Modular compositional learning improves 1D hydrodynamic lake model performance by merging process-based modeling with deep learning

Robert Ladwig[1], A Daw[2], E A Albright[1], C Buelo[1], A Karpatne[2], M F Meyer[1,3], A Neog[2], P C Hanson[1], and H A Dugan[1]

[1]Center for Limnology, University of Wisconsin-Madison
[2]Department of Computer Sciences, Virginia Tech
[3]U.S. Geological Survey, Observing Systems Division

August 7, 2023

Modular compositional learning improves 1D hydrodynamic lake model performance by merging process-based modeling with deep learning

R. Ladwig[1*], A. Daw[2], E.A. Albright[1], C. Buelo[1], A. Karpatne[2], M.F. Meyer[3,1], A. Neog[2], P.C. Hanson[1], H.A. Dugan[1]

[1]Center for Limnology, University of Wisconsin-Madison, WI, 53706 USA
[2]Department of Computer Sciences, Virginia Tech, VI, 24061 USA
[3]U.S. Geological Survey, Observing Systems Division, Madison, WI, 53726 USA

Disclaimer

This preprint is currently under review at Journal of Advances in Modeling Earth Systems (JAMES), and has been peer reviewed and approved for publication consistent with U.S. Geological Survey Fundamental Science Practices (https://pubs.usgs.gov/circ/1367/)

Key Points:

- Deep learning models were pretrained on process-based lake water temperature model output and fine-tuned on observed high-frequency data.
- Fine-tuned deep learning model was integrated into process-based model creating the hybrid model.
- Hybrid model outperformed process-based model and two alternative deep learning models in projecting hydrodynamic lake characteristics.

*Current address: 680 N. Park Street, Madison, 53706 WI, USA

Corresponding author: R. Ladwig, `rladwig2@wisc.edu`

Abstract

Hybrid Knowledge-Guided Machine Learning (KGML) models, which are deep learning models that utilize scientific theory and process-based model simulations, have shown improved performance over their process-based counterparts for the simulation of water temperature and hydrodynamics. We highlight the modular compositional learning (MCL) methodology as a novel design choice for the development of hybrid KGML models in which the model is decomposed into modular sub-components, that can either be process-based models and/or deep learning models. We develop a hybrid MCL model that integrates a deep learning model into a modularized, process-based model. To achieve this, we first train individual deep learning models with the output of the process-based models. In a second step, we fine-tune one deep learning model with observed field data. In this study, we replaced process-based calculations of vertical diffusive transport with deep learning. Finally, this fine-tuned deep learning model is integrated into the process-based model, creating the hybrid MCL model with improved overall projections for water temperature dynamics compared to the original process-based model. We further compare the performance of the hybrid MCL model with the process-based model and two alternative deep learning models and highlight how the hybrid MCL model has the best performance for projecting water temperature, Schmidt stability, buoyancy frequency, and depths of different isotherms. Modular compositional learning can be applied to existing modularized, process-based model structures to make the projections more robust and improve model performance by letting deep learning estimate uncertain process calculations.

Plain Language Summary

Lake models based on physical processes are powerful tools for investigating how lakes and reservoirs respond to local weather and for projecting lake responses to long-term climate change. Historically, physical processes are the basis for designing these models. Due to an abundance of long-term and high-frequency data, deep learning models are used more frequently although they do not reflect our domain expertise about hydrodynamics and heat transport. Recently, the modeling community is focusing on merging models based on physical processes with deep learning. We are highlighting a novel methodology, modular compositional learning, that merges different modeling types in a modularized framework. Our resulting hybrid model outperformed the original model based on physical processes as well as alternative deep learning models regarding the simulation of various lake variables related to water temperature, and showed physically valid results. We are further showing various ways on how modular compositional learning can improve future lake model development and applications.

1 Introduction

The conceptual model of aquatic ecosystem dynamics as a linked set of physical, chemical, and biological processes is fundamental to lake ecosystems research (Håkanson, 2009) and has led to rapid development of hydrodynamic-water quality simulation models in the past couple of decades (Mooij et al., 2010). Given the importance of temperature as a "master variable" in ecosystems (Magnuson et al., 1979; Read et al., 2019), hydrodynamic simulations are used to model the physical environment in which biogeochemical processes occur. As climate patterns become more uncertain and a growing human population increases the need for reliable, expediant assessments of freshwater resources, process-based aquatic ecosystem models can be used to understand and explore how these stressors can impact aquatic ecosystems by detailing the mechanistic effects of external and internal forcings (Janssen et al., 2015). Lake hydrodynamic models may be used to understand how warming air temperatures will alter lake thermal structure (Woolway et al., 2021) and linked biogeochemical conditions such as dissolved oxygen concentrations (Jane et al., 2022). Due to their low computational costs but sufficient replication of lake mixing dynamics (Ishikawa et al., 2022), one-dimensional hydrodynamic lake models are commonly employed to project changes in water temperature (Moore et al., 2021), when it is reasonable to neglect horizontal mixing and focus only on resolving the vertical transport.

Although the low spatial dimensionality provides for computational efficiency, 1D hydrodynamic lake models have a number of drawbacks. First, overparameterization of processes can make calibration challenging (e.g., Guerrero et al. (2017)), especially when field data are sparse. Second, integration of a multitude of processes (atmospheric fluxes, vertical transport, inflow entrainment, etc.) can lead to model equifinality (i.e., alternative model parameterizations can result in the same model output, see also Beven (2006)). Third, implementations of these models tend to be stiff in their structures, creating a high cost to exploring the model's underlying assumptions and utilizing information in observational data that may improve model skill. For example, the MyLake model neglects the influence of internal oscillations by seiches on mixing (Saloranta & Andersen, 2007), whereas the Simstrat model parameterizes turbulent kinetic energy production by such internal movements (Goudsmit et al., 2002). Consequently, a stiff model structure makes incorporating additional data in certain models without major revisions to the source code impossible.

To overcome the limitations of process-based models, the paradigm of "Knowledge-Guided Machine Learning" (KGML) focuses on creating hybrid models that combine process-based principles (or theory) with data-driven deep learning models (Karpatne et al., 2017; Appling et al., 2022). Whereas deep learning models alone need extensive data for training, neglect fundamental physical principles, and can have poor performance when projecting outside of the training range, hybrid KGML models balance design with discovery (Appling et al., 2022), in essence how much prior knowledge we specify for the deep learning models (design) with the potential for learning useful, and sometimes novel, relationships between input data and target (discovery). Initial work in developing KGML models has revealed how physical laws can be encoded as loss terms in deep learning models to make projections more physically valid (Daw et al., 2021), and how recurrent neural networks can improve model performance (Jia et al., 2021). Comparisons of hybrid model performance with both process-based and purely data-driven deep learning models for data-sparse experimental conditions highlight that the hybrid models outperformed their counterparts (Read et al., 2019; Jia et al., 2020). Even predicting water temperature dynamics outside of monitored lake sites has been achieved using a hybrid model, which performed better than the process-based counterpart (Willard et al., 2021). The applications of such hybrid models have extended to water quality modeling, e.g., phosphorus simulations in Hanson et al. (2020). In general, the current generation of KGML hybrid models for water temperature projections have common characteristics: (1) using recurrent neural networks, (2) pretraining of these neural networks using process-based model output, (3) fine-tuning of neural network weights using *in-situ* temperature data, and (4) incorporating of physical laws as loss terms. Pretraining is the initialization of the deep learning network structure using process-based model output in advance of fine-tuning with true observations, and has been shown to vastly improve the accuracy and generalizability of model projections (Read et al., 2019). Here, both pretraining and fine-tuning are training steps for the deep learning model, but the former acts on an uninitialized model structure, whereas the latter trains an already trained model.

As highlighted above, most current hybrid KGML models incorporate only one-way feedbacks from the process-based to the deep learning side, with additional expertise added through loss functions in the training step. This structure highlights how most hybrid KGML models are primarily engineered from a deep learning model with add-ons for a specific physical or theoretical process. Although this architecture works well for single target studies (e.g., water temperature), water quality modeling includes multiple target variables that need to be accounted for in a flexible framework. Most process-based 1D water quality models have a modularized model structure to account for alternative configurations of biogeochemical, ecological and food web-related interactions (e.g., FABM in Bruggeman and Bolding (2014), GLM-AED2 in Hipsey et al. (2019)). Here, a flexible framework, in which individual building blocks, or modules, can be replaced with process-based or data-driven calculations can potentially enhance design options for future KGML studies. Calculations of uncertain processes can be replaced by deep learning to improve overall model confidence. The concept of applying modularized frameworks for compositional learning (modular compositional learning, or MCL) states that domain expertise (scientific knowledge) can be used to decompose the overall modeling goal into modular sub-aspects, hence into a combination of multiple deep learning models and/or process-based models (Karpatne et al., 2017). MCL advances current hybrid KGML model designs, which generally focus on developing a single deep learning model that incorporates process knowledge. Here, hybrid MCL models would have a design similar to the modularized structure of process-based lake models, and individual sub-parts could either be process-based or deep learning models. This MCL approach allows the modeler to balance between framework design (how much prior knowledge is inserted into the model formulation) and chances for the discovery of relationships.

To further the union of process-based and deep learning models, we develop and test the MCL concept for hydrodynamic modeling focusing on projecting water temperature dynamics. Each hydrodynamic, process-based calculation can be envisioned as a module that is linked to other modules. The complexity of each module's process description depends on the assumptions and relative complexity given to a particular process. For example, the effects of atmospheric surface heat fluxes on lake temperature are well described using similarity theory *sensu* Monin-Obukhov, whereas vertical transport through turbulent diffusion can be parameterized using alternative approaches, i.e., integral energy approach as in GLM (Hipsey et al., 2019) vs. turbulence-based approach as in Simstrat (applies k-ϵ turbulence closure scheme, Goudsmit et al. (2002)). By replacing a process-based module with a deep learning model, we can feed additional data into the model without the need for formulating a process relationship. Consequently, the link between modules will ensure that the overall hybrid MCL model will produce physically-valid results.

To test MCL, we are replacing the process-based modules of a 1D hydrodynamic lake model, which modularizes vertical heat transport by sequentially accounting for (a) heat generation, (b) ice and snow formation, (c) vertical diffusion, and (d) convective overturn, with individual deep learning models. Each deep learning model is used to represent each 1D process-based model component. These deep learning models are pretrained on the input data going into the process-based model as well as the process-based model output temperatures. Subsequently, one deep learning module is fine-tuned on observed high-frequency water temperature data to improve model performance. In this study, we replaced the diffusion model with a deep learning model to improve the overall accuracy of the model to capture vertical transport processes. This fine-tuned deep learning module is plugged back into the process-based modular framework creating the hybrid MCL model. The performance of the hybrid MCL in replicating hydrodynamic characteristics of Lake Mendota, USA, is compared with the original process-based model, a deep learning model that does not use any process-based information, and a pretrained deep learning model that incorporates process-based information. These alternative deep learning models reflect different design and discovery ideas; whereas the deep learning model with no process information can be used for discovering relationships between input data and targets, the pretrained deep learning model with process information is configured to reflect physical processes. With this test of the novel MCL methodology for water temperature simulations, we aim to highlight how the hybrid model incorporates potentially both design and discovery, while allowing lake modelers the flexibility of replicating future, more complex, aquatic ecosystem structures.

