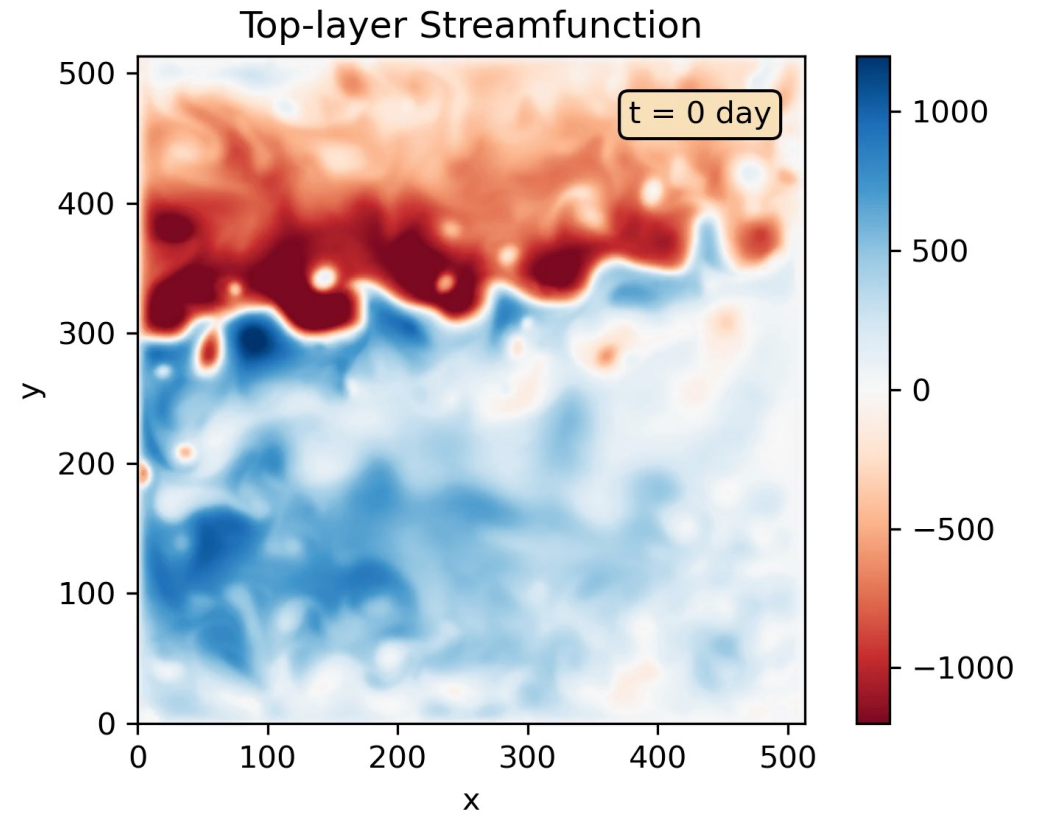
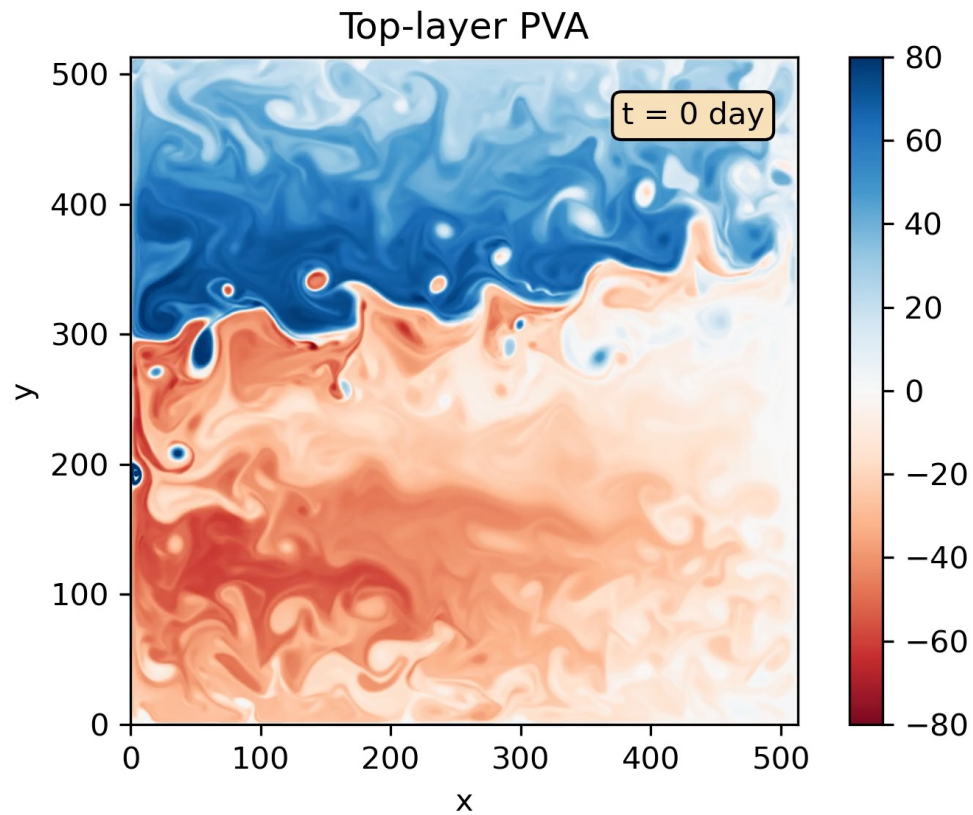
Dataset: Double-gyre model

- Mid-latitude ocean circulation model in North Atlantic (Berloff, 2015).
- Dynamics governed by three-layer quasi-geostrophic potential vorticity anomaly (PVA) equations:

$$\begin{aligned}\frac{\partial q_1}{\partial t} + J(\psi_1, q_1) + \beta \frac{\partial \psi_1}{\partial x} &= \frac{1}{\rho_1 H_1} W + \nu \nabla^4 \psi_1 & q_1 &= \nabla^2 \psi_1 + S_1(\psi_2 - \psi_1), \\ \frac{\partial q_2}{\partial t} + J(\psi_2, q_2) + \beta \frac{\partial \psi_2}{\partial x} &= \nu \nabla^4 \psi_2 & q_2 &= \nabla^2 \psi_2 + S_{21}(\psi_1 - \psi_2) + S_{22}(\psi_3 - \psi_2), \\ \frac{\partial q_3}{\partial t} + J(\psi_3, q_3) + \beta \frac{\partial \psi_3}{\partial x} &= -\gamma \nabla^2 \psi_3 + \nu \nabla^4 \psi_3 & q_3 &= \nabla^2 \psi_3 + S_3(\psi_2 - \psi_3),\end{aligned}$$

