# On the Generalization of Deep Neural Networks for Optimal Sensor Placement in Global Ocean Forecasting

**Alexander Lobashev**[1]
Alexander.Lobashev@skoltech.ru

**Nikita Turko**[2]
NikitaTurko@yandex.ru

**Konstantin Ushakov**[2,3]
ushakovkv@mail.ru

**Maxim Kaurkin**[2]
sherema@yandex.ru

**Rashit Ibrayev** [4,2,3]
ibrayev@mail.ru

[1]Skolkovo Institute of Science and Technology, Moscow 121205, Russia
[2]Shirshov Institute of Oceanology, Russian Academy of Sciences, Moscow 117997, Russia
[3]Moscow Institute of Physics and Technology, Dolgoprudny 141701, Russia
[4]Marchuk Institute of Numerical Mathematics, Russian Academy of Sciences,
Moscow 119333, Russia

## Abstract

The focus of this study is on the generalization of neural networks, particularly in the context of sensor placement for global climate models' forecasts. The goal is to determine if sensor placement strategies derived through training a deep learning model, which is tasked with reconstructing a physical field from a set of measurements, can be effectively applied to a real high-resolution ocean global circulation model. The research compares different sensor placement methods, including one achieved using the Concrete Autoencoder method. Through modeling under varied initial conditions of the World Ocean state, it was found that sensor placements informed by deep learning methods outperformed others in forecast accuracy when using a comparable number of sensors. This finding underscores the potential of deep learning-informed sensor placement as a powerful tool for refining the predictive capabilities of global climate models and accelerating the data assimilation system without extensive revisions to their source code.

## 1 Introduction

**Problem of Integration of Deep Learning Approaches into Climate Models**. Integrating deep learning models into contemporary climate modeling systems presents considerable challenges. Most of these climate models, including the Community Earth System Model (CESM) (1; 2; 3), the Energy Exascale Earth System Model (E3SM) (4; 5; 6), and the Earth System Model 4 (ESM4) (7; 8), have been developed in Fortran. They are optimized for running on CPU clusters, employing from thousands to hundreds of thousands of cores. Implementing deep learning methodologies, particularly for aspects such as turbulence parametrization, within these established systems is technically daunting. This complexity stems from the extensive development time required for a single climate model, often spanning 5 to 20 years, leading to a massive codebase with millions of lines of code. Integrating deep learning approaches necessitates a significant overhaul of sections related to memory and parallelization management to ensure compatibility and enable the integration of GPU accelerators. Specifically, the incorporation of more recent techniques, such as physics-

DLDE-III Workshop in the 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

informed neural networks or deep learning-based solvers, would necessitate substantial effort and resources.

**Potential of Integrating Deep Learning Approaches into the Data Assimilation Subsystem of Climate Models**. There exists a crucial component within climate models — the data assimilation system — that has a significant impact on enhancing the accuracy of model solutions. Historical research indicates that a climate model equipped with a data assimilation system, even with less sophisticated turbulence parametrizations, often outperforms models with advanced turbulence parametrizations but no data assimilation (9; 10). Data assimilation systems are deeply integrated with climate models' atmospheric and ocean modules, complicating the incorporation of deep learning without extensive revisions. Consequently, ocean models like HYCOM (11; 12) continue using traditional methods such as the ensemble Kalman filter (EnKF), while atmospheric models like those at the ECMWF (13; 14) employ 4D variational assimilation (4D-Var). As the volume of data from diverse sources such as satellites, weather stations, ships, and buoys grows, the speed of data assimilation is increasingly becoming a constraining factor. This issue is especially pronounced for ocean models, as the majority do not include adjoint models necessary for the 4D-Var method. The complexity of ocean dynamics and the corresponding computational intensity make the integration of adjoint models particularly challenging. Consequently, the EnKF method is often favored for these models despite 4D-Var's suitability for assimilating large volumes of observations. In practice, to mitigate this limitation, contemporary global ocean models often perform data assimilation just once per simulation day, processing the cumulated data, a compromise that could potentially affect the forecast's precision. Although there have been some efforts to harness deep learning to improve the speed of these data assimilation systems (15; 16), these have been confined to simpler models, like the Lorenz system or a two-layer quasi-geostrophic channel model.

**Accelerating Data Assimilation via Feature Selection.** One method of accelerating data assimilation systems, without compromising the ability to process a large number of measurements, involves preprocessing the data to filter out irrelevant measurements. For instance, if one obtains a 2D field of sea surface temperature from satellite observations, covering the entire global ocean, the challenge of speeding up the data assimilation system reduces to a problem of optimal sensor placement. However, the classical formulation of the optimal sensor placement problem aims to minimize the reconstruction error and reduce the number of sensors, thereby improving data efficiency (17). Notably, the reconstruction error is computed simultaneously with the time the measurements are taken. This scenario is distinct from the traditional data assimilation workflow. In typical settings, data assimilation, serving as the correction of physical fields, is performed once per simulation day. The solver for the thermohydrodynamic equations then operates for the next 24 simulation hours. After these 24 hours, the forecast error is evaluated. This practice stands in contrast with optimal sensor placement techniques, which determine reconstruction errors right as measurements are taken, marking a clear methodological difference. This contrast raises the critical research question:

- **Generalization:** Does sensor placement, determined by *immediate reconstruction errors*, lead to improved *forecast accuracy* in real-world ocean forecasting?

