Using Machine Learning to predict and understand AMOC circulation strength

Faculty of Science
University of Copenhagen
Copenhagen, Denmark
June 2026

Elena De Francesco (rcp370)
Giulia Tea Menghini (swn831)
Isak Rockström (tlp693)
Rafael Pérez Guardiola (gzb538)

Atlantic Meridional Overturning Circulation

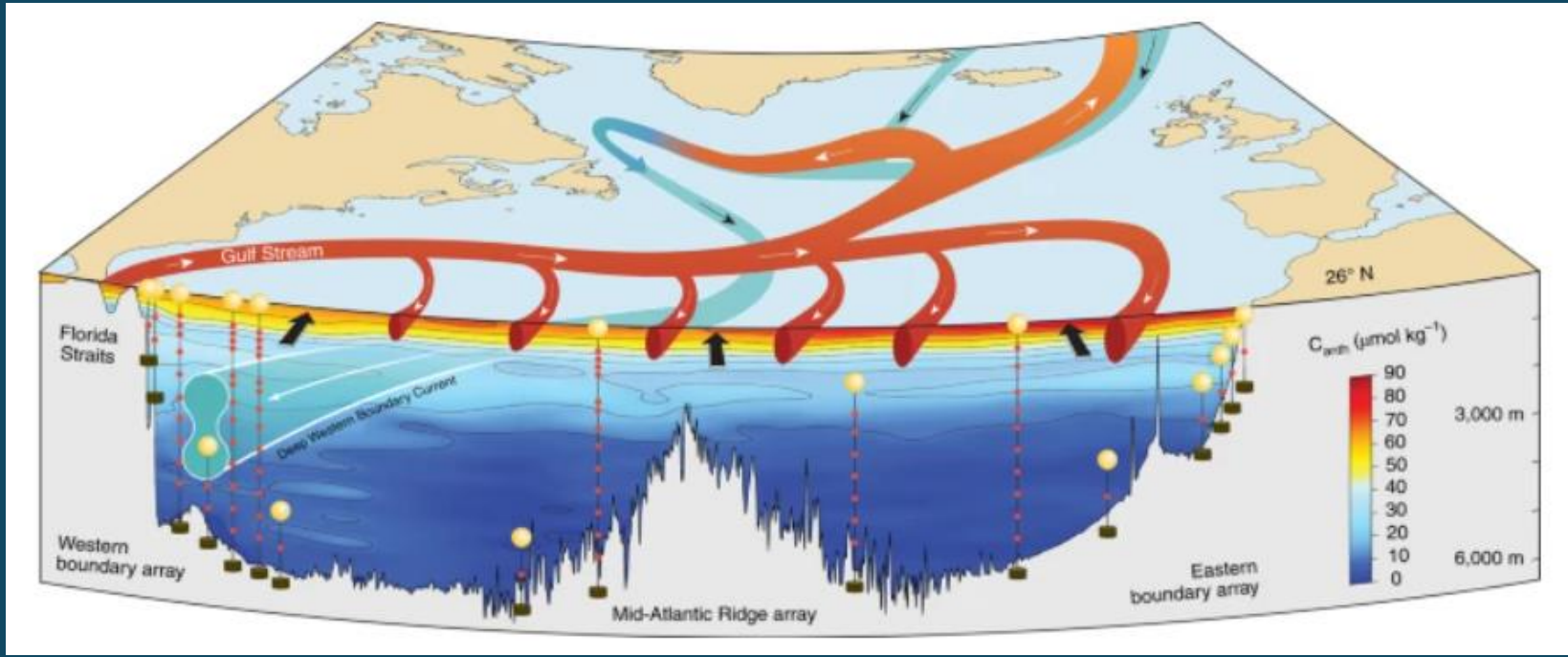
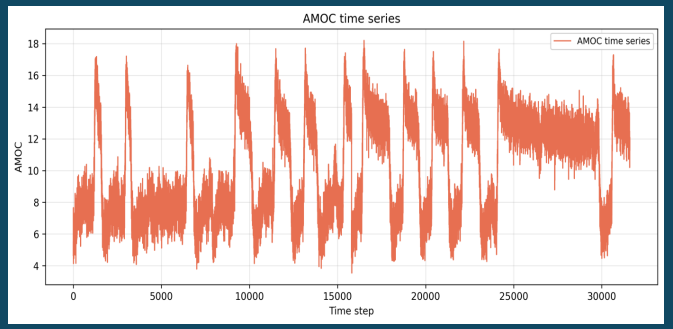


Fig.1 : The Atlantic circulation at 26°N: red arrows indicate the warm surface flow (Gulf Stream), while the blue-green arrow shows the return of cold waters at depth (from "*Circulation-driven variability of Atlantic anthropogenic carbon transports and uptake*" https://doi.org/10.1038/s41561-021-00774-5).

- Large-scale **ocean circulation** transporting warm water northward and cold water southward across the Atlantic
- Key role in **regulating climate**
- **Bistable behavior**: two stable states (strong vs. collapsed)

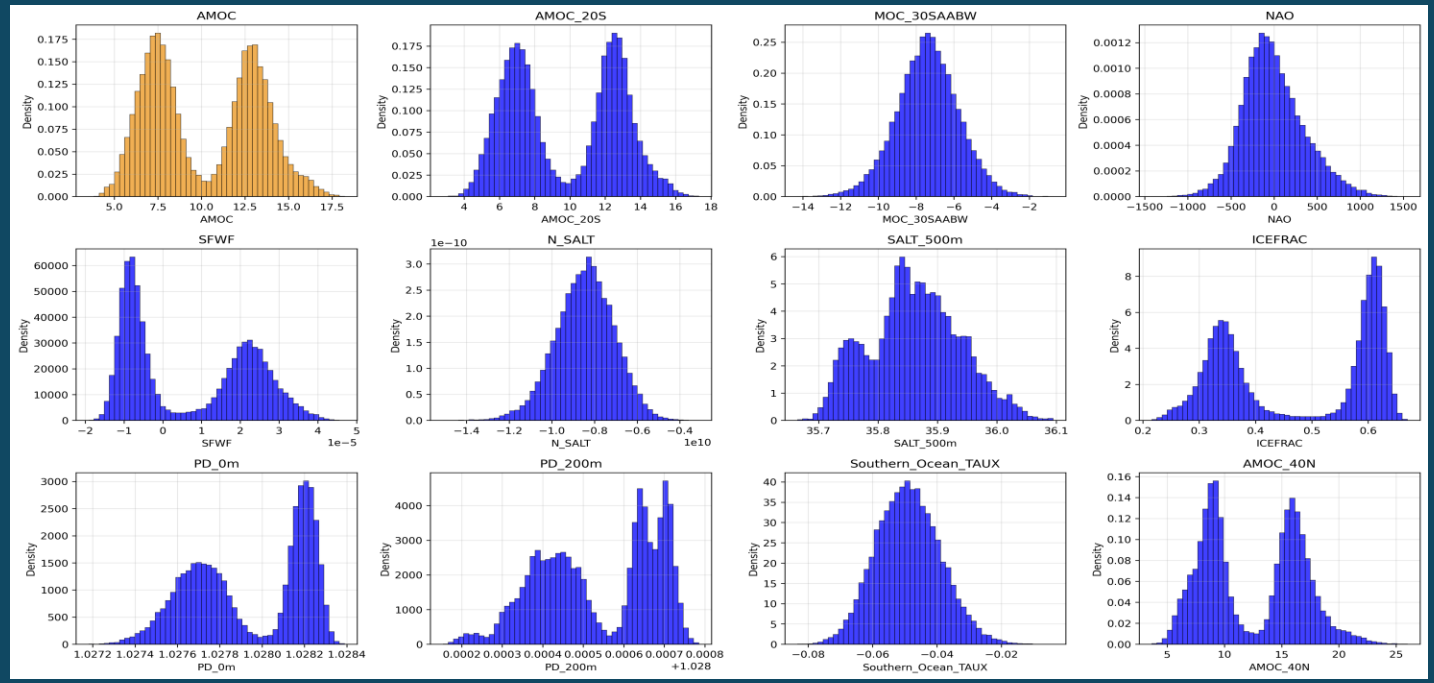


Can ML forecast AMOC evolution and critical transition?

Dataset from NCAR Community Earth System Model

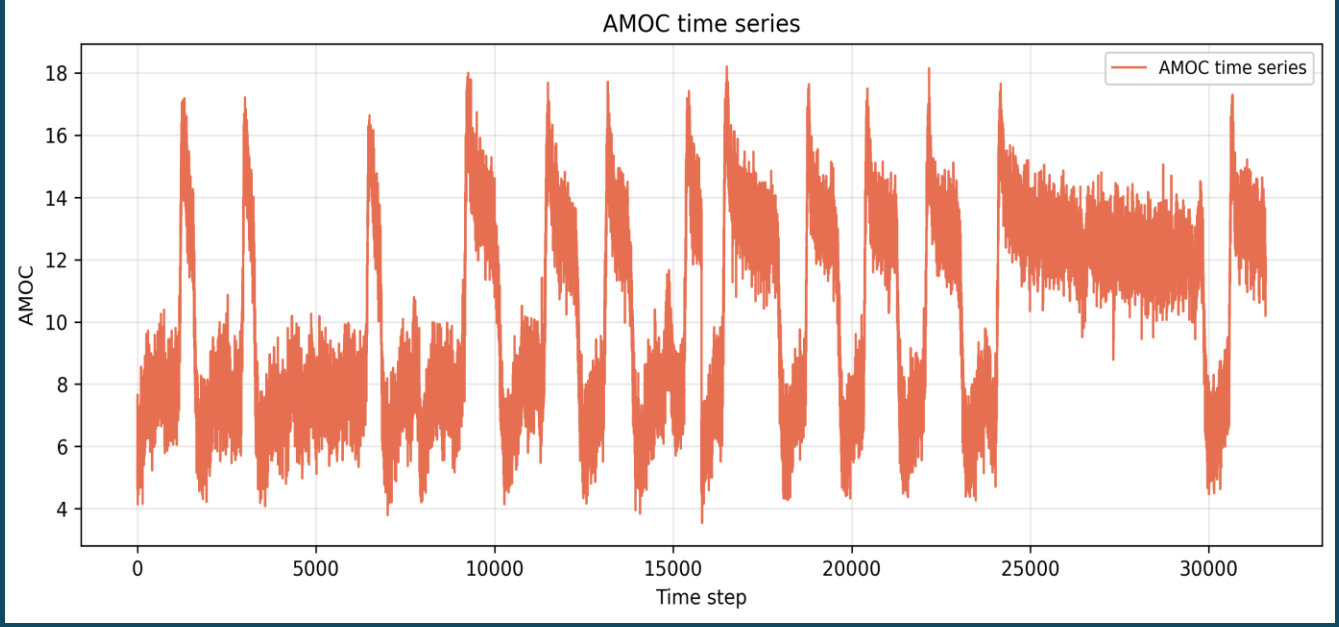
time step	SFWF	PD_200m	ICEFRAC	N_SALT	NAO	...	AMOC
0	-0.0	1.0284	0.6507	-9507178496.0	-120.4375	...	5.4831
1	-0.0	1.0284	0.6483	-8816568320.0	-146.1562	...	4.6295
2	-0.0	1.0284	0.6465	-9958822912.0	0.7969	...	5.9217
3	-0.0	1.0284	0.6547	-6875847680.0	-18.1875	...	6.003
4	-0.0	1.0284	0.6567	-8184853504.0	-46.9297	...	7.6616
...	...	...	...	...	...	...	...
31591	0.0	1.0286	0.3671	-8546611200.0	-478.8203	...	11.2885

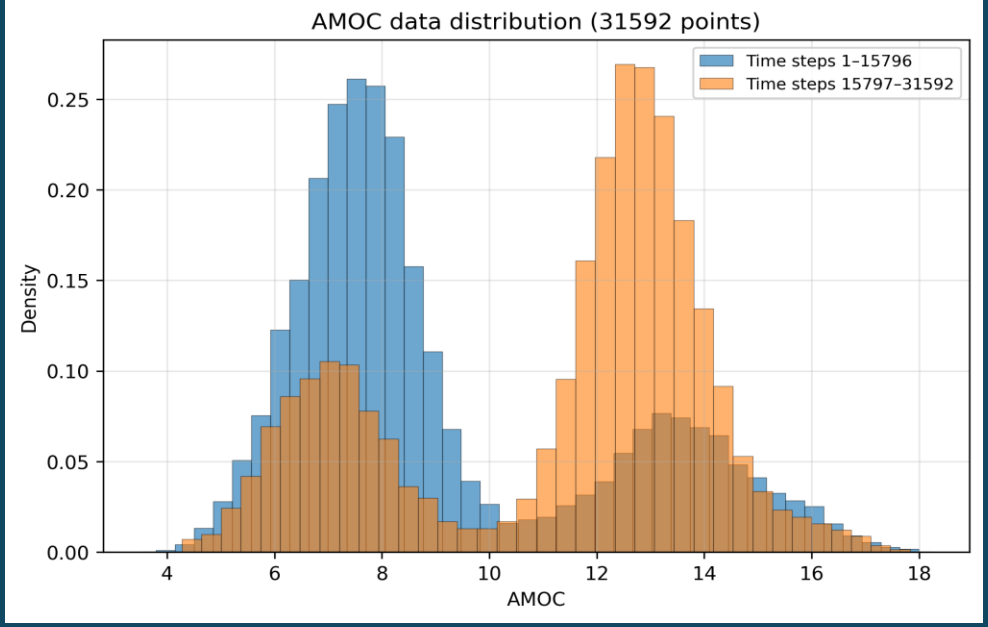
Property	Large dataset
Source	CESM
Format	CSV file
Observations	31.592 time steps
Variables	12
Data type	Floating-point values
Target	AMOC



The datasets: main challenges

- ❑ Only a **limited number of AMOC transitions** are available in the dataset.
- ❑ Variable and target **distributions change** significantly across the time series.
- ❑ The data contain strong **noise**/show **internal variability**.





Research questions

Architectures

Increasing difficulty

Q1: can we make *teacher forced* predictions on AMOC strength?

ESN

Q2: can a model learn to emulate the dynamics of the data for *auto-regressive predictions*?

ESN

GRU

Q3: can a model reliably do *auto-regressive forecasts* on individual transitions?