2 Data

As a test site for model development, we chose Lake Mendota (Wisconsin, USA), which has been the subject of many limnological and modeling studies over the last century (e.g., Snortheim et al. (2017); Magee et al. (2016); Ladwig et al. (2021)). Lake Mendota is a 3,961 ha lake with a maximum depth of 25 m that stratifies during the summer and winter seasons. Lake Mendota has a residence time of 4.3 years (McDonald & Lathrop, 2017), which allows us to assume that inflows and outflows would have a minor effect on in-lake water temperatures. All in-lake measurements were collected in the center of the lake (43.0988N, -89.4054W) and include: depth-discrete measurements of water temperature (collected fortnightly, when ice-free, to monthly, when ice-covered by the North Temperate Lake Long-term Ecological Research program [NTL-LTER] since 1995), and high-frequency water temperature data collected by a sensor-chain connected to a buoy (data since 2006 for the ice-free season) (Magnuson, J.J. and Carpenter, S.R. and Stanley, E.H., 2023b, 2023a). The 1-min high-frequency data were averaged to hourly values. Biweekly measurements were taken with a `YSI Pro-ODO` meter (YSI, resolution of 0.1 °C). For high-frequency measurements, `TempLine` loggers (Apprise Tech, resolution of 0.1 °C) were used in 2006, and since 2007 `Concerto` loggers (RBR, resolution of < 0.00005 °C) are used. Data loggers were placed every 0.5 m from the surface through 7 m, and every meter from 7 m to 15 m in 2006, and are placed every 0.5 m from the surface through 2 m, and every meter from 2 m to 20 m since 2007. Fortnightly and high-frequency temperature data were merged into one data set, in which missing hourly and depth-discrete data were extrapolated using cubic-spline interpolation. This temperature data set consists of hourly, depth-discrete (every 0.5 m) water temperature data at the lake's deepest site. As data during the ice-covered period were sparse and interpolation was high, we set all water temperatures of the layer closest to the surface to a freezing temperature of 0 °C whenever air temperature were ≤ 0 °C.

Meteorological forcing data were obtained from the second phase of the North American Land Data Assimilation System (NLDAS-2; Xia et al., 2012). The NLDAS-2 data are at an hourly resolution and a grid cell that covered most of Lake Mendota's surface area was selected (Mitchell, 2004). Meteorological parameters used in this study included wind speed, air temperature, specific humidity, surface pressure, surface downward short- and longwave radiation. Relative humidity was calculated as a function of specific humidity, air temperature, and surface pressure. Cloud cover was calculated as a function of air temperature, relative humidity, shortwave radiation, latitude and longitude, and elevation above sea level. Air vapor saturation was calculated as a function of relative humidity and air temperature.

3 Methods

In the following sections, we highlight the equations and workflow of the process-based lake model, which provided synthetic data for training and testing of deep learning model architectures. To develop a hybrid MCL model (Fig. 1), we first show the necessary steps of modular compositional learning, pretraining and fine-tuning, and how each step performed for the training and testing period against two alternative deep learning models (Fig. 2): one without process information and one without modularisation. Finally, we describe the design ideas behind the hybrid MCL model that combines a fine-tuned deep learning model in the process-based model. To highlight how the hybrid MCL model performs in capturing key physical limnological lake characteristics, we ran the hybrid MCL model against the process-based model, the deep learning with no process, and the deep learning with no modularisation in a time period that was not used previously for neither training nor testing. To avoid confusion, we are using the following nomenclature in this study:

- **KGML**: Knowledge-Guided Machine Learning, a modeling paradigm that aims to combine process-based knowledge and modeling with deep learning models (Karpatne et al., 2017, 2022).
- **MCL**: Modular Compositional Learning, a KGML methodology in which the overall model is decomposed into modular sub-aspects, each modular sub-aspect can be a deep learning model or a process-based model (Karpatne et al., 2017).
- **Pretraining**: Training of an uninitialized deep learning model using process-based model data as targets.
- **Fine-tuning**: Training of an already trained/initialized deep learning model using observed field data as targets.
- **Process-based model**: 1D hydrodynamic lake model that decomposes each time step into the modeling of heat generation, ice and snow formation, vertical diffusive transport, and convective overturn.
- **Hybrid MCL model**: A process-based model in which one (or more) modular sub-aspect is modeled through a deep learning model. In this study, we replaced the process-based diffusive transport calculations with a deep learning model. This deep learning model was pretrained on the process-based model data, and subsequently fine-tuned on observed field data. The hybrid MCL includes process information and is modularised.
- **Deep learning model with no process information**: Deep learning model that is trained on observed field data with general data inputs, e.g., meteorology and lake characteristics.
- **Pretrained deep learning model with no modularisation**: Deep learning model that is pretrained on process-based model output, and subsequently fine-tuned on observed field data. This model has no feedback between the deep learning model and any process-based model.

3.1 Process-based model

A one-dimensional hydrodynamic lake model was developed to simulate the temperature, heat flux and stratification dynamics in a lake. The algorithms are based on the eddy diffusion approach *sensu* Henderson-Sellers (1985) and the MyLake (Saloranta & Andersen, 2007) model. Using the one-dimensional temperature diffusion equation for heat transport, we neglected any inflows and outflows, mass losses due to evaporation and water level changes:

$$\frac{\partial h}{\partial t} = 0 \tag{1}$$

$$A\frac{\partial T}{\partial t} = A\frac{\partial}{\partial z}(K_z\frac{\partial T}{\partial z}) + \frac{\partial}{\partial z}\frac{AH(z)}{\rho_w c_p} + \frac{\partial}{\partial z}\frac{AH_{geo}(z)}{\rho_w c_p} \tag{2}$$

where h is the water level (m), A is lake area (m^2), T is water temperature (°C), t is time (s), K_z is the vertical diffusion coefficient m^{-2}s^{-1}), H is internal heat generation due to incoming solar radiation (W m^{-2}), ρ_w is water density (kg m^{-3}), c_p is specific heat content of water (J kg^{-1} °C^{-1}), and H_{geo} is internal geothermal heat generation (W m^{-2}). Internal heat generation is implemented based on Beer-Lambert law for attenuation of short-wave radiation as a function of a constant light attenuation coefficient:

$$H(z) = (1-\alpha)\, I_s \exp\left(-k_d z\right) \tag{3}$$

where α is the albedo $(-)$, I_s is total incident short-wave radiation (W m^{-2}), and k_d is a light attenuation coefficient (m^{-1}). For the boundary conditions, we assume a Neumann type for the temperature diffusion equation at the atmosphere-surface boundary, and a zero-flux Neumann type at the bottom:

$$\rho_w c_p (K_z \frac{\partial T}{\partial z})_{surface} = H_{net} \tag{4}$$

$$K_z (\frac{\partial T}{\partial z})_{bottom} = 0 \tag{5}$$

where H_{net} is the net heat flux exchange between atmosphere and water column (W m^{-2}). The neat heat flux exchange consisted of four terms:

$$H_{net} = H_{lw} + H_{lwr} + H_v + H_c \tag{6}$$

where H_{lw} is the incoming long-wave radiation (W m^{-2}), H_{lwr} is emitted radiation from the water column (W m^{-2}), H_v is the latent heat flux (W m^{-2}), and H_c is the sensible heat flux (W m^{-2}). Incoming and outgoing long-wave heat fluxes were derived using the formulations from Livingstone and Imboden (1989) and Goudsmit et al. (2002). The latent and sensible heat fluxes were calculated taking into account atmospheric stability using the algorithm by Verburg and Antenucci (2010).

The calculation of a temperature profile at every time step is modularized into four steps: (a) heat generation from boundary conditions, (b) ice and snow formation, (c) vertical diffusion, and (d) convective overturn. The one-dimensional temperature diffusion equation was discretized using the implicit Crank-Nicolson scheme (Press et al., 2007), which being second-order in both space and time allows the modeling time step to be dynamic without numerical stability issues. The model was implemented in Python 3.7 with a time step of $\Delta t = 3,600$ s and a spatial discretization of $\Delta z = 0.5$ m.

3.1.1 Heat generation from boundary conditions (a)

In the first step, the heat fluxes H, H_{geo} and H_{net} are applied over the vertical water column. For Lake Mendota, we set the constant light extinction coefficient k_d to 0.4 m^{-1} based on the upper end of observed Secchi depth measurements since 1995 (Magnuson, J.J. and Carpenter, S.R. and Stanley, E.H., 2023c).

3.1.2 Ice, snow, and snow ice formation (b)

In the second step, the ice and snow cover algorithm from MyLake (Saloranta & Andersen, 2007) was applied to the model. Whenever water temperatures were equal or below the freezing point of water (set to 0 °C), ice formation was triggered. All layers with water temperatures below the freezing point were set to 0 °C, and the heat deficit from atmospheric heat exchange was converted into latent heat of ice formation. Stefan's law was applied to calculate ice thickness when air temperatures were below freezing point triggering ice formation (e.g., Leppäranta (1993)). The formation of a snow layer on top of the ice layer depended on the amount of precipitation. Further, whenever the weight of snow exceeded the buoyancy capacity of the ice layer, enough water to offset the exceedance forms a snow ice layer with the same properties as ice. When air temperatures were above the freezing point, ice and snow growth ceased, and snow and ice melting were initiated with ice melt requiring no snow to exist. Here, total energy of melting was taken from the total heat flux H_{net}. Once the ice layer has disappeared, the default model routine continued. For more details, we refer the reader to Saloranta and Andersen (2007).

3.1.3 Vertical (turbulent) diffusion (c)

In the third step, vertical turbulent diffusion between adjacent grid cells was calculated. Here, we applied a centered difference approximation for temperature at the next time step. The vertical turbulent diffusion coefficients, K_t, were calculated under non-neutral conditions in relation to the Richardson number (Henderson-Sellers, 1985):

$$K_t = \frac{kw^* z}{P_0\left(1 + 37R_i^2\right)} \exp\left(-k^* z\right) \tag{7}$$

where k is the Karman constant ($k = 0.4$), w^* is the wind friction velocity (m s^{-1}) , P_0 is the turbulent Prandlt number ($P_0 = 1.0$), R_i is the Richardson number, and k^* is a function of wind speed and latitude. Friction velocity was calculated as:

$$w^* = C_D U_2 \tag{8}$$

where the drag coefficient C_D was set to 1.3 x 10^{-3}, and U_2 is the wind speed at 2 m above surface (m s^{-1}). The Richardson number was quantified as:

$$R_i = \frac{-1 + \left[1 + 40N^2k^2z^2 / \left(w^{*2} \exp(-2k^* z)\right)\right]^{(1/2)}}{20} \tag{9}$$

with the squared buoyancy frequency, $N^2 = \frac{g}{\rho_w}\frac{\partial \rho_w}{\partial z}$ (s^{-2}). All values of N^2 less than 7.0 x 10^{-5} s^{-2} were set to 7.0 x 10^{-5} s^{-2} (Hondzo & Stefan, 1993).