Our research is aimed to address this specific question. The contents below are organized as follows: Firstly, we examine existing methods for optimal sensor placement and propose a modification to the Concrete Autoencoder (18). This modified version is capable of optimizing sensor location on high-resolution grids and automatically selecting the optimal number of sensors. Next, will test the proposed modification on a small-scale dataset of 2D temperature and salinity fields near the Svalbard group of islands and compare it with the baselines. Following this, we implement the proposed modification on a large-scale sensor optimization problem. This application focuses on the temperature field for the global ocean, with a resolution of 0.25 degrees or a grid size of 1440 by 720. Lastly, we conduct a series of simulations with the global ocean model, incorporating data assimilation at the optimal locations we discovered. We then assess the forecast accuracy resulting from these simulations and discuss the obtained results.

## 2 Feature Selection for High-Dimensional Spatio-Temporal Data

The full state of a typical global ocean model has about 100 million dimensions for a 0.25-degree resolution and about 1 billion dimensions for a 0.1-degree resolution. Classical methods for optimal sensor placement, which are based on Singular Value Decomposition such as PCA-QR (17; 19), are not suitable for problems of this scale. Currently, deep learning-based approaches such as Concrete Autoencoder (18) or Feature-Selection Networks (20) have only been tested for significantly lower resolution datasets, with data dimensionality of order $10^4$ to $10^5$. In this section, we propose a modification of the Concrete Autoencoder architecture and demonstrate the effectiveness of this modified version on a 2D temperature field at a 0.25-degree resolution, which has a dimensionality of $10^6$. The proposed modifications are as follows:

- **Replace the Concrete Layer by the Straight-Through Gradient Estimator.** Sensor locations are parameterized by a scalar field. In the forward pass, we apply a step function to the scalar field and multiply an input physical field by the obtained binary mask. In the backward pass, we replace the ill-defined gradients of the step function with the gradients of the identity map and apply gradient clipping. After multiplication by the binary mask input physical field is fed to the UNet model which tries to reconstruct the initial field.
- **Least Square GAN Loss Enhances Spatial Sensor Separation.** The loss function consists of three terms: Least Square GAN loss, Pixel-Wise L2 loss, and the mean value of the binary mask parameterizing sensor locations. As shown in Figure 1, both the PCA-QR method and our Concrete Autoencoder (referred to as CA) struggle with closely placed sensors, which leads to correlated measurements. However, the addition of the LSGAN loss qualitatively changes the sensor pattern, allowing for more separated sensors. It also exhibits the lowest reconstruction error, as seen in Table 1.
- **Dynamic Increase of the Sensor Loss Term.** The weight of the loss term, equal to the mean value of the binary mask $\lambda_{\text{mask}}$, changes dynamically during training. In the first warm-up stage, $\lambda_{\text{mask}} = 0$; this is performed to initially achieve good reconstruction quality. Next, during the sparsification stage, $\lambda_{\text{mask}}$ is increased by $\Delta\lambda_{\text{mask}}$ at every epoch. This allows the Concrete Autoencoder to minimize the number of sensors without significantly increasing the reconstruction error.

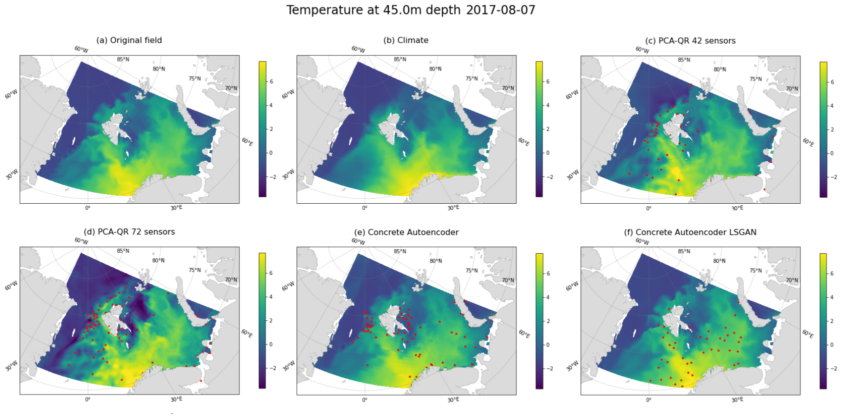

Figure 1: (a) Ground truth temperature field at 45 m depth at 00:00 $7^{th}$ August 2017. The reconstructed fields: (b) Climate at 00:00 $7^{th}$ August, no sensors. (c) PCA-QR, 42 sensors. (d) PCA-QR, 72 sensors. (e) Concrete Autoencoder (CA), 72 sensors. (f) CA with LSGAN, 42 sensors.

Based on the RMSE values from Table 1, we have chosen the configuration of the Concrete Autoencoder with the LSGAN loss term for the high-resolution experiments at the grid size of 1440 x 720. Since the PCA-QR method is computationally expensive for this resolution, we compared the reconstruction method with the nearest neighbor interpolation on a regular grid with 1,000 and 8,000 sensors.

## 3 Generalization of Sensor Placement in Operational Ocean Forecasting

**Overview of the Experiment with Global Ocean Circulation Model and Data Assimilation.** A simulation was conducted to assess the impact of sensor placement on forecast accuracy, simulating a situation where the initial global ocean state substantially deviates from the ground truth. In this

Table 1: Reconstruction Errors for Different Sensor Placement Methods for Temperature and Salinity Fields at 3m and 45m Depth with Grid Size 104 x 284, and for Temperature Field at 9m Depth with Grid Size 1440 x 720. The reanalysis produced by the INMIO ocean model was used for training and validation.