GRU

Prediction Methods

Teacher-Forced:

W_1: $D_1, D_2, D_3, \ldots, D_W \Rightarrow P_1$

W_2: $D_2, D_3, D_4, \ldots, D_W \Rightarrow P_2$

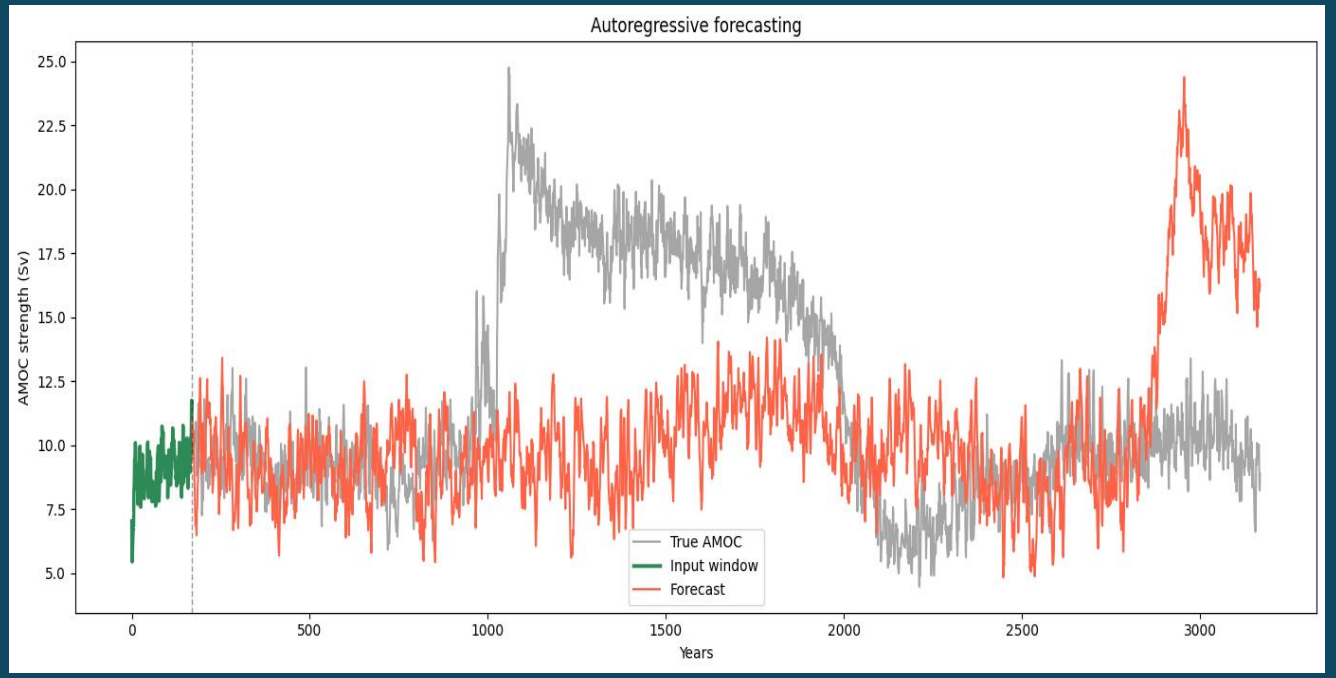
W_3: $D_3, D_4, D_5, \ldots, D_W \Rightarrow P_3$

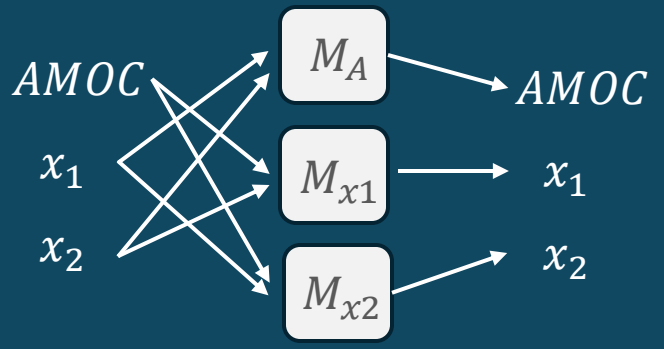
Auto-Regressive:

W_1: $D_1, D_2, D_3, \ldots, D_W \Rightarrow P_1$

W_2: $D_2, D_3, D_4, \ldots, P_1 \Rightarrow P_2$

W_3: $D_3, D_4, \ldots, P_1, P_2 \Rightarrow P_3$



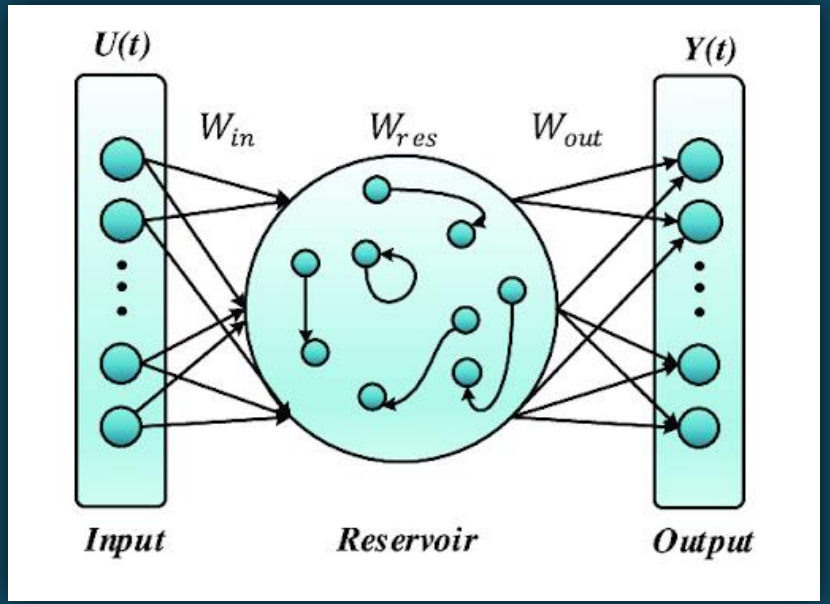


$$Noise(t) = k * MVG(AMOC(t)) + \rho * Noise(t-1)$$

ESN: Echo State Network

Reservoir computing for temporal dynamics

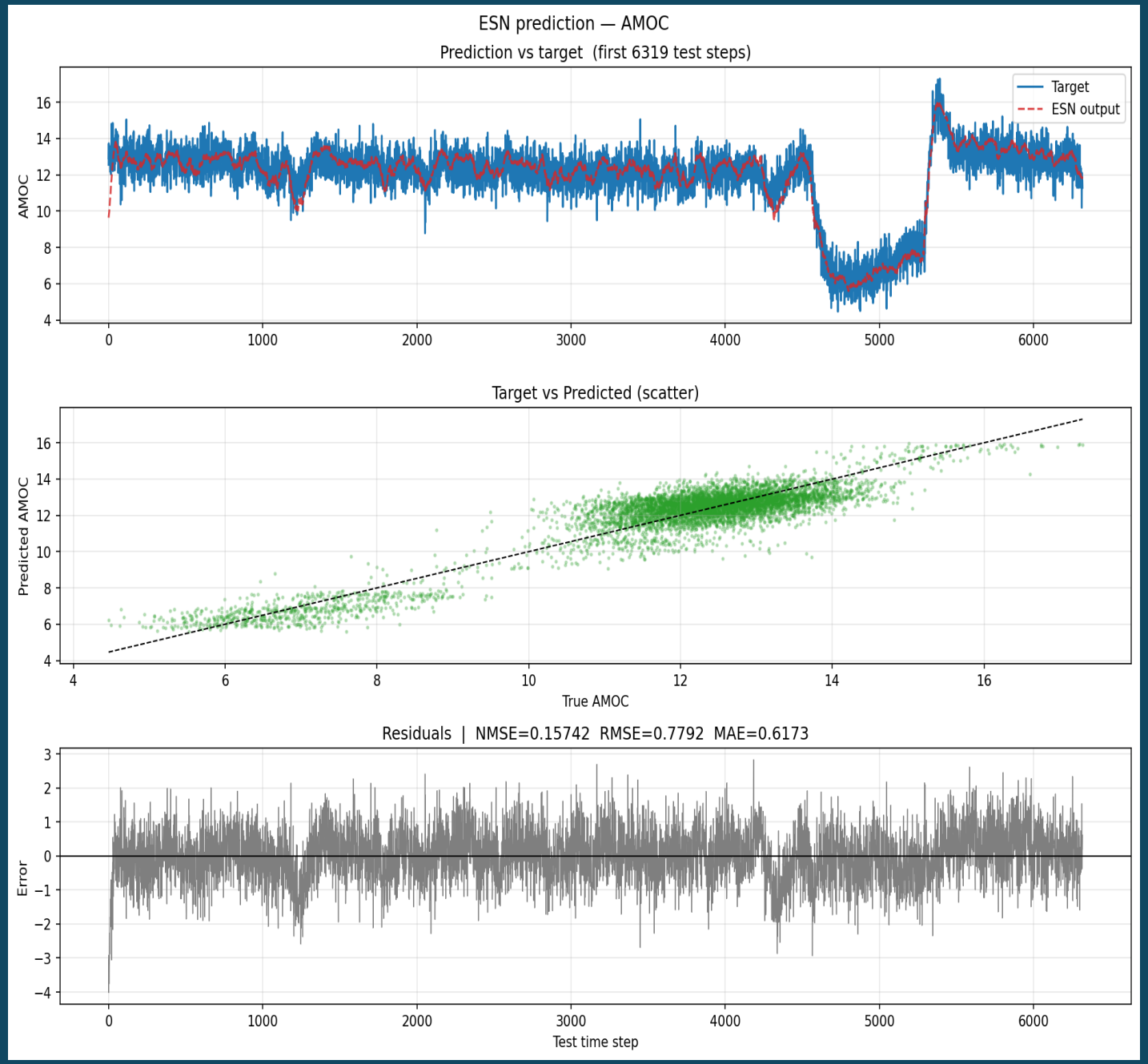
- The input is projected into a **high-dimensional dynamical space**.
- The internal recurrent network, called the **reservoir**, is randomly initialized and kept fixed.
- Only the output layer, the **readout**, is trained.



PROS:
- Efficient for **testing forecasting strategies**.
- Fast to train

Features used:

$$\texttt{AMOC}(t+1) \leftarrow [\texttt{PD\_200m}(t),\ \texttt{ICEFRAC}(t)]$$
$$\texttt{SFWF}(t+1) \leftarrow [\texttt{AMOC}(t),\ \texttt{ICEFRAC}(t)]$$
$$\texttt{PD\_200m}(t+1) \leftarrow [\texttt{AMOC}(t),\ \texttt{SFWF}(t),\ \texttt{ICEFRAC}(t)]$$
$$\texttt{ICEFRAC}(t+1) \leftarrow [\texttt{AMOC}(t),\ \texttt{PD\_200m}(t),\ \texttt{SFWF}(t)]$$



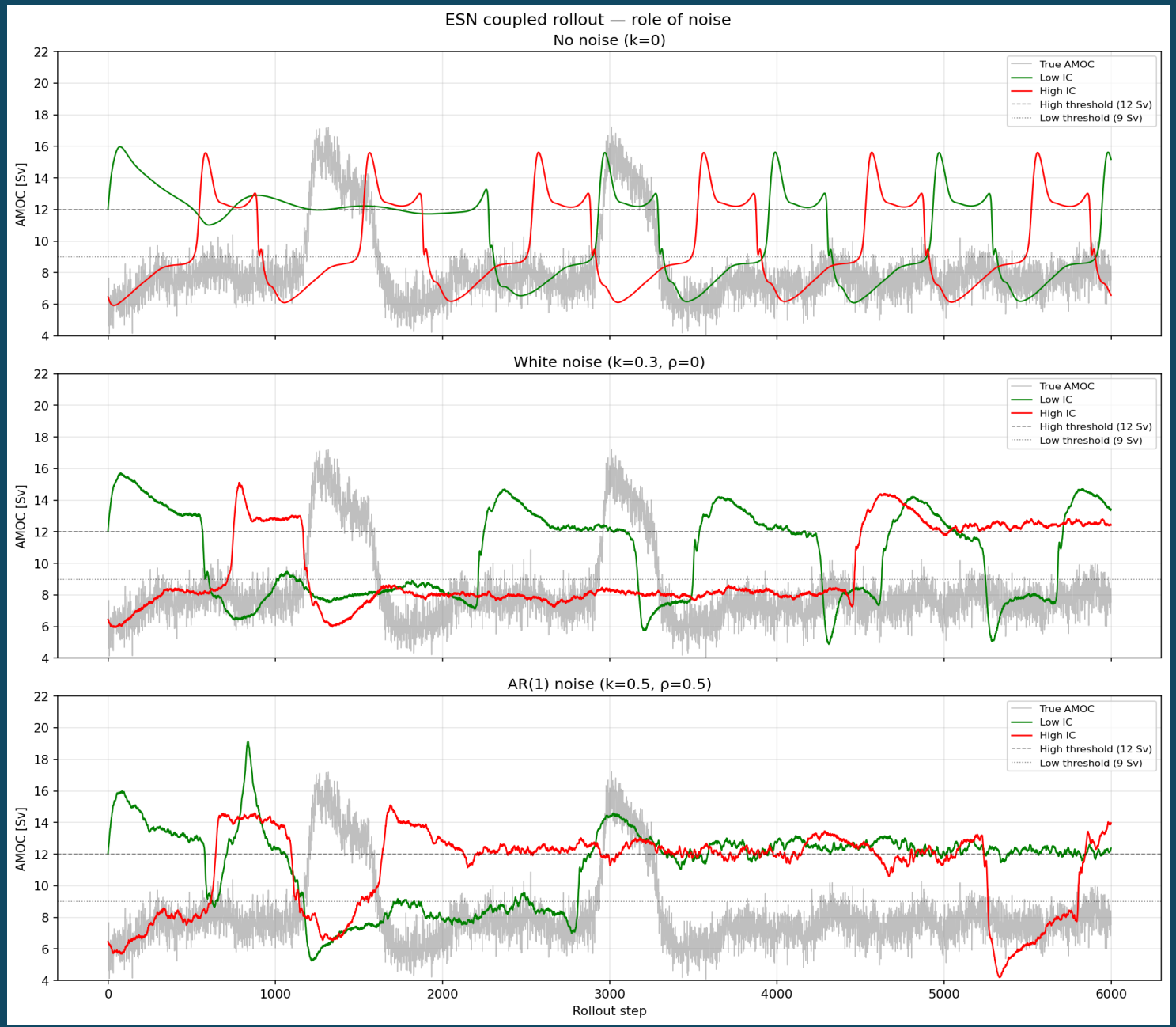
The model: Single ESN V1

Input variables:
- AMOC
- ICEFRAC
- PD_200m

Forecasting setup:
- Teacher-Forced

Noise modelling:
- No noise considerations for preliminary testing

Conclusion:
- Without noise, the model learns the transition trends between the two states
- With simulation data AMOC can be predicted accurately



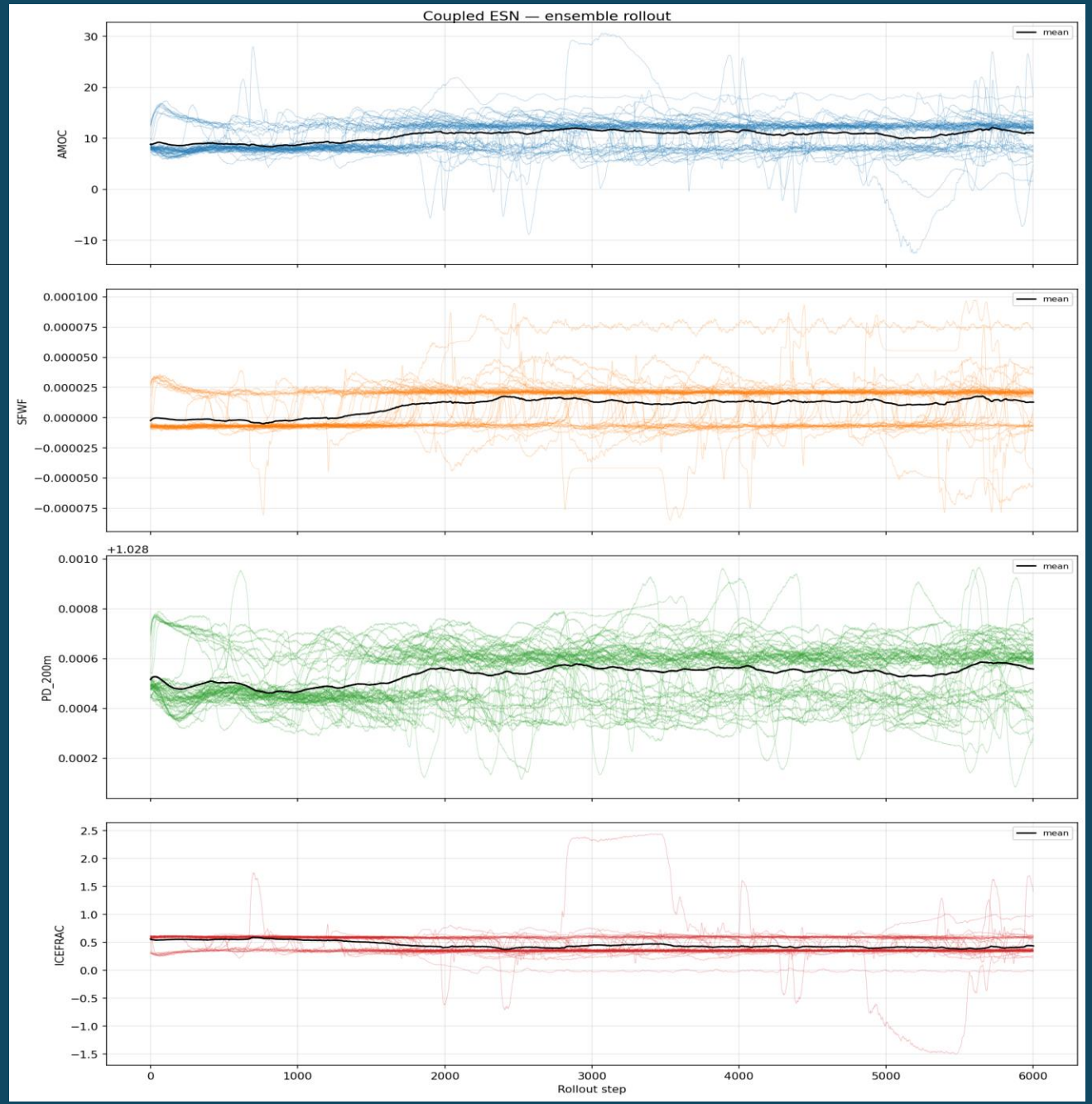
The model: Coupled ESN V2

Input variables:
- AMOC
- ICEFRAC
- PD_200m
- SFWF

Forecasting setup:
- Auto-regressive Forecasting

Noise modelling:
- Multivariate Gaussian noise for on/off states
- Samples constrained within 1σ for each variable

