KNOWLEDGE-GUIDED ASSIMILATION: BRIDGING THE GAP BETWEEN SENSING AND MODELING WITH INDIRECT LABELS FOR GLOBAL CARBON MONITORING

Anonymous authors
Paper under double-blind review

ABSTRACT

Advanced air-borne and in-situ sensing platforms have generated invaluable observations of the Earth systems and offer exciting opportunities in enhancing the monitoring and forecasting capabilities to tackle challenges such as global warming. While process-based models have been developed for decades, they have limited ability to incorporate real-world observations to further enhance the prediction ability, especially to correct simplified sub-processes that tend to cause deviations from the observations. In particular, many process-based models rely on itemized lower-level processes, whereas the sensors very often can only collect aggregated mixed-up information, constraining the use of these observations to improve the modeling. Existing works on knowledge-guided learning mainly focus on connecting process-based and data-driven methods via directly matched variables, using physical rules and simulations to constrain the training process. We propose a knowledge-guided assimilation approach to integrate process-based and learning models to improve the utilization of large-scale simulations with aggregated indirect observations. To evaluate approach, we carry out a global-scale case study with ecosystem models that are widely used in carbon monitoring. The results on global-scale benchmark data show that knowledge-guided integration of indirect labels can significantly enhance prediction skills compared to existing learning methods.

1 INTRODUCTION

Advanced air-borne systems such as remote sensing satellites and in-situ platforms such as networks of monitoring stations have generated invaluable observations of the Earth systems and offer exciting opportunities in enhancing the monitoring capabilities. Such abilities are essential to improve the solutions for tackling grand challenges such as carbon neutralization, global warming, extreme events, etc. In carbon monitoring, for example, Earth monitoring satellites provide global coverage of forest ecosystems, providing critical information such as forest coverage, plant function types (PFTs), and canopy heights. On the other hand, in-situ sites consisting networks of carbon flux towers offer years of highly dynamic observations of key carbon variables. The increasing availability of sensing-based observations has also led to major advances of process-based models to better integrate these valuable information into the modeling to enhance the estimation. In forest ecosystems, the Ecosystem Demography (ED) model is a new generation of models that has the ability to incorporate many of the observations (e.g., billions of height measurements from NASA GEDI) at the global scale, and is serving important roles in the Global Carbon Budget (Friedlingstein et al., 2023; 2024), NASA Carbon Monitoring System (Hurtt et al., 2019), etc.

Despite the promising potential, there are several challenges integrating sensing and process-based modeling. First, most of the models follow rule-based processes. While the knowledge structure does not require extensive training, it also significantly constrains the models' ability to refine the fixed structure (e.g., structures with simplifying assumptions) using rich real observations. Second, many process-based models rely on itemized lower-level processes, whereas the sensors very often can only collect aggregated mixed-up information. As a concrete example, in the ED model, the overall ecosystem process is modeled as multiple sub-streams of processes corresponding to different forest ages, where each experiences different competition between different PFTs such as deciduous trees, evergreens, or grass/shrubs. Due to the complexity of the cross-PFT competition, PFT proportions generated by ED may deviate from the true values, propagating to errors in its estimates of carbon variables. In such cases, it will be ideal if the PFT observations from remote sensing satellites can be leveraged to correct the values dynamically. However, the satellite-based observations are aggregated results from all the sub-stream processes (e.g., often tens or hundreds of them depending on the granularity), making the observations not compatible with the lower-level modeling. Third, the inputs and outputs from the process-based models at large scale (e.g., global-scale) often come with large volume. As a result, domain scientists normally do not further save intermediate results from different sub-streams and it is highly expensive to regenerate the results, making it harder to learn for data-driven models. Finally, the in-situ observations are often constrained by their temporal coverage compared to the multi-decade scope of Earth system modeling, due to sensor installation time, maintenance, etc.

Existing works on knowledge-guided machine learning (ML) mainly focus on connecting process-based and data-driven methods via directly matched variables, using physical rules and simulations to constrain the training process (Willard et al., 2022). Earlier approaches were typically trained to estimate the residual between observations and the outputs of physics-based simulations (Forssell & Lindskog, 1997; Xu & Valocchi, 2015; Wan et al., 2018), which was later extended into hybrid approaches that integrate results from both physics-based and ML models (Karpatne et al., 2017; Yao et al., 2018; Paolucci et al., 2018). However, in these cases the process-based and ML models still operate separately and do not exploit complementary benefits. More recently, studies have highlighted the promise of using physical knowledge to inform and guide the training of ML models. Such efforts include the design of loss functions that enforce compliance with known physical laws (Li et al., 2024; Jia et al., 2020; Fioretto et al., 2020; Karpatne et al., 2017; Read et al., 2019; Stewart & Ermon, 2017; Yu et al., 2024a; Raissi et al., 2019), strategies for initializing models through knowledge transferred from simulations (Jia et al., 2020; Read et al., 2019; Hurtado et al., 2018; Sultan et al., 2018; McCabe et al., 2023), and development of model architectures that explicitly encode physical symmetries (Satorras et al., 2021; Batzner et al., 2022) and general physical relationships such as mass conservation (Shen et al., 2023; Daw et al., 2019; Muralidhar et al., 2020; Ling et al., 2016; Zhang et al., 2018; Schütt et al., 2017; Hettige et al., 2024). These models demonstrated that ML models can acquire more generalizable abilities with limited observations by using knowledge from process-based models. However, these models focus on training with matched variables and do not consider or address the utilization of satellite-based indirect labels. The models with knowledge-guided architecture also rely on intermediate outputs from smaller and less-expensive process-based models that are often unavailable in large scale problems due to the excessive storage cost as well as the expensive computation to re-run the middle outputs.

We propose a knowledge-guided assimilation learning framework to integrate process-based and learning models with the utilization of higher-level indirect observations from sensing platforms at large scale. Our contributions are:

- We propose a knowledge-guided assimilation framework with a learned decomposition-and-resembling (DERE) process: (1) Knowledge-aligned decomposition of end-to-end simulation data to represent intermediate, sub-stream processes of a process-based model, as a preparation for integration of indirect higher-level labels. This is necessary as there is often no intermediate modeling results saved for large-scale process-based models, which are costly both in space and computation. (2) Knowledge-aligned resembling of decomposed intermediate, sub-stream processes to enable supervision from indirect higher-level labels available at large-scale to constrain the sub-stream process. This data-driven sub-stream process calibration can significantly enhance model generalizability using limited in-situ flux tower observations of final carbon variables.

- We propose a probabilistic label expansion module to increase the temporal coverage of in-situ observations for finetuning, with explicitly learned uncertainty-awareness to leverage the generated probabilistic labels.