| Physical Field | Grid Size | Reconstruction Method | Placement Method | Number of Sensors | MED(Bias) | MED(RMSE) |
|---|---|---|---|---|---|---|
| Temp. 3m | 104x284 | Climate | - | 0 | −0.19 | 0.98 |
| Temp. 3m | 104x284 | PCA-QR | PCA-QR | 77 | 0.13 | 1.03 |
| **Temp. 3m** | **104x284** | **CA** | **CA** | **77** | **−0.07** | **0.73** |
| Temp. 45m | 104x284 | Climate | - | 0 | −0.09 | 0.88 |
| Temp. 45m | 104x284 | PCA-QR | PCA-QR | 72 | 0.11 | 1.10 |
| Temp. 45m | 104x284 | CA | CA | 72 | −0.05 | 0.83 |
| **Temp. 45m** | **104x284** | **CA-GAN** | **CA-GAN** | **42** | **0.07** | **0.73** |
| Salin. 3m | 104x284 | Climate | - | 0 | 0.58 | 0.84 |
| Salin. 3m | 104x284 | PCA-QR | PCA-QR | 57 | −0.03 | 0.66 |
| **Salin. 3m** | **104x284** | **CA** | **CA** | **57** | **0.05** | **0.53** |
| Salin. 45m | 104x284 | Climate | - | 0 | 0.59 | 0.72 |
| **Salin. 45m** | **104x284** | **PCA-QR** | **PCA-QR** | **61** | **0.02** | **0.30** |
| Salin. 45m | 104x284 | CA | CA | 61 | 0.26 | 0.41 |
| Temp. 9m | 1440x720 | Climate | - | 0 | −0.224 | 0.903 |
| Temp. 9m | 1440x720 | Near. Neib. | Regular | 1326 | −0.007 | 1.246 |
| Temp. 9m | 1440x720 | Near. Neib. | Regular | 8671 | 0.005 | 0.598 |
| Temp. 9m | 1440x720 | Near. Neib. | CA | 1236 | 0.26 | 1.90 |
| **Temp. 9m** | **1440x720** | **CA** | **CA** | **1236** | **−0.20** | **0.69** |

experiment, initial conditions for the ocean and ice models were altered while atmospheric conditions from the control experiment were maintained. In the series s** experiments, the state of the ocean on September 11, 2019, was chosen as the initial conditions. The experiments were conducted with atmospheric forcing that began on September 11, 2020, and taken from the control experiment a01. The model then assimilated data from the control experiment a01 temperature state at 9m depth at sensor locations. Forecast accuracy of the 3D global ocean temperature state in the top 100 meters served as the primary metric for sensor placement quality. The results are presented in Table 2.

Table 2: Summary of experiments, sensor details, and forecast accuracy compared to control experiment a01. Forecast accuracy is averaged over the top 100 meters and over time for the first 20 days.

| Experiment | Placement method | Sensors | Bias (Avg), °C | RMSE (Avg), °C |
|---|---|---|---|---|
| s00 | - | 0 | −0.222 | 1.128 |
| s23 | ARGO, 1 day | 302 | −0.213 | 1.175 |
| s24 | ARGO, 4 days | 1299 | −0.223 | 1.170 |
| s25 | ARGO, non-stationary placement | ∼280-380 | −0.146 | 1.226 |
| s36 | Regular | 1306 | −0.078 | 1.149 |
| s37 | Regular | 3379 | −0.102 | 1.138 |
| s38 | Regular | 6792 | −0.106 | 1.127 |
| s39 | Regular | 13581 | −0.100 | 1.127 |
| **s44** | **Concrete Autoencoder (CA)** | **1160** | **−0.100** | **1.139** |
| s53 | Regular (3443) from CA reconstruction | 964 (3443) | −0.185 | 1.141 |
| s54 | Regular (3443) from CA reconstruction | 1236 (3443) | −0.202 | 1.144 |
| a01 | Reanalysis with data assimilation, temperature and salinity profiles, sea ice compactness, sea level anomaly | ∼ 11200 | - | - |

**Results and Discussion.** The results in Table 2 suggest that when comparing experiments s24, s36, and s44, each with approximately one thousand sensors, the s44 arrangement, which was obtained using the Concrete Autoencoder method, has the lowest RMSE value (Bias $-0.100$, RMSE $1.139$). A similar forecast accuracy value is found in the s37 arrangement with data assimilation from 3,379 regularly placed sensors (Bias $-0.102$, RMSE $1.138$). This suggests a positive response to the research question from the introduction section, considering the Concrete Autoencoder was trained to restore the temperature field concurrently with the measurements.

However, our approach has several limitations. In experiments s53 and s54, we used the fields reconstructed from 964 and 1236 measurements by the Concrete Autoencoder for data assimilation instead of the ground truth fields resampled at a regular grid. We didn't observe significant improvements in this case, but there also wasn't a drop in accuracy. Both of these methods still outperform regular and ARGO placement methods. Additionally, to speed up the ocean model runs, we implemented aggressive data assimilation which resulted in moderate data assimilation shocks. As a result, low values of RMSE were observed for experiment s00, which was conducted without data assimilation. Despite this, our choice allowed us to conduct a significantly larger number of experiments.

## References

[1] Hurrell, J. W., M. M. Holland, P. R. Gent, et al. The community earth system model: a framework for collaborative research. *Bulletin of the American Meteorological Society*, 94(9):1339–1360, 2013.

[2] Kay, J. E., C. Deser, A. Phillips, et al. The community earth system model (cesm) large ensemble project: A community resource for studying climate change in the presence of internal climate variability. *Bulletin of the American Meteorological Society*, 96(8):1333–1349, 2015.

[3] Danabasoglu, G., J.-F. Lamarque, J. Bacmeister, et al. The community earth system model version 2 (cesm2). *Journal of Advances in Modeling Earth Systems*, 12(2):e2019MS001916, 2020.

[4] Leung, L. R., D. C. Bader, M. A. Taylor, et al. An introduction to the e3sm special collection: Goals, science drivers, development, and analysis. *Journal of Advances in Modeling Earth Systems*, 12(11):e2019MS001821, 2020.

