Domain-Adaptive Continual Meta-Learning for Modeling Dynamical Systems: An Application in Environmental Ecosystems

Yiming Sun* Runlong Yu* Runxue Bao* Yiqun Xie† Ye Ye* Xiaowei Jia*

Abstract

Environmental ecosystems exhibit complex and evolving dynamics over time, making the modeling of non-stationary processes critically important. However, traditional methods often rely on static models trained on entire datasets, failing to capture the non-stationary and drastically fluctuating characteristics. Dynamically adjusting models to evolving data is challenging, as they can easily either lag behind new trends or overfit newly received data. To address these challenges, we propose Domain-Adaptive Continual Meta-Learning (DACM) method, aiming to automatically detect distribution shifts and adapt to newly emergent domains. In particular, while DACM continuously explores the sequential temporal data, it also exploits historical data that are similar in distribution to the current observations. By striking a balance between temporal exploration and distributional exploitation, DACM quickly adjusts the model to stay up-to-date with new trends while maintaining generalization ability to data with similar distributions. We demonstrate the effectiveness of DACM on a real-world water temperature prediction dataset, where it outperforms diverse baseline models and shows strong adaptability and predictive performance in non-stationary environments.

1 Introduction

Environmental ecosystems involve complex dynamical processes, such as water, energy, and carbon cycles, that evolve over time. Accurate modeling of these processes is essential for understanding their dynamics and interactions, and informing important management decisions and policies. Recent years have witnessed a paradigm shift from traditional process-based physical models [1–4] to machine learning (ML)-based data driven models [5–7], which have shown promise in capturing complex temporal patterns of key variables in environmental ecosystems.

Although temporal ML models, such as long-short term memory (LSTM), have shown encouraging results in many environmental applications, they remain limited in generalizing to real scenarios with non-stationary patterns. In particular, traditional temporal ML models assume that the conditional probability distribution $p(y \mid x)$, representing the probability of the target variable y given input features x, remains constant over time. This assumption is critical for the generalizability of ML models, as it implies that the patterns learned from the training data are applicable to unseen testing data. However, the processes in real environmental ecosystems are inherently non-stationary over time [8,9]. For example, $p(y|x)$ can be subject to complex and unpredictable changes due to changing weather and environmental conditions. This phenomenon, known as temporal concept drift, can be commonly observed in many scientific applications. For example, crop production is highly affected by advances in seed quality and management practices over years, and energy consumption patterns vary with both short-term weather conditions and long-term climate trends.

Prior work has investigated several approaches to address temporal concept drift. One prominent method is continual learning [10, 11], which enables ML models to incrementally update their knowledge over new learning tasks. The central idea is to keep the model updated with new data patterns while retaining previously acquired information. Consequently, this approach can help maintain good performance over different learning tasks despite shifts in the underlying data distribution. However, real-world dynamical processes often involve significant temporal variations in the data distribution. For example, the same input features may map to entirely different labels across different time domains. Given such potential mapping conflicts across time, it is unrealistic to expect a single model to maintain good performance across all the time periods, as assumed in conventional continual learning methods. Another possibility is to leverage domain adaptation methods, which aim to transfer knowledge from a source domain to a target domain to facilitate model adaptation [12–16]. Although these methods allow parameter adaptation across time domains, they are not designed for continuous model adaptation over multiple time domains during long time periods. Moreover, they require pre-defined domains, which are not existent in temporal

*University of Pittsburgh. Email:{yimingsun, ruy59, runxue.bao, yey5, xiaowei}@pitt.edu

†University of Maryland. Email:xie@umd.edu

data, and do not consider the dynamic data distribution shifts due to the change of environmental conditions.

In this work, we hypothesize that while the conditional distribution of data changes continuously over time, incorporating previous data points that share a similar distribution with the current data can enhance the learning process. To achieve this, we propose **D**omain-**A**daptive **C**ontinual **M**eta-Learning (DACM), a method that enables models to detect distribution changes and adaptively adjust to the current distribution. As shown in Figure 1, DACM balances between temporal exploration—leveraging continuous temporal data to capture new patterns—and distributional exploitation—utilizing historical data similar in distribution to the current observations.

DACM employs a data-driven approach to group data into multiple domains based on their mutual similarity. Similar to other continual learning methods, it maintains a memory buffer to store data points from previously encountered domains. When a new data point arrives, the method selects data points from the buffer that exhibit a distribution similar to the current one. By exploiting this similar-in-distribution buffer, the model can quickly adapt to new patterns during temporal exploration, achieving a balance between accuracy and generalization. This approach allows DACM to effectively handle temporal distribution shifts inherent in non-stationary environments. This methodology not only improves the model's ability to generalize across different temporal distributions but also demonstrates potential applicability to various domains where data distribution shifts over time.