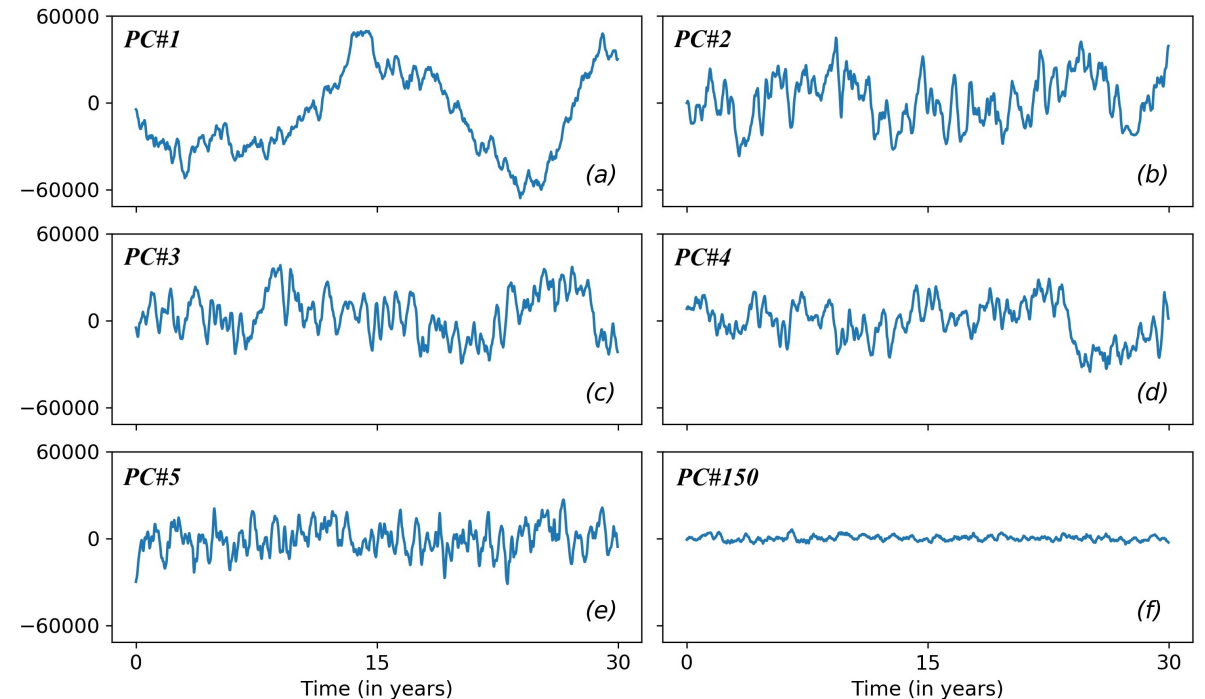
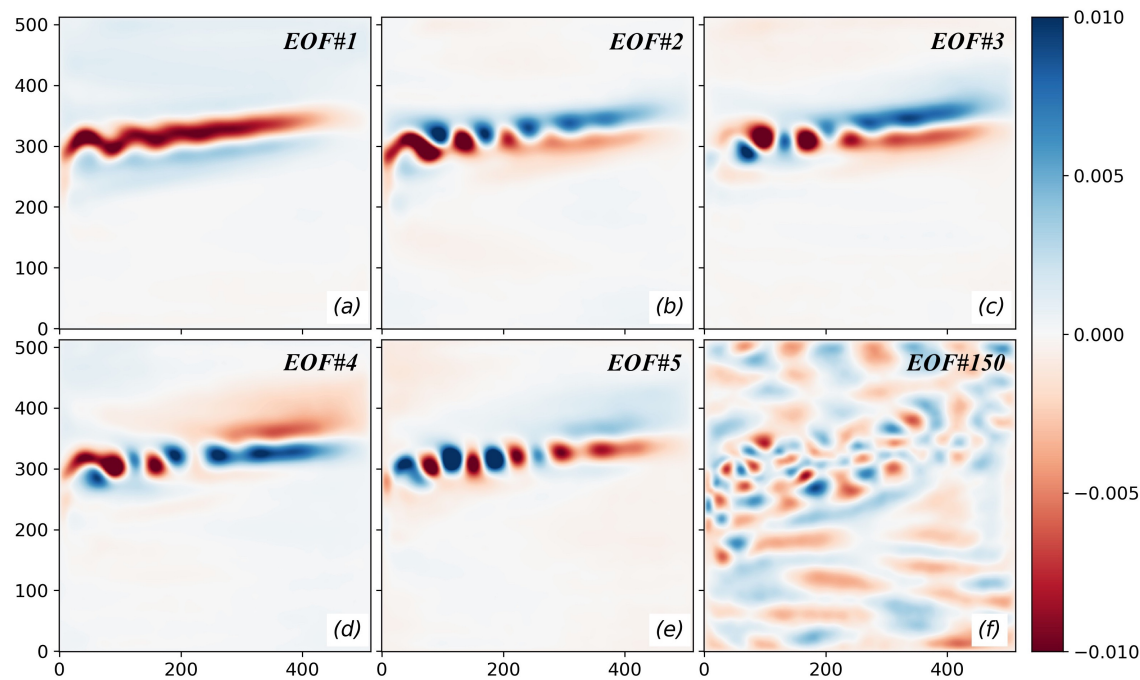
- Solved to obtain PVA.
- Inverted to get the geostrophic streamfunction (ψ).
- Snapshots are saved after every 10 days for a total of 500,000 days (≈ 1370 years)

Upper ocean snapshots:



Dataset continued...