[5] Golaz, J.-C., L. P. Van Roekel, X. Zheng, et al. The doe e3sm model version 2: Overview of the physical model and initial model evaluation. *Journal of Advances in Modeling Earth Systems*, 14(12):e2022MS003156, 2022.

[6] Rasch, P., S. Xie, P.-L. Ma, et al. An overview of the atmospheric component of the energy exascale earth system model. *Journal of Advances in Modeling Earth Systems*, 11(8):2377–2411, 2019.

[7] Dunne, J. P., L. Horowitz, A. Adcroft, et al. The gfdl earth system model version 4.1 (gfdl-esm 4.1): Overall coupled model description and simulation characteristics. *Journal of Advances in Modeling Earth Systems*, 12(11):e2019MS002015, 2020.

[8] Adcroft, A., W. Anderson, V. Balaji, et al. The gfdl global ocean and sea ice model om4. 0: Model description and simulation features. *Journal of Advances in Modeling Earth Systems*, 11(10):3167–3211, 2019.

[9] Kalnay, E. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

[10] Compo, G. P., J. S. Whitaker, P. D. Sardeshmukh, et al. The twentieth century reanalysis project. *Quarterly Journal of the Royal Meteorological Society*, 137(654):1–28, 2011.

[11] Chassignet, E. P., H. E. Hurlburt, O. M. Smedstad, et al. The hycom (hybrid coordinate ocean model) data assimilative system. *Journal of Marine Systems*, 65(1-4):60–83, 2007.

[12] Chassignet, E. P., H. E. Hurlburt, E. J. Metzger, et al. Us godae: global ocean prediction with the hybrid coordinate ocean model (hycom). *Oceanography*, 22(2):64–75, 2009.

[13] Molteni, F., R. Buizza, T. N. Palmer, et al. The ecmwf ensemble prediction system: Methodology and validation. *Quarterly journal of the royal meteorological society*, 122(529):73–119, 1996.

[14] Andersson, E., J. Thépaut. Ecmwf's 4d-var data assimilation system–the genesis and ten years in operations. *ECMWF Newsletter*, 115:8–12, 2008.

[15] Wu, P., X. Chang, W. Yuan, et al. Fast data assimilation (fda): Data assimilation by machine learning for faster optimize model state. *Journal of Computational Science*, 51:101323, 2021.

[16] Farchi, A., P. Laloyaux, M. Bonavita, et al. Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, 147(739):3067–3084, 2021.

[17] Manohar, K., B. W. Brunton, J. N. Kutz, et al. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

[18] Abid, A., M. F. Balin, J. Zou. Concrete autoencoders for differentiable feature selection and reconstruction, 2019.

[19] Manohar, K., B. W. Brunton, J. N. Kutz, et al. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

[20] Singh, D., H. Climente-González, M. Petrovich, et al. Fsnet: Feature selection network on high-dimensional biological data. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2023.

[21] Ronneberger, O., P. Fischer, T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[22] Bengio, Y., N. Léonard, A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[23] Isola, P., J.-Y. Zhu, T. Zhou, et al. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134. 2017.

[24] Bryan, K. A numerical method for the study of the circulation of the world ocean. *Journal of computational physics*, 135(2):154–169, 1997.

[25] Lebedev, V. I. Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. i. *USSR Computational Mathematics and Mathematical Physics*, 4(3):69–92, 1964.

[26] —. Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. ii. *USSR Computational Mathematics and Mathematical Physics*, 4(4):36–50, 1964.

[27] Mesinger, F., A. Arakawa. Numerical methods used in atmospheric models. 1976.

[28] Ushakov, K., R. Ibrayev. Assessment of mean world ocean meridional heat transport characteristics by a high-resolution model. *Russian Journal of Earth Sciences*, 18(1):1–14, 2018.

[29] Hunke, E. C., W. H. Lipscomb, A. K. Turner, et al. Cice: the los alamos sea ice model documentation and software user's manual version 4.1 la-cc-06-012. *T-3 Fluid Dynamics Group, Los Alamos National Laboratory*, 675:500, 2010.

[30] Killworth, P. D., D. J. Webb, D. Stainforth, et al. The development of a free-surface bryan–cox–semtner ocean model. *Journal of Physical Oceanography*, 21(9):1333–1348, 1991.

[31] Griffies, S. M., R. W. Hallberg. Biharmonic friction with a smagorinsky-like viscosity for use in large-scale eddy-permitting ocean models. *Monthly Weather Review*, 128(8):2935–2946, 2000.

[32] Munk, W. H. Note on the theory of the thermocline. *J. Mar. Res.*, 7:276–295, 1948.

[33] Griffies, S. M., A. Biastoch, C. Boning, et al. Coordinated ocean-ice reference experiments (cores). *Ocean modelling*, 26(1-2):1–46, 2009.

[34] Hersbach, H., B. Bell, P. Berrisford, et al. The era5 global reanalysis, qj roy. meteor. soc., 146, 1999–2049, 2020.

[35] Kaurkin, M. N., R. A. Ibrayev, K. P. Belyaev. Data assimilation in the ocean circulation model of high spatial resolution using the methods of parallel programming. *Russian Meteorology and Hydrology*, 41(7):479–486, 2016. Cited By :12.

[36] Kaurkin, M., R. Ibrayev, A. Koromyslov. Enoi-based data assimilation technology for satellite observations and argo float measurements in a high resolution global ocean model using the cmf platform. In V. Voevodin, S. Sobolev, eds., *Supercomputing*, pages 57–66. Springer International Publishing, Cham, 2016.

[37] Evensen, G. *Data Assimilation*. Springer Berlin Heidelberg, 2009.

[38] Ryan, A., C. Regnier, P. Divakaran, et al. Godae oceanview class 4 forecast verification framework: global ocean inter-comparison. *Journal of Operational Oceanography*, 8(sup1):s98–s111, 2015.