Further, we applied the turbulent eddy diffusivity modifications from Gu et al. (2015) for the Henderson-Sellers parameterization and a lake depth between 15 to 150 m:

$$K_t = \begin{cases} 10^2 K_t, & \text{if } T_{surface} > 4°\text{C} \\ 10^4 K_t, & \text{if } 0°\text{C} < T_{surface} \leq 4°\text{C} \\ 0, & \text{if } T_{surface} \leq 0°\text{C} \end{cases} \tag{10}$$

To replicate a lag in the mixing dynamics, we set the values of K_t to the average between the current profile and the one from the previous time step (Piccolroaz & Toffolon, 2013). The vertical diffusion coefficient was calculated as:

$$K_z = K_t + K_m \tag{11}$$

with the molecular diffusivity K_m set to 1.4 x 10^{-7} m^2 s^{-1}.

3.1.4 Convective overturn (d)

In the final step, any density instabilities over the vertical water column were mixed with the first stable layer below an unstable layer. Here, we applied the area weighed mean of temperature between two layers to calculate the new temperature of the previously unstable grid cell. Density differences between two layers were averaged until the difference was equal or less than 1 x 10^{-3} kg m^{-3}.

3.2 Modular compositional learning workflow

MCL aims to merge process-based modeling and deep learning to create an overall flexible model with improved performance in which individual processes and state variables are linked through a modularized approach (Fig. 1 1A). The steps to create a hybrid MCL model consisted of:

1. Developing and running a process-based model (see subplots 1A and 2A in Fig. 1)
2. Pretraining step: sequence of deep learning models are pretrained with simulated data to replicate the process-based model output data (see subplots 1B and 2B in Fig. 1, Tab. 1)
3. Fine-tuning step: the pretrained deep learning model surrogating the performance of the diffusion module was trained on the observed water temperature data (see subplots 1C and 2C in Fig. 1, Tab. 1)

4. Developing the hybrid MCL model (described in more detail in Section 3.3): the fine-tuned deep learning model mimicking the diffusion module is put back into the process-based model (see subplots 1D and 2D in Fig. 1)

The process-based model descriptions are detailed above. We note that the process-based model was not thoroughly calibrated using e.g., an automatic optimization algorithm. Only the light extinction coefficient was modified to represent field clarity conditions. For deep learning, we used multi-layer perceptrons (MLP) with 2 hidden layers each with 32 neurons, respectively, and Gaussian Error Linear Units (GELU) activation functions. The selection of the number of layers and neurons per layer often involves a hyperparameter search with a validation set. However, our experiments revealed that a simple configuration of 2 layers with 32 neurons each is capable of effectively capturing the data. Incorporating more layers could potentially yield slightly improved performance. However, we intentionally opted for a simpler model design for each module within our modular compositional framework. Seven years of data (2011-12-31 01:00:00 to 2017-12-28 23:00:00) were used to train and test the process-based model (Fig. 1 1A). During pretraining (Fig. 1 1B), the training and testing data were split 60-40 and each deep learning model was trained for 100 epochs to replicate the temperature output of its respective process-based counterpart (for all inputs and targets see Tab. 1). Training and testing data for the MLP models consisted of hourly, depth-discrete ($\Delta z = 0.5$ m) simulated data from the process-based model output. Although each MLP was trained individually (Fig. 1 1B), overall model performance was evaluated by linking each MLP to the next one (Fig. 1 2B), similar to the process-based model (Fig. 1 2A), i.e., the projected temperature from the first deep learning model replaced the respective input temperature of the next deep learning model (Fig. 1 2B).

In the fine-tuning step (Fig. 1 1C, 7 years of data from 2011-12-31 01:00:00 to 2017-12-28 23:00:00 with a split of 60-40 % for training and testing), each trained deep learning model got linked to mimic the process-based model. The linked deep learning models did not include recurrent information because for every time step, the initial temperature got derived from the the process-based model output and not from the previously projected final temperature profile (see Fig. 1 2C - there is no loop between final projected temperature and the initial temperature of the next time step because the latter was taken from the process-based model simulations). To improve model performance, we only fine-tuned the weights of the third module mimicking the vertical diffusive transport (the weights of all other MLP's were unaltered during fine-tuning). Fine-tuning was done using high-frequency observed water temperature data with 1,000 epochs. We chose to fine-tune the diffusion module as among state-of-the-art hydrodynamic lake models, different approaches are taken in their parameterization of diffusive transport dynamics. Eddy-diffusion models quantify the eddy diffusivity coefficients for turbulent transport as a function of the gradient Richardson number (e.g., Henderson-Sellers (1985) or Hostetler and Bartlein (1990) models), whereas turbulence-based models use additional equations to quantify production and dissipation of turbulent kinetic energy, e.g., the $k-\epsilon$ approach as in Simstrat (Goudsmit et al., 2002) and LAKE2.0 (Stepanenko et al., 2016). For our calculations, we chose the method *sensu* Henderson-Sellers (1985) and parameterised the vertical turbulent diffusivity coefficients as a function depending on the gradient Richardson number, Eq. 9, in which external wind energy is directly used to compute turbulent transport. Although these alternative calculations have common physical assumptions and foundations, the degree to which they replicate the complexity of a specific lake's hydrodynamics is uncertain. Therefore, by letting deep learning estimate the turbulent diffusive transport, we are actively reducing the process uncertainty in the hybrid MCL model.

We further tested the performance of two alternative deep learning models (Fig. 2, Tab. 1). One model, deep learning (no process, Fig. 2 A), acted as our test case to investigate if deep learning with no process-based information in its input data would perform as well as the hybrid MCL model. The other model, pretrained deep learning (no modularisation, Fig. 2 B), had a similar setup as the hybrid MCL model (including pretraining on process-based model simulations and fine-tuning on observed data, Fig. 1 2D), but without the deep learning model being part of the modularised workflow. This model was tested to see if the feedbacks between the process-based modules and the deep learning were improving model performance and stability. The two deep learning models consisted of:

1. Deep learning model (no process): a MLP was trained on the observed data (Fig. 2 A). The deep learning model had the same amount of hidden layers as the hybrid MCL model, and the input data for training were lake characteristics and meteorological driver data (Tab. 1).
2. Pretrained deep learning model (no modularisation): a MLP was pretrained with the final simulation output from the process-based model (Tab. 1). The deep learning model had the same amount of hidden layers as the hybrid MCL model, and the input data for training were lake characteristics, initial projected process-based water temperature and meteorological driver data. The pretrained deep learning model was subsequently fine-tuned on the observed water temperature data (Fig. 2 B)

In total, model performance for the training and testing periods were evaluated for the process-based model, the intermediate MCL steps of pretraining and fine-tuning, the hybrid MCL model, the deep learning model (no process), and the the pretrained deep learning model (no modularisation) on how well they replicated the observed water temperature data for training and testing periods (quantified using the root-mean squared error, RMSE).

3.3 Hybrid MCL model

To develop a hybrid MCL model (Fig. 1 2D) in which the fine-tuned MLP mimicking vertical diffusive transport was integrated into the modularised process-based model, we had to acknowledge the architecture of the pretraining and fine-tuning steps (Fig. 1.2 B+C). We recall that each deep learning model used the projected output of its previous process-based module counterpart during pretraining, and that during fine-tuning the first deep learning model (acting as substitute for heating) received its initial water temperature profile from the process-based model projections, whereas subsequent deep learning models used the simulated output from their deep learning predecessors as input (Fig. 1 2C). This bias of pretraining and fine-tuning to rely on process-based information needed to be replicated in the hybrid MCL model. To ensure the inclusion of unbiased process-based initial values in the hybrid MCL model, we ran two parallel paths inside the hybrid MCL model at each iteration (Fig. 1 2D). For each iteration (representing the calculations for one time step), a complete process-based model calculation is performed (heating, ice, diffusion, convection) and the final process-based model output is used as the input temperature profiles for the next time step (process-based model path in Fig. 1 2D). In parallel, after accounting for ice and snow formation, the fine-tuned deep learning model is run with its output further processed by a process-based convection module. This final temperature profile is stored as the ”true” model output (hybrid path in Fig. 1 2D), which is not used as the initial profile for the next time step. Eventually, these parallel processes mean that at each iteration the process-based modules are run to create a biased, process-based initial temperature profile for the next iteration. But on top of this loop, the deep learning model calculates the final modeled temperature profile for each time step in parallel. This diffusive adjustment by the deep learn-

Configuration	Input data	Pretraining target	Fine-tuning target
Modular compositional learning			
Deep learning model for heating (a)	depth, air temp., longwave radiation, sensible heatflux, latent heatflux, short-wave radiation, light extinction, area, ice, snow, snow ice, initial process-based temperature, day of year, time of day	process-based heating (a) temperature	-
Deep learning model for ice and snow formation (b)	depth, ice, snow, snow ice, initial process-based temperature, process-based heating temperature (a), day of year, time of day	process-based ice temperature (b)	-
Deep learning model for diffusion (c)	depth, area, wind speed, process-based buoyancy profile, process-based diffusivity coefficient values, ice, snow, snow ice, initial process-based temperature, process-based heating temperature (a), process-based ice temperature (b), day of year, time of day	process-based diffusion temperature (c)	observed temperature
Deep learning model for convection (d)	depth, area, ice, snow, snow ice, initial process-based temperature, process-based heating temperature (a), process-based ice temperature (b), process-based diffusion temperature (c), day of year, time of day	process-based convection temperature (d)	-
Alternative deep learning models			
Deep learning model (no process)	depth, air temp., longwave radiation, sensible heatflux, latent heatflux, short-wave radiation, light extinction, area, wind speed, day of year, time of day	observed temperature	-
Pretrained deep learning model (no modularisation)	depth, air temp., longwave radiation, sensible heatflux, latent heatflux, short-wave radiation, light extinction, area, wind speed, day of year, time of day, initial process-based temperature	process-based convection temperature (d)	observed temperature

Table 1. Overview of the deep learning models regarding input data and target variable