- Performed an EOF decomposition of the upper-ocean streamfunction field and considered the leading 150 EOFs/PCs (out of 50,000).
- 150 EOFs/PCs represent 90% of the total variance.



Workframe:



- Model the time series of the PCs ($y_i, i = 1, 2, 3, \dots, 150$).
- Obtain the forecasts and project them back to the physical space by multiplying with the EOFs.
- Use the reconstructed physical fields and the modelled PCs to examine the statistical skills.

Models: Linear Regression (LR)

- Baseline model

$$\frac{d\mathbf{y}}{dt} = \mathbf{L}\mathbf{y}$$

where, $\mathbf{y} = [y_1, y_2, \dots, y_N]$ is the multivariate PC ($N = 150$ here)

$\mathbf{L}$ is $N \times N$ matrix of regression coefficients.

LR + Additive Noise (LR-AN)

- LR model additionally equipped with additive noise : White and Red.
- Additive White Noise (LR-AWN):

$$d\mathbf{y} = \mathbf{L}\mathbf{y}dt + \mathbf{Q}d\mathbf{W};$$

where, $\mathbf{W} = [W_1 \ W_2 \ W_3 \dots W_N]$ is a multivariate Wiener process
with $d\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \sigma^2)$

σ : standard deviation of LR residuals $\mathbf{r} = d\mathbf{y}/dt - \mathbf{L}\mathbf{y}$

$\mathbf{Q}$: Lower triangular matrix such that $\mathbf{Q}\mathbf{Q}^T = \mathbf{C}$

$$\mathbf{C} = \text{Cov}([r_1 \ r_2 \ r_3 \ \dots \ r_N])$$

$$\mathbf{Q}, \mathbf{C} \in \mathbb{R}^{N \times N}$$

LR + Additive Noise (LR-AN)

- Additive Red Noise (LR-ARN):
- Red noise (η) modeled as an AR(1) process to give the coupled system:

$$\begin{aligned}\frac{d\mathbf{y}}{dt} &= \mathbf{L}\mathbf{y} + \mathbf{Q}\eta; \\ \frac{d\eta}{dt} &= \frac{-1}{\tau}\eta + \zeta;\end{aligned}$$

where, $\eta = [\eta_1, \eta_2, \eta_3, \dots, \eta_N]$ is the multivariate red noise.

τ is the decorrelation time scale of $\mathbf{y}$, $\zeta \sim \mathcal{N}(\mathbf{0}, \sigma^2)$, $\sigma^2 = \frac{\mathbb{E}[\mathbf{r}^2](1 - \mathbf{c}^2)}{(\Delta t)^2}$

$$\mathbb{E}[\mathbf{r}^2] = \text{Var}(\mathbf{r}), \mathbf{r} = \frac{d\mathbf{y}}{dt} - \mathbf{L}\mathbf{y}, \mathbf{c} = 1 - \frac{\Delta t}{\tau}$$

Multi-level LR (ML-LR)

- Regression residuals are not immediately replaced by noise but are modeled on a stack of levels using the state variables and the residuals.

$$\text{Level 1: } dy = \mathbf{L}ydt + \mathbf{r}(t)dt,$$

$$\text{Level 2: } d\mathbf{r} = \mathbf{M}^{(1)}[\mathbf{y}; \mathbf{r}]dt + \mathbf{r}^{(1)}(t)dt,$$

$$\text{Level 3: } d\mathbf{r}^{(1)} = \mathbf{M}^{(2)}[\mathbf{y}; \mathbf{r}; \mathbf{r}^{(1)}]dt + \mathbf{r}^{(2)}(t)dt,$$

.....

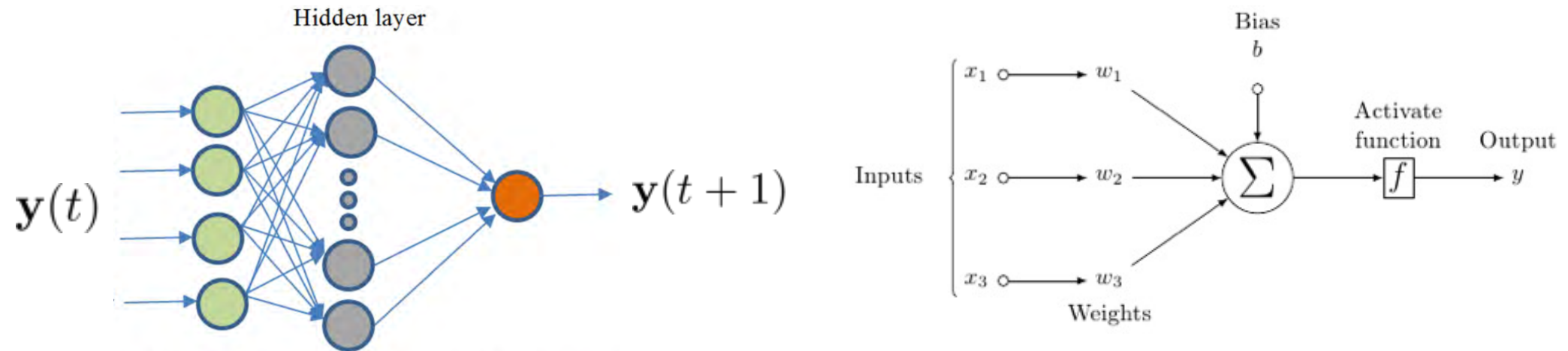
$$\text{Level L: } d\mathbf{r}^{(L-2)} = \mathbf{M}^{(L-1)}[\mathbf{y}; \mathbf{r}; \mathbf{r}^{(1)}; \dots; \mathbf{r}^{(L-2)}]dt + \mathbf{Q}d\mathbf{W},$$

 Up to here in our model

where, $d\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ and $\mathbf{Q}$ is the same as in LR-AN.

Artificial Neural Network (ANN)

- Feed-forward ANN - Deep learning family of model.



- 3 hidden layers, 100 neurons each, tanh activation, MAE as the loss function
- ANN-AWN: $y(t+1) = ANN(y(t)) + \zeta$; where $\zeta \sim \mathcal{N}(0, Q^T Q)$,
- Q is obtained from the Cholesky decomposition of $r(t) = y(t+1) - ANN(y(t))$.