Our contributions are summarized as follows:

- We introduce DACM, a novel method designed to handle temporal distribution shifts in non-stationary environments. DACM balances between temporal exploration and distributional exploitation, allowing the model to adapt to new data patterns over time while leveraging historical data similar in distribution to the current observations.
- We present a data-driven approach to group data into multiple domains based on their mutual similarity. This grouping facilitates efficient selection of relevant historical data, improving the model's exploitation of past experiences that are most beneficial for current predictions.
- We conduct comprehensive experiments on a real-world stream temperature dataset. The results demonstrate the superiority of DACM over diverse baselines, including LSTM and Transformer-based models, as well as other continual learning approaches, in terms of both prediction accuracy and adaptability to temporal distribution shifts.

2 Related Work

2.1 Continual Learning. Continual Learning [10] aims to learn sequentially from a stream of tasks without forgetting prior knowledge. One of the key challenges is catastrophic forgetting, where the model loses performance on earlier tasks when adapting to new ones. Various methods have been proposed to mitigate this issue, typically with three categories: regularization-based, replay-based and optimization-based methods.

Regularization-based methods constrain the model's parameters to preserve performance on prior tasks. This is typically done by storing a frozen copy of previous models and penalizing changes to important parameters [17, 18]. These approaches mitigate task interference by ensuring that specific neurons remain dedicated to earlier tasks, thereby reducing the likelihood of catastrophic forgetting.

Replay-based methods maintain a memory buffer containing samples from previous tasks. Models like Gradient Episodic Memory (GEM) [19] and Averaged GEM (A-GEM) [20] use these memory samples to constrain gradient updates, minimizing conflicts between old and new tasks during training. Meta-Experience Replay (MER) [21] further improves this process by encouraging gradient alignment between past and present tasks through replay buffers.

Optimization-based methods focus on redesigning the optimization process to prevent catastrophic forgetting. Online-aware Meta Learning (OML) [22] incorporates a meta-learning objective to learn representations that are robust to interference during online updates, promoting future learning. Look-ahead Meta Learning (La-MAML) [23] optimizes the OML objective by introducing a learnable learning rate, further reducing interference between old and new tasks.

2.2 Meta-Learning. Meta-learning, often referred to as "learning to learn", is designed to learn from previous learning episodes to improve future learning performance. Instead of learning a specific task, meta-learning models aim to understand the underlying structure of tasks so the models can quickly adapt to new, unseen tasks with minimal data. This approach is particularly valuable in scenarios where data is scarce or expensive to obtain, such as few-shot learning problems.

One of the most well-known approaches is Model-Agnostic Meta-Learning (MAML) [24], which is designed to find a set of model parameters that serve as a good initialization point for learning new tasks. The core idea is to optimize the model's initial parameters such that a small number of gradient updates with respect to a new task's loss function will produce effective generalization on that task. Numerous exten-

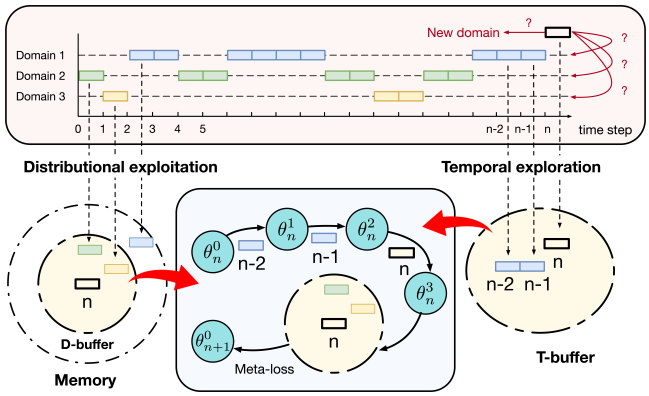


Figure 1: Overall framework of our DACM method.

sions of MAML have been developed to enhance computational efficiency and stability [25, 26], as well as to broaden its applications, including in environmental modeling [27, 28].

3 Problem Definition

In the water temperature prediction scenario, our objective is to learn a model $\mathcal{F}_\theta : \mathcal{X} \to \mathcal{Y}$ to predict the water temperature $y \in \mathcal{Y}$ for a specific stream, given its input features $x \in \mathcal{X}$ at each daily time step $t \in \{1, \ldots, T\}$. Specifically, the input feature space encompasses the physical variables that drive the basin system dynamics, and includes both meteorological features (e.g., solar radiation, rainfall, air temperature) and hydrological conditions. We consider stream data arriving in small batches. For example, water data is typically collected and processed at regular intervals, often requiring cleaning before analysis. Hence, we divide the available data sequence $\{x_t, y_t\}_{t=1}^{T}$ into several intervals, with each interval containing k data points. Each interval is indexed by the starting time step, as $(X_j, Y_j) = \{(x_{jk+l}, y_{jk+l})\}_{l=1}^{k}$.