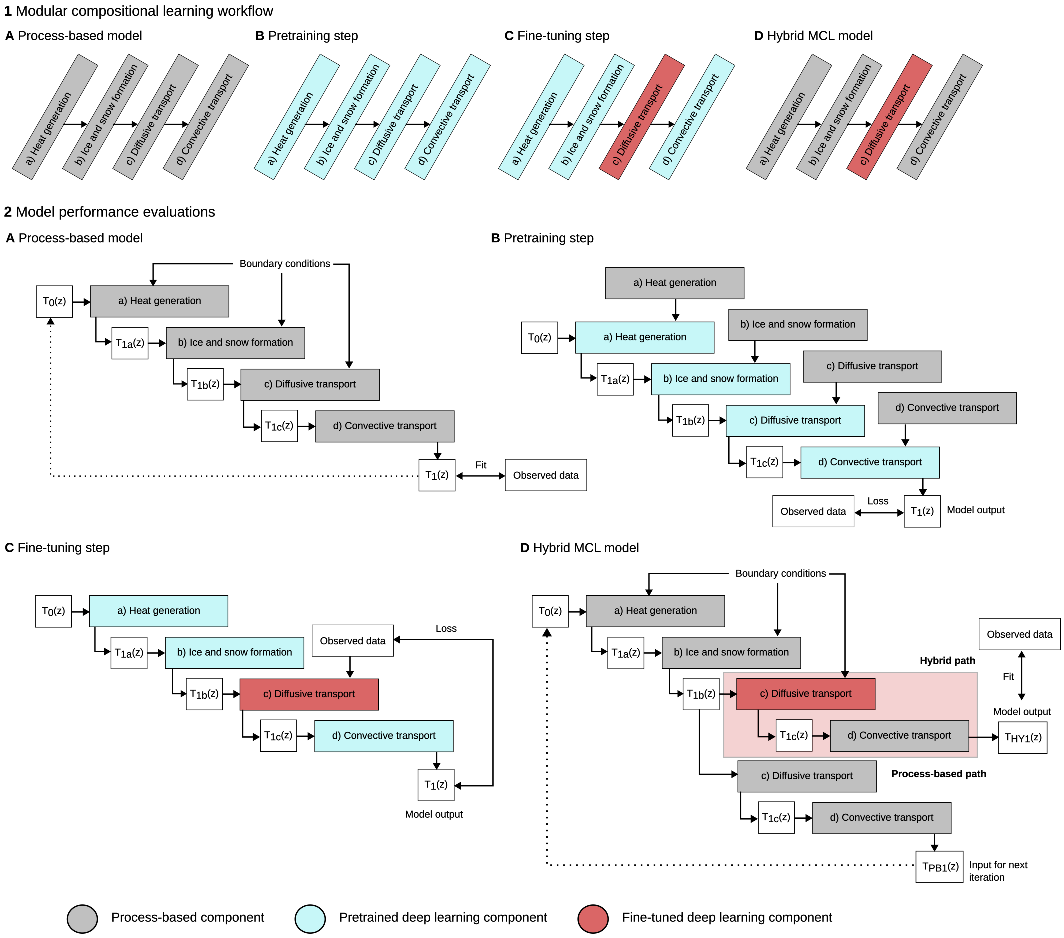

Figure 1. Design process for modular compositional learning. Gray boxes represent process-based modules, cyan boxes represent pretrained deep learning models, and red boxes represent fine-tuned deep learning modules. **1**: Workflow to create the hybrid MCL model consisting of **(A)** the process-based model, **(B)** pretraining step (deep learning models learn to surrogate performance of process-based counterparts), **(C)** fine-tuning step (one deep learning model is trained on observed data), and the **(D)** hybrid MCL model. **2**: Evaluation of model performance for the **(A)** process-based model, the **(B)** pretraining step, the **(C)** fine-tuning step, and the **(D)** hybrid MCL model.

ing module using data-driven information of the temperature profile provided by the process-based model was due to inherent numerical instabilities of the hybrid MCL model. Initial, non-published experiments highlighted that a hybrid MCL framework in which the final projected temperature profile of the hybrid path would be the input for the next iteration was susceptible to numerical oscillations, which over time could develop into unrealistic water temperature values. Therefore, in our hybrid MCL model, the process-based model is used as the structural backbone (for heat fluxes, ice/snow formation and convection and also for temporal evolution), but the diffusive processes are "adjusted" using the deep learning model, which was trained on observed water temperature data.

The hybrid MCL model is tested against the performance of the process-based model, the deep learning model (no process), and the pretrained deep learning model (no modularisation) in replicating the observed data for the period from 2018-04-26 00:00:00 to

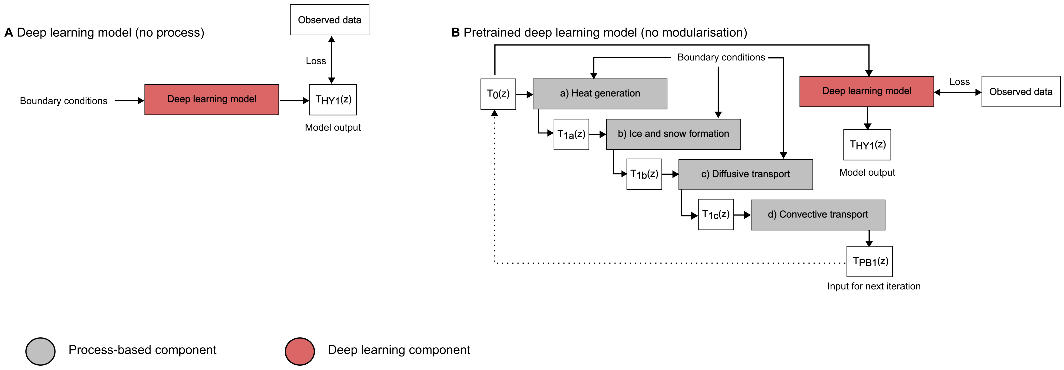

Figure 2. Architectures of alternative deep learning models: the gray boxes represent process-based modules, and red boxes represent trained models. **A**: Evaluation of the deep learning model with no process information performance. **B**: Evaluation of the pretrained deep learning model (no modularisation) performance.

2019-10-27 11:00:00, which was outside of the training-testing period (2011-12-31 01:00:00 to 2017-12-28 23:00:00). We quantified model fits using the RMSE and the Nash-Sutcliffe coefficient of efficiency (NSE). We evaluated model performance for a set of hydrodynamic metrics: whole water temperature profiles over the time period, near-surface water temperatures (depth at 0.5 m), near-bottom water temperature (depth at 24 m), thermocline depths, upper and lower metalimnion depths, the isothermals of 13, 15 and 17 °C over time, max. buoyancy frequency, and Schmidt stability. Thermocline depths represent if the model can accurately replicate vertical transport dynamics as well as internal mixing processes due to oscillations or entrainment, whereas the isothermals at temperatures close to the thermocline explore the ability of the model to mimick oscillations due to internal waves. Maximum buoyancy frequency was quantified as the maximum value of each hour's profile of the Brunt–Väisälä frequency, or squared buoyancy frequency N^2 (Lerman et al., 1995):

$$N^2 = \frac{g}{\rho_0}\frac{\partial \rho}{\partial z} \tag{12}$$

where g is gravitational acceleration (m s^{-2}). The Schmidt stability, St (J m^{-2}) (Idso, 1973; Schmidt, 1928), quantifies the amount of external energy needed to mix the entire water column without affecting the amount of stored internal energy:

$$St = \frac{g}{A_0}\int_0^{z_{max}} (z - z_g)\,(\rho_z - \hat{\rho_z})\,A_z dz \tag{13}$$

where z is the depth referenced from the water surface, z_g is the depth of the center of mass, and $\hat{\rho_z}$ is the mean density.

Further, we investigated if the respective models produced unstable density profiles over the water column. For this, we calculated average epilimnion and metalimnion densities, respectively, and compared their differences over time for the process-based model, the hybrid MCL model, the deep learning model (no process), and the pretrained deep learning model (no modularisation). To understand if the hybrid MCL model and the two deep learning models projected unrealistic fluctuations around their output variables, we quantified monthly signal-to-noise ratios by dividing the monthly mean by its standard deviation for surface water temperature, bottom water temperature and Schmidt stability.

Model	Train RMSE (°C)	Test RMSE (°C)
Process-based model	5.31	4.46
Pretraining step	4.78	5.27
Fine-tuning step	1.31	1.94
Hybrid MCL model	1.97	1.60
Deep learning model (no process)	**0.83**	2.10
Pretrained deep learning model (no modularisation)	1.42	**1.42**

Table 2. Performance of models in recreating the full observed temperature profiles of the training (2011-12-31 to 2015-08-05) and test periods (2015-08-06 to 2017-12-28). Bold numbers highlight the best performance metric.

3.4 Computational implementation

The process-based, the deep learning models, and the hybrid MCL model were developed and run in Python 3.7. Deep learning models were trained using `PyTorch` v1.11.0 (Paszke et al., 2019). All calculations to assess the performance of the hybrid models and its competitors were done using R v4.3.1 (R Core Team, 2023) and the package `rLakeAnalyzer` (Read et al., 2011).

4 Results

4.1 Performance of modular compositional learning

The process-based model had shortcomings replicating the field temperature dynamics of the test period (Tab. 2, Fig. 3 A+D). The deep-water heat transport is underestimated, resulting in bottom temperatures being approx. 5 °C colder than what the field data suggest. Further, observed temperature fluctuations near the thermocline were not replicated by the process-based model. The pretraining step, with four individual deep learning models surrogating the performance of their process-based counterparts, achieved a similar performance as the process-based model (Tab. 2, Fig. 3 A+E1). Once trained or fine-tuned on observed data, the deep learning model (no process), the pretrained deep learning model (no modularisation), the fine-tuning step, and the hybrid MCL model are able to capture the thermal dynamics of the observed data with RMSE's for the test period of 2.10, 1.42, 1.94, and 1.60 °C, respectively (Tab. 2, Fig. 3 B+C, E2+F). Training performance suggests that all fine-tuned deep learning models had a very similar performance (1.94 and 1.42 °C for fine-tuning step and pretrained deep learning with no modularisation, respectively). The deep learning model with no process information had a better performance during training than testing. The combination of pretraining and fine-tuning caused the error to decrease from an initial RMSE of 4.46 °C (process-based model) to 1.60 °C (hybrid MCL model). Past modeling studies (process-based and hybrid KGML models) achieved a similar performance for water temperature simulations, i.e., 1.96 °C in Ladwig et al. (2021) and 1.56 °C in Read et al. (2019).

4.2 Performance of the hybrid MCL model

The performance of the hybrid MCL model was further evaluated against the process-based model, the deep learning model (no process), and the pretrained deep learning model (no modularisation) for the time period 2018-04-26 00:00:00 to 2019-10-27 11:00:00 (data that were previously not used in training/testing). The hybrid MCL model vastly out-