Long Short Term Memory (LSTM)

- Recursive NN (RNN) class of deep learning methods.
- Passes the information explicitly between successive time steps, and thus accounts for long term correlations (i.e., memory effects) in the model states.
- For this reason, LSTMs are preferred for modelling time series.
- Configuration: 3 hidden layers, densely connected,
50 hidden units per layer, “Adam” optimizer, MAE as the loss
- Expression: $\mathbf{y}(t + 1) = LSTM(\mathbf{y}(t))$
- LSTM-AWN: $\mathbf{y}(t + 1) = LSTM(\mathbf{y}(t)) + \zeta$, where ζ = spatially correlated white noise (same as ANN)

LR + Deep Learning = Hybrid model

- LR : provides the linear dynamical core,
- Deep Learning models : provides non-linear corrections + memory effects
- Hybrid models may potentially preserve some core dynamics and benefit from simpler algorithm and architectures.
- If f : Deep learning model, then

$$\begin{aligned} dy(t) &= Ly(t)dt + r(t)dt, \\ r(t+1) &= f(r(t), y(t)) \end{aligned}$$

LR-ANN, LR-LSTM

$$\begin{aligned} dy(t) &= Ly(t)dt + r(t)dt, \\ r(t+1) &= f(r(t), y(t)) + \xi, \quad \xi \sim \mathcal{N}(0, Q^T Q). \end{aligned}$$

LR-ANN-AWN, LR-LSTM-AWN

Training and Testing

- Length of the PCs : 50,000 records
- Training length : 40,000 records
- Test dataset length : 10,000 records
- Forecast length:
 - Short timescale: 10 records (i.e., 100 days)
 - Long timescale: 20,000 records (i.e., 200,000 days, or, 547 years)
- Number of initial conditions:
 - Short timescale forecast: 9990
 - Long timescale forecast: 1
- Ensembles for stochastic models:
 - Short timescale forecast: 100
 - Long timescale forecast: 1

Assessment Metric 1: RMSE

- Used for assessing short timescale forecasts
- Root Mean Square Error (RMSE)

$$E(t) = \sqrt{\frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m \left(\psi_1^{ref}(x_i, y_j, t) - \psi_1^{fcast}(x_i, y_j, t) \right)^2},$$

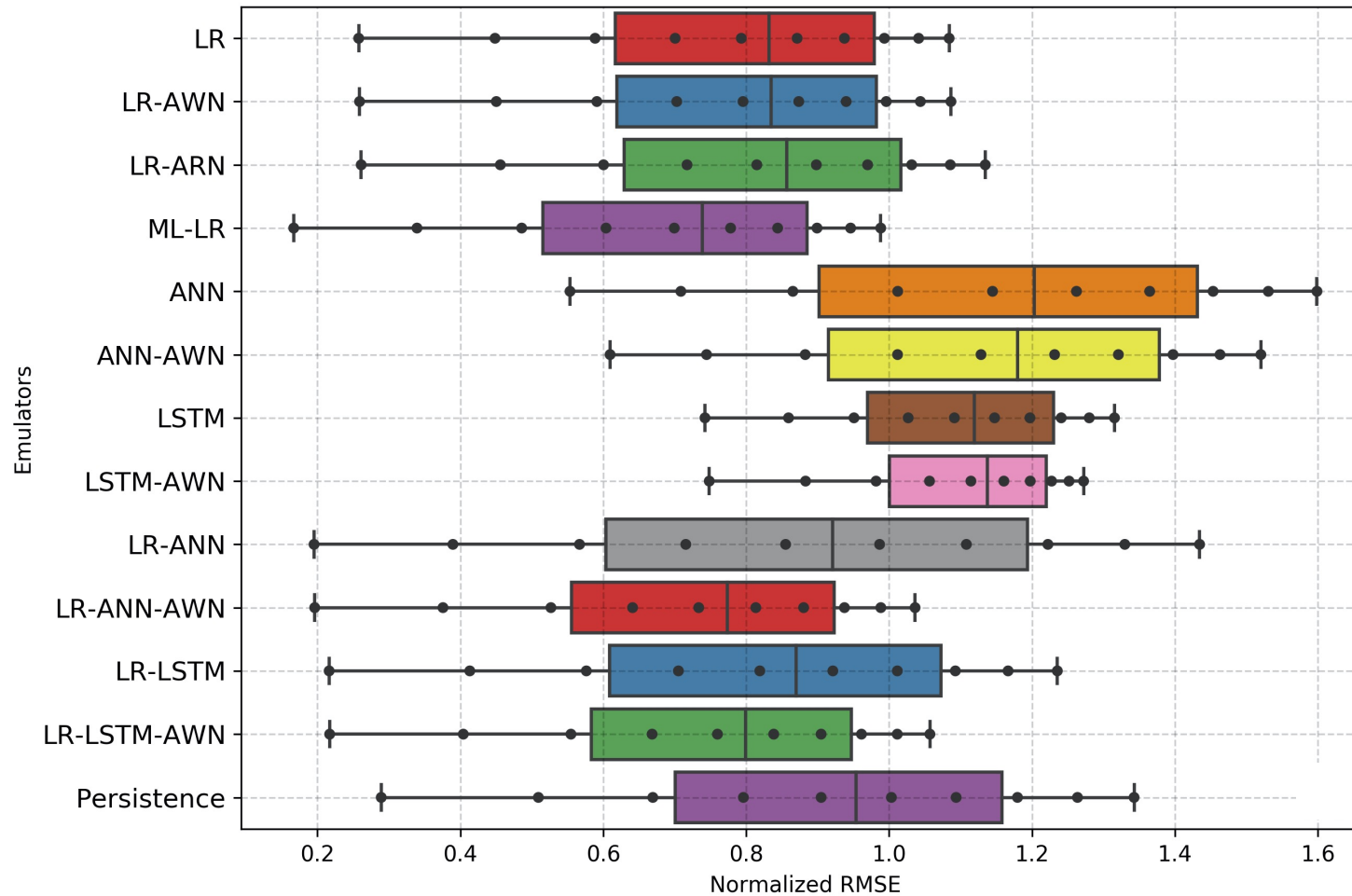
where,

$\psi_1^{ref}(x, y, t)$: reference truth spanned by 'true' 150 PCs

$\psi_1^{fcast}(x, y, t)$: reconstructed spatio-temporal field using the forecasted PCs