Conclusion:
- Noiseless model leads to oscillator
- White Noise disrupts oscillations and leads to unrealistic AMOC predictions
- AR(1) noise and white noise is able to have varying oscillation regimes



The model: Coupled ESN V2

Input variables:
- AMOC
- ICEFRAC
- PD_200m
- SFWF

Forecasting setup:
- Auto-regressive Forecasting

Noise modelling:
- Multivariate Gaussian noise for on/off states
- Samples constrained within 1σ for each variable

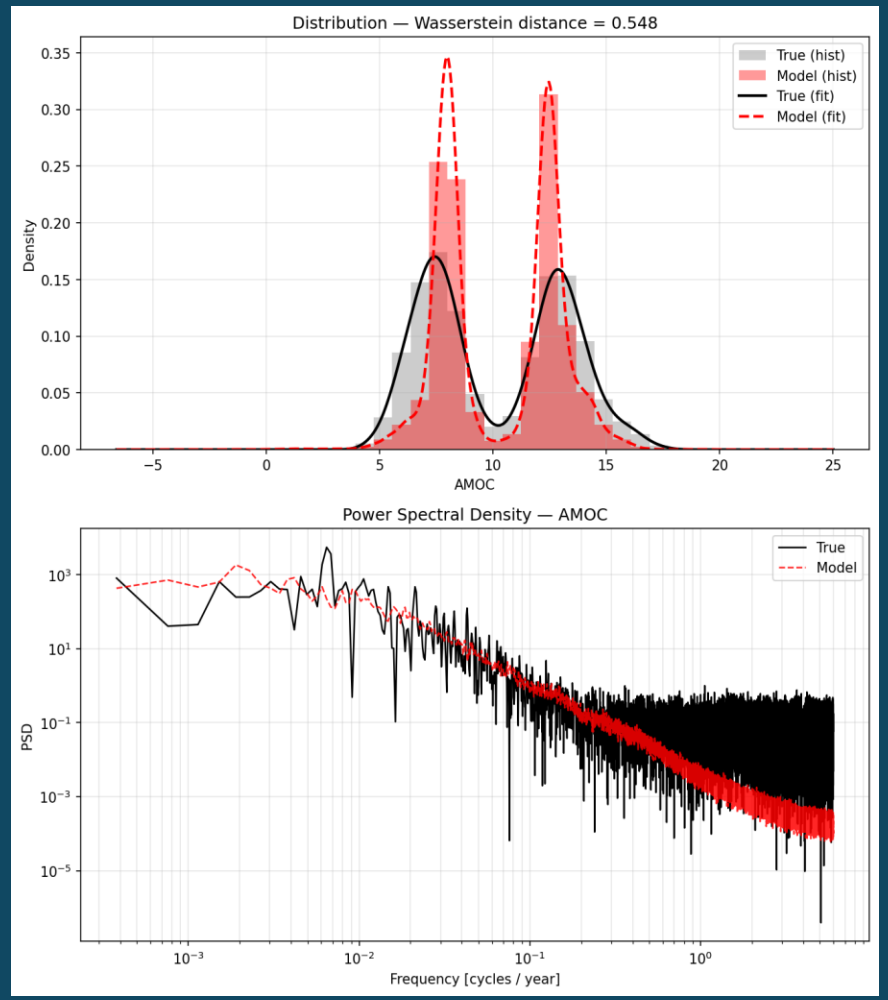
Conclusion:
- There are a few problems with the runs
  - Unrealistic feature values
  - Likely caused by too much noise inputted

Assessment of ESN Coupled V2

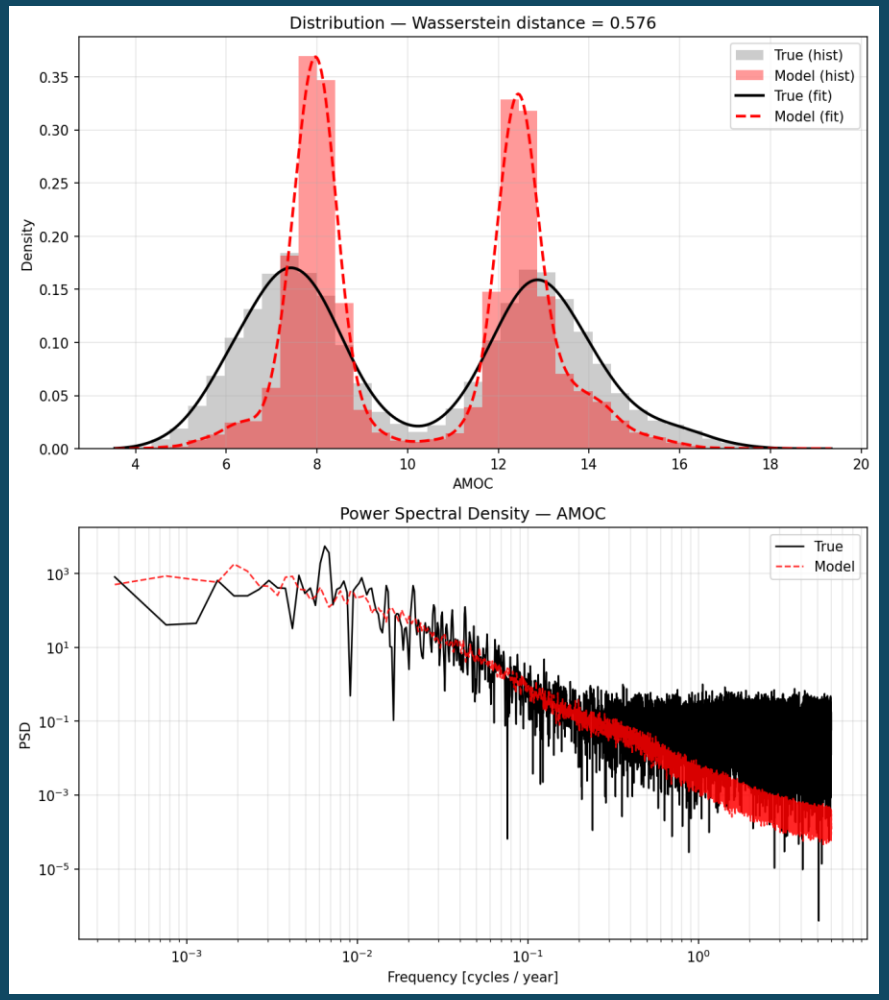
Diagnostics:

- Noise level is less than the true signal
- Dominant spectral peaks lie around 10^{-3} cycles/year
- Cleaning the data decreases the WD

Raw Trajectories



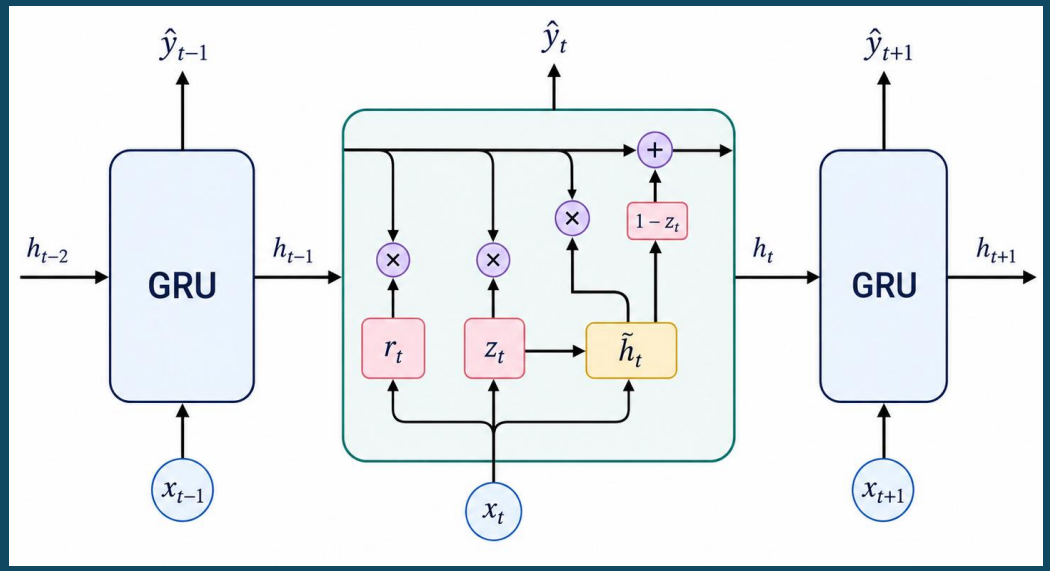
Cleaned Trajectories



GRU: Gated Recurrent Unit

Recurrent neural network for gated memory dynamics

- GRUs are recurrent neural networks designed for sequential data.
- The hidden state acts as a memory of past time steps.
- Two gates control how information is updated:
  - The **reset gate** filters past information;
  - The **update gate** balances old memory and new input.



PROS:

- Learns long-term temporal dependencies in climate time series.
- Selectively preserves relevant past information through gated memory.

Features used:

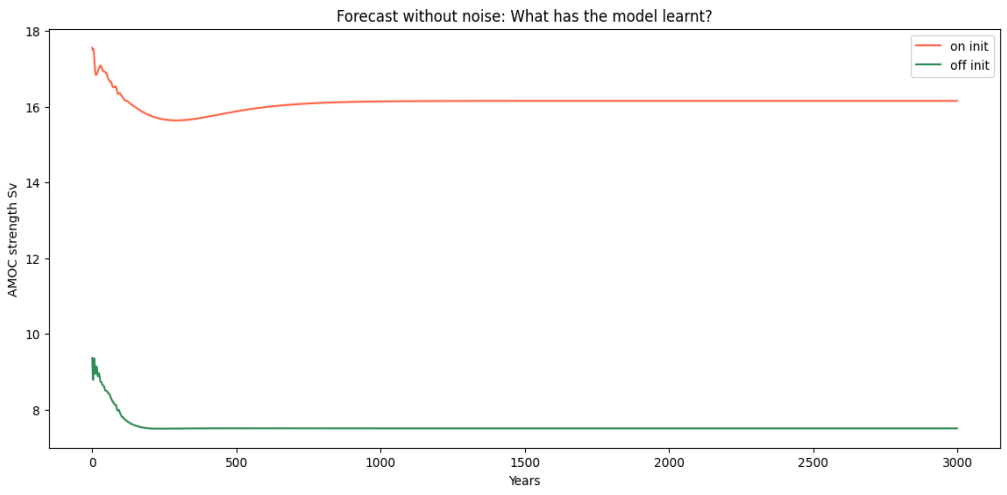
$$\text{AMOC}(t+1) \leftarrow [\text{AMOC}(t),\ \text{PD\_200m}(t),\ \text{SFWF}(t)]$$

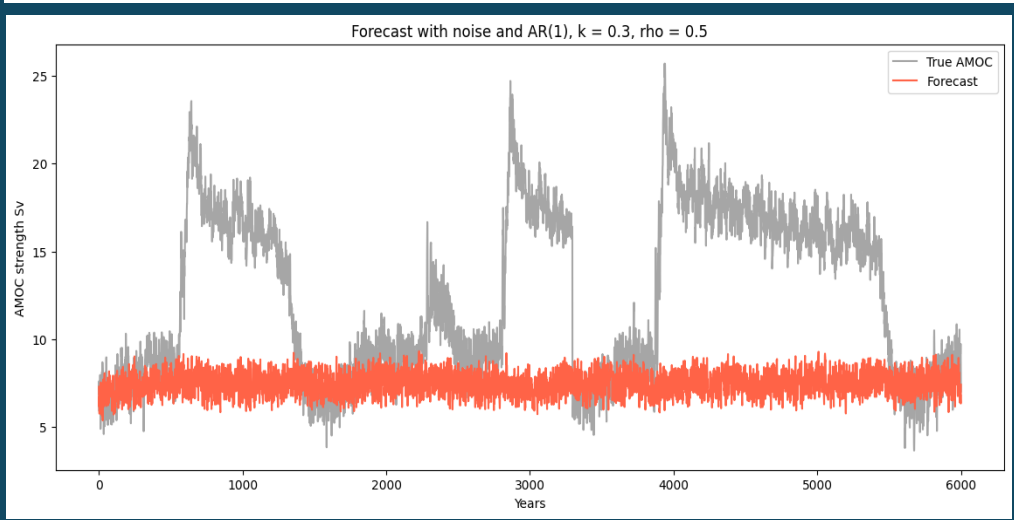
$$\text{SFWF}(t+1) \leftarrow [\text{SFWF}(t),\ \text{AMOC}(t),\ \text{PD\_200m}(t)]$$

$$\text{PD\_200m}(t+1) \leftarrow [\text{PD\_200m}(t),\ \text{AMOC}(t),\ \text{SFWF}(t)]$$

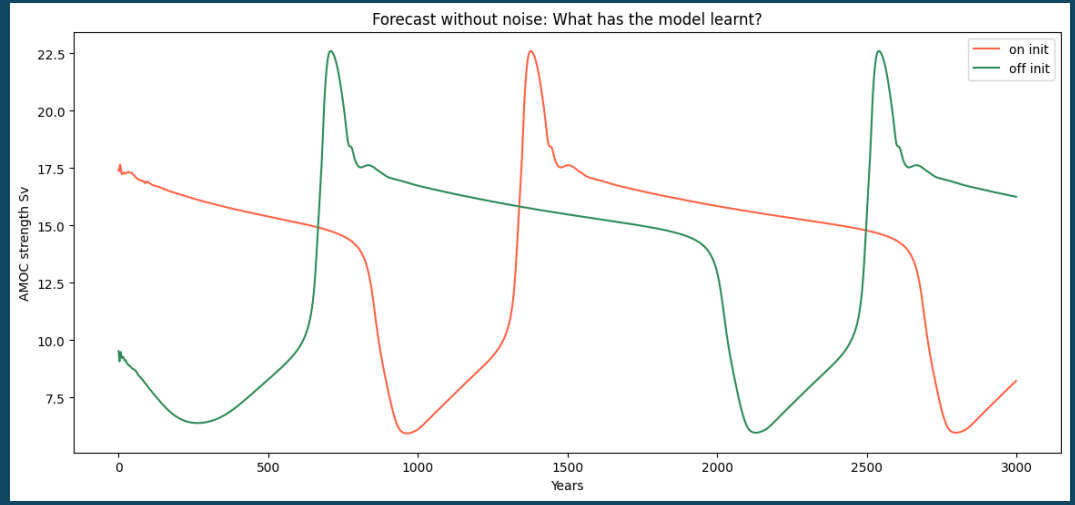
Auto-Regressive Forecasting – Tricky Training