The aim of our work is to consistently predict the water temperature for a future k-length interval, $Y_{j+1} = \{y_{(j+1)k+l}\}_{l=1}^{k}$, given the input of that interval $X_{j+1} = \{x_{(j+1)k+l}\}_{l=1}^{k}$, and both the input and labels of its previous intervals, $\{(X_m, Y_m)\}_{m=1}^{j}$. Here we assume the access of observed labels for previous intervals when predicting for the next interval. Real environmental data often exhibit temporal concept shifts due to changes in environmental conditions. Hence, $P(Y_j \mid X_j)$ is non-stationary as time progresses (i.e., j increases). This implies that the parameters θ in $\mathcal{F}_\theta$ need to be adaptively adjusted over time.

4 Method

In this section, we provide details for the proposed Domain-Adaptive Continual Meta-Learning (DACM) method. The objective of the method is to dynamically update a given temporal learning model to adapt to the shifting environment. The learning model is utilized to automatically identify concept drifts and segment temporal data into homogeneous domains. We will then keep the learning model updated by leveraging the knowledge learned from both current evolving data and historical data with similar patterns (Figure 1). Here we adopt the LSTM as the base learning model due to its popularity in aquatic science [6, 29–33] and the superiority over advanced models (e.g., Transformer-based models) as reported in prior work [29, 32]. However, the proposed method can be generally applied to other temporal models.

In the following, we first discuss how we automatically segment temporal data into homogeneous domains using mutual similarity in distribution space. We then introduce how the learning model can be continuously updated using a meta-learning approach that balances between exploring temporal continuous data and exploiting historical data with similar distributions. The detailed process is outlined in Algorithm 1.

4.1 Domain Segmentation.

Environmental data are continuously collected over time while also exhibiting severe concept drifts due to temporal environmental changes. It is critical for the temporal model to be aware of significant distribution shifts so that it can smoothly adapt to the new environment. However, such temporal domain boundaries are not pre-defined and cannot be directly obtained based on prior knowledge.

To address this challenge and facilitate the learning of the evolving data distribution, we propose a new domain segmentation mechanism to automatically group intervals into several domains based on the properties of the temporal learning model $\mathcal{F}$. Ideally, intervals within the same domain are expected to share similar distributions, while intervals in different domains should follow different distributions. Specifically, for two intervals i and j within the same domain, we assume that $P(Y_i \mid X_i) \simeq P(Y_j \mid X_j)$. We define a domain $\mathbf{D}_n$ as a collection of intervals $\{(X_i, Y_i)\}$, where $i \in \{i_1^n, \ldots, i_{l_n}^n\}$, representing the indices of intervals in domain $\mathbf{D}_n$. It is important to note that these indices $\{i_1^n, \ldots, i_{l_n}^n\}$ are not necessarily consecutive, which allows non-contiguous intervals to be grouped into the same domain. Intuitively, we wish to identify and group distributionally similar intervals over long time periods, which may not necessarily be contiguous over time, e.g., intervals in winter time from different years.

To group intervals into these domains, we consider a similarity measure based on their contribution to model training. In particular, for the learning model $\mathcal{F}_\theta$, we

measure its gradients of loss $\mathcal{L}_i = \frac{1}{k}||\mathcal{F}_\theta(X_i) - Y_i||^2$ with respect to its parameters θ, as $\frac{\partial \mathcal{L}_i}{\partial \theta}$. The gradients reflect the model adjustments needed to capture patterns in the interval (X_i, Y_i). Then we measure the cosine similarity between the gradients of two intervals (X_i, Y_i) and (X_j, Y_j), i.e., $\cos(\frac{\partial \mathcal{L}_i}{\partial \theta}, \frac{\partial \mathcal{L}_j}{\partial \theta})$. Higher similarity indicates that the model is learning similar relationships between input features and the output variable from two intervals. It is also worthwhile to mention that observations could be collected only on certain time steps in real environmental ecosystems, and the loss $\mathcal{L}_i$ will be computed only on available observations.

In our problem setting, as new data interval arrives, we calculate the gradient for the new interval and compare it against existing domains. For each of the existing domains $\mathbf{D}_n$, we maintain a memory $\mathbf{R}_n$ that contains $|\mathbf{R}_n|$ intervals that were previously added to the domain. To measure the similarity between a newly arrived interval i and an existing domain $\mathbf{D}_n$, we compute the average cosine similarity between the gradients of the interval i and different intervals in $\mathbf{R}_n$, as follows:

$$\text{sim}((X_i, Y_i), \mathbf{D}_n) = \frac{1}{|\mathbf{R}_n|} \sum_{j \in \mathbf{R}_n} \cos(\frac{\partial \mathcal{L}_i}{\partial \theta}, \frac{\partial \mathcal{L}_j}{\partial \theta}). \tag{4.1}$$

If the gradient of the new interval (X_i, Y_i) aligns with at least one of the existing domains, i.e., the cosine similarity exceeds a certain threshold λ, we assign the interval to the domain with the highest similarity. Conversely, if the gradient of the new interval differs significantly from all existing domains (low or negative values of cosine similarity), it suggests a shift to an unseen data distribution. We will then create a new domain starting from the new interval (X_i, Y_i).