m : number of spatial grid points in x, y direction

Results: RMSE

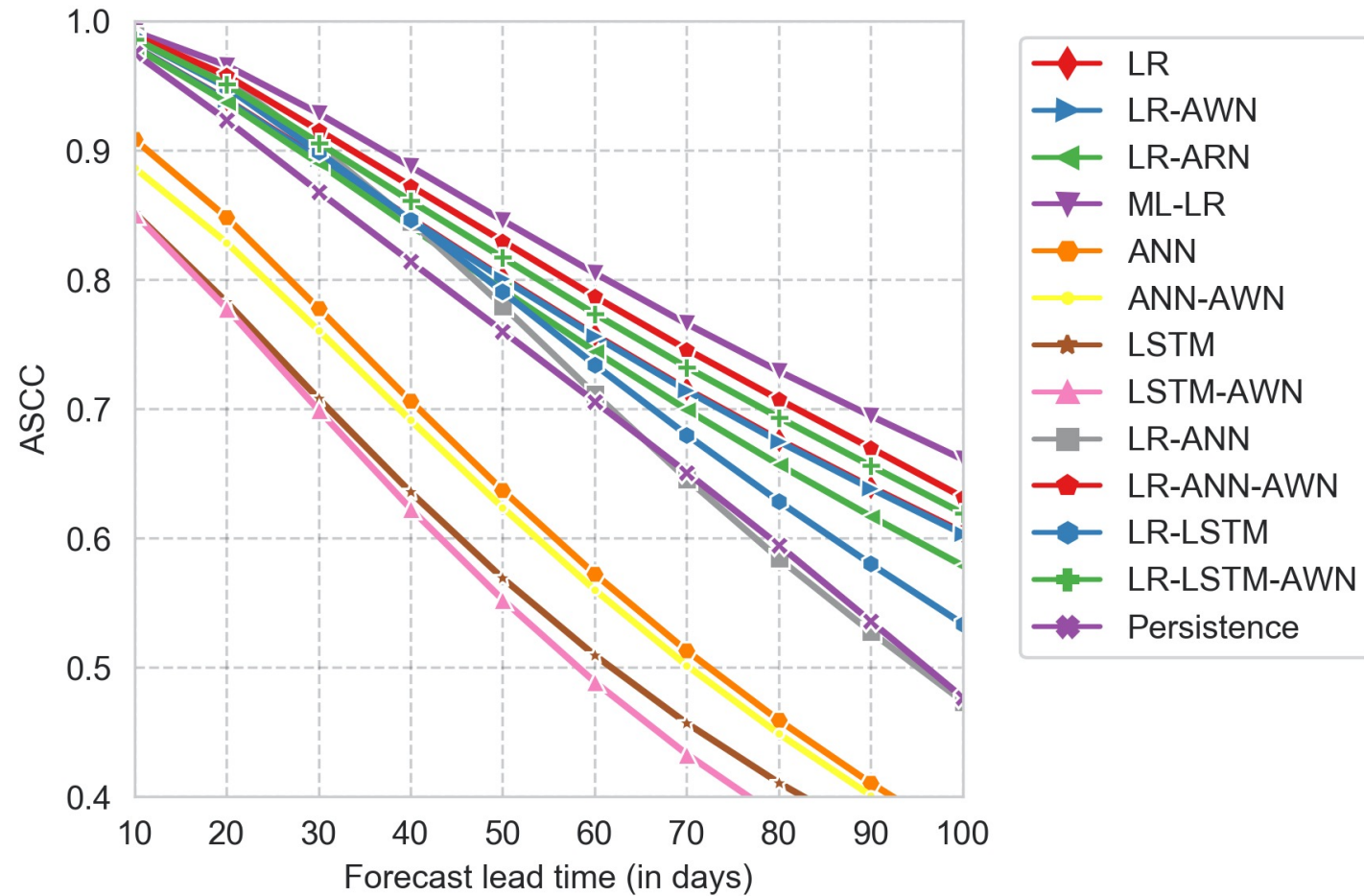


Persistence: The model state remains constant with time – equal to the initial condition