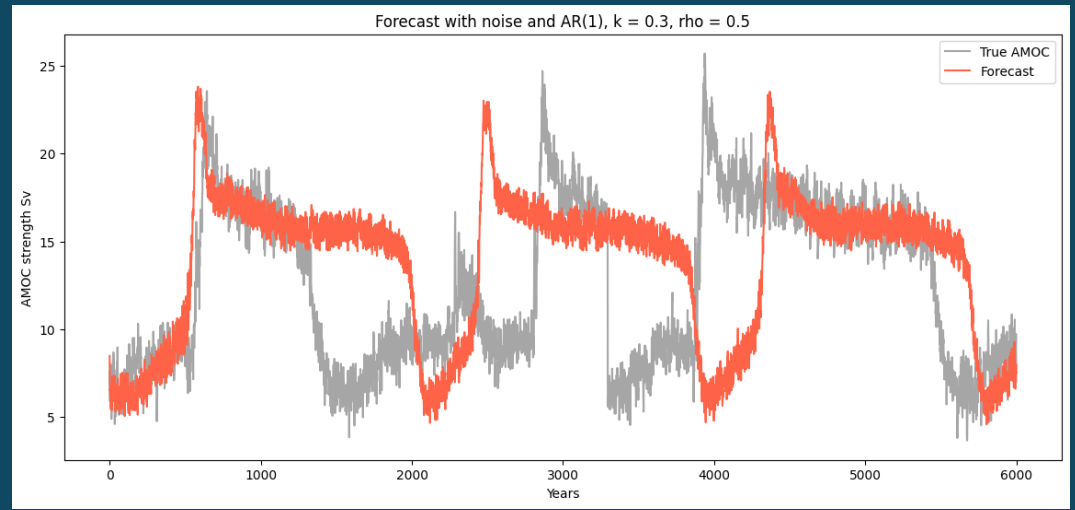
Short training -> Bistable





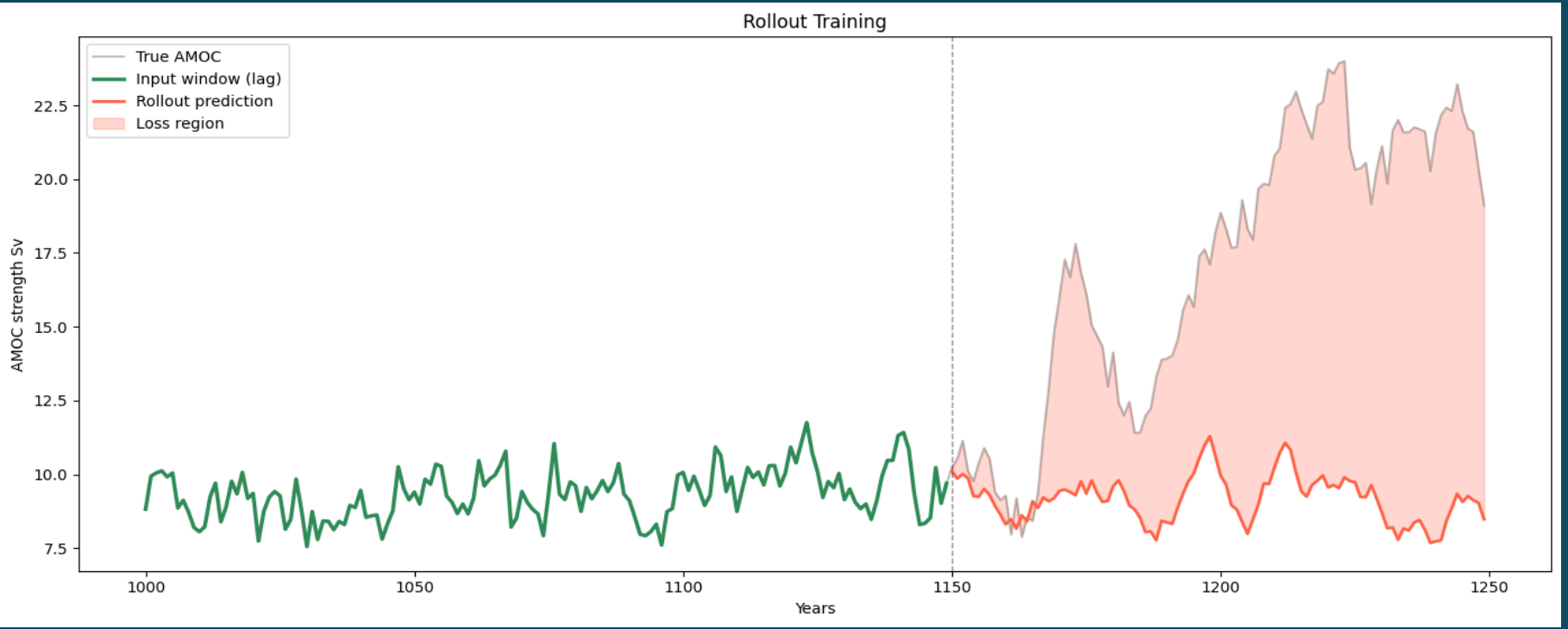
Longer training -> Limit cycle





GRU Coupled V1 - Rollout training

1: Learn the broad dynamics first with brief teacher forced training
2: Fine tune with rollout training



- Improve exposure bias
- Stabilize AR forecasts by learning from its own errors
- Many new hyper parameters!

Optimizing

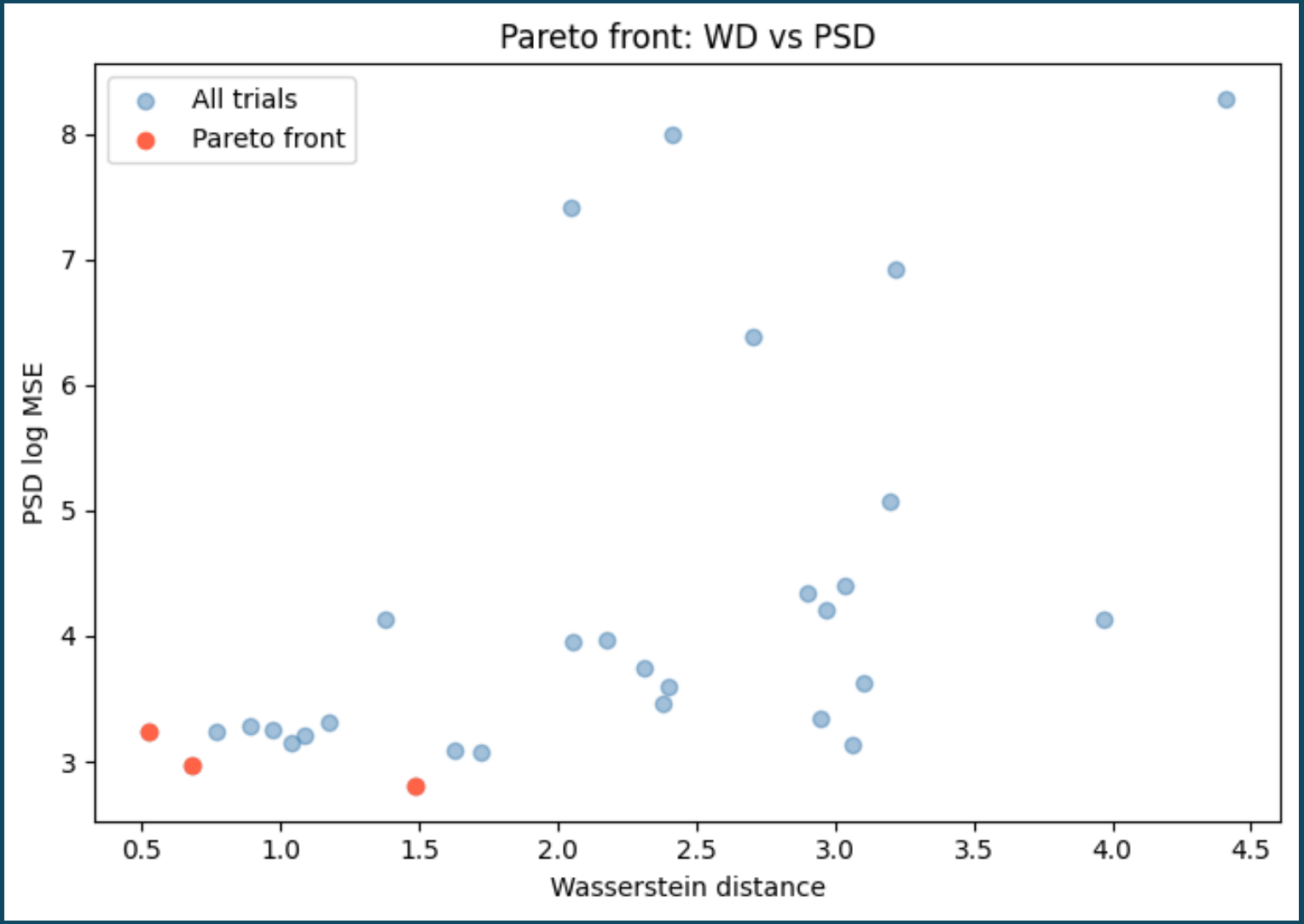
We now have a lot of HPs…
- Model: ~5
- Training: ~4
- Prediction: ~2

Targets:
- Wasserstein distance
- Power spectral density

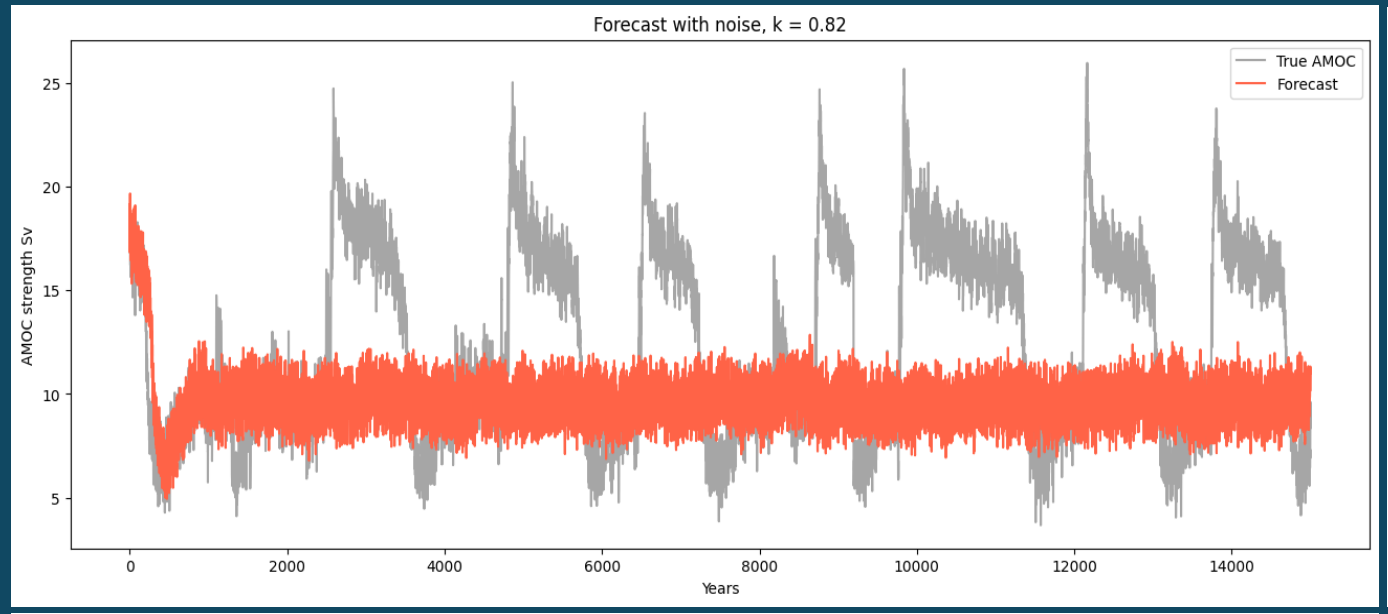
Results were impossible to reproduce!

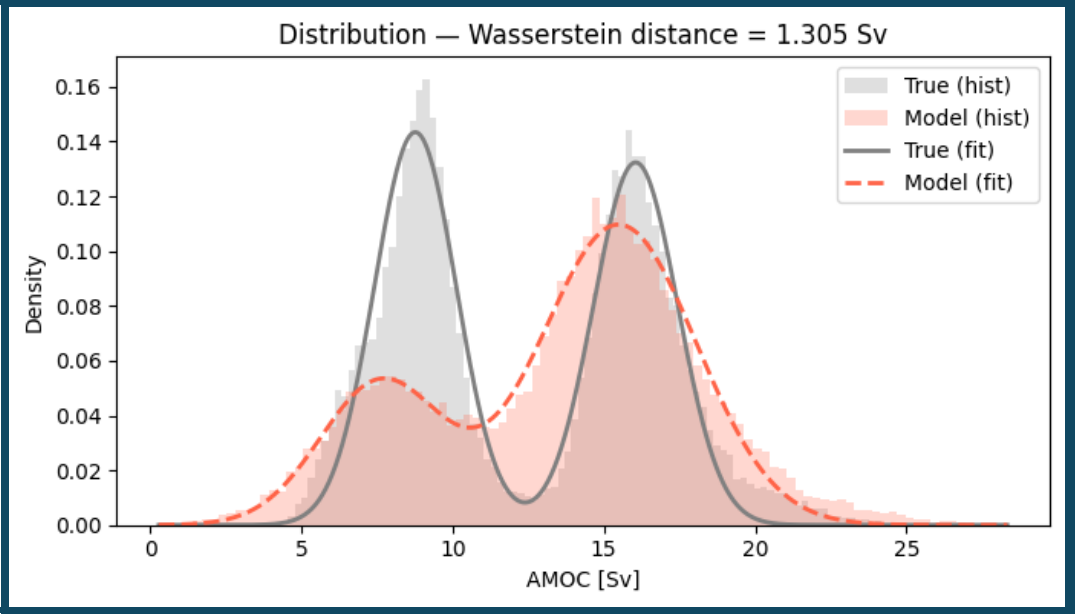
Seed controlled random sources:
- Rollout window sampling
- Prediction noise

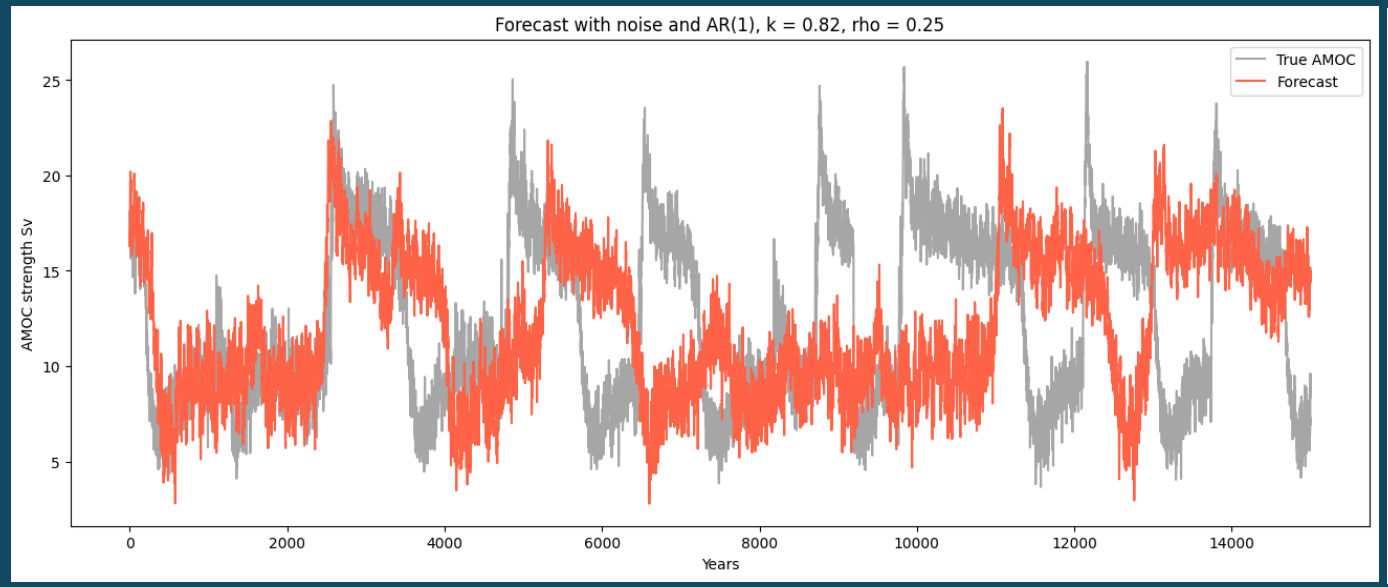
Uncontrolled random sources:
- Weight initialization
- Pretraining shuffling by torch
- Dropout

