

Summary

The authors explore whether deep-learning emulators can be used to replace the marine biogeochemical model ERSEM in a one-dimensional GOTM–FABM–ERSEM configuration at station L4 in the western English Channel. The motivation is that coupling full biogeochemical models is expensive, and AI based tools could provide a lightweight alternative with better forecasting capabilities. Although 1D models don't need emulation, this work aims to serve as a proof of concept. The manuscript presents two emulators: an LSTM that predicts a selected set of surface and vertically integrated state variables, and a physics-informed 1D CNN that predicts the vertical state of 52 pelagic variables. The emulators are evaluated in 10-day forecasts, in a 22-year run, and in a 7-year forecast after training on a reanalysis. The authors also test the robustness of the emulators to noisy physics inputs, and use a saliency analysis to interpret what drives the predictions of the LSTM.

General assessment

I believe this is a timely and relevant study. Representing marine biogeochemical models as emulators using AI is a clear future pathway to enable coupling to ocean physics models. AI emulation is in its early stages, and this work aims to demonstrate and report such coupling in a 1D setup as a proof of concept. The openly available code is transparent (although the link in the manuscript does not seem to work: <https://github.com/JOZSKA/AI-1D-model-emulators-at-L4>). I believe the work itself is a valuable contribution and a clear step forward in this field. The work demonstrates that it can at the very least reproduce the free run. The manuscript is also well written. However, I think several of the main claims regarding the outperforming of the forecasting system, the transferability to 3D systems and the ability to predict extreme events are not yet fully supported by the analyses presented. This is not to imply that the emulator work itself is incomplete or not well executed, but rather that some of the statements in the abstract and conclusions seem to go beyond what the results currently show.

General comments

1. Benchmarking against persistence at 10 days

The 10-day forecasts of the emulator (Fig. 10) are only compared against persistence. The persistence does not account for the seasonal change over the forecast window, so for a strongly seasonal site such as L4 and using a 10-days window, how much of the skill of the emulator could come from know to which season its heading? The authors note themselves that the seasonal signal is "an easily learnable signal" (Sect. 2.1.4). A forecast that includes a seasonal signal does not require an AI emulator, an adding such benchmark (e.g. climatology) would show what the emulator has learned beyond this and it would be valuable to see how the emulator performs against such a baseline in addition to persistence.

A comparison with the L4 observations would also help to put the emulation errors into perspective. If the differences between the emulator and ERSEM are small compared with the differences between ERSEM and the observations, it can strengthen the case for the emulator, as it would

predict the real ocean almost as well as ERSEM itself, while, in the case of the LSTM, being a reduced version of it.

2. Comparison with the reanalysis-trained forecast

I think the comparison with the reanalysis-trained forecast is not as strong as it can be. If I understand correctly, it compares an emulator trained on the reanalysis against a free-running simulation started from the reanalysis. ERSEM has seasonal biases, which are acknowledged by the authors, and these can re-emerge within months to years. In that case, the comparison mainly measures what the emulator has learned from the bias correction in the reanalysis, while the free run only starts from a corrected start point. This could mean that the results come at least partly from bias correction, which could also be obtained without an AI emulator. Perhaps a fairer benchmark would be to have the free run corrected for its mean seasonal bias relative to the reanalysis. This would show whether the emulator learns more than a fixed bias correction, for example that bloom size depends on the conditions in a given year.

Comments 2. Is not to imply that the emulators or their implementation is incorrect. It is rather that the numbers stated in the abstract and the conclusion are likely depend on the baseline it is tested against. It would be valueable to this work to see how the emulator would perform against a bias-corrected baseline, which is likely also a more realistic version of a forecast to compete against, compared to a free run.

3. Extreme events

The abstract states that the performance of the emulators is evaluated on their ability to reproduce extreme events, but I could not clearly find such an evaluation in the manuscript. I appreciate that the forcings are hindcast, and extreme events are indeed present in the forcing and in the simulator output, but whether the emulator reproduces does not seem specifically evaluated in the results. The authors argue that RMSE emphasises extreme events in Sect. 2.4, but a lower RMSE does not show that extreme events are reproduced, while the abstracts suggest that such analysis will be explicitly done. For example, the bloom peaks appear clipped in the emulator. In Fig 3, the ERSEM surface chlorophyll peaks reach roughly 3.5–4.0 mg m⁻³, whereas the LSTM peaks, where visible, appear noticeably lower.