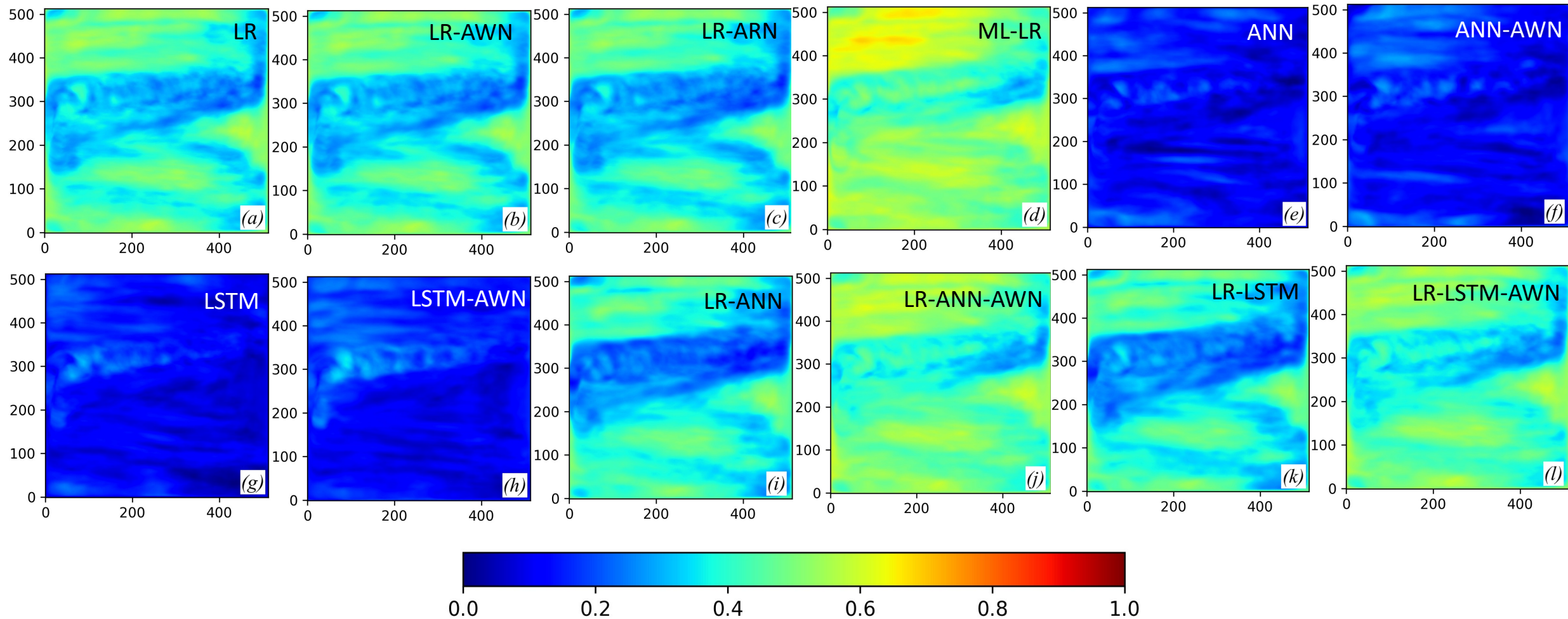
Assessment Metric 2: ACC

- Anomaly Correlation Coefficient (ACC)
- Used for assessing short timescale forecasts.
- Cross-correlation between the spatio-temporal datasets $\psi_1^{ref}(x, y, t)$ and $\psi_1^{fcst}(x, y, t)$
- Snapshot-wise correlation: ASCC
 - Conveys the spatial structure similarity between the snapshots of the two datasets
 - $ASCC = ASCC(t)$
- Temporal correlation: ATCC
 - Grid-point-wise zero-lag cross-correlation between the forecasts and reference truth
 - $ATCC = ATCC(x, y)$

Results: ASCC



Results: ATCC



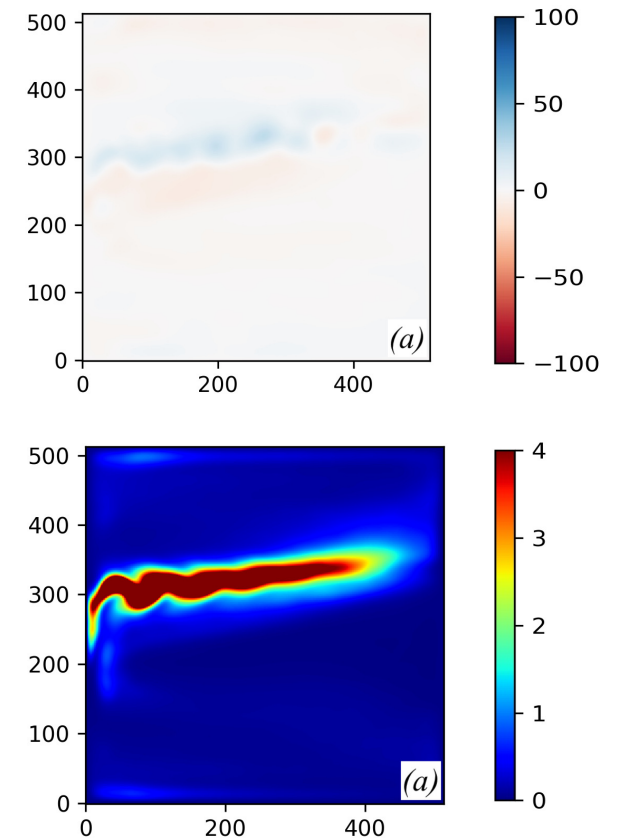
Assessment Metric 3: Climatology & Variance

- Used to assess long timescale forecasts
- Forecast settings: 200K days, 1 initial condition, and 1 stochastic realization

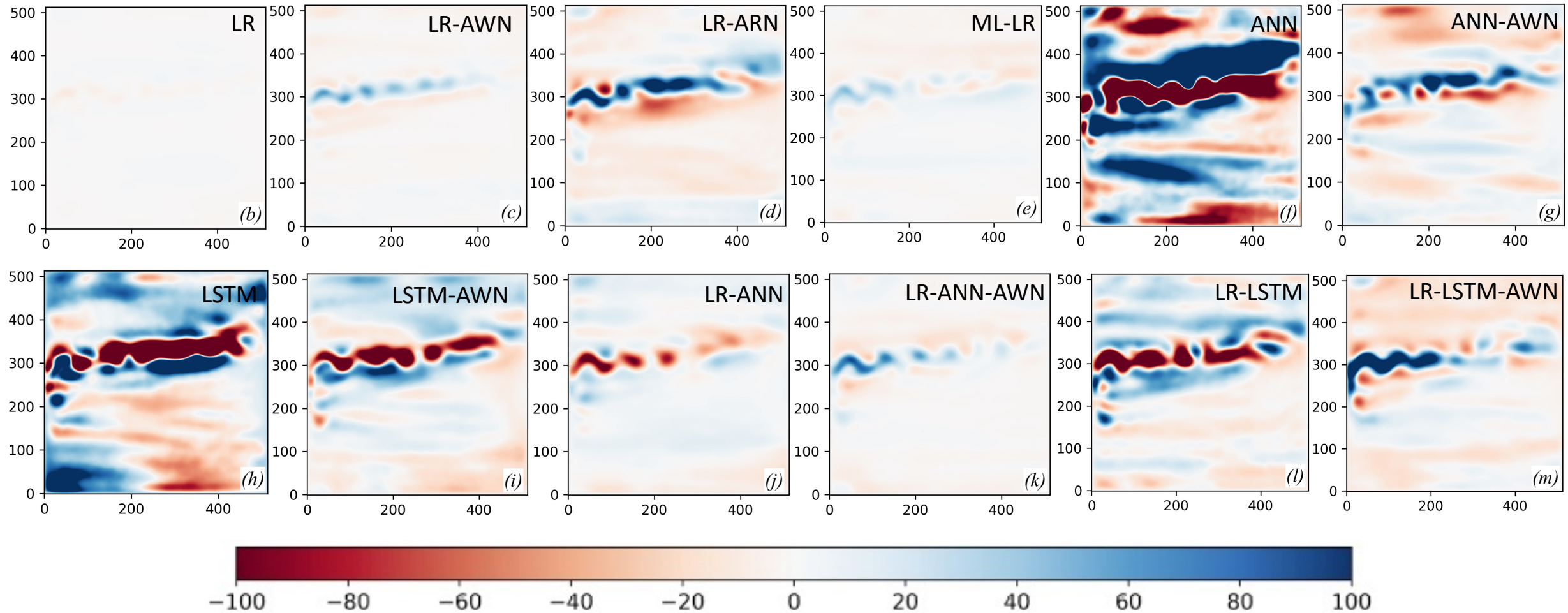
Climatology:
$$\langle \psi_1^{fcast} \rangle = \frac{1}{N} \sum_{n=1}^N \psi_1^{fcast}(x, y, t_n),$$