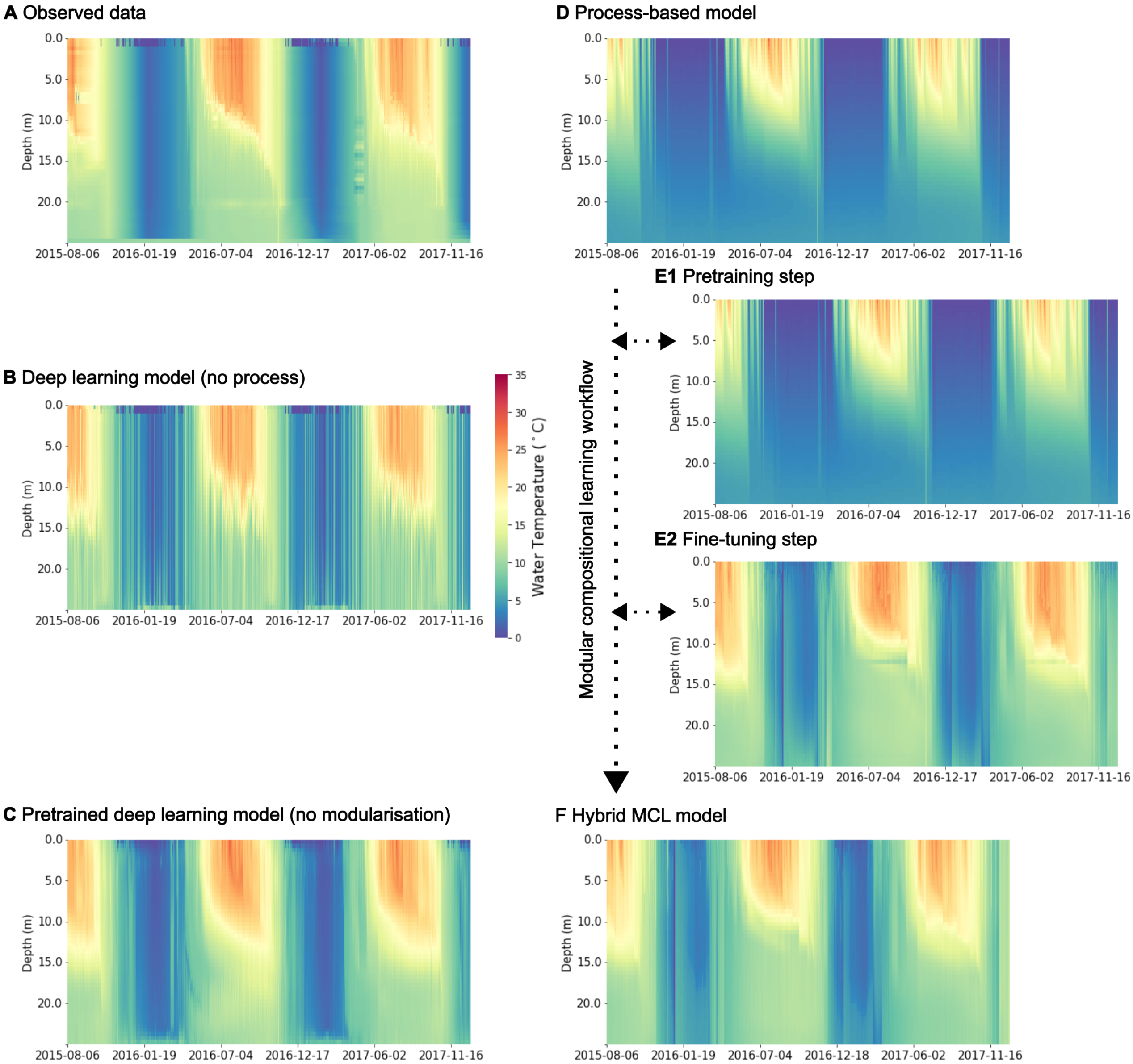

Figure 3. Model performance for replicating thermodynamics during the test period (2015-08-06 to 2017-12-28) visualised through heatmaps of vertical water temperature profiles over time. **A**: Observed data. **B**: Output from deep learning model with no process information (Fig. 2 A) with erratic oscillations. **C**: Output from pretrained deep learning model with no modularisaton (Fig. 2 B). **D**: Output from process-based model highlighting the shortcomings of the process-based model to replicate deep-water heat transport (Fig. 1 2A). **E1**: Output from pretraining step, which is almost identical to process-based model performance as the process-based output was used to train the deep learning models (Fig. 1 2B). **E2**: Output from fine-tuning step, almost identical to hybrid output for the testing period (Fig. 1 2C). Model performance improved for deep-water heat transport compared to process-based model. **F**: Output from hybrid MCL model (Fig. 1 2D).

performed the process-based model (Tab. 3) regarding the replication of total temperature dynamics (RMSE of 2.14 and 4.63 °C, respectively), water column stability (i.e., NSE of Schmidt stability of 0.92 and 0.79, respectively), and density layer fluctuations (i.e., NSE of 15 °C isotherm depth of 0.68 and -1.09, respectively). The improved performance of the hybrid MCL model compared to the process-based model is further highlighted in the time series of projected surface and bottom water temperature dynamics (Fig. 4 A+B). The process-based model is not able to capture deep-water heat transport (revealed by its nearly constant bottom water temperature through summer stratification), and its earlier decline of surface water temperatures. Further, the process-based model generally underestimated the depths of the thermocline, upper metalimnion, lower metalimnion and all investigated isotherms (Tab. 3, Fig. 4).

The hybrid MCL model and the pretrained deep learning model with no modularisation performed similarly for water temperature dynamics, energy budgets, and density layer depths. However, the pretrained deep learning (no modularisation) has a better projection of surface water temperatures than the hybrid MCL model, RMSE of 1.57 to 2.12 °C and NSE of 0.97 to 0.94, respectively. Conversely, the hybrid MCL model better projected overall heat budgets the both alternative deep learning models: NSE of Schmidt stability of 0.92, and NSE of maximum buoyancy frequency of 0.25 (Tab. 3, Fig. 4 C-D). Although performance metrics suggest that the deep learning model (no process) has good projections regarding thermocline and metalimnion depths (i.e., RMSE of thermocline depth of 3.16 m, which is similar to the RMSE of 2.71 m by the hybrid MCL model), subplots E+F as well as H+I in Fig. 4 reveal that the deep learning model has profound high-frequency noise around its projections. Overall, the deep learning model with no process information performed worse than the other deep learning models for temperature projections, which is further evident in the depths of the isotherms, e.g., RMSE for the 17 °C isotherm of 1.57 m and 2.97 m for hybrid MCL model and deep learning (no process), respectively. The hybrid MCL model and the pretrained deep learning model (no modularisation) failed to project surface temperatures close to 0 °C during the winter season of 2018-2019 (Fig. 4 A), similar to the interpolated observed data. Compared to the results presented in Section 4.1, the performance of the deep learning with no process has a worse performance for the evaluation period 2018-2019 than during the training and testing periods. Contrary, the performance of the hybrid MCL model improved during this additional verification period compared to its initial performance during training and testing.

To investigate whether the models project unrealistically long unstable conditions, with the assumption that any density instability in the water column would be resolved rather quickly under field conditions, we compared depth-integrated epilimnetic and metalimnetic water densities over time (Fig. 5). The hybrid MCL model occasionally projected density instabilities (meaning that average metalimnion density was lower than average epilimnion density), which were resolved quickly and only occurred during the winter ice-covered period. Both other deep learning models produced more frequent density instabilities (Fig. 5), especially during turnover conditions (before and after summer stratification). The process-based model did also produce density instabilities during turnover conditions as the convective mixing algorithm only considered a threshold of equal or less than 1 x 10^{-3} kg m^{-3} for mixing, which, when integrated over a layer, can result in occasional unstable profiles. However, the models' susceptibility to generate unstable conditions was not critical (max. density differences were $< 2 \times 10^{-1}$ kg m^{-3}).

The signal-to-noise ratios were generally low for the deep learning model (no process) during the winter seasons for surface and bottom water temperature (Fig. 6 A+B). The deep learning model with no process further produced low signal-to-noise ratios for bottom water temperatures (Fig. 6 B). Regarding water column stability, all three deep learning models had similar signal-to-noise ratios (Fig. 6 C). Some of the models occasionally projected unstable density profiles that resulted in negative Schmidt stability

Variable	Process-based model	Hybrid MCL model	Deep learning (no process)	Pretrained deep learning (no modularisation)
		RMSE (NSE)		
Whole water temperature profiles (°C)	4.63 (0.53)	2.14 (**0.90**)	4.14 (0.62)	**2.11** (**0.90**)
Surface temperature (°C	4.34 (0.77)	2.12 (0.94)	5.99 (0.57)	**1.57** (**0.97**)
Bottom temperature (°C)	4.02 (-1.35)	**1.73** (**0.56**)	3.10 (-0.40)	2.10 (0.35)
Schmidt Stability (J m^{-2})	139.92 (0.79)	**85.43** (**0.92**)	255.54 (0.31)	85.85 (**0.92**)
Max. buoyancy frequency (s^{-2})	0.006 (-0.14)	**0.004** (**0.25**)	0.006 (-0.19)	0.005 (-0.02)
Thermocline depth (m)*	5.63 (-4.64)	2.71 (-0.31)	3.16 (-0.77)	**2.65** (**-0.25**)
Upper metalimnion depth (m)*	6.93 (-5.21)	4.05 (-1.13)	**3.60** (**-0.68**)	4.71 (-1.88)
Lower metalimnion depth (m)*	3.28 (-2.00)	**1.64** (**0.24**)	2.88 (-1.31)	2.53 (-0.78)
13 °C isotherm depth (m)	4.40 (-0.76)	**1.94** (**0.65**)	3.64 (-0.20)	3.88 (-0.37)
15 °C isotherm depth (m)	4.34 (-1.09)	**1.68** (**0.68**)	2.94 (0.03)	2.58 (0.25)
17 °C isotherm depth (m)	4.16 (-1.94)	**1.57** (**0.57**)	2.97 (-0.50)	2.34 (0.08)

Table 3. Performance of the process-based model, the hybrid MCL model, and deep learning model with no process information, and the pretrained deep learning model with no modularisation regarding the replication of a set of water quality variables for the period 2018-04-26 00:00:00 to 2019-10-27 11:00:00. RMSE is given outside of brackets, whereas NSE is given inside of brackets. Bold numbers highlight the best performance metric for each variable of interest.

*Variable fit was calculated from June to September to avoid a bias to strong density fluctuations during overturn.

values, hence a negative signal-to-noise ratio. The hybrid MCL model and the pretrained deep learning model with no modularisation simulated consistently higher signal-to-noise ratios than the deep learning model (no process). The lower signal-to-noise ratios for the deep learning model (no process) are potentially caused by the spurious oscillations of its water temperature projections (Fig. 3 B, Fig. 4).

5 Discussion and conclusions

A hybrid MCL model incorporating process-based formulations and trained deep learning models through MCL improved overall model performance and provided physically sound results. Compared to three alternative configurations, a process-based model, a deep learning model with no process information, and a pretrained deep learning model with no modularisation, the hybrid MCL model had the overall best replications of physical limnological characteristics of Lake Mendota. Through fine-tuning, the overall hybrid MCL model learned how to enhance deep-water heat transport as well as thermocline oscillations compared to the original process-based model. The alternative deep learning model with no process information (a design heavily inspired by discovery applications, hence learning novel relationships between input data and target without hard-wiring potential interactions) performed very well during training, but had several short-