4. Mixed results of the reanalysis-trained emulators (Fig. 12)

"The deep learning emulators trained on the reanalysis and run for the 2019-2025 7-year forecast are inherently correcting the ERSEM seasonal biases, and therefore produce much more skilled forecast than the free run simulator, when compared to the assimilated satellite observations, or the reanalysis (Fig.11:A-B and Fig.12:A,C)" (Sect. 3.3)

The text states that the emulators are better than the free run for most of the predicted variables (Fig. 12). However, it seems to me that the LSTM is worse than or almost equal to the free run for 5 of the 9 variables shown, most notably nitrate and silicate, with smaller differences for phosphate, surface oxygen and bottom oxygen.

Could the authors discuss why? (Does it have to do with the fact that the nutrients have been relaxed towards the climatology?)

5. Transfer of the conclusions to the three-dimensional case

In the introduction, the authors state that they expect that most of the conclusions derived for the one-dimensional demonstrator presented in this study will translate into the three-dimensional case.

This statement however stays implicit, and it is not explained why these results would translate to three-dimensional case. I am not implying that they do not translate, but a discussion on why the authors believe this would significantly strengthen these statements. The missing processes (horizontal advection and river discharge) are acknowledged, but their consequences for the emulator are not discussed. For example

Temperature and salinity are relaxed monthly towards the L4 observations, and the four nutrients towards an L4 seasonal climatology (Sect. 2.1.3). The nutrient relaxation keeps the nutrients close to the same seasonal cycle every year, damping their year-to-year variability and removing trends; the authors attribute the small winter nutrient variability to this (Sect. 2.1.4). The relaxation terms act as sources and sinks (standing in for river input, lateral transport, perhaps sediment-water exchange) that are not resolved by the model and are not supplied to the emulators. In 3D, these terms are replaced by explicit, time-varying advection and river input, which an emulator would also need as inputs. The authors acknowledge that "the modelled seasonality is likely exaggerated due to the absence of horizontal transport and riverine inputs.". I think this can be a reason why the step to 3D may be harder, as the strong seasonal cycle is part of what makes the 1D problem easy to learn.

The LSTM, which carries most of the positive results, emulates only 20 surface and vertically averaged variables. So as far as I understand it can therefore not be coupled to a 3D physics model as a replacement for ERSEM. The CNN provides the full state, but it is also less robust. Could the authors discuss which emulator they seem fit for a for the 3D case and why and what would and what would not translate to a 3D case

6. The 1D CNN

Most of the results rest on the LSTM emulator, but some results are also attributed to the 1D CNN. The authors present "a single well-performing 1D CNN model", chosen from a set of runs that differ substantially, which is thus likely a best-case scenario. It would help to report:

- how many seeds were trained
- the spread in short-range and decadal skill across the seeds
- and which data were used to select the presented run

The conclusions state that the CNN runs are "capable of robust short- to medium-term predictions". However, I believe Fig. A9 seems to show that the CNN is worse than persistence at nearly all lead times for nearly all variables.

7. The noise experiment (Fig. A8)

Looking at Fig. A8A, the normalized SST seems keeps its phase. Is the noise only applied in amplitude? If so, this does not fully represent the errors that a physics model also make, including as phase errors.

If only applied to the amplitude, I believe this connects to the authors' own interpretation of SST as a "proxy for seasonality" (Sect. 3.4). If the LSTM mainly uses the physics inputs as when things happen rather than how much, you would expect what was found: relative insensitivity to changes in the amplitude. Perhaps a test would be to shift the physics inputs in time and examine the effect on, for example, the timing of the blooms.

8. The explainability analysis

The XAI analysis is a nice idea, but I think the saliency figures (Fig. 13 and Fig. A10) need careful explanation, as they are easy to over-read. The inputs are not all independent predictors: some of them are driven by others. The authors acknowledge this specifically for the biogeochemical inputs, but not specifically for the physical inputs. For example, the MLD results from atmospheric forcing.