Variance:
$$\sigma^2(\psi_1^{fcast}) = \frac{1}{N} \sum_{n=1}^N \left(\psi_1^{fcast}(x, y, t_n) - \langle \psi_1^{fcast} \rangle \right)^2.$$

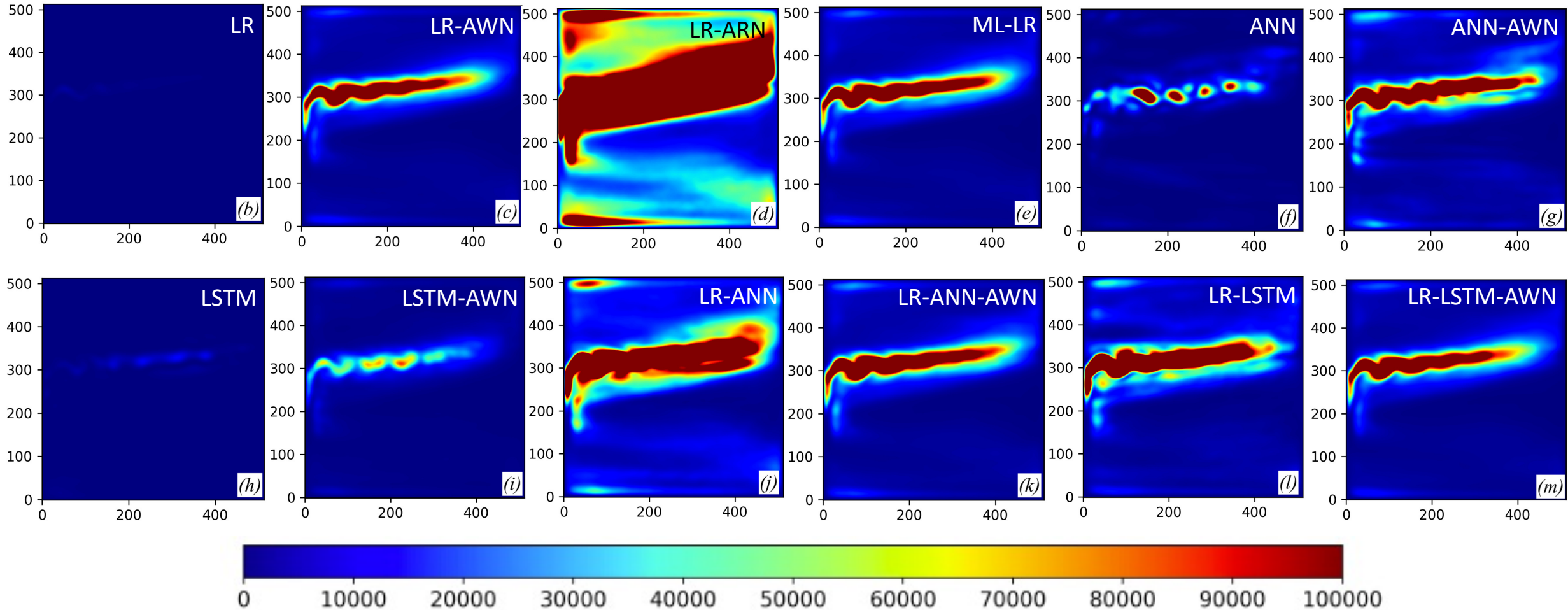
- True climatology is zero, as we deal with mean subtracted PCs.