- We carry out large-scale experiments with multiple carbon monitoring network datasets using the most recent ICLR CarbonSense benchmark data (Fortier et al., 2025). Extensive comparisons with time-series models and their knowledge-guided extensions demonstrated the effectiveness of our DERE-based knowledge-guided assimilation framework.

2 Problem Definition

2.1 General formulation

Our problem is formulated with the following inputs and outputs:

Inputs: (1) Physical conditions $\boldsymbol{x}_{s,t}$ that are needed to infer target output variables at each location s in a spatial domain $\mathcal{S}$ (e.g., global) along each time step t in a time-series $\mathcal{T} = \{1, ..., T\}$. (2) Initial conditions/states $\boldsymbol{c}_k$ (from a set of possible conditions $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_K\}$) and their weights α_k at the beginning of $\mathcal{T}$.

Outputs: Predicted target variables $\hat{\boldsymbol{y}}_{s,t}$ for the same set of locations and time-series. The ground truth $\boldsymbol{y}_{s,t}$ from in-situ data are available at a limited number of locations $\mathcal{S}' \subset \mathcal{S}$ for a subset of temporal periods $\mathcal{T}' \subset \mathcal{T}$.

In addition, there are the following auxiliary information related to process-based modeling:

- **Estimates of target output variables**, $\boldsymbol{y}^P_{s,t}$, from a process-based model $\mathcal{M}^P$ based on the physical conditions $\boldsymbol{x}_{s,t}$ in $\mathcal{S}$ and $\mathcal{T}$. In the problem setting, $\boldsymbol{y}^P_{s,t}$ represents end-results out of the process-based model and does not contain intermediate results due to the excessive cost of space and computation in large-scale applications. In other words, domain scientists often do not save the intermediate results due to the storage cost and it is also too expensive to re-run the model to generate them.
- **Indirect higher-level observations**, $\boldsymbol{z}_{s,t}$ (e.g., from satellites), on outputs of the intermediate, sub-stream processes as defined in Def. 1. These observations are indirect and cannot be used in the process-based model because they are mixed at the aggregated level and are not compatible with the sub-stream processes.

Definition 1 (Intermediate, sub-stream processes) *Denote $\mathcal{M}^P$ as the entire set of functions from a process-based model, with $(\boldsymbol{y}^P_{s,t})_k = \mathcal{M}^P(\boldsymbol{x}_{s,t}, \boldsymbol{c}_k)$. As a clarification, when we use subscript k on $\boldsymbol{y}^P_{s,t}$ (i.e., $(\boldsymbol{y}^P_{s,t})_k$) it means the result simulated for the initial condition $\mathbf{c}_k$; otherwise, it means final result aggregated over different initial conditions using α_k (observation $\mathbf{y}_{s,t}$ is always aggregated). In process-based modeling, $\mathcal{M}^P$ often consists of a set of intermediate and sub-stream processes. An **intermediate process** generates intermediate results from part of the physical system, and the results are fed into other parts to complete the simulation. For example, we can have $\mathcal{M}^P(\cdot) = \mathcal{M}^P_1(\mathcal{M}^P_2(\cdot))$, where $\mathcal{M}^P_1$ and $\mathcal{M}^P_1$ are intermediate processes. In addition, an intermediate process can further contain **sub-stream processes**, which run in parallel and aggregate into the complete intermediate process. For example, we can have $\mathcal{M}^P_1(\mathcal{M}^P_2(\boldsymbol{x}_{s,t})) = \mathcal{M}^P_1\big(\mathcal{M}^P_2(\boldsymbol{x}_{s,t}, \boldsymbol{c}_1), \mathcal{M}^P_2(\boldsymbol{x}_{s,t}, \boldsymbol{c}_2), \mathcal{M}^P_2(\boldsymbol{x}_{s,t}, \boldsymbol{c}_3), ...\big)$. Sub-stream processes often have the same function form, take the same set of input variables, and generate the same set of output variables. The difference between sub-streams is the initial condition or model state $\mathbf{c}_k$. For example, a forest often contains cohorts with different initial ages where each age corresponds to a different state for the simulation.*

2.2 Real-world example: Global carbon monitoring

Here we provide a concrete and important real-world example in global forest carbon monitoring to better illustrate the problem. In carbon monitoring, the input physical conditions $\boldsymbol{x}_{s,t}$ include meteorological variables (e.g., temperature, precipitation), soil properties and more, where each time step may correspond to a month or shorter for monitoring over multiple decades at the global scale. The initial condition $\boldsymbol{c}_k$ corresponds to the initial age of trees at the beginning of the process. The output carbon variables $\boldsymbol{y}_{s,t}$ include gross primary production (GPP), Net Ecosystem Exchange (NEE), Ecosystem Respiration (RECO), etc.

The auxiliary information from process-based modeling include: (1) Carbon variables $\boldsymbol{y}^P_{s,t}$ estimated from the ED model (Hurtt et al., 1998; Moorcroft et al., 2001; Ma et al., 2022) on variables such as GPP, NEE and RECO, which do not contain intermediate results due to the excessive storage and re-computation cost. (2) Indirect observations $\boldsymbol{z}_{s,t}$ from remote sensing satellites, which provide observations of forest PFTs (e.g., deciduous, evergreen, shrubs) at large geographic scale that are important for cross-PFT competition modeling. However, in ecological modeling, such natural competitions need to be modeled as multiple sub-streams of different forest processes (i.e., based on different initial ages $\boldsymbol{c}_k$), whereas the sensing-based PFTs cannot distinguish between these sub-processes. As a result, existing efforts have not been able to leverage these indirect observations to enhance the prediction quality. We will also use this example later on to help illustrate components of the method section.