# 4 Supplementary Material

## 4.1 Algorithms

Our Concrete Autoencoder architecture consists of a trainable binary mask and a reconstructing image-to-image neural network with a U-Net architecture (21) and bilinear upsampling. The U-Net takes, as input, a physical field multiplied by the binary mask and predicts the reconstructed field in the full computational domain. Our binary mask representing sensor locations is parametrized by the scalar field of parameter $w$ via a step function

$$\text{mask} = \text{step}(w).$$

To determine the optimal number of sensors, we optimize the parameters of the binary mask using a straight-through gradient estimator proposed in (22) with gradient clipping. In the backward pass, we replace the ill-defined gradients of the step function with the gradients of the identity function. The straight-through gradient estimator allows us to use a single matrix $w$ of parameters for different numbers of sensors.

The loss function of the Concrete Autoencoder has three main terms

$$\mathcal{L}_{CA}(\theta) = \mathcal{L}_{\text{LSGAN}} + \mathcal{L}_{\text{pixel-wise}} + \lambda_{\text{mask}}\mathcal{L}_{\text{sensors}}, \tag{1}$$

where $\lambda_{\text{mask}}$ dynamically changes during training. In the first warm-up stage, $\lambda_{\text{mask}} = 0$, this is performed to initially achieve good reconstruction quality. Next, at the sparsification stage, $\lambda_{\text{mask}}$ is increased by $\Delta\lambda_{\text{mask}}$ at every epoch. This allows the Concrete Autoencoder to minimize the number of sensors without a significant increase in the reconstruction error. A similar procedure was proposed in the original version of the Concrete Autoencoder (18) where the annealing procedure was applied to the temperature parameter of the Gumbel-softmax distribution.

Consider a random batch $t_1, \ldots, t_K$ sampled from $\{1, \ldots, T\}$. Following Pix2Pix framework (23), the loss function $\mathcal{L}_{\text{LSGAN}}$ requires that the Concrete Autoencoder produces physical fields that are indistinguishable by the PatchGAN discriminator from the real fields

$$\mathcal{L}_{\text{LSGAN}} = \frac{1}{K}\sum_{i=1}^{K} \text{MSE}(D_\phi(CA_\theta(\text{mask} \odot \text{F}^{(t_i)})), \mathbb{I})$$

In addition to the requirement for the realism of the generated physical fields, we require an element-by-element correspondence of the reconstructed physical fields with the ground truth ones. The $L_2$-norm is used to measure pixel-wise error

$$\mathcal{L}_{\text{pixel-wise}} = \frac{1}{K}\sum_{i=1}^{K} ||CA_\theta(\text{mask} \odot \text{F}^{(t_i)}) - \text{F}^{(t_i)}||_{L_2}$$

The sparsification of sensors is achieved by adding the average value of the binary mask to the loss function

$$\mathcal{L}_{\text{sensors}} = \mathbb{E}||\text{mask}||_{L_1}$$

We require the discriminator to distinguish the physical fields produced by the Concrete Autoencoder from the real ones, thus defining its loss as

$$\mathcal{L}_D(\phi) = \frac{1}{2}\frac{1}{K}\sum_{i=1}^{K} \Big[\text{MSE}(D_\phi(CA_\theta(\text{mask} \odot \text{F}^{(t_i)})), \mathbb{O}) + \text{MSE}(D_\phi(\text{F}^{(t_i)}), \mathbb{I})\Big]$$

The full training procedure for the Concrete Autoencoder is outlined in Algorithm 1.

**Algorithm 1:** Concrete Autoencoder Training.

```
Input:
    $\mathcal{F} = \{\mathrm{F}^{(t)}\}$—train dataset of historical values of a physical field, $t \in \{1, \dots, T\}$ denotes a time moment
    CA$_\theta$—randomly initialized Concrete Autoencoder network with parameters $\theta$, which reconstructs physical field from sparse measurements; parameters of the binary mask are also included in $\theta$
    $D_\phi$—randomly initialized PatchGAN Discriminator with parameters $\phi$
    mask—binary mask defining initial sensor locations sampled proportionally to the information entropy field
    $N_{\text{warm-up steps}}$—number of training steps on the warm-up stage
    $N_{\text{sparsification steps}}$—number of training steps on the sensor sparsification stage
    $\mathbb{I}$ - matrix filled with ones
    $\mathbb{O}$ - matrix filled with zeros
    $\lambda_{\text{mask}} = 0$—weight of the mask term in CA loss
    $\Delta\lambda_{\text{mask}} \geq 0$—weight increase of $\lambda_{\text{mask}}$ in sparsification stage
for $m = 1$ to $N_{\text{warm-up steps}} + N_{\text{sparsification steps}}$ do
    choose $t_1, \dots, t_K$ randomly from $\{1, \dots, T\}$
    Update the Concrete Autoencoder weights:
    $\mathcal{L}_{\text{sensors}} = \mathbb{E}||\text{mask}||_{L_1}$
    $\mathcal{L}_{\text{pixel-wise}} = \frac{1}{K}\sum_{i=1}^{K} ||CA_\theta(\text{mask} \odot \mathrm{F}^{(t_i)}) - \mathrm{F}^{(t_i)}||_{L_2}$
    $\mathcal{L}_{\text{LSGAN}} = \frac{1}{K}\sum_{i=1}^{K} \text{MSE}(D_\phi(CA_\theta(\text{mask} \odot \mathrm{F}^{(t_i)})), \mathbb{I})$
    $\mathcal{L}_{CA}(\theta) = \mathcal{L}_{\text{LSGAN}} + \mathcal{L}_{\text{pixel-wise}} + \lambda_{\text{mask}}\mathcal{L}_{\text{sensors}}$
    $\theta \leftarrow \text{Adam}(\theta, \nabla_\theta \mathcal{L}_{CA}(\theta))$
    Update the discriminator weights:
    $\mathcal{L}_D(\phi) = \frac{1}{2}\frac{1}{K}\sum_{i=1}^{K}\Big[\text{MSE}(D_\phi(CA_\theta(\text{mask} \odot \mathrm{F}^{(t_i)})), \mathbb{O}) + \text{MSE}(D_\phi(\mathrm{F}^{(t_i)}), \mathbb{I})\Big]$
    $\phi \leftarrow \text{Adam}(\phi, \nabla_\phi \mathcal{L}_D(\phi))$
    if $m \geq N_{\text{warm-up steps}}$ then
        Increase the sparsification weight:
        $\lambda_{\text{mask}} \leftarrow \lambda_{\text{mask}} + \Delta\lambda_{\text{mask}}$
    end if
end for
Output:
    CA$_{\theta^*}$— Concrete Autoencoder network with optimal parameters $\theta^*$
    mask$^*$—binary mask with optimal sensor placement
```