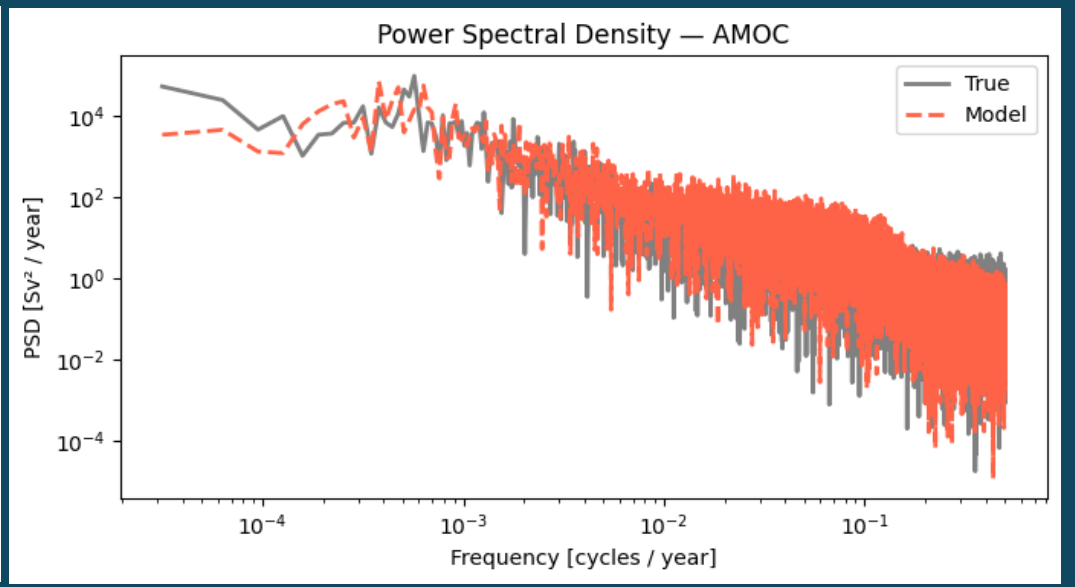


GRU Single (W. Rollout)

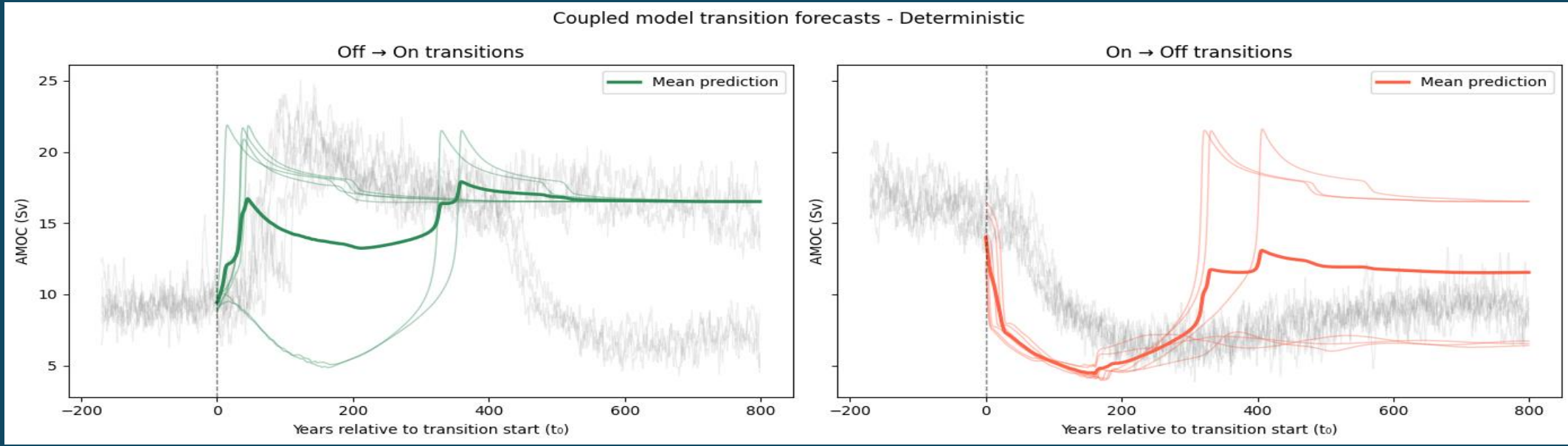




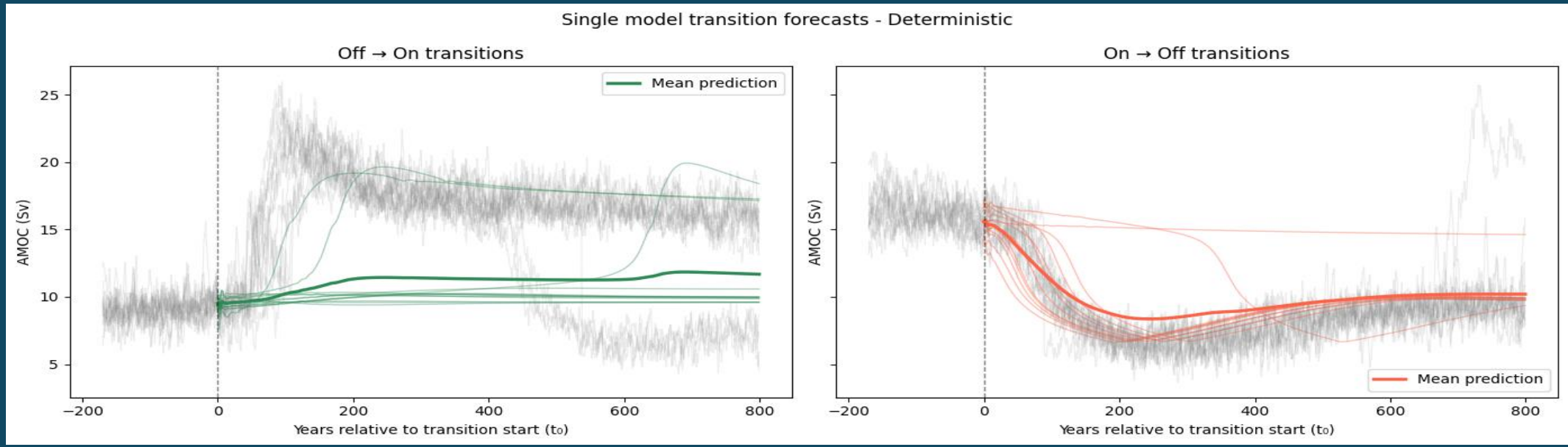




GRU Single:



GRU Single (With Rollout training):



Conclusions

Teacher Forced predictions:

- A **high level of accuracy** is achieved for teacher forced predictions for all coupled models
  - If the data for the predictors is easier to simulate, then AMOC predictions can be made that will be accurate
- Introducing **noise** and AR(1) noise **makes results worse**, as expected

Auto-regressive emulation:

- Models **learn bistable or limit-cycle dynamics**
- Auto-regressive noise can push a bistable model between **on and off states**, and pace transitions in a model with limit cycles
- **Rollout training** may help to **compromise** between bi-stable and limit cycle behaviour

Transition forecasting:

- Models can **learn the dynamics of On -> Off transitions**
- Models **struggle with Off -> On predictions**, suggesting that they are more noise driven
- Data leakage is a bigger issue here
- **ESN** is able to explore **both regimes** and transition between them although values may sometimes be unrealistic

Thank you for your attention

Appendix

All group members contributed equally

Important note on data leakage!

Training models for auto-regressive forecasting does not include any train/test splits.

The reasoning here is that we are targeting emulation and not prediction. We want to see if an ML model can learn to replicate the AMOC dynamics from the data.

During auto-regressive predictions the model will quickly forget whatever data it was given as an initial state, meaning its predictions reflect whatever dynamics it has learnt, and not what it remembers should come next, making data leakege less of a concern. There is however some indirect data leakage, the dynamics learnt by the model may only fit the data and not transfer to other regimes.

One could have used a train/test split, and done Wasserstein distance and PSD evaluations on the test set. However, as the data distribution varies strongly through different sections of the time series this does not make sense, it would mean trying to fit the model on data very different from what it was trained on.

The place where this data leakage becomes more of a concern is slide 20 where we discuss transition forecasting. Here the whole point is that the model should infer something meaningful from its initial conditions. Having already seen the transitions this is a meaningful data leakage, but since the models struggle to forecast transitions either way, this has been accepted in our project:)

Architectures

The two architectures employed in this work are governed by fundamentally different update mechanisms: the GRU (left) learns to selectively retain and update information through trainable reset and update gates, while the ESN (right) projects the input into a fixed high-dimensional reservoir and trains only the linear readout via ridge regression - trading expressive gating for computational efficiency

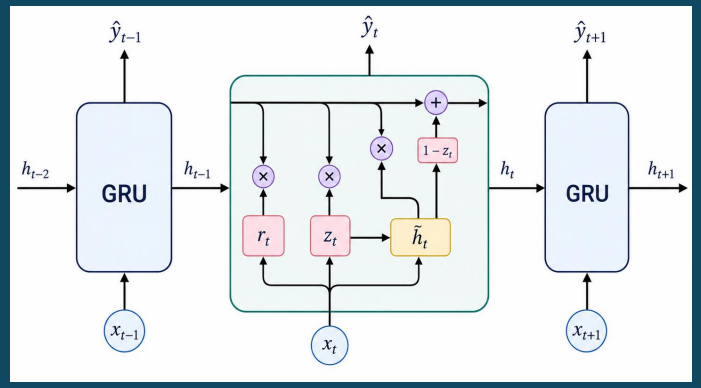
GRU

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right)$$

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right)$$

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

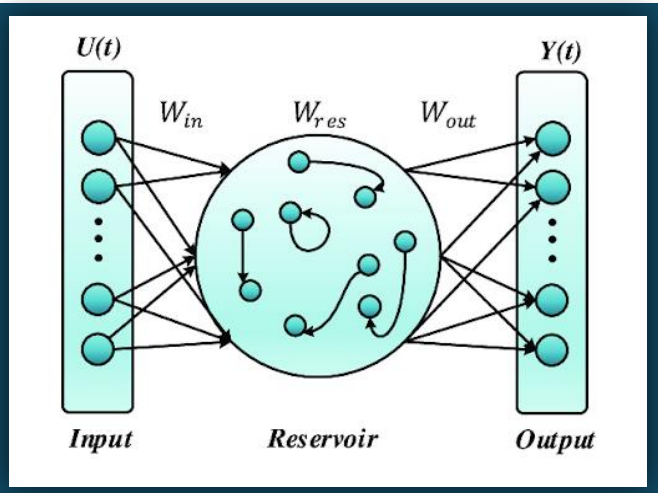


ESN

$$\boldsymbol{x}(t) = f\left(W_{\text{in}}\boldsymbol{u}(t) + W_{\text{res}}\boldsymbol{x}(t-1)\right)$$

$$\boldsymbol{y}(t) = W_{\text{out}}\boldsymbol{x}(t)$$

$$W_{\text{out}} = Y_{\text{target}}X^T\left(XX^T + \beta I\right)^{-1}$$



ESN: Hyperparameters

- Reservoir size (N)
  - Dimensionality of the dynamic state space.
- Leaking Rate (α)
  - Controls the timescale of the reservoir state update.
- Spectral Radius (ρ)
  - Controls the balance between memory capacity and sensitivity to new inputs.
- Ridge Regularisation (λ)
  - Prevents overfitting by penalising large readout weights.
- Input Scaling (s)
  - Controlling the strength with which the input drives the reservoir.
- Washout Period (t_wash)
  - Number of initial time steps discarded during training to allow the reservoir to forget its initial state.
- Sparsity
  - Connectivity sparsity (fraction of zero weights).

ESN: Baseline Hyperparameters

Hyperparameter	Value
Reservoir size (N)	1000
Spectral radius (ρ)	0.9
Sparsity	0.9
Leaking rate (α)	0.05
Input scaling	0.5
Ridge regularisation (λ)	10^{-1}
Washout period	500

Feature	Autocorrelation Timescale (steps)
AMOC	~301
PD_200m	~306
ICEFRAC	~362
SFWF	~343

Choice Criteria

- Autocorrelation timescales for features are around 300 steps which affects the leaking rate
- Standard conventions taken from literature based on problem size
- Mantas Lukoševičius' A Practical Guide to Applying Echo State Networks:
  - DOI: 10.1007/978-3-642-35289-8_36

Forecasting strategies (appendix)

- Teacher Forced:

$$\begin{bmatrix} x_1(\mathrm{t}) \\ x_2(\mathrm{t}) \\ x_3(\mathrm{t}) \end{bmatrix} = M(Data_1\,(t-W:t-1),\ Data_2\,(t-W:t-1),\ Data_3\,(t-W:t-1)) + \text{Noise}$$

- Auto-Regressive Forecasting:

- Single Model: $\begin{bmatrix} x_1(\mathrm{t}) \\ x_2(\mathrm{t}) \\ x_3(\mathrm{t}) \end{bmatrix} = M(x_1\,(t-W:t-1),\ x_2\,(t-W:t-1),\ x_3\,(t-W:t-1)) + \text{Noise}$