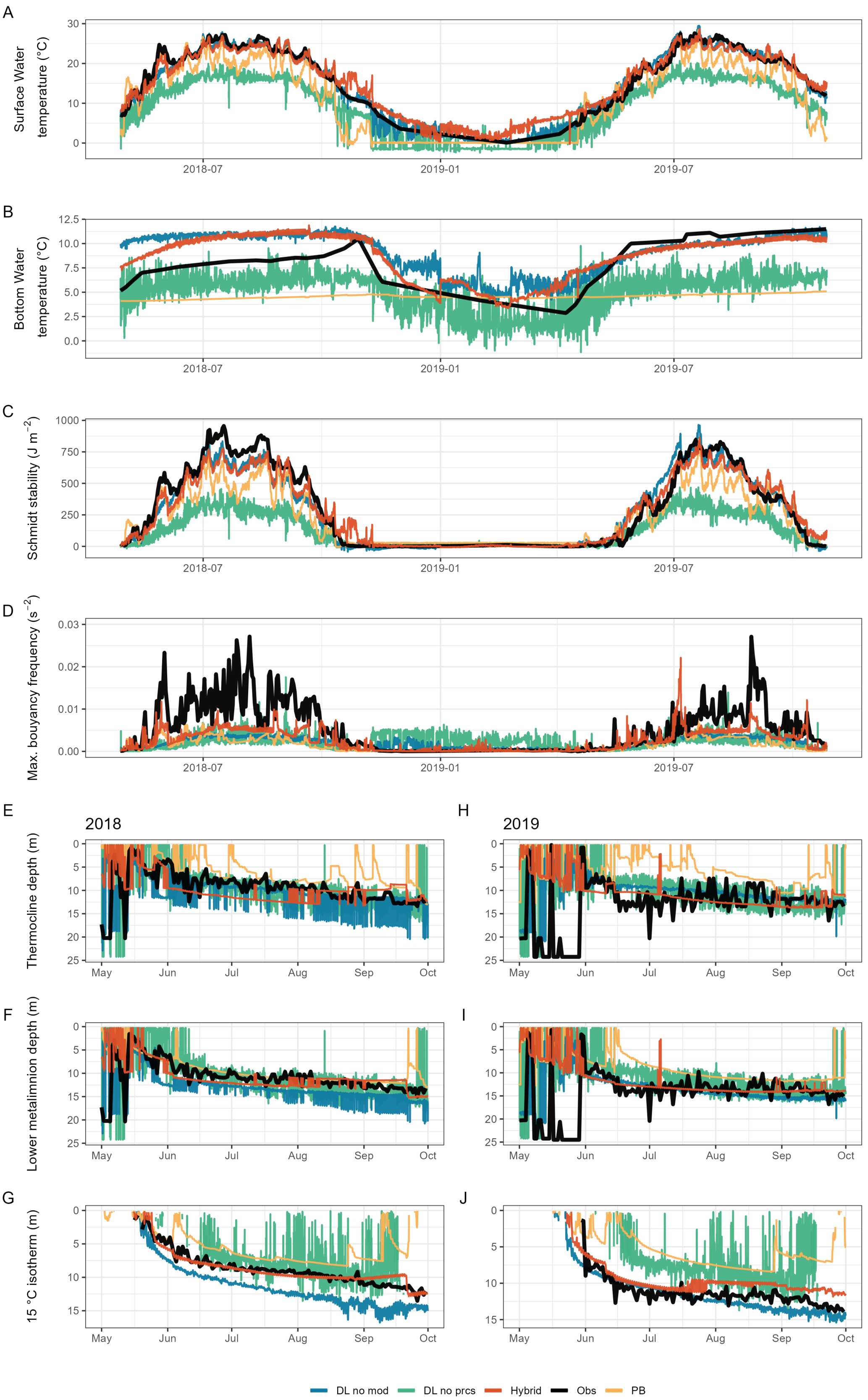

Figure 4. Performance analysis of the four models to replicate physical metrics: blue represents the pretrained deep learning model with no modularisation, green represents the deep learning model with no process information, orange represents the hybrid MCL model, black represents observed data, yellow line represents the process-based model. **A**: Surface water temperature dynamics. **B**: Bottom water temperature dynamics. **C**: Schmidt stability dynamics. **D**: Max. buoyancy frequency dynamics. **E**: Thermocline dynamics in 2018. **F**: Lower metalimnion depth dynamics in 2018. **G**: 15 °C isotherm dynamics in 2018. **H**: Thermocline dynamics in 2019. **I**: Lower metalimnion depth dynamics in 2019. **J**: 15 °C isotherm dynamics in 2019.

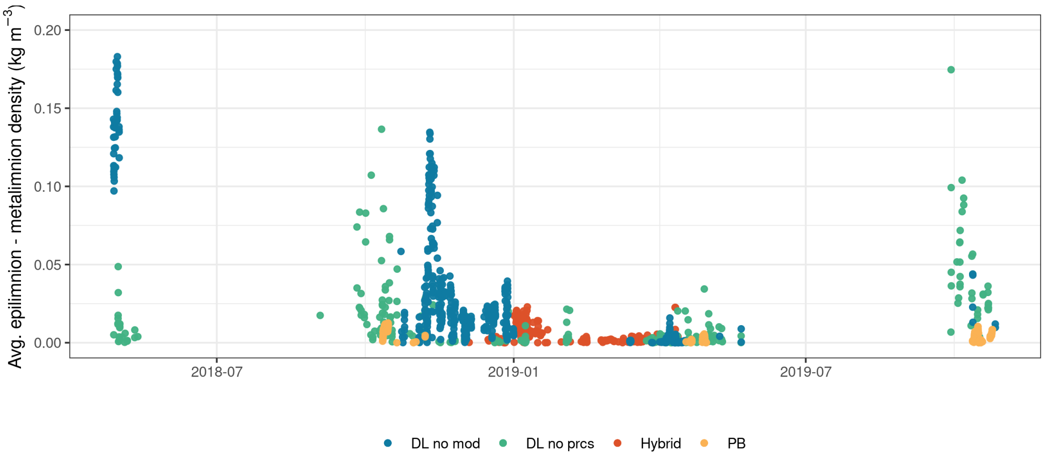

Figure 5. Density differences between averaged epilimnion and metalimnion layers. Here, only density violations ($\frac{\rho_{Epi}}{\rho_{Meta}} > 1$) are highlighted: blue represents the pretrained deep learning model with no modularisation, green represents the deep learning model with no process information, orange represents the hybrid MCL model, yellow represents the process-based model.

comings during the model application to a new time frame: (a) worse performance than the hybrid MCL model and the pretrained deep learning model (no modularisation), (b) more physically-unrealistic density profiles, and (c) lower signal-to-noise ratio. A pretrained deep learning model that lacked any modularisation between the process-based component and the deep learning component performed similarly to our hybrid MCL model, but eventually also produced occasional and more pronounced unstable density profiles.

In this study, we merged a simple, process-based 1D hydrodynamic model with a fine-tuned MLP both (1) to highlight the potential of the MCL methodology and (2) to develop a hybrid MCL model. Notably, there are several shortcomings in the current methodology. The process-based model was not thoroughly calibrated. This step was omitted intentionally to highlight MCL's potential to improve the performance of a broadly uncalibrated process-based model. Regarding process uncertainty, a more advanced process-based formulation, i.e. as in LAKE2.0 (Stepanenko et al., 2016), that is better guided by physical theory, or straightforward improvements like transient light extinction coefficients instead of a constant value could improve the pretraining of the deep learning models, which would improve overall hybrid MCL model performance. Further, memory of past events is only included in the hybrid MCL model through process-based calculations; memory could likewise be added to the deep learning through recurrent deep learning models. Data uncertainty is confounding our results, as observed data were sparse during ice-covered conditions and hence interpolated. The pretraining did not consider any testing data to tune the deep learning model because we used test data exclusively only for tuning hyperparameters and validating model performance. Potential bias could also originate from interpolating the original observed water temperature data to match the resolution and time step of the process-based model. Data interpolation can add fictitious trends to depth-discrete water temperature time series data, as well as add interpolation artifacts. However, all these limitations do not weaken the overall study outcome of highlighting how MCL can guide future hybrid KGML model developments.

Balancing between design and discovery, and best practises to incorporate process knowledge into deep learning models, will guide future developments of hybrid KGML

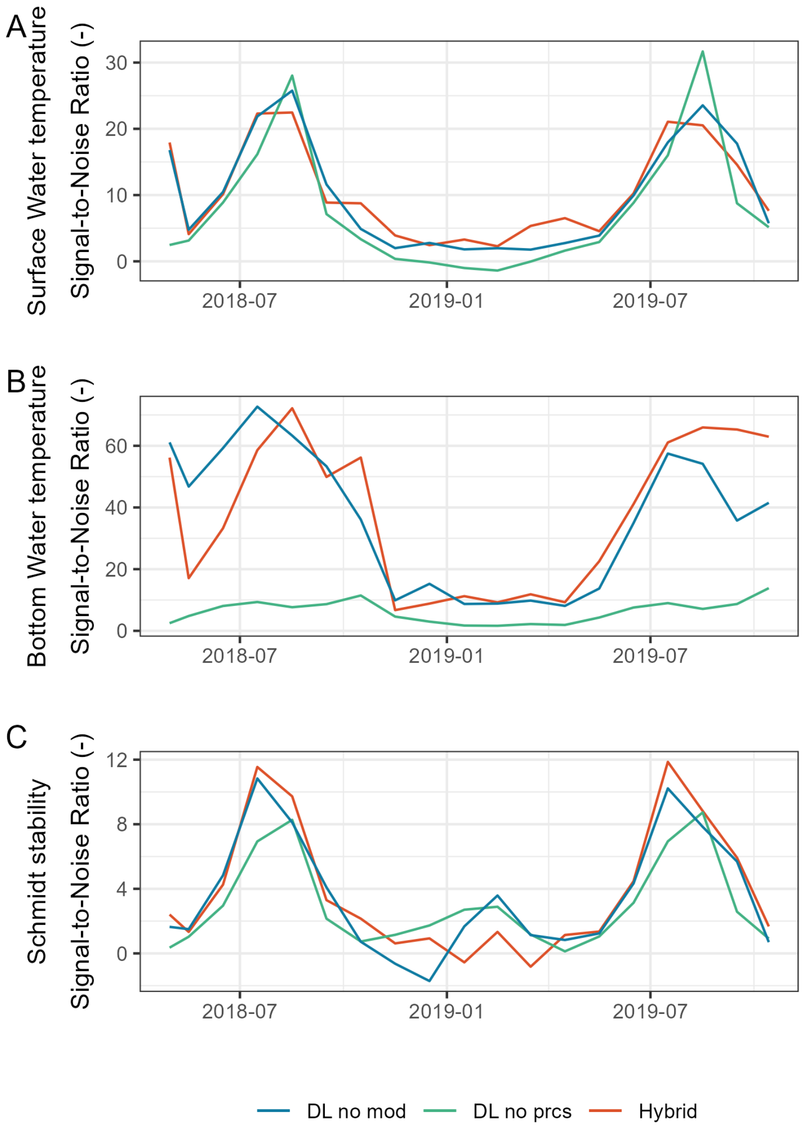

Figure 6. Signal-to-Noise ratios (mean divided by standard deviation) for monthly values: blue represents the pretrained deep learning model with no modularisation, green represents the deep learning model with no process information, orange represents the hybrid MCL model. **A**: Surface water temperature dynamics. **B**: Bottom water temperature dynamics. **C**: Schmidt stability dynamics.

models. In this study, we have highlighted the potential of MCL, in which the final hybrid MCL model directly couples process-based modules with a fine-tuned deep learning model. The worse performance of the deep learning model with no process information reveals again the importance of including discipline-specific expertise in the design of machine learning models. Although we used MLPs and only accounted for recurrence through process information, these results highlight future potentials for aquatic ecosystem modeling. Similar to the approach in this study, future hybrid MCL models could incorporate *in-situ* data to fine-tune incorporated deep learning models for different modular sub-aspects (Fig. 7 A). Further, coupled ecosystem models could improve projections of complex food web dynamics by driving zooplankton projections using long-term and high-frequency data through a deep learning model, whereas other components, e.g., nutrient dynamics and phytoplankton, could be simulated using process-based modeling (Fig. 7 A). On a technical level, deep learning models can utilize data from remote sensing to incorporate spatial information, which is challenging to be included in 1D process-based models due to static input-output relationships (Fig. 7 A).