Consider a random batch $t_1, \dots, t_K$ sampled from $\{1, \dots, T\}$. Following Pix2Pix framework (23), the loss function

$$\mathcal{L}_{\text{LSGAN}}$$

requires that the Concrete Autoencoder produces physical fields that are indistinguishable by the PatchGAN discriminator from the real fields

$$\mathcal{L}_{\text{LSGAN}} = \frac{1}{K}\sum_{i=1}^{K} \text{MSE}(D_\phi(CA_\theta(\text{mask} \odot \mathrm{F}^{(t_i)})), \mathbb{I}).$$

In addition to the requirement for the realism of the generated physical fields, we require an element-by-element correspondence of the reconstructed physical fields with the ground truth ones. The $L_2$-norm is used to measure pixel-wise error

$$\mathcal{L}_{\text{pixel-wise}} = \frac{1}{K}\sum_{i=1}^{K} ||CA_\theta(\text{mask} \odot \mathrm{F}^{(t_i)}) - \mathrm{F}^{(t_i)}||_{L_2}.$$

The sparsification of sensors is achieved by adding the average value of the binary mask to the loss function

$$\mathcal{L}_{\text{sensors}} = \mathbb{E}||\text{mask}||_{L_1}.$$

We require the discriminator to distinguish the physical fields produced by the Concrete Autoencoder from the real ones, thus defining its loss as

$$\mathcal{L}_D(\phi) = \frac{1}{2}\frac{1}{K}\sum_{i=1}^{K}\Big[\mathrm{MSE}(D_\phi(CA_\theta(\text{mask} \odot \mathrm{F}^{(t_i)})), \mathbb{O}) + \mathrm{MSE}(D_\phi(\mathrm{F}^{(t_i)}), \mathbb{I})\Big]$$

.

### 4.2 Sensor Locations and Training Dynamics

The training dynamics of the Concrete Autoencoder on the grid 1440 times 720 are shown in Figure 5. The sensor locations used in experiments s44 and s53 are shown in 3 and 4 correspondingly.

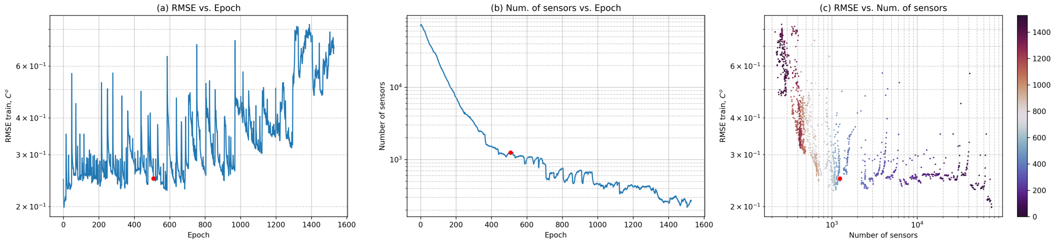

Figure 2: The training dynamics of the Concrete Autoencoder at the sensor sparsification stage. (**a**) RMSE as a function of epoch, (**b**) number of sensors as a function of epoch, and (**c**) RMSE as a function of the number of sensors, with the epoch indicated by the color bar. The red dots in (**a–c**) denotes the Concrete Autoencoder checkpoint at the epoch 508 used in further analysis.

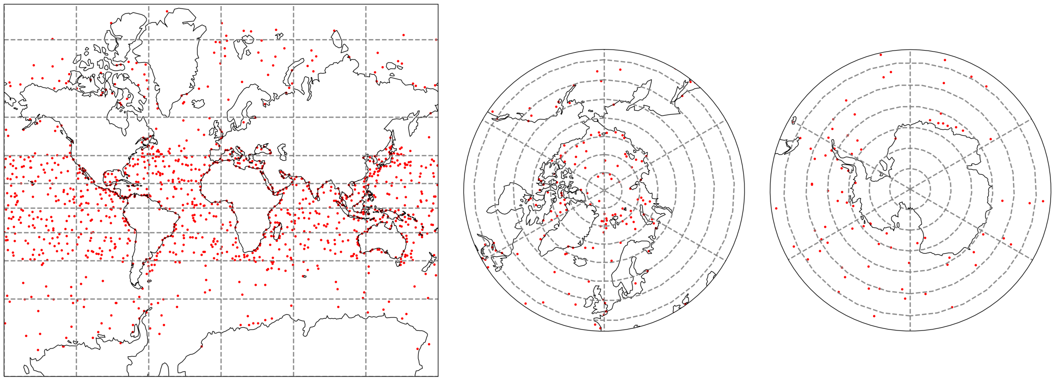

Figure 3: Sensor locations in the experiments s44. Learned by the Concrete Autoencoder at epoch 508.