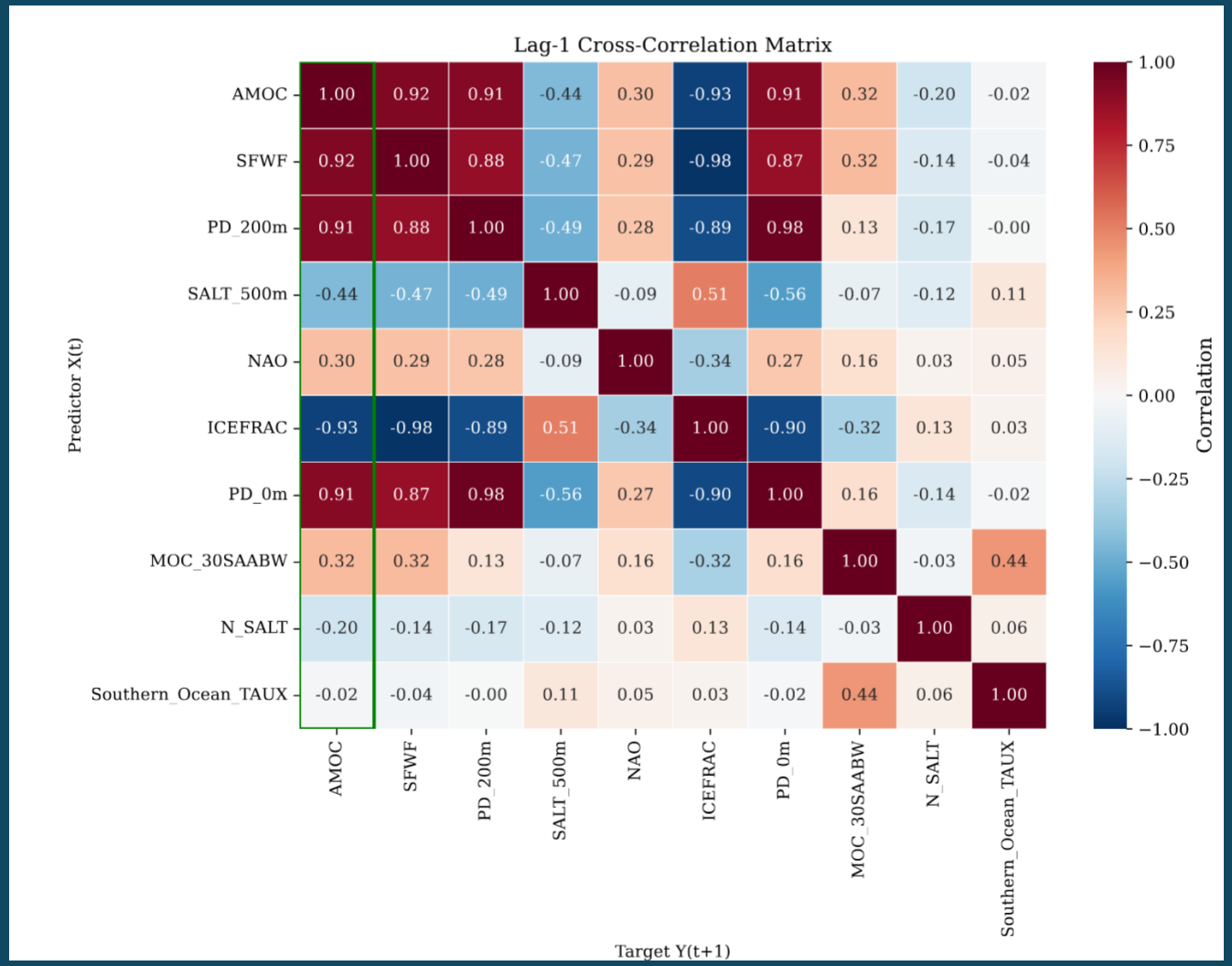
- Coupled Model:

$$\begin{cases} x_1(t) = M_1(x_2\,(t-W:t-1), x_3\,(t-W:t-1)) + \text{Noise} \\ x_2(t) = M_2(x_1\,(t-W:t-1), x_3\,(t-W:t-1)) + \text{Noise} \\ x_3(t) = M_3(x_1\,(t-W:t-1), x_2\,(t-W:t-1)) + \text{Noise} \end{cases}$$

- Noise (One MVG for on and off state independently):

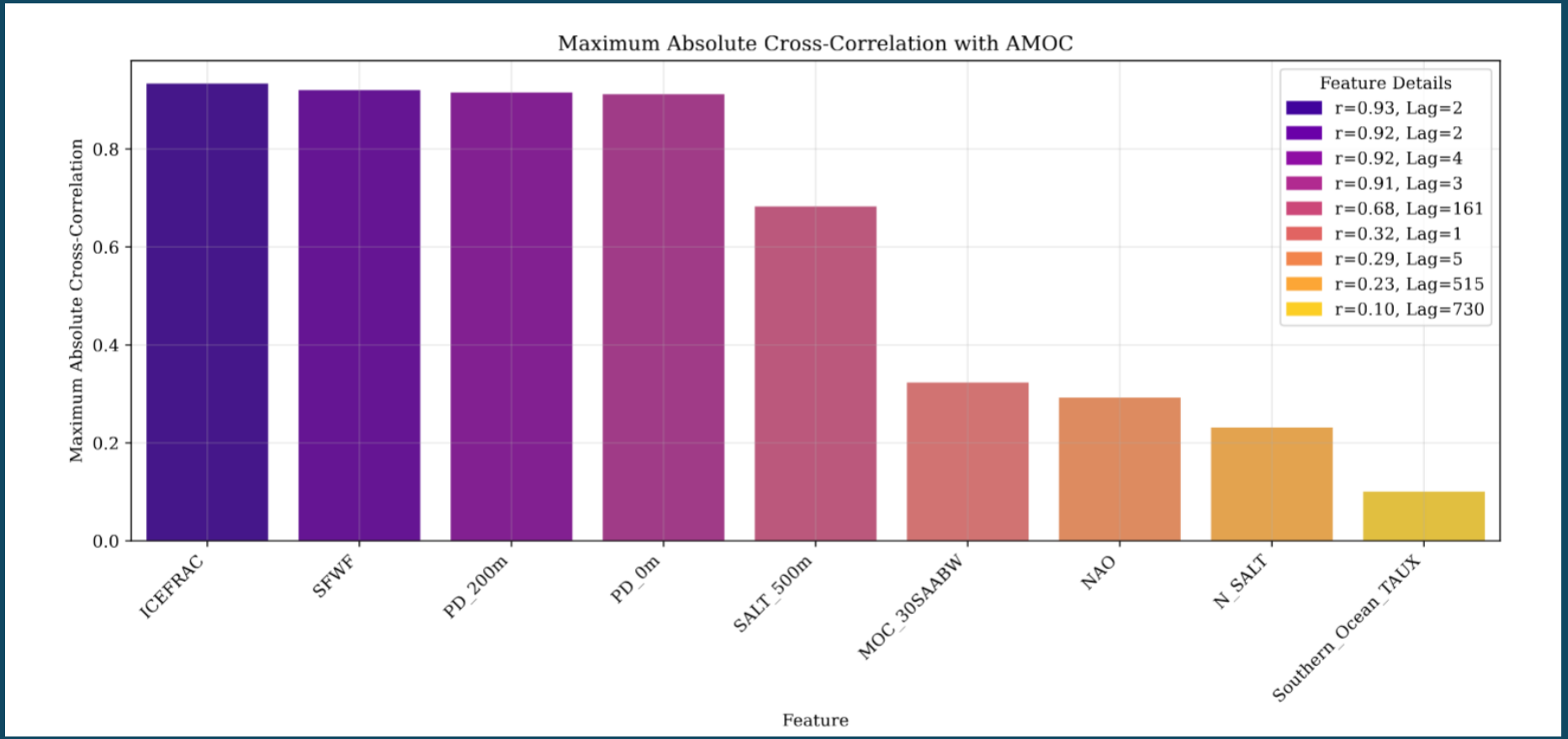
$$Noise(t) = k * MVG(AMOC(t)) + \rho * Noise(t-1)$$

Correlation matrix



Lag-1 cross-correlation matrix between features in dataset.

Maximum absolute cross-correlation



Maximum absolute cross-correlation between each candidate feature and AMOC. The lag at which the peak occurs is reported in the legend. ICEFRAC, SFWF, PD_200m, and PD_0m peak at short lags (2–4 years) with correlations above 0.90, while the remaining features show weak or delayed relationships and were excluded from further analysis.

Feature Selection Quantitative Metrics

Feature	MI Score	Feature Kept
PD_200m	0.8549	✓
ICEFRAC	0.7986	✓
PD_0m	0.7823	✓
SFWF	0.7637	✓
SALT_500m	0.4833	✓
NAO	0.1087	✗
MOC_30SAABW	0.0702	✗
N_SALT	0.0511	✗
Southern_Ocean_TAUX	0.0048	✗

Mutual information scores between each candidate feature at time t and AMOC at time t+1, computed using a k-nearest-neighbours estimator. Features with MI score below 0.11 (NAO, MOC_30SAABW, N_SALT, Southern_Ocean_TAUX) were excluded from further analysis.

Feature	VIF	Feature Kept
ICEFRAC	446.31	✓
PD_0m	280063012.78	✗
PD_200m	288493731.69	✓
SALT_500m	354122.16	✗
SFWF	31.28	✓

Variance Inflation Factors (VIF) for the high-MI candidate features, assessing multicollinearity. PD_0m was excluded due to near-perfect collinearity with PD_200m (VIF ~280M), and SALT_500m was excluded due to its high VIF and unsuitable lag structure for one-step-ahead prediction.

Model-Independent Feature Rankings (appendix)

- Tests Performed
  - Visual Inspection of Feature Distribution
  - Lag-1 Cross-Correlation
  - Peak Lag Cross-Correlation
  - Mutual Information (MI)
  - VIF

Dropped Features

Feature	Reason
PD_0m	Near-perfect collinearity with PD_200m (VIF ∼280M); lower MI
SALT_500m	Peak predictive relationship at lag 161, not lag-1; VIF 354K
NAO	Weak MI (0.109), weak lag-1 correlation (0.298)
N_SALT	Weak MI (0.051)
MOC_30SAABW	Weak MI (0.070)
Southern_Ocean_TAUX	Near-zero lag-1 correlation (-0.017)

Retained Features

Feature	Reason
AMOC	Primary target variable; defines regime state
ICEFRAC	Lag-1 $r = -0.933$ (strongest of all candidates), MI = 0.799, bimodal distribution
PD_200m	Highest MI (0.855), lag-1 $r = 0.914$, physically relevant at thermocline depth
SFWF	Lag-1 $r = 0.923$, MI = 0.764, bimodal distribution

Ablation Feature Test Results

Tables below present the full ablation results for the ESN architecture across all four predicted variables - AMOC, SFWF, PD_200m, and ICEFRAC - reporting mean ± standard deviation of NMSE, RMSE, and MAE over twenty random seeds. For each target, all subsets of the remaining candidate features were evaluated, with feature combinations ranked by mean NMSE. In every case, ICEFRAC appears in the best-performing input set, consistently improving predictive skill despite its high multicollinearity with other variables. The optimal input structure differs across targets, motivating the use of per-variable configurations in the coupled model

Table 12: Prediction performance for AMOC. Lower values indicate better predictive skill.

Feature Set	NMSE	RMSE	MAE
PD_200m + ICEFRAC	0.14499 ± 0.00267	0.7478 ± 0.0068	0.5932 ± 0.0050
SFWF + PD_200m + ICEFRAC	0.14922 ± 0.00478	0.7586 ± 0.0121	0.6013 ± 0.0081
SFWF + PD_200m	0.17387 ± 0.00347	0.8189 ± 0.0082	0.6516 ± 0.0068
SFWF + ICEFRAC	0.19294 ± 0.00569	0.8626 ± 0.0127	0.6874 ± 0.0102

Table 13: Prediction performance for SFWF. Lower values indicate better predictive skill.

Feature Set	NMSE	RMSE	MAE
AMOC + ICEFRAC	0.18945 ± 0.00090	$(4.63 \pm 0.01) \times 10^{-6}$	$(3.54 \pm 0.01) \times 10^{-6}$
PD_200m + ICEFRAC	0.19360 ± 0.00128	$(4.68 \pm 0.02) \times 10^{-6}$	$(3.57 \pm 0.01) \times 10^{-6}$
AMOC + PD_200m + ICEFRAC	0.19511 ± 0.00158	$(4.70 \pm 0.02) \times 10^{-6}$	$(3.58 \pm 0.02) \times 10^{-6}$
AMOC + PD_200m	0.23669 ± 0.00471	$(5.17 \pm 0.05) \times 10^{-6}$	$(3.94 \pm 0.06) \times 10^{-6}$

Table 14: Prediction performance for PD_200m. Lower values indicate better predictive skill.

Feature Set	NMSE	RMSE	MAE
AMOC + SFWF + ICEFRAC	0.16913 ± 0.01698	$(4.87 \pm 0.24) \times 10^{-5}$	$(3.50 \pm 0.13) \times 10^{-5}$
AMOC + SFWF	0.20006 ± 0.02602	$(5.27 \pm 0.34) \times 10^{-5}$	$(4.11 \pm 0.32) \times 10^{-5}$
AMOC + ICEFRAC	0.21468 ± 0.02184	$(5.46 \pm 0.27) \times 10^{-5}$	$(3.99 \pm 0.16) \times 10^{-5}$
SFWF + ICEFRAC	0.45439 ± 0.04530	$(7.99 \pm 0.39) \times 10^{-5}$	$(7.01 \pm 0.31) \times 10^{-5}$

Table 15: Prediction performance for ICEFRAC. Lower values indicate better predictive skill.

Feature Set	NMSE	RMSE	MAE
AMOC + PD_200m + SFWF	0.10660 ± 0.00127	0.0278 ± 0.0002	0.0209 ± 0.0002
AMOC + SFWF	0.10935 ± 0.00077	0.0281 ± 0.0001	0.0212 ± 0.0001
PD_200m + SFWF	0.11438 ± 0.00151	0.0288 ± 0.0002	0.0218 ± 0.0002
AMOC + PD_200m	0.14237 ± 0.00417	0.0321 ± 0.0005	0.0239 ± 0.0004

Ablation Study Feature Rankings

- Different subsets of the retained features used to predict all other features
- Asymmetric coupled model is recovered
- Novel prediction features for AMOC (ICEFRAC instead of SFWF)

$$\texttt{AMOC}(t+1) \leftarrow [\texttt{PD\_200m}(t),\ \texttt{ICEFRAC}(t)]$$
$$\texttt{SFWF}(t+1) \leftarrow [\texttt{AMOC}(t),\ \texttt{ICEFRAC}(t)]$$
$$\texttt{PD\_200m}(t+1) \leftarrow [\texttt{AMOC}(t),\ \texttt{SFWF}(t),\ \texttt{ICEFRAC}(t)]$$
$$\texttt{ICEFRAC}(t+1) \leftarrow [\texttt{AMOC}(t),\ \texttt{PD\_200m}(t),\ \texttt{SFWF}(t)]$$

Target	Best Input Features	NMSE
AMOC	PD_200m + ICEFRAC	0.14499 ± 0.00267
SFWF	AMOC + ICEFRAC	0.18945 ± 0.00090
PD_200m	AMOC + SFWF + ICEFRAC	0.16913 ± 0.01698
ICEFRAC	AMOC + PD_200m + SFWF	0.10660 ± 0.00127

Feature Specific ESN Hyperparameter Optimization

Hyperparameter	Value
Reservoir size (N)	2000
Spectral radius (ρ)	0.5
Sparsity	0.9
Leaking rate (α)	0.05
Input scaling	1.0
Ridge regularisation (λ)	10^{-1}
NMSE	0.04751 ± 0.00045
RMSE	0.9663
MAE	0.7543

Table 8: Optimal ESN hyperparameters for `AMOC` prediction.

Hyperparameter	Value
Reservoir size (N)	500
Spectral radius (ρ)	0.7
Sparsity	0.9
Leaking rate (α)	0.1
Input scaling	1.0
Ridge regularisation (λ)	10^{-1}
NMSE	0.05761 ± 0.00021
RMSE	0.0000
MAE	0.0000

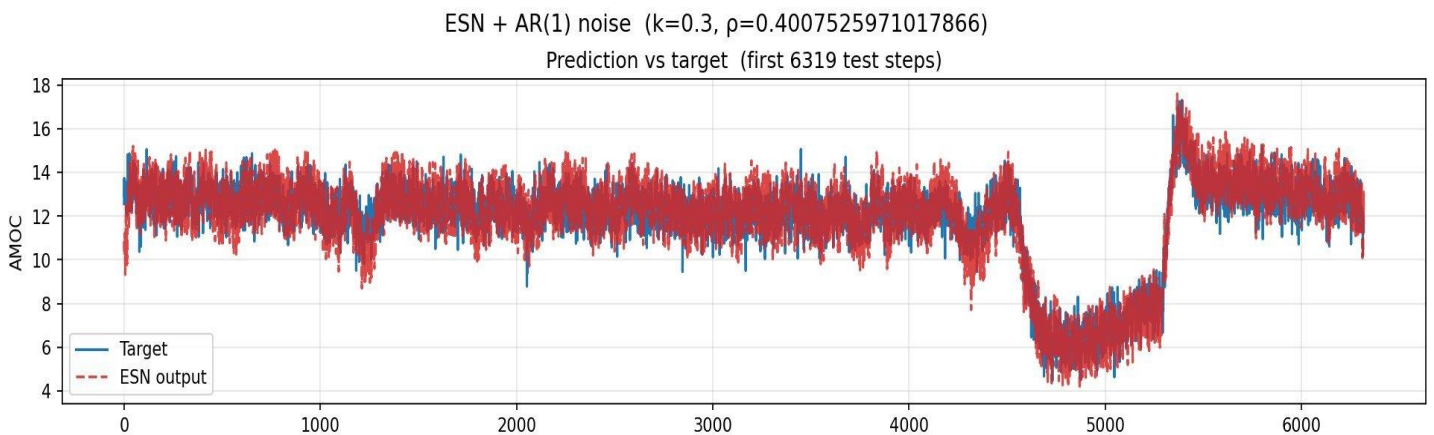
Table 9: Optimal ESN hyperparameters for `SFWF` prediction.