3 Related Work

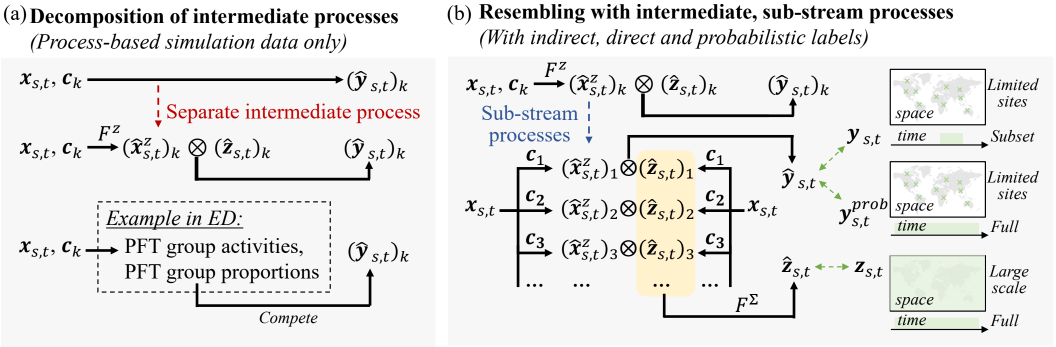
Time-series forecasting. Deep learning models for time-series forecasting are natural data-driven frameworks to model the input-output relationships in our problem setting. Transformer-based architectures have become widely adopted for tasks with long sequences thanks to their ability to capture long-range dependencies (Vaswani et al., 2017; Devlin et al., 2018; Dosovitskiy et al., 2021) compared to earlier models based on recurrent structures (Wang et al., 2023; Chen et al., 2023a; Xu et al., 2024; Lai et al., 2018). Numerous adaptations of transformers have been introduced for further improvements, including ProbSparse self-attention in Informer (Zhou et al., 2021), frequency-domain attention in FEDFormer (Zhou et al., 2022), cross-feature as well as cross-time dependency modeling in Crossformer (Zhang & Yan, 2023), exogenous feature integration in TimeXer (Wang et al., 2024a), time-variable inversion in iTransformer (Liu et al., 2023), and multivariate SimpleTM (Chen et al., 2025). While these models have shown promising performances in general forecasting tasks, they are by design data-driven methods that do not consider physical guidance from process-based models, limiting their performance when only limited in-situ observations are available for large-scale applications.

Data-driven emulation of process-based model. Recent studies have also explored variants of forecasting models as learning-based emulators to approximate process-based models. For example, these models have been developed to emulate climate models (Yu et al., 2024b; Rasp et al., 2018; Mooers et al., 2021; Wang et al., 2022) and weather forecasting models (Lam et al., 2023; Kurth et al., 2023; Bonev et al., 2023) to improve the scalability for higher resolution tasks. In addition, deep learning surrogates have been widely explored for process-based simulations involving the solution of partial differential equations (Obiols-Sales et al., 2020; Sirignano et al., 2020; Karniadakis et al., 2021). However, these emulators mainly aim to enhance the computational efficiency of process-based models instead of combining physical knowledge with real observations to further improve the prediction quality.

Knowledge-guided machine learning. There has been growing efforts on incorporating physics into ML models to enhance both predictive performance and generalizability for solving scientific problems (Willard et al., 2022). Early studies mainly consider residual modeling, where simple regression models (Forssell & Lindskog, 1997; Xu & Valocchi, 2015) or recurrent networks (Wan et al., 2018) are used to infer differences between process-based models and ground truth. However, these methods are unable to enforce physics-based constraints and can only make use of the simulation results when corresponding observations are simultaneously available. More recent strategies start focusing on deeper integration of knowledge, and common strategies include modifying layer architectures based on process-based models (Anderson et al., 2019; Muralidhar et al., 2018; Feng et al., 2022), pretraining with simulation data (Read et al., 2019; Ham et al., 2019; Sultan et al., 2018; Hurtado et al., 2018; Yu et al., 2025), and adding physics-constrained loss functions (Jia et al., 2020; Read et al., 2019; Li et al., 2024; Yu et al., 2024a; Raissi et al., 2019). Variants have also been developed to learn from multiple process-based models (Chen et al., 2023b; Jia et al., 2021), performing model selection (Chen et al., 2023a), integrating simulation data to reduce bias (Wang et al., 2024b; He et al., 2023), etc. However, these models focus on training with matched variables and do not consider indirect labels that are not directly usable with process-based models but are often available at large scale. The models with knowledge-guided architecture also rely on intermediate outputs from smaller and less-expensive process-based models that are often unavailable in large scale problems due to the expensive storage and re-generation cost.

4 Knowledge-Guided Assimilation with Indirect Labels

The knowledge-guided framework has a decomposition-and-resembling structure to allow integration of indirect labels at large scale to provide guidance to intermediate, sub-stream processes. Sec. 4.1 and 4.2 discuss details of the designs, and Sec. 4.3 presents a probabilistic label expansion module to increase the temporal coverage of in-situ labels and uncertainty-aware finetuning. Fig. 1 shows the overall framework.



(a) **Decomposition of intermediate processes**
(Process-based simulation data only)

(b) **Resembling with intermediate, sub-stream processes**
(With indirect, direct and probabilistic labels)

$\boldsymbol{x}_{s,t}, \boldsymbol{c}_k$
$(\hat{\boldsymbol{y}}_{s,t})_k$
Separate intermediate process
F^z
$(\hat{\boldsymbol{x}}^z_{s,t})_k \otimes (\hat{\boldsymbol{z}}_{s,t})_k$
Example in ED:
PFT group activities,
PFT group proportions
Compete
Sub-stream processes
$\boldsymbol{c}_1$
$\boldsymbol{c}_2$
$\boldsymbol{c}_3$
$(\hat{\boldsymbol{x}}^z_{s,t})_1 \otimes (\hat{\boldsymbol{z}}_{s,t})_1$
$(\hat{\boldsymbol{x}}^z_{s,t})_2 \otimes (\hat{\boldsymbol{z}}_{s,t})_2$
$(\hat{\boldsymbol{x}}^z_{s,t})_3 \otimes (\hat{\boldsymbol{z}}_{s,t})_3$
$\boldsymbol{x}_{s,t}$
...
F^Σ
$\boldsymbol{y}_{s,t}$
$\hat{\boldsymbol{y}}_{s,t}$
$\boldsymbol{y}^{prob}_{s,t}$
$\hat{\boldsymbol{z}}_{s,t}$
$\boldsymbol{z}_{s,t}$
space
time
Limited sites
Subset
Full
Large scale

Figure 1: Overview of the decomposition and resembling framework.

4.1 Knowledge-aligned data decomposition

As explained in Def. 1, the entire physical process $\mathcal{M}^P$ contains intermediate, sub-stream processes. These are often not readily available for data-driven models to learn right off the simulation data, as the intermediate results are often not saved due to the excessive size at large scale and high cost of re-generation. Thus, a necessary initial step is to separate out the intermediate process of interest as a preparation to incorporate the indirect labels (Sec. 2.1) in the later resembling step. This is the goal of knowledge-aligned decomposition. For clarity: (1) the decomposition only separates out an intermediate process and does not consider sub-stream processes that will be addressed in the resembling step; and (2) the decomposition step uses only the simulation data $\boldsymbol{y}^P_{s,t}$ from $\mathcal{M}^P$ and does not use observations.