### 4.3 Dataset Description for Training the Concrete Autoencoder

**INMIO Ocean general circulation model.** The system of equations of three-dimensional ocean dynamics and thermodynamics in the Boussinesq and hydrostatic approximations is solved by the finite volume method (24) on the type B grid (25; 26; 27). The ocean model INMIO (28) and the sea ice model CICE (29) operate on the same global tripolar grid with a nominal resolution of $0.25°$. The vertical axis of the ocean model uses $z$-coordinates on 49 levels with a spacing from 6 m in the upper layer to 250 m at the depth. The barotropic dynamics are described with the help of a two-dimensional system of shallow water equations by the scheme (30). The horizontal turbulent mixing of heat and salt is parameterized with a background (time-independent) diffusion coefficient equal to the nominal value at the equator and scaled toward the poles proportionally to the square root of the grid cell area. To ensure numerical stability in the equations of momentum transfer,

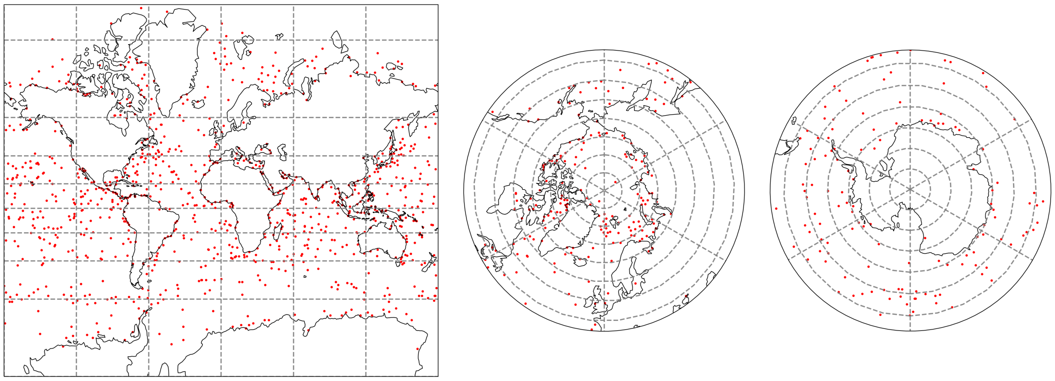

Figure 4: Original sensor locations in the experiment s53. Learned by the Concrete Autoencoder at epoch 634.

the biharmonic filter is applied with a background coefficient scaled proportionally to the cell area to the power 3/2 and with the local addition by Smagorinsky scheme in formulation (31) for maintaining sharp fronts. Vertical mixing is parameterized by the Munk–Anderson scheme (32), with convective adjustment performed in the case of an unstable vertical density profile. At the ocean–atmosphere interface, the nonlinear kinematic-free surface condition is imposed with heat, water, and momentum fluxes calculated by the CORE bulk formulae (33). Except for vertical turbulent mixing, all the processes were described using time-explicit numerical methods, which allow simple and effective parallel scaling. The time steps of the main cycle for solving model equations are equal for the ocean and the ice. The ocean model, within the restrictions of its resolution, implements the eddy-permitting mode by not using the laplacian viscosity in the momentum equations.

**The sea ice model CICE v. 5.1.** For this experiment, an elastic-viscous-plastic rheology model was applied to parameterize the ice dynamics, and zero-layer approximation was used for thermodynamics calculations. To explicitly resolve elastic waves, a subcycle with small time steps was set. The simulation mode includes the processing of five categories of ice thickness and one category of snow thickness using an upwind transport scheme and a description of melt ponds.

**ERA5 atmospheric forcing.** We used the ERA5 reanalysis (34) for the period 2004–2020 as the external forcing to determine the water and momentum fluxes on the ocean–atmosphere and ice–atmosphere interfaces. Wind speed at 10 m above sea level and temperature and dew point temperature at 2 m were transmitted to the ice–ocean system every 3 h. In addition, the accumulated fluxes of incident solar and long-wave radiation and precipitation (snow and rain) were also read with the same period.

**Ensemble Optimal Interpolation (EnOI).** A detailed description of the EnOI method is presented in the work (35). For our calculation, we used the original parallel realization of the EnOI method for data assimilation in the INMIO model (36).

The basic equations of the EnOI method are as follows (37):

$$F_a = F_b + K(y_{obs} - HF_b), \tag{2}$$

where $K$ is defined as

$$K = BH^T(HBH^T + R)^{-1}, \tag{3}$$

$F_b$ (background) and $F_a$ (analysis) are vectors of size $n$ representing the model solution before and after data assimilation, respectively; $n$ is the number of model grid points weighted by the number of model variables to be corrected (temperature, salinity, sea level, etc.); $y_{obs}$ is the vector of observations of size $m$; $m$ is the total number of observation points where various data were obtained; $K(n \times m)$ is the gain matrix; $R(m \times m)$ is the covariance matrix of observation errors, it is assumed that the matrix $R = rI$ is the identity matrix multiplied by a scalar parameter $r$; $H(m \times n)$ is the matrix, representing the projection operator of model values into the observational data space; $B(n \times n)$ is the covariance matrix of model errors.

The EnOI method belongs to a group of assimilation methods that rely on some approximation of matrix $B$ based on an ensemble of model solution vectors. This approximation allows for the estimation of the covariance matrix of model errors. In practice, the ensemble is used to approximate matrix $HBH^T$ of size $m \times m$. The inverse $(HBH^T + R)^{-1}$ is then computed using SVD, which can be a limiting factor in the assimilation of a large number of observations.