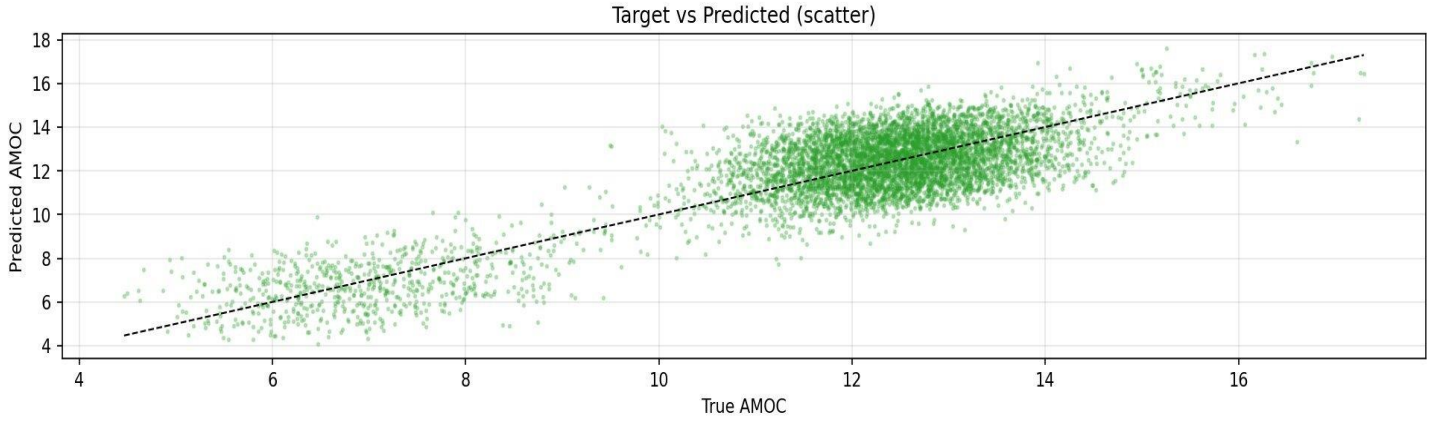
Hyperparameter	Value
Reservoir size (N)	500
Spectral radius (ρ)	0.7
Sparsity	0.9
Leaking rate (α)	0.1
Input scaling	1.0
Ridge regularisation (λ)	10^{-1}
NMSE	0.03551 ± 0.00095
RMSE	0.0255
MAE	0.0186

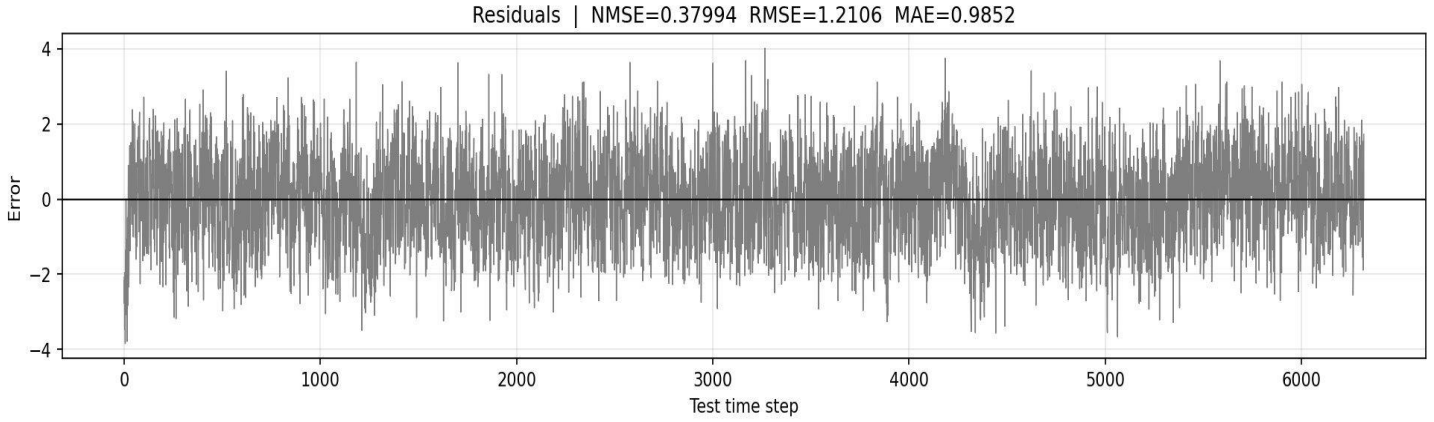
Table 10: Optimal ESN hyperparameters for `ICEFRAC` prediction.

Hyperparameter	Value
Reservoir size (N)	500
Spectral radius (ρ)	0.95
Sparsity	0.9
Leaking rate (α)	0.01
Input scaling	1.0
Ridge regularisation (λ)	10^{0}
NMSE	0.03454 ± 0.00569
RMSE	0.0000
MAE	0.0000

Table 11: Optimal ESN hyperparameters for `PD_200m` prediction.







The model: AR(1) Single ESN V1

Input variables:
- AMOC
- ICEFRAC
- PD_200m

Forecasting setup:
- Teacher forced
- Auto-regressive Forecasting

Noise modelling:
- Multivariate Gaussian noise for on/off states
- Samples constrained within 1σ for each variable

Conclusion:
- With noise, the model still learns the transition trends between the two states and an autoregressive model can be achieved although a worse accuracy is obtained as the NMSE has increased.

Coupled ESN V2 - Optimizing

Investigating the interplay between k and ρ is well worth our time to observe how well the model performs after tuning

Observed AMOC Fraction: 0.486

Configuration	Mean Fraction	Std. Fraction	Mean Transitions
No noise	0.361	0.000	12.0
White noise ($k = 0.3$)	0.376	0.145	7.1
AR(1) ($k = 0.3$, $\rho = 0.5$)	0.320	0.137	6.8
AR(1) ($k = 0.5$, $\rho = 0.5$)	0.495	0.146	4.3

- 50 runs were used for each of the previously observed scenarios to gauge where the optimum may lie
- To quantify the accuracy, the distributions between the real AMOC data and the predictions were compared, to observe when they are in each regime

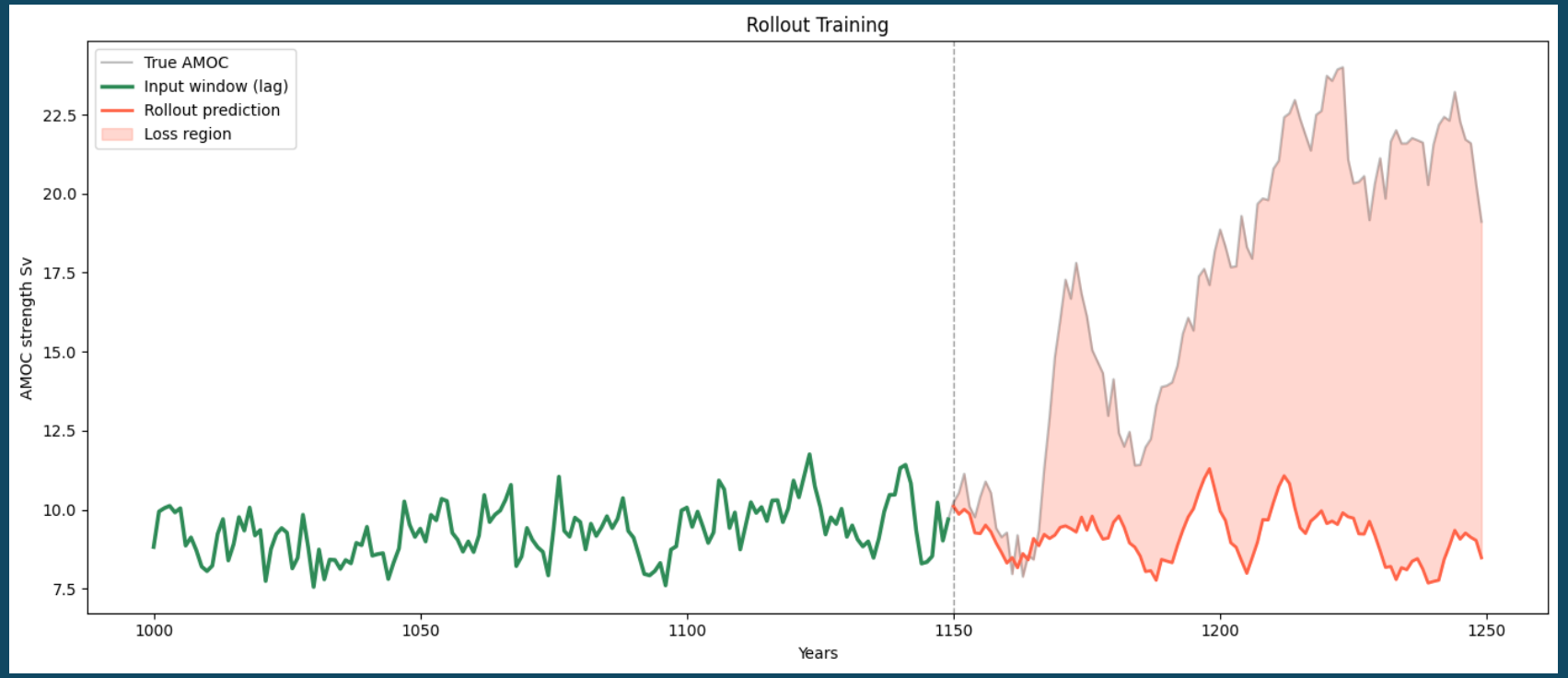
GRU Single - Rollout training

1: Learn the broad dynamics first with brief teacher forced training
2: Fine tune with rollout training

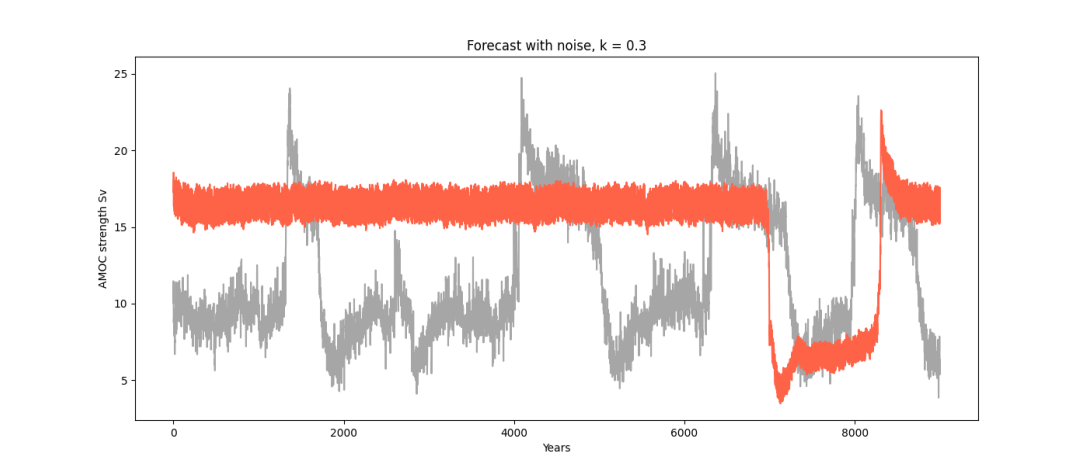
At each rollout training step:
- Sample a window containing a transition
- Let the model predict X steps autoregressively and include all errors in the loss

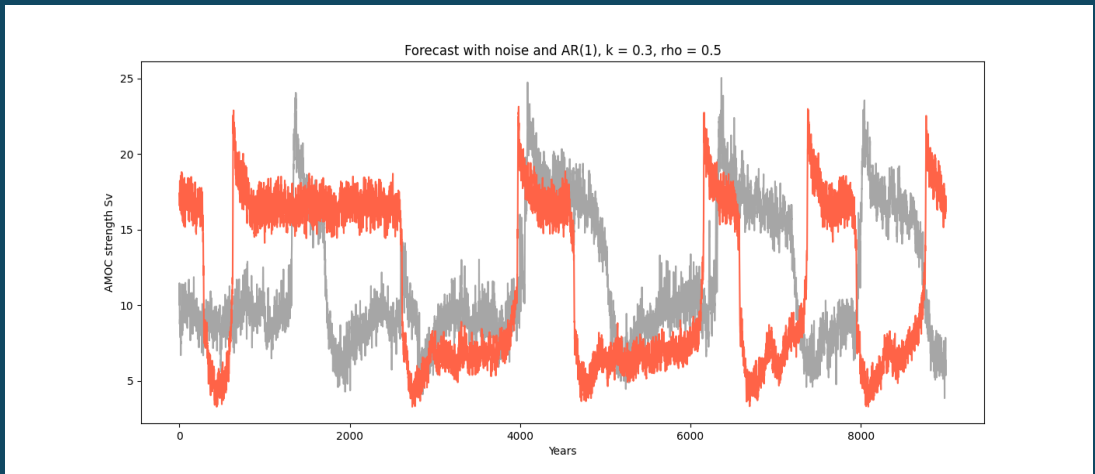
=> Fix exposure bias, learn from it's own errors, & hopefully teach the model what is important for a transition?