Specifically, As shown in Fig. 1(a), the decomposition step splits out an intermediate process, approximated by a learned function $\mathcal{F}^z(\boldsymbol{x}_{s,t}, \boldsymbol{c}_k)$ to generate $((\hat{\boldsymbol{x}}^z_{s,t})_k, (\hat{\boldsymbol{z}}_{s,t})_k)$ before reaching $(\hat{\boldsymbol{y}}_{s,t})_k$. Most importantly, the key objective is to explicitly represent $(\hat{\boldsymbol{z}}_{s,t})_k$ so later on it can be connected to the indirect labels during resembling. As the decomposition only concerns simulation data, here $\boldsymbol{c}_k$ can be any condition available from the simulation data. The functional relationship between $((\hat{\boldsymbol{x}}^z_{s,t})_k, (\hat{\boldsymbol{z}}_{s,t})_k)$ and $(\hat{\boldsymbol{y}}_{s,t})_k$, denoted by "$\otimes$", can be defined either using a pre-defined, differentiable function based on the physical process (when such functions are direct and simple) or using another learned function, or a combination of the two. Using ecosystem carbon monitoring as an example (Fig. 1(a), bottom), $(\hat{\boldsymbol{x}}^z_{s,t})_k$ represents outputs of carbon variables from groups of trees with different PFTs (e.g., deciduous) and $(\hat{\boldsymbol{z}}_{s,t})_k$ is a vector containing the proportions of different PFTs. They can then be combined into final carbon variables by linear combination (i.e., a predefined weighted sum based on proportions) plus non-linear competition (a learned function). All the learned functions will be pre-trained using the simulation data, which will be finetuned later with stronger constraints.

4.2 Knowledge-aligned resembling of intermediate, sub-stream processes

The resembling process will build on the separated intermediate process and integrate it with different types of real observations. As shown in Fig. 1(b), the resembling process will first explicitly integrate all sub-stream processes, governed by different initial conditions/states $\boldsymbol{c}_k$ (each location or spatial unit can have multiple different states such as different tree ages). Here the function $\mathcal{F}^z(\boldsymbol{x}_{s,t}, \boldsymbol{c}_k)$ is responsible for generating all pairs of $((\hat{\boldsymbol{x}}^z_{s,t})_k, (\hat{\boldsymbol{z}}_{s,t})_k)$ corresponding to different initial conditions, and these sub-stream results are then combined into the final prediction $\hat{\boldsymbol{y}}_{s,t}$ (no longer subscripted by k as all sub-streams have been combined), either by prefixed physical relationships or another learned function.

The most important part of the resembling is the integration of direct labels $\boldsymbol{y}_{s,t}$ (e.g., observations from in-situ flux towers on carbon variables) and indirect labels $\boldsymbol{z}_{s,t}$ (e.g., mixed PFTs across ages from satellite observations). In particular, the key is to enable the use of indirect labels $\boldsymbol{z}_{s,t}$ that are often available at much larger scales compared to direct labels $\boldsymbol{y}_{s,t}$ as shown by the illustrative maps in Fig. 1(b); for example, satellite-based $\boldsymbol{z}_{s,t}$ tends to have global coverage. With the decomposition of intermediate processes and the resembling of sub-stream processes, the indirect labels $\boldsymbol{z}_{s,t}$ can be compared with the integrated $\mathcal{F}^{\Sigma}(\{(\hat{\boldsymbol{z}}_{s,t})_k\}_{k=1...K})$ from all the sub-stream processes. Here it will be best if $\mathcal{F}^{\Sigma}$ can be determined by prefixed aggregation functions when applicable, and can be learned when necessary. In the carbon monitoring example, $\mathcal{F}^{\Sigma}$ is prefixed by a simple sum, $\sum_i^K((\hat{\boldsymbol{z}}_{s,t})_k \cdot \alpha_k)$, where α_k is the weight of an initial condition that is used to aggregate the PFT proportions across the conditions.

Furthermore, the final predictions $\hat{\boldsymbol{y}}_{s,t}$ are constrained by the direct labels $\boldsymbol{y}_{s,t}$ as a regular part of a training process, which are often available at a more limited number of locations (e.g., carbon flux towers that are expensive to build). We also enhance it with a probabilistic label tuning method in the next section. During training, each batch contains samples intended both for the indirect label comparison and direct label comparison.

4.3 Probabilistic label expansion and uncertainty-aware finetuning

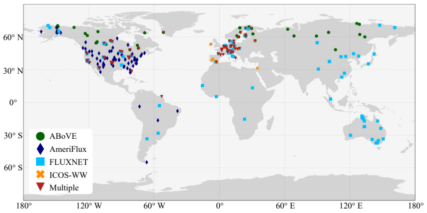


Figure 2: Distribution of in-situ sites.

As direct labels $\boldsymbol{y}_{s,t}$ tend to be limited in the temporal domain in many Earth monitoring tasks (e.g., carbon flux towers at many sites are recent and only provides a few years of coverage), we further develop a probabilistic label expansion strategy and uncertainty-aware tuning to enrich the usable labels. For example, Fig. 2 shows several examples of observations at carbon flux tower sites, where the measurements are only available for a subset of years. To address this, we propose a simpler prediction task to expand the labels, where the goal is to predict the missing labels at each site leveraging existing labels as additional inputs. Comparing to the original task that aims to predict $\boldsymbol{y}_{s,t}$ using $\boldsymbol{x}_{s,t}$ and $\boldsymbol{c}_k$, this task has significantly reduced difficulty as it only aims to make predictions at sites where a set of labels is already known in a time window, and those labels are given as part of the inputs. This makes it feasible to leverage these predictions to facilitate the model tuning in our original task. Furthermore, we make the predictions probabilistic so the finetuning step can explicitly utilize the uncertainty to determine whether or not a prediction should be used. Specifically, we adopt conditional diffusion (Tashiro et al., 2021) to generate the missing measurements, where the existing observations from the same time-series can be added as conditions, and the variance can be obtained. During training, we set a subset of observations as part of the given conditions while using the remaining observations for loss evaluations. Denote $\boldsymbol{y}_{s,t}^{m}$ as the masked observations for prediction, $\boldsymbol{y}_{s,t}^{m'}$ as those given as conditions, and j as the position in the denoising sequence, we have $p_\theta((\boldsymbol{y}_{s,t}^{m})_{j-1} \mid (\boldsymbol{y}_{s,t}^{m})_j, \boldsymbol{y}_{s,t}^{m'}) = \mathcal{N}\Big((\boldsymbol{y}_{s,t}^{m})_{j-1}; \mu_\theta\big((\boldsymbol{y}_{s,t}^{m})_j, \boldsymbol{y}_{s,t}^{m'}\big), \sigma_j^2 I\Big)$. Once trained, we generate 100 samples per site to estimate the variance and confidence interval.