A next step for hybrid MCL models could be to add a water quality modules, such as coupling a dissolved oxygen concentrations to the hydrodynamic model. For example, dissolved oxygen models account for lake metabolism through:

$$A\frac{\partial DO}{\partial t} = A\frac{\partial}{\partial z}(K_z\frac{\partial DO}{\partial z}) + A\frac{\partial}{\partial z}NEP \tag{14}$$

where DO is the dissolved oxygen concentration (kg m^{-3}), and NEP is net ecosystem production (kg m^{-3} d^{-1}), which is the difference between an ecosystem's gross primary production and its respiration (Hoellein et al., 2013). Metabolism formulations account for the atmospheric exchange and the sediment oxygen demand (a term included in ecosystem respiration) as boundary conditions near the surface and near the sediment, respectively (e.g., Perga et al. (2023); Ladwig et al. (2022)). Using a MCL approach, both, temperature and oxygen calculations could be hybrid MCL models, in which diffusion is estimated by deep learning model (and as an input for the vertical dissolved oxygen transport), but net ecosystem production could be estimated by a deep learning model for the water quality side. Here, additional *in-situ* data (e.g., zooplankton biomass, depth of the photic zone, Chlorophyll-a concentrations) can be used to get improved forecasts of net ecosystem production (Fig. 7 B). We envision that the formulation of future hybrid MCL models has the potential to not only improve model performance through better designed KGML models but also through the discovery of feedback relationships between input data and target variables.

In this study we have highlighted the potential of MCL, which is a novel design philosophy for creating hybrid KGML models consisting of process-based and deep learning models. The developed hybrid MCL model, which has process-based formulations for heating, snow and ice formation, and convection, that was coupled to a pretrained and fine-tuned deep learning model to account for diffusive transport, produced improved projections for lake thermal variables compared to the original process-based model, a deep learning model with no process information, and a pretrained deep learning model without modularisation. Past hybrid KGML models have focused mostly on developing a single, deep learning model to project one target variable (Read et al., 2019; Hanson et al., 2020). The breaking down of processes into modules and assigning them either process-based calculations if the domain expertise is high or deep learning if process formulation is uncertain allows modelers flexibility in balancing design choices with opportunities for discovery. Further, by ensuring that deep learning is an integrated part of a process-based model, the results are physically valid and to a certain extent explainable as the whole model is not a black box. As the deep learning side is pretrained with synthetic data, the overall hybrid MCL model can be used under (observed) data-scare conditions. Further development of hybrid KGML models that support understanding, exploring, and mitigating the impacts of global changes on water resources could support water resources managers and other decision makers.

6 Open Research

All development code and input data for the models as well as output are available at `https://github.com/robertladwig/LakePIAB` with scripts in R and Python (relying on PyTorch for training the deep learning models) under a GNU General Pub-

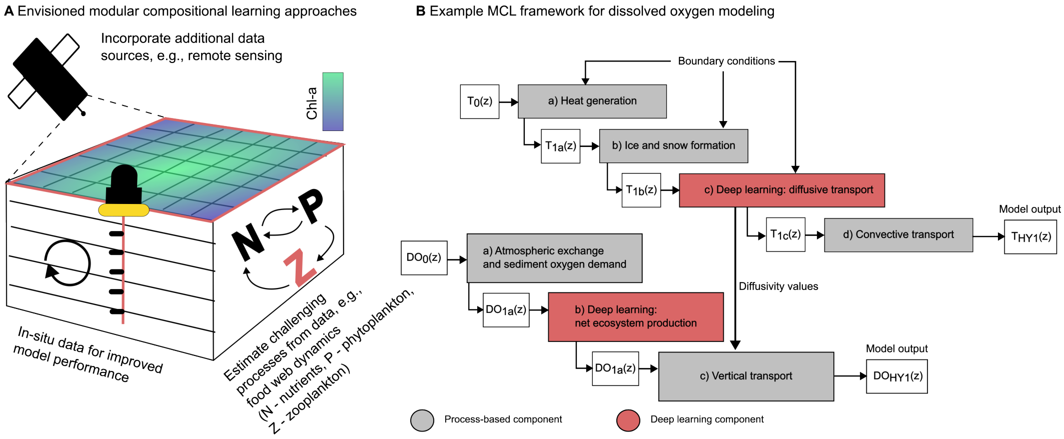

Figure 7. Outlook for future modular compositional learning (MCL) approaches. **A**: Future MCL approaches can directly incorporate *in-situ* data to improve model performance, replace uncertain process formulations in food web modeling with deep learning, and/or directly incorporate additional data like remote sensing in the deep learning model. **B**: Proposed framework for a coupled water temperature - dissolved oxygen hybrid MCL model in which dissolved oxygen calculations are coupled to temperature through simulated diffusivity coefficients for vertical transport. On the dissolved oxygen side, atmospheric exchange and sediment oxygen demand calculations are process-based, whereas net primary production is data-driven through deep learning.

lic License Version 2.0 (GPL-2.0). All data and scripts will be archived at Zenodo data repository once manuscript is accepted.

Acknowledgments

Lake data were obtained from the North Temperate Lakes Long-Term Ecological Research program (#DEB-1440297 and #DEB-2025982). The project was supported through a United States National Science Foundation (NSF) ABI development grant (#DBI 1759865), UW-Madison Data Science Initiative grant, and NSF MSB grant (#2213549). RL was funded by UW-Madison's Integrative Biology Postdoctoral Fellowship. MFM was funded by a Mendenhall Fellowship from the U.S. Geological Survey Water Mission Area. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government. EAA was supported by the NSF Graduate Research Fellowship Program (DGE-1747503) with additional funding from the Wisconsin Alumni Research Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We are thankful for the feedback from USGS-initiated peer reviews, which vastly improved this manuscript.

References

Appling, A. P., Oliver, S. K., Read, J. S., Sadler, J. M., & Zwart, J. A. (2022). Machine Learning for Understanding Inland Water Quantity, Quality, and Ecology. In *Encyclopedia of Inland Waters* (pp. 585–606). Elsevier. Retrieved 2023-05-01, from `https://linkinghub.elsevier.com/retrieve/pii/B9780128191668001213` doi: 10.1016/B978-0-12-819166-8.00121-3

Beven, K. (2006, March). A manifesto for the equifinality thesis. *Journal of Hydrology*, *320*(1), 18–36. Retrieved 2021-03-05, from https://www.sciencedirect.com/science/article/pii/S002216940500332X doi: 10.1016/j.jhydrol.2005.07.007

Bruggeman, J., & Bolding, K. (2014, November). A general framework for aquatic biogeochemical models. *Environmental Modelling & Software*, *61*, 249–265. Retrieved 2023-05-01, from https://www.sciencedirect.com/science/article/pii/S1364815214001066 doi: 10.1016/j.envsoft.2014.04.002

Daw, A., Karpatne, A., Watkins, W., Read, J., & Kumar, V. (2021). Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv*.

Goudsmit, G.-H., Burchard, H., Peeters, F., & Wüest, A. (2002). Application of k-ε turbulence models to enclosed basins: The role of internal seiches. *Journal of Geophysical Research: Oceans*, *107*(C12), 23–1–23–13. Retrieved 2020-08-31, from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001JC000954 doi: 10.1029/2001JC000954

Gu, H., Jin, J., Wu, Y., Ek, M. B., & Subin, Z. M. (2015, April). Calibration and validation of lake surface temperature simulations with the coupled WRF-lake model. *Climatic Change*, *129*(3), 471–483. Retrieved 2023-05-01, from https://doi.org/10.1007/s10584-013-0978-y doi: 10.1007/s10584-013-0978-y

Guerrero, J.-L., Pernica, P., Wheater, H., Mackay, M., & Spence, C. (2017, December). Parameter sensitivity analysis of a 1-D cold region lake model for land-surface schemes. *Hydrology and Earth System Sciences*, *21*(12), 6345–6362. Retrieved 2023-05-25, from https://hess.copernicus.org/articles/21/6345/2017/ doi: 10.5194/hess-21-6345-2017

Hanson, P. C., Stillman, A. B., Jia, X., Karpatne, A., Dugan, H. A., Carey, C. C., ... Kumar, V. (2020, August). Predicting lake surface water phosphorus dynamics using process-guided machine learning. *Ecological Modelling*, *430*, 109136. Retrieved 2020-06-11, from http://www.sciencedirect.com/science/article/pii/S0304380020302076 doi: 10.1016/j.ecolmodel.2020.109136

Henderson-Sellers, B. (1985, December). New formulation of eddy diffusion thermocline models. *Applied Mathematical Modelling*, *9*(6), 441–446. Retrieved 2023-05-01, from https://www.sciencedirect.com/science/article/pii/0307904X85901106 doi: 10.1016/0307-904X(85)90110-6

Hipsey, M. R., Bruce, L. C., Boon, C., Busch, B., Carey, C. C., Hamilton, D. P., ... Winslow, L. A. (2019, January). A General Lake Model (GLM 3.0) for linking with high-frequency sensor data from the Global Lake Ecological Observatory Network (GLEON). *Geoscientific Model Development*, *12*(1), 473–523. Retrieved 2019-08-22, from https://www.geosci-model-dev.net/12/473/2019/ doi: https://doi.org/10.5194/gmd-12-473-2019

Hoellein, T. J., Bruesewitz, D. A., & Richardson, D. C. (2013). Revisiting Odum (1956): A synthesis of aquatic ecosystem metabolism. *Limnology and Oceanography*, *58*(6), 2089–2100. Retrieved 2020-11-24, from https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.4319/lo.2013.58.6.2089 doi: https://doi.org/10.4319/lo.2013.58.6.2089

Hondzo, M., & Stefan, H. G. (1993, November). Lake Water Temperature Simulation Model. *Journal of Hydraulic Engineering*, *119*(11), 1251–1273. Retrieved 2023-07-03, from https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9429%281993%29119%3A11%281251%29 doi: 10.1061/(ASCE)0733-9429(1993)119:11(1251)

Hostetler, S. W., & Bartlein, P. J. (1990, October). Simulation of lake evaporation with application to modeling lake level variations of Harney-Malheur Lake, Oregon. *Water Resources Research*, *26*(10), 2603–2612. Retrieved 2023-05-25, from http://doi.wiley.com/10.1029/WR026i010p02603 doi:

10.1029/WR026i010p02603

Håkanson, L. (2009). Modeling of lake ecosystems. In G. E. Likens (Ed.), *Encyclopedia of inland waters* (p. 441-447). Oxford: Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B978012370626300212X doi: https://doi.org/10.1016/B978-012370626-3.00212-X

Idso, S. B. (1973, July). On the concept of lake stability. *Limnology and Oceanography*, *18*(4), 681–683. Retrieved 2020-02-18, from http://doi.wiley.com/10.4319/lo.1973.18.4.0681 doi: 10.4319/lo.1973.18.4.0681

Ishikawa, M., Gonzalez, W., Golyjeswski, O., Sales, G., Rigotti, J. A., Bleninger, T., … Lorke, A. (2022, March). Effects of dimensionality on the performance of hydrodynamic models for stratified lakes and reservoirs. *Geoscientific Model Development*, *15*(5), 2197–2220. Retrieved 2023-01-18, from https://gmd.copernicus.org/articles/15/2197/2022/ doi: 10.5194/gmd-15-2197-2022

Jane, S. F., Mincer, J. L., Lau, M. P., Lewis, A. S. L., Stetler, J. T., & Rose, K. C. (2022, December). Longer duration of seasonal stratification contributes to widespread increases in lake hypoxia and anoxia. *Global Change Biology*, gcb.16525. Retrieved 2022-12-07, from https://onlinelibrary.wiley.com/doi/10.1111/gcb.16525 doi: 10.1111/gcb.16525