As I understand it, gradient-based saliency measures how the output responds to a change in one input while all other inputs are kept fixed. How does this work for inputs that depend on each other? It would be good to further explain this so that the result can be interpreted better, especially as understanding what drives what is an important point of using such coupled biogeochemical-physical ocean models. For example, the MLD is largely set by the atmospheric forcing (wind, heat flux and freshwater input), accumulated over the preceding days to weeks. If the MLD carries this information, does the network not need the wind or heat flux separately? This would make their sensitivities small even they are underlying drivers. The authors write that "the relatively high importance assigned by the LSTM to mixed-layer depth, compared with atmospheric forcing variables, may indicate that bloom timing is more strongly linked to water column structure than to short-term variations in atmospheric forcing" (Sect. 3.4). Could this not mean that the MLD is simply a more convenient summary of the atmospheric forcing for the emulator to learn?

If so, it would help to indicate which inputs are physically coupled, for example by reporting the correlations between the inputs, or by grouping related inputs (e.g., MLD with wind and heat flux).

9. Deviations in 2025 consistency

The year 2025 is left outside of (some) the analysis in Figures 6, 9 and 10 due to the degraded emulator run, but are kept in the trend lines of Figs. 7, 8, A5 and A6. Perhaps 2025 can be treated more consistently?

Specific comments

- "highly skilful in reproducing ERSEM simulations in both decadal climate projection" (abstract)

The 2004–2025 run is a hindcast forced by ERA5, with the physics prescribed from the simulator. I believe this is different than a climate projection.

- "the emulator's ability to accurately predict the timing of phytoplankton Spring blooms several years in advance" (abstract)

This is conditional on high-quality known future physics and weather (this is acknowledged in the conclusion, but not in the abstract), and the timing is not predicted equally well in all years.

- "the emulators can outperform ERSEM forecast skill score for several key variables, including phytoplankton and zooplankton, by 50–60%" (abstract)

Relative to..? The baseline (a 7-year free run without data assimilation) should be specified.

Also, does this hold only for the LSTM? For the CNN (Fig. 12C), surface zooplankton improves by only ~20%, and vertically averaged zooplankton is about 2.5 times worse than the free run.

- "including the ability to reproduce daily anomalies and extreme events" (abstract)

Extreme events are not specifically evaluated.

- "we expect the key conclusions of this study to remain valid" (conclusions, regarding the 3D case)

This is not supported by evidence or a discussion

- "by elucidating the key drivers of Spring bloom formation and hypoxic conditions" (introduction)

As stated in Sect. 3.4, L4 does not experience hypoxia, so I believe the analysis rather looks at the seasonal cycle of bottom oxygen than hypoxia

- Beyond having the potential to be better at forecasting, a main argument for AI emulators is their ability to reduce computational expense in a coupled framework. This is acknowledged by the authors in the introduction: "One key motivation for deep learning-based forecasting systems is their ability to substantially reduce computational expense" (introduction). A 1D model does not need emulation, and the authors acknowledge this also: "Although one-dimensional model is computationally cheap and does not need to be emulated" (introduction). However, if there is already a meaningful difference, it would be interesting to see and report the time it takes to run the setup with the emulator and the fully coupled run. If there is a difference, it could strengthen this statement in the introduction.
- "For each prediction, the model receives a sequence of the previous 30 daily average values (lookback window = 30) of both state and forcing variables" (section 2.2.1)

How was this window chosen, and were other windows considered? Is this a sweets pot for capturing the seasonal signal, and if so, how would the emulator respond to other windows?

Technical corrections

Figures

- (see main comment 2) In Fig. 3, the bloom peaks appear clipped in the emulator. Unless I missed it, this underestimation is not mentioned in the text.
- Figs. A5 and A6: the captions appear shifted. Fig. A5 shows yearly correlations, so it is the CNN version of Fig. 7, not Fig. 6. Fig. A6 shows the normalized RMSE, so it is the version of Fig. 8, not Fig. 7.
- In Fig. 7: the correlation for bottom oxygen increases with lead time, while it decreases for all other variables. This seems to be driven mainly by a single low value in 2018. Could the

authors comment on this, and give the uncertainty of the fitted trends, which are based on only ten yearly values (including the anomalous year of 2025, see main comment 9)?

- Fig. A9: the y-axis label reads MAE (LSTM/persistence), but the caption describes the RMSE for the CNN.
- Figs. 1 and A1: the red solid and black dashed lines are not defined.

Text

- pycnocline → pycnocline (Sect. 2.1.4)
- biogeochemistry → biogeochemistry (Sect. 3.4)

Other

The github link currently in the manuscript does not seem to work for me: <https://github.com/JOZSKA/AI-1D-model-emulators-at-L4>