Given the predictions with variance, we include an uncertainty-aware tuning module as part of the resembling process, where a learned sub-network is used to adaptively determine the weight of each predicted label based on its value and variance. Overall, our DERE model is trained with direct labels, probabilistic labels, and indirect labels.

5 Experiments

5.1 Data

We conducted extensive experiments at the global scale using the collection of datasets from the most recent ICLR CarbonSense benchmark data (Fortier et al., 2025) developed for the important carbon monitoring problem. Specifically, CarbonSense includes various datasets representing different in-situ carbon flux observation networks under different conditions, including AmeriFlux, FLUXNET, the ICOS-2023, ICOS-WW, and a mixed set. Through data inspection, we found that ICOS-2023 only have a few sites with one year of data. This leaves very few data points for testing that can cause highly instable results. Thus, we replaced it (ICOS-WW still contains data from the network) with the recent ABoVE dataset covering the broad Arctic region (Bill et al., 2023). Fig. 2 shows the geographic distribution of the sites from different datasets. We used 3 key variables GPP, RECO, and NEE for the evaluation, which are available across all the datasets. For model training

Table 1: GPP result comparison (top results in bold; runner-ups with underlines).

	Methods	ABoVE		AmeriFlux		FLUXNET		ICOS-WW		Multiple		Top-2
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	count
Physical	ED	0.427	0.703	0.883	1.419	0.822	1.110	0.713	0.968	0.678	1.010	0/10
Baseline	Transformer	0.502	0.684	1.030	1.405	0.850	1.147	0.955	1.235	0.736	0.997	0/10
	Informer	0.523	0.729	1.093	1.465	0.882	1.180	0.859	1.096	0.817	1.065	0/10
	FEDformer	0.451	0.626	0.917	1.347	0.768	1.025	1.154	1.524	0.790	1.080	0/10
	iTransformer	0.561	0.662	1.007	1.480	0.887	1.162	1.009	1.340	0.851	1.149	0/10
	TimeXer	0.685	1.001	1.250	1.939	1.156	1.602	1.451	1.860	1.100	1.547	0/10
	SimpleTM	0.631	0.782	1.057	1.549	0.972	1.251	1.132	1.434	1.007	1.346	0/10
KGML	Transformer	0.352	0.541	0.651	1.033	0.699	0.951	0.711	1.014	0.636	0.919	2/10
	Informer	0.373	0.555	**0.604**	**0.947**	0.756	1.011	0.839	1.141	0.618	**0.888**	**4/10**
	FEDformer	0.369	0.531	0.769	1.115	0.726	0.926	0.830	1.121	0.701	1.003	0/10
	iTransformer	0.363	0.508	0.678	1.027	0.654	0.885	**0.641**	0.919	0.621	0.909	3/10
	TimeXer	0.375	0.566	0.657	1.036	0.649	0.879	0.791	1.050	0.678	0.990	1/10
	SimpleTM	0.406	0.587	0.737	1.102	0.669	**0.855**	0.651	0.916	0.680	0.976	3/10
Proposed	DERE	**0.302**	**0.485**	0.663	1.068	**0.598**	0.863	0.682	**0.888**	**0.585**	0.890	**7/10**

Table 2: RECO result comparison (top results in bold; runner-ups with underlines).

	Methods	ABoVE		AmeriFlux		FLUXNET		ICOS-WW		Multiple		Top-2
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	count
Physical	ED	0.431	0.657	0.603	0.855	0.603	0.833	**0.464**	**0.620**	0.457	0.649	**3/10**
Baseline	Transformer	0.602	0.857	0.804	1.040	0.559	0.743	0.698	0.886	0.488	0.709	0/10
	Informer	0.498	0.719	0.720	0.941	0.546	0.749	0.697	0.927	0.530	0.740	0/10
	FEDformer	0.431	0.611	0.616	0.867	0.482	0.638	0.832	1.054	0.574	0.804	0/10
	iTransformer	0.412	0.526	0.543	0.774	0.510	0.662	0.679	0.969	0.535	0.752	0/10
	TimeXer	0.607	0.854	0.946	1.279	0.713	0.960	1.386	1.770	0.784	1.031	0/10
	SimpleTM	0.503	0.660	0.620	0.865	0.561	0.709	0.862	1.126	0.626	0.859	0/10
KGML	Transformer	0.328	0.467	0.526	0.705	0.438	0.583	0.586	0.756	0.466	0.655	**3/10**
	Informer	0.354	0.495	0.448	**0.614**	0.475	0.626	0.559	0.716	0.463	0.638	**3/10**
	FEDformer	**0.319**	**0.443**	0.533	0.736	0.506	0.645	0.582	0.781	0.541	0.727	2/10
	iTransformer	0.348	0.492	**0.439**	0.650	0.445	0.585	0.595	0.796	0.537	0.733	2/10
	TimeXer	0.388	0.543	0.591	0.769	0.512	0.640	0.739	0.914	0.555	0.744	0/10
	SimpleTM	0.385	0.555	0.492	0.694	0.487	0.643	0.519	0.707	0.545	0.733	0/10
Proposed	DERE	0.320	0.472	0.465	0.671	**0.393**	**0.557**	0.502	0.639	**0.442**	**0.627**	**7/10**

and testing, we spatially split each dataset with 80% for training and 20% for testing. The spatial split ensures that there is no site-overlap between training and test sets.

For data on the process-based model side, we used the simulations from the ED model (Hurtt et al., 1998; Moorcroft et al., 2001; Ma et al., 2022), which has global coverage and the simulation results fully covered the temporal range of the in-situ observations. For the indirect labels, we used the satellite-derived PFT information from the ESA CCI PFT dataset (Harper et al., 2023), which also has global coverage for the temporal range.

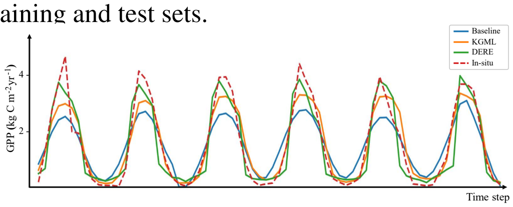


Figure 3: Comparison of GPP over time steps. Both the baseline and the KGML model here are from the Transformer.

Table 3: NEE result comparison (top results in bold; runner-ups with underlines).