Janssen, A. B. G., Arhonditsis, G. B., Beusen, A., Bolding, K., Bruce, L., Bruggeman, J., … Mooij, W. M. (2015, December). Exploring, exploiting and evolving diversity of aquatic ecosystem models: a community perspective. *Aquatic Ecology*, *49*(4), 513–548. Retrieved 2020-08-31, from http://link.springer.com/10.1007/s10452-015-9544-1 doi: 10.1007/s10452-015-9544-1

Jia, X., Willard, J., Karpatne, A., Read, J. S., Zwart, J. A., Steinbach, M., & Kumar, V. (2021, May). Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles. *ACM/IMS Transactions on Data Science*, *2*(3), 20:1–20:26. Retrieved 2023-05-01, from https://dl.acm.org/doi/10.1145/3447814 doi: 10.1145/3447814

Jia, X., Zwart, J., Sadler, J., Appling, A., Oliver, S., Markstrom, S., … Kumar, V. (2020, December). *Physics-Guided Recurrent Graph Networks for Predicting Flow and Temperature in River Networks.* arXiv. Retrieved 2023-05-01, from http://arxiv.org/abs/2009.12575 (arXiv:2009.12575 [physics]) doi: 10.48550/arXiv.2009.12575

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., … Kumar, V. (2017, October). Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*, *29*(10), 2318–2331. doi: 10.1109/TKDE.2017.2720168

Karpatne, A., Kannan, R., & Kumar, V. (2022). *Knowledge Guided Machine Learning: Accelerating Discovery using Scientific Knowledge and Data.* Retrieved 2023-05-31, from https://www.routledge.com/Knowledge-Guided-Machine-Learning-Accelerating-Discovery-using-Scientific/Karpatne-Kannan-Kumar/p/book/9780367693411

Ladwig, R., Appling, A. P., Delany, A., Dugan, H. A., Gao, Q., Lottig, N., … Hanson, P. C. (2022). Long-term change in metabolism phenology in north temperate lakes. *Limnology and Oceanography*, *67*(7), 1502–1521. Retrieved 2022-07-19, from https://onlinelibrary.wiley.com/doi/abs/10.1002/lno.12098 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/lno.12098) doi: 10.1002/lno.12098

Ladwig, R., Hanson, P. C., Dugan, H. A., Carey, C. C., Zhang, Y., Shu, L., … Cobourn, K. M. (2021). Lake thermal structure drives inter-annual variability in summer anoxia dynamics in a eutrophic lake over 37 years. *Hydrology and Earth System Sciences*, *25*(2), 1009–1032. doi: https://doi.org/10.5194/

hess-25-1009-2021

Leppäranta, M. (1993, March). A review of analytical models of sea-ice growth. *Atmosphere-Ocean*, *31*(1), 123–138. Retrieved 2023-05-01, from https://doi.org/10.1080/07055900.1993.9649465 doi: 10.1080/07055900.1993.9649465

Lerman, A., Imboden, D., & Gat, J. (1995). *Physics and Chemistry of Lakes.* Springer-Verlag Berlin Heidelberg.

Livingstone, D. M., & Imboden, D. M. (1989, December). Annual heat balance and equilibrium temperature of Lake Aegeri, Switzerland. *Aquatic Sciences*, *51*(4), 351–369. Retrieved 2023-05-01, from https://doi.org/10.1007/BF00877177 doi: 10.1007/BF00877177

Magee, M. R., Wu, C. H., Robertson, D. M., Lathrop, R. C., & Hamilton, D. P. (2016, May). Trends and abrupt changes in 104 years of ice cover and water temperature in a dimictic lake in response to air temperature, wind speed, and water clarity drivers. *Hydrology and Earth System Sciences*, *20*(5), 1681–1702. Retrieved 2019-07-05, from https://www.hydrol-earth-syst-sci.net/20/1681/2016/ doi: 10.5194/hess-20-1681-2016

Magnuson, J. J., Crowder, L. B., & Medvick, P. A. (1979, 08). Temperature as an Ecological Resource. *American Zoologist*, *19*(1), 331-343. Retrieved from https://doi.org/10.1093/icb/19.1.331 doi: 10.1093/icb/19.1.331

Magnuson, J.J. and Carpenter, S.R. and Stanley, E.H. (2023a). *North Temperate Lakes LTER: High Frequency Water Temperature Data - Lake Mendota Buoy 2006.* Retrieved 2023-08-07, from https://doi.org/10.6073/pasta/2d6db053cfe03be2ddd3fbc0d86a6fb3. (Publication Title: Environmental Data Initiative) doi: https://doi.org/10.6073/pasta/2d6db053cfe03be2ddd3fbc0d86a6fb3.

Magnuson, J.J. and Carpenter, S.R. and Stanley, E.H. (2023b). *North Temperate Lakes LTER: Physical Limnology of Primary Study Lakes 1981 - current.* Retrieved 2023-08-07, from https://doi.org/10.6073/pasta/be287e7772951024ec98d73fa94eec08. (Publication Title: Environmental Data Initiative) doi: https://doi.org/10.6073/pasta/be287e7772951024ec98d73fa94eec08.

Magnuson, J.J. and Carpenter, S.R. and Stanley, E.H. (2023c). *North Temperate Lakes LTER: Secchi Disk Depth; Other Auxiliary Base Crew Sample Data 1981 - current.* Retrieved 2023-08-07, from https://doi.org/10.6073/pasta/4c5b055143e8b7a5de695f4514e18142. (Publication Title: Environmental Data Initiative) doi: https://doi.org/10.6073/pasta/4c5b055143e8b7a5de695f4514e18142.

McDonald, C. P., & Lathrop, R. C. (2017, April). Seasonal shifts in the relative importance of local versus upstream sources of phosphorus to individual lakes in a chain. *Aquatic Sciences*, *79*(2), 385–394. Retrieved 2020-01-14, from http://link.springer.com/10.1007/s00027-016-0504-1 doi: 10.1007/s00027-016-0504-1

Mitchell, K. E. (2004). The multi-institution North American Land Data Assimilation System (NLDAS): Utilizing multiple GCIP products and partners in a continental distributed hydrological modeling system. *Journal of Geophysical Research*, *109*(D7), D07S90. Retrieved 2019-12-03, from http://doi.wiley.com/10.1029/2003JD003823 doi: 10.1029/2003JD003823

Mooij, W. M., Trolle, D., Jeppesen, E., Arhonditsis, G., Belolipetsky, P. V., Chitamwebwa, D. B. R., ... Janse, J. H. (2010, September). Challenges and opportunities for integrating lake ecosystem modelling approaches. *Aquatic Ecology*, *44*(3), 633–667. Retrieved 2020-08-31, from https://doi.org/10.1007/s10452-010-9339-3 doi: 10.1007/s10452-010-9339-3

Moore, T. N., Mesman, J. P., Ladwig, R., Feldbauer, J., Olsson, F., Pilla, R. M., ... Read, J. S. (2021, May). LakeEnsemblR: An R package that facilitates ensemble modelling of lakes. *Environmental Modelling & Software*, 105101. Retrieved 2021-06-08, from https://linkinghub.elsevier.com/retrieve/pii/S1364815221001444 doi: 10.1016/j.envsoft.2021.105101

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Perga, M., Minaudo, C., Doda, T., Arthaud, F., Beria, H., Chmiel, H. E., ... Bouffard, D. (2023, March). Near-bed stratification controls bottom hypoxia in ice-covered alpine lakes. *Limnology and Oceanography*, lno.12341. Retrieved 2023-05-01, from https://aslopubs.onlinelibrary.wiley.com/doi/10.1002/lno.12341 doi: 10.1002/lno.12341

Piccolroaz, S., & Toffolon, M. (2013, December). Deep water renewal in Lake Baikal: A model for long-term analyses: Deep Water Renewal in Lake Baikal. *Journal of Geophysical Research: Oceans*, *118*(12), 6717–6733. Retrieved 2019-07-23, from http://doi.wiley.com/10.1002/2013JC009029 doi: 10.1002/2013JC009029

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (2007). *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press.

R Core Team. (2023). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Read, J. S., Hamilton, D. P., Jones, I. D., Muraoka, K., Winslow, L. A., Kroiss, R., ... Gaiser, E. (2011, November). Derivation of lake mixing and stratification indices from high-resolution lake buoy data. *Environmental Modelling & Software*, *26*(11), 1325–1336. Retrieved 2019-07-16, from http://www.sciencedirect.com/science/article/pii/S136481521100123X doi: 10.1016/j.envsoft.2011.05.006

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., ... Kumar, V. (2019). Process-Guided Deep Learning Predictions of Lake Water Temperature. *Water Resources Research*, *55*(11), 9173–9190. Retrieved 2020-11-18, from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR024922 doi: https://doi.org/10.1029/2019WR024922

Saloranta, T. M., & Andersen, T. (2007, September). MyLake—A multi-year lake simulation model code suitable for uncertainty and sensitivity analysis simulations. *Ecological Modelling*, *207*(1), 45–60. Retrieved 2020-09-02, from http://www.sciencedirect.com/science/article/pii/S0304380007001548 doi: 10.1016/j.ecolmodel.2007.03.018

Schmidt, W. (1928). Über die Temperatur- und Stabilitätsverhältnisse von Seen. *Geografiska Annaler*, *10*, 145. Retrieved 2019-11-11, from https://www.jstor.org/stable/519789?origin=crossref doi: 10.2307/519789

Snortheim, C. A., Hanson, P. C., McMahon, K. D., Read, J. S., Carey, C. C., & Dugan, H. A. (2017, January). Meteorological drivers of hypolimnetic anoxia in a eutrophic, north temperate lake. *Ecological Modelling*, *343*, 39–53. Retrieved 2019-07-15, from http://www.sciencedirect.com/science/article/pii/S030438001630583X doi: 10.1016/j.ecolmodel.2016.10.014

Stepanenko, V., Mammarella, I., Ojala, A., Miettinen, H., Lykosov, V., & Vesala, T. (2016, May). LAKE 2.0: a model for temperature, methane, carbon dioxide and oxygen dynamics in lakes. *Geoscientific Model Development*, *9*(5), 1977–2006. Retrieved 2022-03-15, from https://gmd.copernicus.org/articles/9/1977/2016/ doi: 10.5194/gmd-9-1977-2016

Verburg, P., & Antenucci, J. P. (2010, June). Persistent unstable atmospheric boundary layer enhances sensible and latent heat loss in a tropical great lake: Lake Tanganyika. *Journal of Geophysical Research*, *115*(D11), D11109. Retrieved 2023-05-01, from http://doi.wiley.com/10.1029/2009JD012839 doi: 10.1029/2009JD012839

Willard, J. D., Read, J. S., Appling, A. P., Oliver, S. K., Jia, X., & Kumar, V. (2021, July). Predicting Water Temperature Dynamics of Unmonitored Lakes With Meta-Transfer Learning. *Water Resources Research*, *57*(7). Retrieved 2021-07-19, from https://onlinelibrary.wiley.com/doi/10.1029/2021WR029579 doi: 10.1029/2021WR029579

Woolway, R. I., Sharma, S., Weyhenmeyer, G. A., Debolskiy, A., Golub, M., Mercado-Bettín, D., … Jennings, E. (2021, December). Phenological shifts in lake stratification under climate change. *Nature Communications*, *12*(1), 2318. Retrieved 2021-04-19, from http://www.nature.com/articles/s41467-021-22657-4 doi: 10.1038/s41467-021-22657-4