Results: Climatology

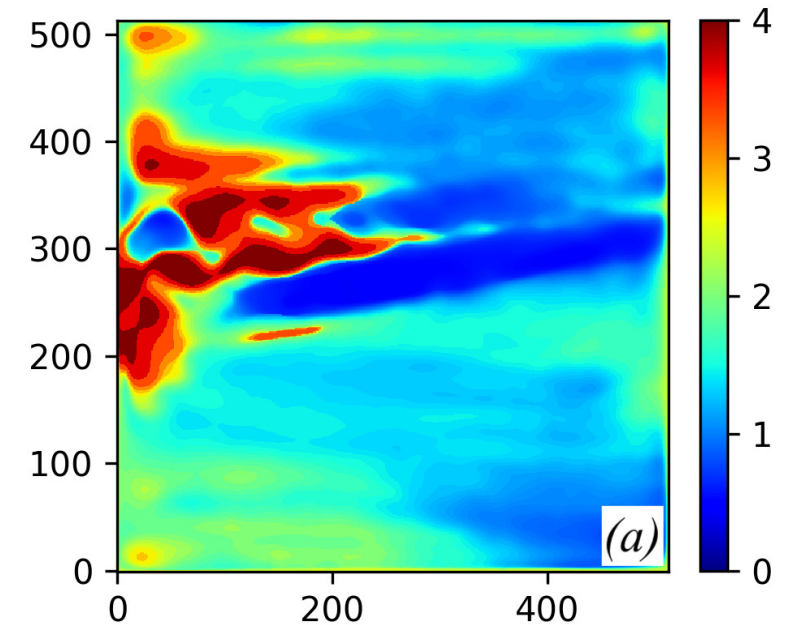


Results: Variance

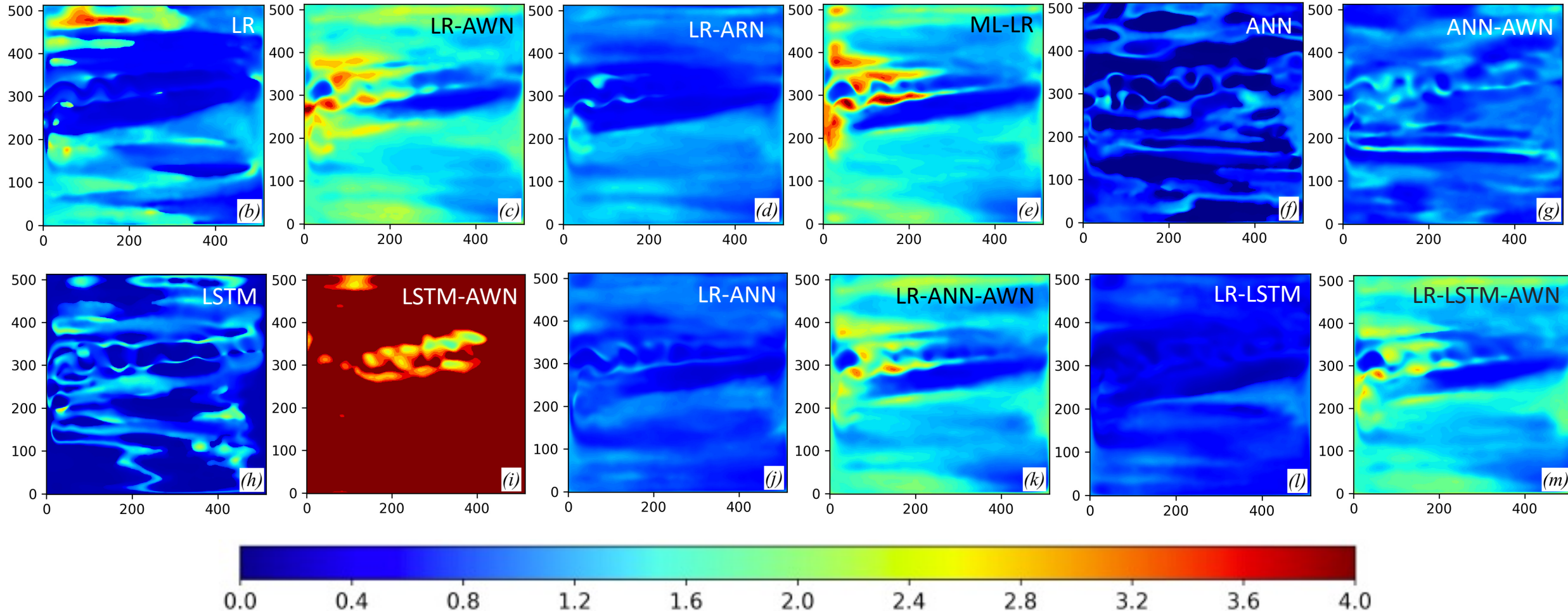


Assessment Metric 4: Frequency map

- Used to assess long timescale forecasts.
- Used to test spectral characteristics of the emulated solutions.
- For each grid cell, we compute frequency as the inverse of the decorrelation time scale.
- Decorrelation time scale: time lag at which the autocorrelation drops by a factor of $\exp$ from the zero-lag value.
- Example: Frequency map for a randomly selected streamfunction dataset of length 20K records.



Results: Frequency Map



Assessment Metric 5: Forecast Horizon

- Forecast Horizon: The time scale for which a model produces stable, non-zero, and finite forecasts.
- Benefits: Helps us in deciding the applicability of a method in short- or long-term forecasts.
- For all methods, the solutions either
 - Decay to zero -- LR
 - Saturate to a steady state -- LSTM
 - produce stable, non-zero, and finite values for the total length of long timescale forecasts – all others

Assessment Metric 6: Running Time

- Training time: time taken to train the model using the training dataset
- Prediction time: time taken to produce one realization of 1000 records long forecast.
- In practice, training is done once, but the predictions are done multiple times.
- We implemented the same level of code optimization for all models and used the same platform, i.e., Python.
- Only ML-LR is used from publicly available MATLAB toolbox.

Results: Forecast Horizon & Running Time

Method	Forecast Horizon (in years)	Training Time (in seconds)	Prediction Time (in seconds)
LR	1.3	1	10^{-1}
LR-AWN	∞	1	10^{-1}
LR-ARN	∞	10^2	10
ML-LR	∞	10	1
ANN	∞, ∞	10^3	1
LSTM	0.3, ∞	10^3	10
LR-ANN hybrid	∞, ∞	10^3	1
LR-LSTM hybrid	∞, ∞	10^3	1

Conclusions

- ML-LR is the most accurate on both short- and long-timescale forecasts, computationally inexpensive and numerically stable.
- DL models perform better in a hybrid configuration with the core dynamics.
- Therefore, the optimal strategy is:

Core dynamics + Memory effects + State-dependent noise
- Hybrid model + noise can produce very general form of state-dependent noise and can potentially learn the ML-LR architecture with more non-linear terms.
- ***Possible Extensions:***
 - Test on more complicated testbed, e.g., GCM outcomes to model, say, MJO, ENSO etc.
 - Include some physical constraints and/or conservation principles in the models.



Thanks

manuscript submitted to *JAMES*

A comparison of data-driven approaches to build low-dimensional ocean models

N. Agarwal¹, D.Kondrashov^{3,5}, P. Dueben², E.Ryzhov^{1,4}, P. Berloff¹

¹Department of Mathematics, Imperial College London, London, SW7 2AZ, UK

²ECMWF, Shinfield Road, Reading, RG2 9AX, UK

³Department of Atmospheric and Oceanic Sciences, University of California, Los Angeles, CA 90095, USA

⁴Pacific Oceanological Institute, Vladivostok, 690041, Russia

⁵Institute of Applied Physics of the Russian Academy of Sciences, 603950, Nizhny Novgorod, Russia