	Methods	ABoVE		AmeriFlux		FLUXNET		ICOS-WW		Multiple		Top-2
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	count
Physical	ED	0.276	0.386	0.599	0.920	0.613	0.889	0.530	0.691	0.537	0.770	0/10
Baseline	Transformer	0.413	0.500	0.615	0.875	0.575	0.790	0.489	0.665	0.532	0.726	0/10
	Informer	0.217	0.287	0.626	0.860	0.561	0.787	**0.466**	**0.589**	0.554	0.741	2/10
	FEDformer	0.201	0.266	0.546	0.805	0.548	0.769	0.535	0.680	0.529	0.735	0/10
	iTransformer	0.365	0.427	0.711	1.026	0.697	0.940	0.585	0.762	0.625	0.855	0/10
	TimeXer	1.288	2.209	0.714	1.129	0.770	1.126	1.340	1.644	0.843	1.474	0/10
	SimpleTM	0.334	0.415	0.692	0.997	0.664	0.912	0.640	0.827	0.655	0.894	0/10
KGML	Transformer	0.177	0.244	0.507	0.733	0.504	0.722	0.511	0.721	0.489	**0.691**	**4/10**
	Informer	0.186	0.252	**0.456**	**0.700**	0.509	0.721	0.519	0.730	0.501	0.699	3/10
	FEDformer	0.197	0.268	0.506	0.741	0.522	0.735	0.475	0.627	0.492	0.698	2/10
	iTransformer	0.179	**0.239**	0.540	0.770	0.600	0.834	0.485	0.641	0.534	0.737	1/10
	TimeXer	0.192	0.264	0.529	0.800	0.532	0.760	0.563	0.735	0.505	0.714	0/10
	SimpleTM	0.187	0.255	0.582	0.842	0.588	0.790	0.527	0.709	0.492	0.698	0/10
Proposed	DERE	**0.175**	0.244	0.496	0.713	**0.478**	**0.683**	0.505	0.696	**0.476**	0.692	**8/10**

5.2 Candidate methods

We consider four different categories of candidate methods: (1) The process-based model ED; (2) Learning-based time-series models as listed below; (3) Knowledge-guided extensions of the time-series models (more details later); (4) Our proposed method DERE. For the training, we used the recommended settings from the papers, and trained till convergence via patience checks on validation data (10% of training data; independent from test data).

- **Transformer** (Vaswani et al., 2017): The vanilla Transformer serves as the fundamental sequence modeling framework with self-attention to help capture long-range temporal dependencies.
- **Informer** (Zhou et al., 2021): A Transformer variant that employs sparse attention and generative-style decoding to enhance long-sequence prediction efficiency.
- **FEDFormer** (Zhou et al., 2022): A Transformer variant using frequency-domain attention to strengthen series decomposition and boost forecasting accuracy.
- **iTransformer** (Liu et al., 2023): A Transformer variant that adopts an inverted design, emphasizing feature dimensions over time steps and helping to exploit correlations among input variables.
- **Timexer** (Wang et al., 2024a): A Transformer variant that refines attention by integrating inter-target and input-target relations to better capture temporal dependencies.
- **SimpleTM** (Chen et al., 2025): A lightweight time series forecasting model that streamlines architecture and improves computational efficiency while maintaining strong predictive performance.

For each forecasting method, we consider two variants: a default data-driven version (baseline) and an extension with knowledge-guided machine learning (KGML). The baseline models were trained directly on in-situ observations as target values, whereas the KGML variants include two generally used strategies: (1) They are pretrained on simulation data from ED and then finetuned with in-situ observations; and (2) The training included physics-constrained loss functions based on the carbon mass balance (e.g., NEE $=$ RECO $-$ GPP). Both are used in our DERE model as well. Finally, the DERE model is implemented with the Transformer baseline as the backbone.

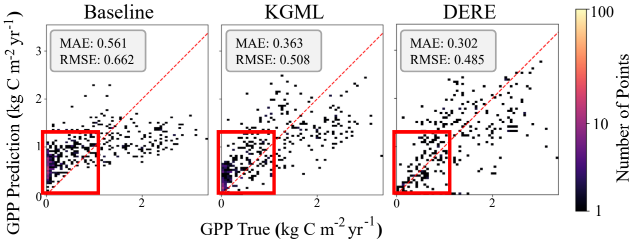


Figure 4: Scatter plots of GPP true / prediction for the ABoVE data. Both the baseline and KGML model here are from the iTransformer.

5.3 Results

Overall Evaluation. Tables 1 to 3 present the overall testing performance using in-situ GPP, RECO, and NEE data, evaluated with MAE and RMSE. We provide the top-2 counts (i.e., number of times a method ranked in top-2 across all columns) for convenience. The proposed DERE method show the best performance compared to the others. The pure data-driven baselines did not outperform the process-based ED model, potentially due to the generalization challenge with limited observations. As a reference, ED is a fairly strong model that has been used in major systems including NASA Carbon Monitoring System. In contrast, KGML-based methods and the proposed approach outperform the baseline, underscoring the importance of incorporating physics knowledge. The enhanced performance of the proposed DERE model can be attributed to its integration of direct, indirect and probabilistic labels in addition to the simulation data. Details of the ablation study will be provided later. This shows the promising potential of integrating indirect labels that are available at large scale.

Visualization. Figure 3 shows the temporal dynamics of GPP using the baseline Transformer, KGML Transformer, the proposed DERE method and in-situ data, as examples to visualize the qualitative performance. The proposed method aligns more closely with in-situ observations than the other models. Both the KGML and proposed methods demonstrate more stable and consistent

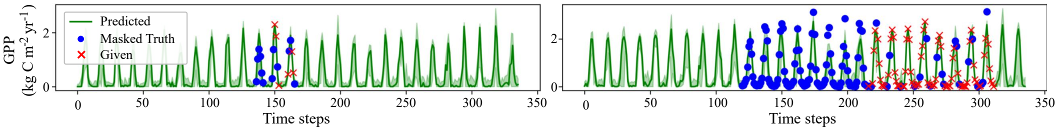


Figure 5: Probabilistic label expansion examples. The green shaded regions represent the 5%-95% confidence interval based on sampled time series.

Table 4: Ablation study (top results in bold; runner-ups with underlines).