### 4.4 Baseline

**Climate**. The simplest baseline in ocean modeling, reconstruction, and forecasting is climate interpolated in time to the correct date. We calculated our climate values on the training set for every day of a year, according to the formula

$$F^{climate}(i,j,d) = \frac{1}{N^{years}} \sum_{y=1}^{N^{years}} F(i,j,y,d), \tag{4}$$

where $F(i,j,y,d)$ is the value of a physical field with coordinates $(i,j)$ at day number $d = \{1,2,\ldots,365\}$ in year $y$ from the training set, $N^{years}$ is the number of years with day $d$ in the training set.

### 4.5 Evaluation Metrics

In this study, we use evaluation metrics based on the GODAE OceanView Class 4 forecast verification framework (38). The bias metric measures the correspondence between the mean forecast and the mean observation. To calculate the spatial and temporal distributions of the bias, we average over time at each spatial location $(i,j)$ using Equation (5) and average over the spatial coordinates at each time point in the test set using Equation (6), respectively.

$$Bias(i,j) = \frac{1}{\#\{\tau \in TestSet\}} \sum_{\tau \in TestSet} (F^{recon}(i,j,\tau) - F^{ref}(i,j,\tau)), \tag{5}$$

where $F^{recon}(i,j,\tau)$ is the reconstructed values of a physical field at a point with coordinates $(i,j)$ and at time moment $\tau$, $F^{ref}(i,j,\tau)$ is the original reanalysis values of a physical field in the same point.

$$Bias(\tau) = \frac{1}{N^i}\frac{1}{N^j}\sum_{i=1}^{N^i}\sum_{j=1}^{N^j}(F^{recon}(i,j,\tau) - F^{ref}(i,j,\tau)), \tag{6}$$

where $N^i \cdot N^j$ is the total number of computational cells for the field considered.

The second metric used is the Root Mean Square Error ($RMSE$). It was calculated for each grid point $(i,j)$ by averaging along the time dimension using Equation (7) and for each time moment in the test set by averaging along the spatial dimensions using Equation (8)

$$RMSE(i,j) = \sqrt{\frac{1}{\#\{\tau \in TestSet\}} \sum_{\tau \in TestSet} (F^{recon}(i,j,\tau) - F^{ref}(i,j,\tau))^2} \tag{7}$$

$$RMSE(\tau) = \sqrt{\frac{1}{N^i}\frac{1}{N^j}\sum_{i=1}^{N^i}\sum_{j=1}^{N^j}(F^{recon}(i,j,\tau) - F^{ref}(i,j,\tau))^2} \tag{8}$$

The scalar metrics presented in Table 1 were calculated by taking the median along the time dimension. The spatial forecast errors for the static concrete autoencoder sensor placement (s44) and the static regular sensor placement (s36) are presented in Figure 5 and Figure 6, respectively. The errors for the real dynamic sensor locations of ARGO buoys (s25) are shown in Figure 7.

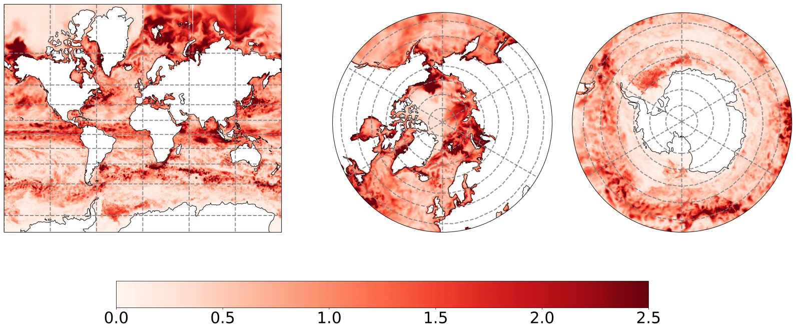

Figure 5: Average RMSE for the temperature field forecast over the top 100m using static concrete autoencoder sensor placement s44 (in Celsius).

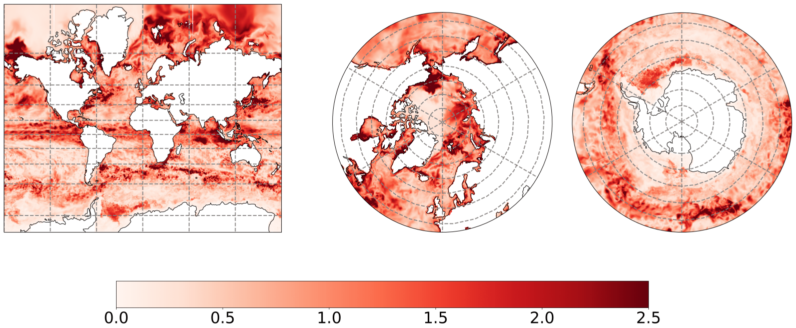

Figure 6: Average RMSE for the temperature field forecast over the top 100m using static regular sensor placement s36 (in Celsius).

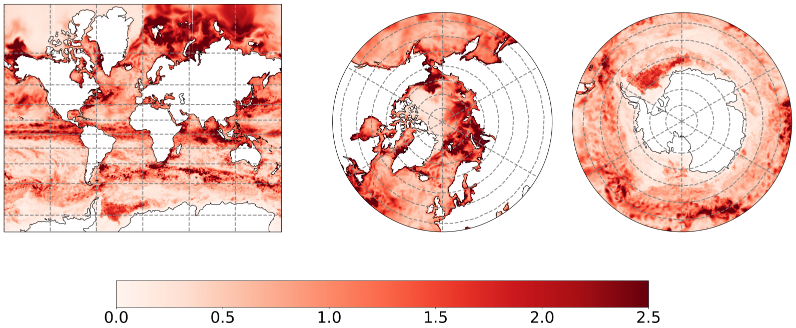

Figure 7: Average RMSE of the temperature field forecast over the top 100m using real dynamic sensor locations of ARGO buoys s25 (in Celsius).