A lot of new hyper parameters....
- Search ahead for transition
- Threshold to identify transition
- BPTT or not (Too noisy?)?
- Rollout Horizon
- Number of Rollout training steps



GRU Single Model prediction results and HPs



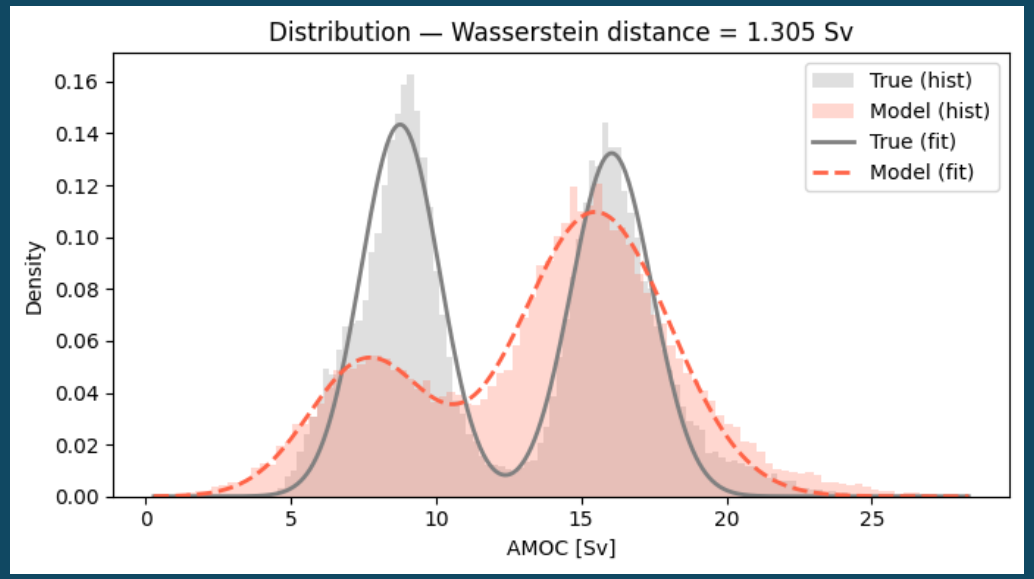


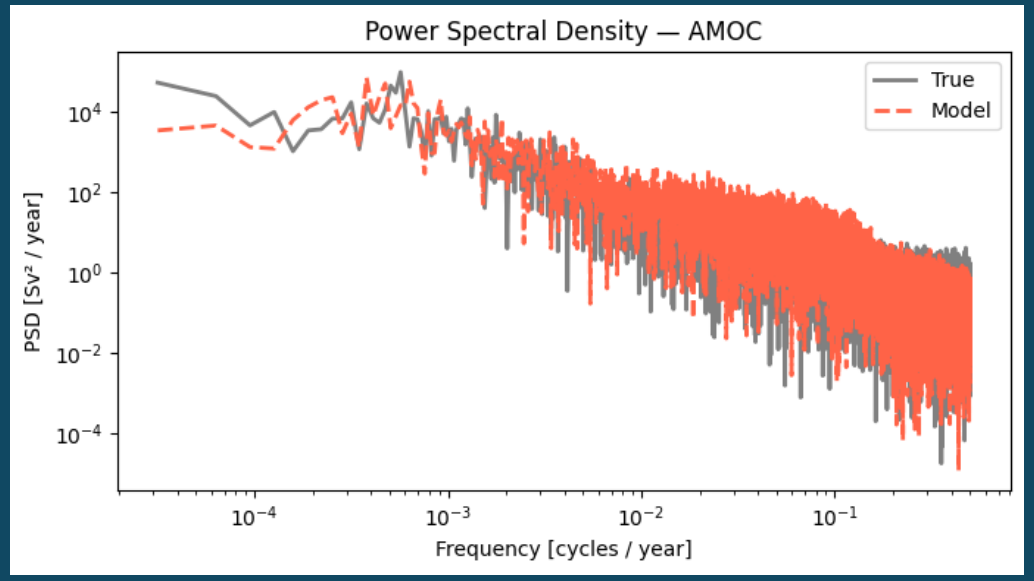
Features	Config	Training	Model HPs	Training HPs	Prediction HPs
AMOC	Window: 150 yrs	Teacher Forced	Hidden size	Epochs	K
SFWF	Horizon: 0 yrs	Rollout	N layers	Rollout window sampling	Rho
PD_200m			Dropout	Rollout train horizon	
			Lr-pretrain	Rollout train iterations	
			Lr-rollout	BPTT	
				Gradient clipping	

GRU Single evaluation

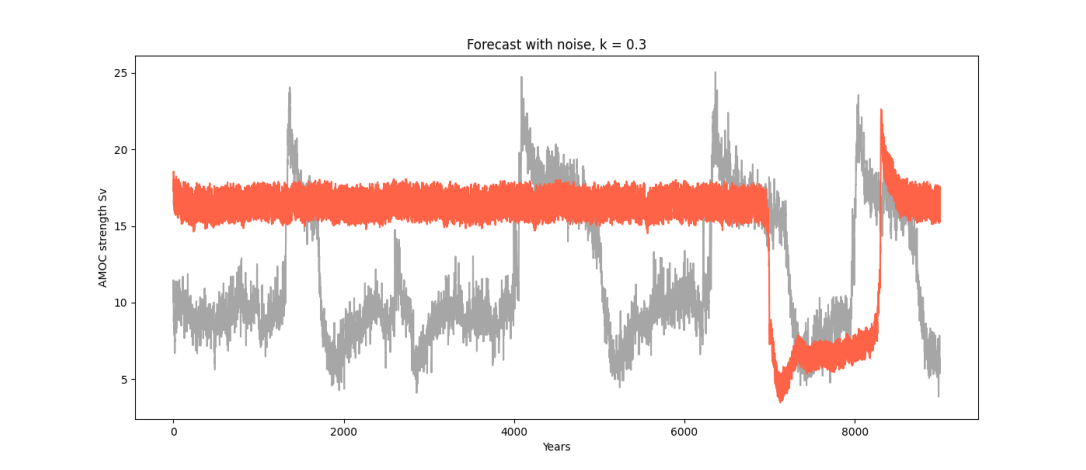
Diagnostics:

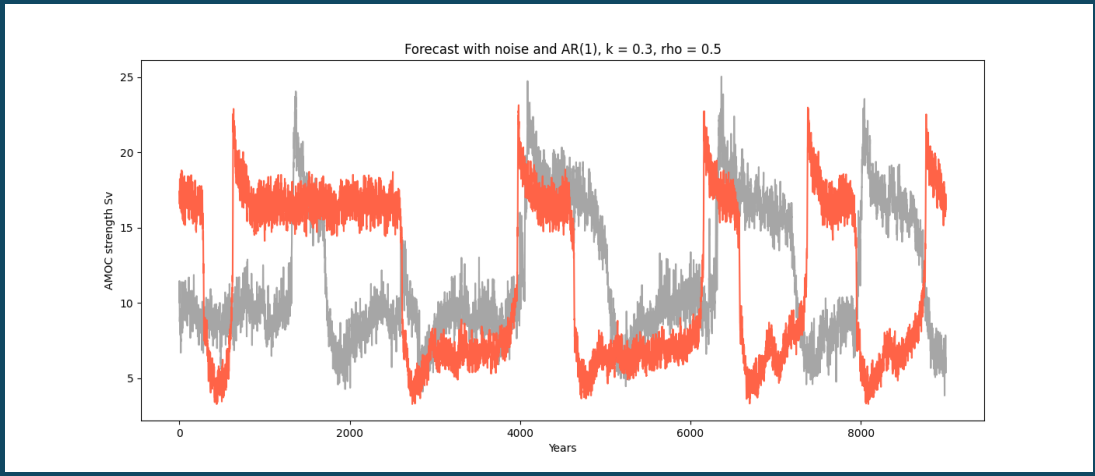
- Model spends too much time in the on state
- Off-state regime is too weakly represented
- Noise level is comparable to the true signal
- Dominant spectral peaks lie slightly below data





GRU Coupled Model prediction results and HPs



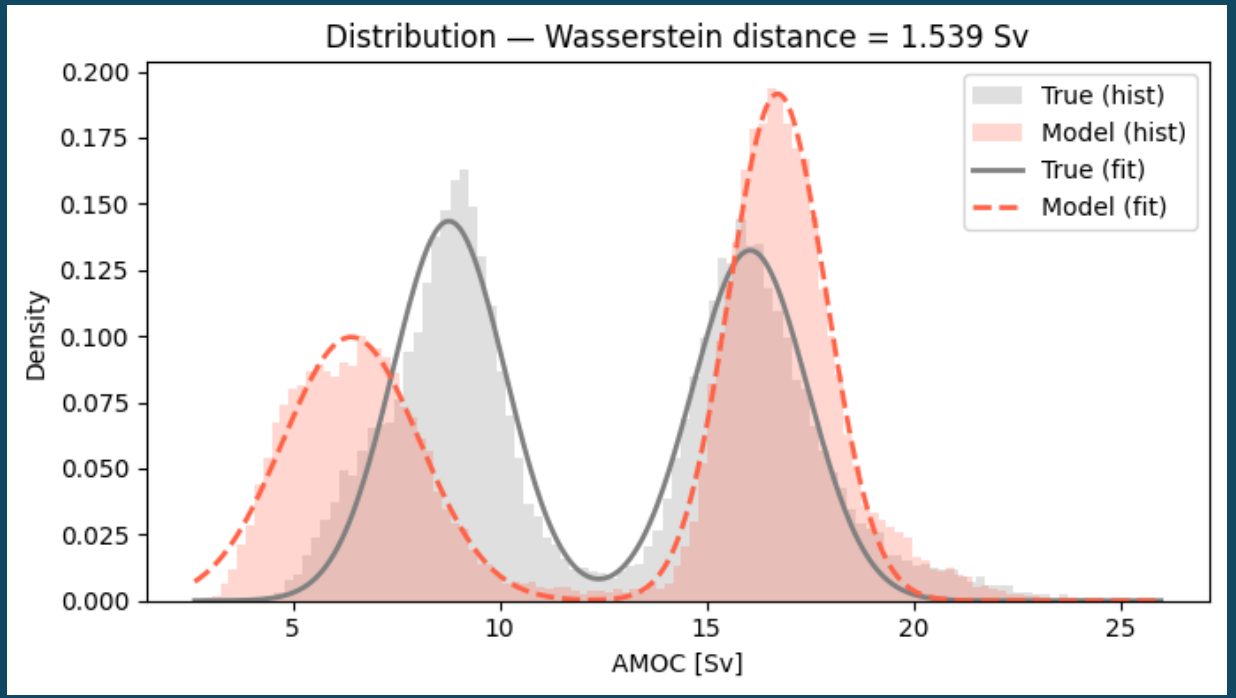


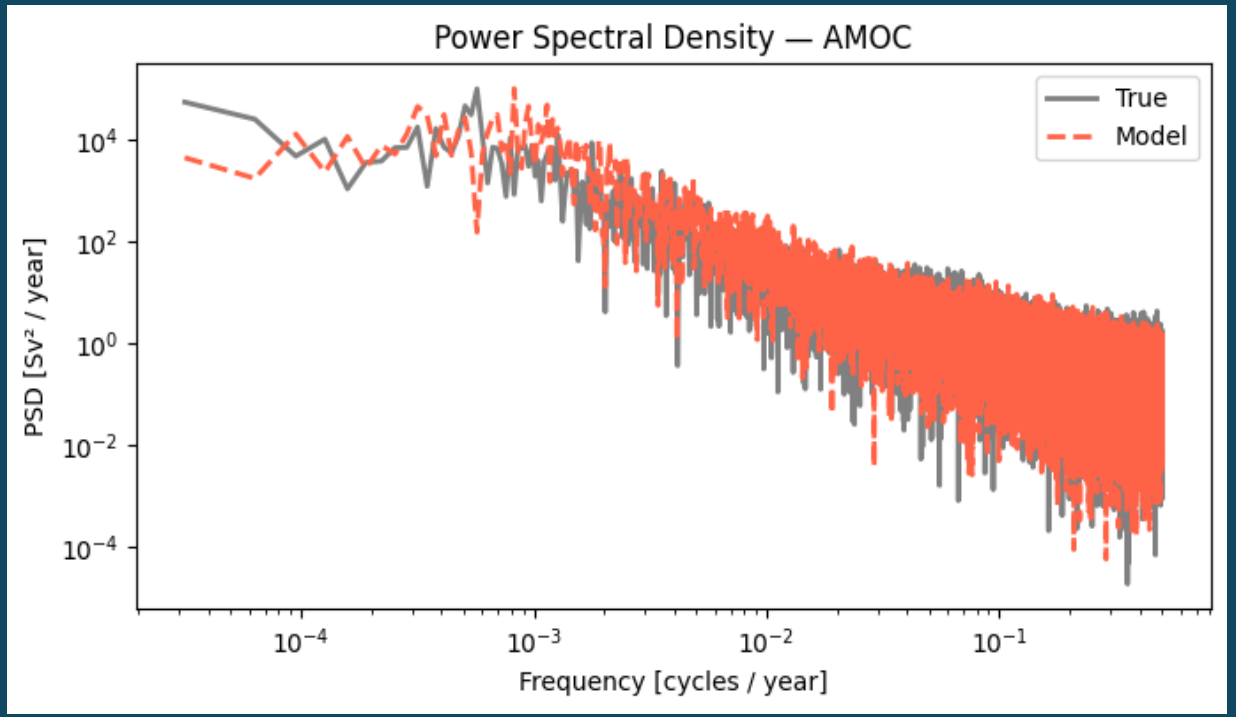
Features	Config	Training	Model HPs	Training HPs	Prediction HPs
AMOC	Window: 100 yrs	Teacher Forced	Hidden size x 3	Epochs x 3	K
SFWF	Horizon: 40 yrs		N layers x 3		Rho
PD_200m			Dropout x 3		
			Lr x 3		

GRU Coupled evaluation

Diagnostics:

- Model spends too much time in the on state
- Off-state regime is too weakly represented
- Noise level is comparable to the true signal
- Dominant spectral peaks lie above data



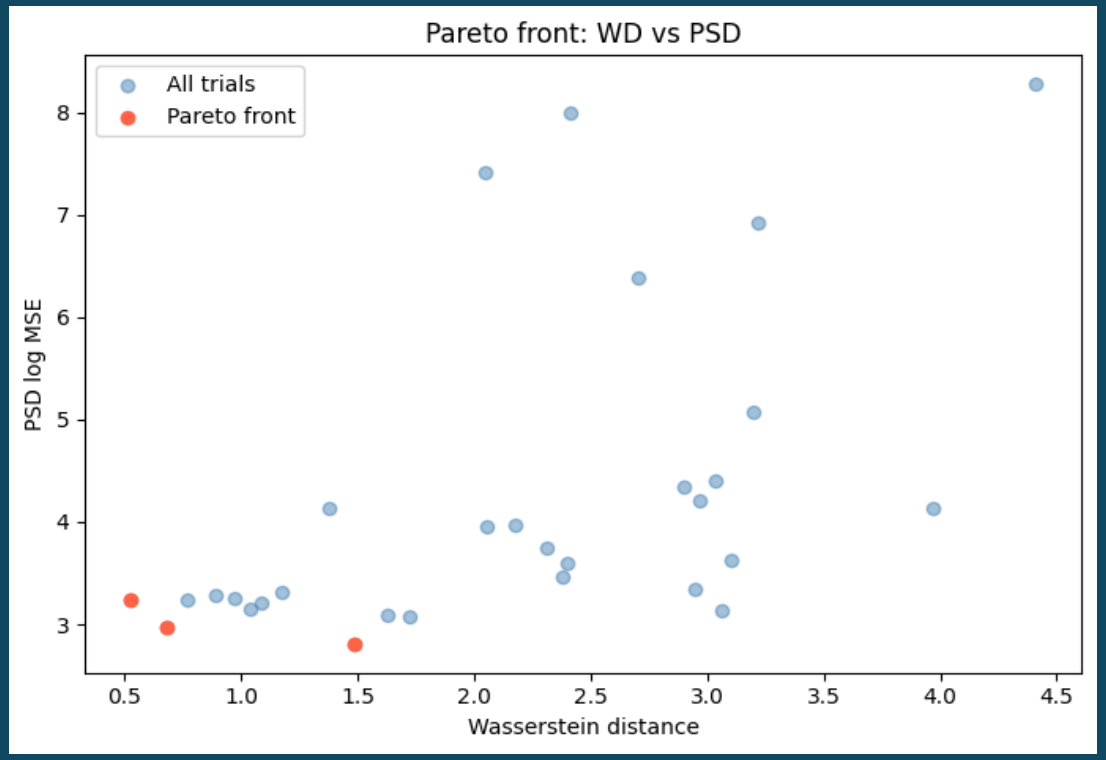


GRU Single – Optuna optimization study

How do we tune all our parameters, and towards what target?

- Does a model that performs well on teacher-forced predictions do well auto-regressively?
- Do we treat all parameters like a dynamical system and tune them all at once?

Model	Training	Prediction
Hidden size	Pretrain epochs	Noise strength (K)
Lr pretrain	Rollout steps	AR (rho)
Lr rollout	Rollout window sampling	
Dropout	Rollout horizon	
Input Window		



- Seeds were used for the rollout window sampling for reproducibility
- The objective function averaged the results over 3 different seeds of 30 000 time step predictions

Optimization results not reproducible

Controlled random sources:
- Rollout windows are controlled by a seed
- Prediction noise is controlled by a seed

Uncontrolled random sources:
- Weight initialization
- Pretraining shuffling by torch
- Dropout

Could this possibly make such a large difference or are we dealing with a very, very sneaky bug?