	Methods	ABoVE		AmeriFlux		FLUXNET		ICOS-WW		Multiple		Top-2
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	Count
GPP	Baseline	0.502	0.684	1.030	1.405	0.850	1.147	0.955	1.235	0.736	0.997	0/10
	KGML	0.352	0.541	**0.651**	**1.033**	0.699	0.951	0.711	1.014	0.636	0.919	2/10
	KGML + indirect labels	0.321	0.536	0.677	1.084	0.619	**0.850**	0.685	**0.888**	**0.573**	**0.888**	8/10
	DERE	**0.302**	**0.485**	0.663	1.068	**0.598**	0.863	**0.682**	0.888	0.585	0.890	**10/10**
RECO	Baseline	0.602	0.857	0.804	1.040	0.559	0.743	0.698	0.886	0.488	0.709	0/10
	KGML	0.328	**0.467**	0.526	0.705	0.438	0.583	0.586	0.756	0.466	0.655	3/10
	KGML + indirect labels	0.334	0.526	0.526	0.721	0.409	0.567	0.508	0.639	**0.402**	**0.596**	7/10
	DERE	**0.320**	0.472	**0.465**	**0.671**	**0.393**	**0.557**	**0.502**	**0.639**	0.442	0.627	**10/10**
NEE	Baseline	0.413	0.500	0.615	0.875	0.575	0.790	**0.489**	**0.665**	0.532	0.726	2/10
	KGML	0.177	0.244	0.507	0.733	0.504	0.722	0.511	0.721	0.489	**0.691**	2/10
	KGML + indirect labels	0.179	**0.242**	**0.475**	0.723	**0.471**	**0.677**	0.505	0.696	0.482	0.698	7/10
	DERE	**0.175**	0.244	0.496	**0.713**	0.478	0.683	0.505	0.696	**0.476**	0.692	**9/10**

temporal patterns than the baseline model. Figure 4 presents more detailed scatter plots of GPP predictions versus ground-truth observations across Above, FLUXNET, and Multiple networks. The proposed DERE method is able to correct large number of deviations highlighted by the red boxes, leading to improved performance.

Probabilistic Label Expansion. Figure 5 visualizes examples of label expansion on the in-situ data with conditional diffusion, where the effect on final predictions are included in the ablation study. The green line indicates the mean of the sampled predictions. In masked regions, the predicted series closely match the hidden in-situ data, demonstrating accurate and reliable GPP imputation. The green shaded region represents the 5%-95% confidence interval across the imputed time series. The variance from the predictions can be leveraged by the uncertainty-aware tuning module to improve the usability of the probabilistic labels. The patterns remain fairly stable with varying proportions of missing data, potentially benefiting from auxiliary environment inputs.

Ablation Study Table 4 provides the ablation study results. The proposed DERE model achieves the best top-2 count across all variables, with stepwise improvements from the baseline to KGML, then KGML with PFT, and finally DERE. Extending KGML with indirect labels helps significantly reduce the errors. The full DERE with probabilistic labels and uncertainty-aware tuning yielded the best overall performance.

6 Conclusion

We presented a knowledge-guided assimilation framework with a decomposition-and-resembling approach that bridges process-based models and learning models to leverage indirect labels from advanced sensing platforms that are available at large scale. We further developed a probabilistic label expansion module to extend the temporal coverage of sparse limited observations with explicit uncertainty awareness. Extensive experiments on global carbon monitoring datasets, including the ICLR CarbonSense benchmark, demonstrate that our proposed DERE-based knowledge-guided assimilation framework can effectively improve prediction quality compared with existing methods. Future work will explore extensions to further consider more challenging scenarios with anomalous or rare conditions.

References

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Advances in Neural Information Processing Systems*, pp. 14510–14519, 2019.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1): 2453, 2022.

KE Bill, C Dieleman, JL Baltzer, GE Degre-timmons, MC Mack, SG Cumming, XJ Walker, and MR Turetsky. Arctic-boreal vulnerability experiment (above). *ORNL Distributed Active Archive Center (DAAC) dataset 10.3334/ORNLDAAC/2235 (2023*, pp. 2235, 2023.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pp. 2806–2823. PMLR, 2023.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shengyu Chen, Nasrin Kalanat, Simon Topp, Jeffrey Sadler, Yiqun Xie, Zhe Jiang, and Xiaowei Jia. Meta-transfer-learning for time series data with extreme events: An application to water temperature prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 266–275, 2023a.

Shengyu Chen, Yiqun Xie, Xiang Li, Xu Liang, and Xiaowei Jia. Physics-guided meta-learning method in baseflow prediction over large regions. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pp. 217–225. SIAM, 2023b.

A Daw, RQ Thomas, CC Carey, JS Read, AP Appling, and A Karpatne. Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. *arXiv:1911.02682*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Dapeng Feng, Jiangtao Liu, Kathryn Lawson, and Chaopeng Shen. Differentiable, learnable, regionalized process-based models with multiphysical outputs can approach state-of-the-art hydrologic prediction accuracy. *Water Resources Research*, 58(10):e2022WR032404, 2022.

Ferdinando Fioretto, Terrence WK Mak, and Pascal Van Hentenryck. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 630–637, 2020.

Urban Forssell and Peter Lindskog. Combining semi-physical and neural network modeling: An example ofits usefulness. *IFAC Proceedings Volumes*, 30(11):767–770, 1997.

Matthew Fortier, Mats Leon Richter, Oliver Sonnentag, and Christopher Pal. Carbonsense: A multimodal dataset and baseline for carbon flux modelling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Pierre Friedlingstein, Michael O'sullivan, Matthew W Jones, Robbie M Andrew, Dorothee CE Bakker, Judith Hauck, Peter Landschützer, Corinne Le Quéré, Ingrid T Luijkx, Glen P Peters, et al. Global carbon budget 2023. *Earth System Science Data*, 15(12):5301–5369, 2023.

Pierre Friedlingstein, Michael O'sullivan, Matthew W Jones, Robbie M Andrew, Judith Hauck, Peter Landschützer, Corinne Le Quéré, Hongmei Li, Ingrid T Luijkx, Are Olsen, et al. Global carbon budget 2024. *Earth System Science Data Discussions*, 2024:1–133, 2024.

Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572, 2019.

K. L. Harper, C. Lamarche, A. Hartley, et al. A 29-year time series of annual 300 m resolution plant-functional-type maps for climate models. *Earth System Science Data*, 15:1465–1499, 2023. doi: 10.5194/essd-15-1465-2023. URL `https://doi.org/10.5194/essd-15-1465-2023`.

Erhu He, Yiqun Xie, Licheng Liu, Weiye Chen, Zhenong Jin, and Xiaowei Jia. Physics guided neural networks for time-aware fairness: an application in crop yield prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 14223–14231, 2023.

Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. Airphynet: Harnessing physics-guided neural networks for air quality prediction. *arXiv preprint arXiv:2402.03784*, 2024.

David Menéndez Hurtado, Karolis Uziela, and Arne Elofsson. Deep transfer learning in the assessment of the quality of protein models. *arXiv preprint arXiv:1804.06281*, 2018.

G Hurtt, M Zhao, R Sahajpal, A Armstrong, R Birdsey, E Campbell, Katelyn Dolan, R Dubayah, JP Fisk, S Flanagan, et al. Beyond mrv: high-resolution forest carbon modeling for climate mitigation planning over maryland, usa. *Environmental Research Letters*, 14(4):045013, 2019.

George C Hurtt, Paul R Moorcroft, Stephen W Pacala And, and Simon A Levin. Terrestrial models and global change: challenges for the future. *Global Change Biology*, 4(5):581–590, 1998.

Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S Read, Jacob A Zwart, Michael Steinbach, and Vipin Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *arXiv preprint arXiv:2001.11086*, 2020.

Xiaowei Jia, Yiqun Xie, Sheng Li, Shengyu Chen, Jacob Zwart, Jeffrey Sadler, Alison Appling, Samantha Oliver, and Jordan Read. Physics-guided machine learning from simulation data: An application in modeling lake and river systems. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 270–279. IEEE, 2021.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

A Karpatne, W Watkins, J Read, and V Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv:1710.11431*, 2017.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference*, pp. 1–11, 2023.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024.

J Ling, A Kurzawski, and J Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech*, 2016.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Lei Ma, George Hurtt, Lesley Ott, Ritvik Sahajpal, Justin Fisk, Rachel Lamb, Hao Tang, Steve Flanagan, Louise Chini, Abhishek Chatterjee, et al. Global evaluation of the ecosystem demography model (ed v3. 0). *Geoscientific Model Development*, 15(5):1971–1994, 2022.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

Griffin Mooers, Michael Pritchard, Tom Beucler, Jordan Ott, Galen Yacalis, Pierre Baldi, and Pierre Gentine. Assessing the potential of deep learning for emulating cloud superparameterization in climate models with real-geography boundary conditions. *Journal of Advances in Modeling Earth Systems*, 13(5):e2020MS002385, 2021.

Paul R Moorcroft, George C Hurtt, and Stephen W Pacala. A method for scaling vegetation dynamics: the ecosystem demography model (ed). *Ecological monographs*, 71(4):557–586, 2001.

N Muralidhar, MR Islam, M Marwah, A Karpatne, and N Ramakrishnan. Incorporating prior domain knowledge into deep neural networks. In *IEEE Big Data*. IEEE, 2018.

Nikhil Muralidhar, Jie Bu, Ze Cao, Long He, Naren Ramakrishnan, Danesh Tafti, and Anuj Karpatne. Phynet: Physics guided neural networks for particle drag force prediction in assembly. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 559–567. SIAM, 2020.

Octavi Obiols-Sales, Abhinav Vishnu, Nicholas Malaya, and Aparna Chandramowliswharan. Cfdnet: A deep learning-based accelerator for fluid simulations. In *Proceedings of the 34th ACM international conference on supercomputing*, pp. 1–12, 2020.

Roberto Paolucci, Filippo Gatti, Maria Infantino, Chiara Smerzini, Ali Güney Özcebe, and Marco Stupazzini. Broadband ground motions from 3d physics-based numerical simulations using artificial neural networksbroadband ground motions from 3d pbss using anns. *Bulletin of the Seismological Society of America*, 108(3A):1272–1286, 2018.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Stephan Rasp, Michael S Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

Jordan S Read, Xiaowei Jia, Jared Willard, Alison P Appling, Jacob A Zwart, Samantha K Oliver, Anuj Karpatne, Gretchen JA Hansen, Paul C Hanson, William Watkins, et al. Process-guided deep learning predictions of lake water temperature. *Water Resources Research*, 55(11):9173–9190, 2019.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, pp. 991–1001, 2017.

Chaopeng Shen, Alison P Appling, Pierre Gentine, Toshiyuki Bandai, Hoshin Gupta, Alexandre Tartakovsky, Marco Baity-Jesi, Fabrizio Fenicia, Daniel Kifer, Li Li, et al. Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, 4(8):552–567, 2023.

Justin Sirignano, Jonathan F MacArt, and Jonathan B Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, 423: 109811, 2020.

Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Mohammad M Sultan, Hannah K Wayment-Steele, and Vijay S Pande. Transferable neural networks for enhanced sampling of protein dynamics. *Journal of chemical theory and computation*, 14(4): 1887–1894, 2018.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

ZY Wan, P Vlachas, P Koumoutsakos, and T Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PloS one*, 2018.

Xin Wang, Yilun Han, Wei Xue, Guangwen Yang, and Guang J Zhang. Stable climate simulations using a realistic general circulation model with neural network parameterizations for atmospheric moist physics and radiation processes. *Geoscientific Model Development*, 15(9):3923–3940, 2022.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024a.

Zhihao Wang, Yiqun Xie, Xiaowei Jia, Lei Ma, and George Hurtt. High-fidelity deep approximation of ecosystem simulation over long-term at large scale. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pp. 1–10, 2023.

Zhihao Wang, Yiqun Xie, Zhili Li, Xiaowei Jia, Zhe Jiang, Aolin Jia, and Shuo Xu. Simfair: Physics-guided fairness-aware learning with simulation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 22420–22428, 2024b.

Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55(4):1–37, 2022.

Lei Xu, Xihao Zhang, Hongchu Yu, Zeqiang Chen, Wenying Du, and Nengcheng Chen. Incorporating spatial autocorrelation into deformable convlstm for hourly precipitation forecasting. *Computers & Geosciences*, 184:105536, 2024.

Tianfang Xu and Albert J Valocchi. Data-driven methods to improve baseflow prediction of a regional groundwater model. *Computers & Geosciences*, 85:124–136, 2015.

Kun Yao, John E Herr, David W Toth, Ryker Mckintyre, and John Parkhill. The tensormol-0.1 model chemistry: a neural network augmented with long-range physics. *Chemical science*, 9(8): 2261–2269, 2018.

Runlong Yu, Chonghao Qiu, Robert Ladwig, Paul C Hanson, Yiqun Xie, Yanhua Li, and Xiaowei Jia. Adaptive process-guided learning: An application in predicting lake do concentrations. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 580–589. IEEE, 2024a.

Runlong Yu, Chonghao Qiu, Robert Ladwig, Paul Hanson, Yiqun Xie, and Xiaowei Jia. Physics-guided foundation model for scientific discovery: An application to aquatic science. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28548–28556, 2025.

Sungduk Yu, Walter Hannah, Liran Peng, Jerry Lin, Mohamed Aziz Bhouri, Ritwik Gupta, Björn Lütjens, Justus C Will, Gunnar Behrens, Julius Busecke, et al. Climsim: A large multi-scale dataset for hybrid physics-ml climate emulation. *Advances in Neural Information Processing Systems*, 36, 2024b.

Linfeng Zhang, Jiequn Han, Han Wang, Wissam Saidi, Roberto Car, and E Weinan. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. In *Advances in Neural Information Processing Systems*, pp. 4436–4446, 2018.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.