4.2 Temporal Meta Learning with Memory. To preserve the information from historical data, we propose to create a temporary D-buffer $\mathbf{S}_D$ for each newly arrived interval (X_i, Y_i). The D-buffer contains historical data intervals with patterns similar to the new interval. This approach facilitates model updates by fully leveraging historical information through distributional exploitation (Figure 1). Specifically, we calculate the cosine similarity between the gradients of the new interval and the gradients of the intervals stored in the memory $\{\mathbf{R}_n\}$ of existing domains. Based on these cosine similarities, the method selects a set of intervals from the previous domains to form the D-buffer, provided that the similarity exceeds a pre-defined threshold. This selection process ensures that only intervals that share similar input-output relationships are included in the D-buffer, thereby promoting effective adaptation and generalization. It also offers flexibility that the D-buffer could contain intervals from different domains.

Next, we create a T-buffer $\mathbf{S}_T$ for the new interval i to enable the temporal exploration of useful patterns over a continuous time period (Figure 1). The T-buffer contains the past τ consecutive intervals and the interval itself, i.e., $\mathbf{S}_T = \{(X_{i-\tau}, Y_{i-\tau}), ..., (X_i, Y_i)\}$. The update of the model $\mathcal{F}_\theta$ for the new interval (X_i, Y_i) essentially balances between distributional exploitation and temporal exploration, by exploring recent temporal patterns while also leveraging most beneficial knowledge from the historical data. As inspired by prior work [23], we introduce a meta-learning process, which tunes the model with the intervals in the updated T-buffer while also optimizing the consistency between the tuned model and the D-buffer. In particular, the model is continuously updated using each interval $(X_{i-\tau+t}, Y_{i-\tau+t})$, for $t = 0$ to τ. After each update step t, we measure the performance of the tuned model on the D-buffer $\mathbf{S}_D$ as $\mathcal{L}_s(i, t)$, given by:

$$\begin{aligned} \mathcal{L}_s(i,t) &= \frac{1}{|\mathbf{S}_D|} \sum_{i' \in \mathbf{S}_D} \mathcal{L}(\mathcal{F}_{\theta_i^t}(X_{i'}), Y_{i'}), \\ \theta_i^t &= \theta_i^{t-1} - \alpha \nabla_\theta \mathcal{L}(\mathcal{F}_{\theta_i^{t-1}}(X_{i-\tau+t}), Y_{i-\tau+t}), \end{aligned} \tag{4.2}$$

where $\mathcal{L}$ represents the squared error, and α denotes the learning rate of gradient descent on each sample in the new interval. The initial parameter value θ_i^0 is set to be the previous parameters of the model $\mathcal{F}$ before the arrival of the interval (X_i, Y_i).

Then we summarize the inconsistency with the D-buffer by summing $\mathcal{L}_s(i, t)$ over all the intermediate update steps using intervals in the T-buffer, and minimize the overall inconsistency error, as $\min_{\theta_i^0} \sum_{t=0}^{\tau} L_s(i, t)$.

Here we update the initial parameters θ_i^0 to optimize the inconsistency error. This method provides twofold benefits. First, by penalizing on the deviation from the knowledge learned from historical data, it helps mitigate the overfitting on the new interval. Second, it allows adjustment of the initial parameter values θ_i^0, which were originally learned from previous time. This could help fix the model bias given new data samples and prevent the model from perpetuating the errors.

In summary, the proposed DACM method combines the domain segmentation mechanism and the meta-learning-based model update. Furthermore, if the new interval is assigned to the same domain as its previous interval, it suggests that they share similar patterns. In this case, we only update the model using the meta learning approach. In contrast, if the new interval is grouped into a different domain, it indicates a shift in patterns. In this case, the method needs to revisit information from the emerging domain using the D-buffer,

Algorithm 1 Domain-Adaptive Continual Meta-Learning

Input: Sequence of intervals $\{(X_i, Y_i)\}, i \in 0, 1, \ldots,$ similarity threshold λ, length of exploration window τ. Initialize $\mathbf{R}, \mathbf{S}_T, \mathbf{S}_D, \mathcal{M}_{I2D}(\cdot)$.

while *receive new interval* $\{(X_i, Y_i)\}$ **do**
- $\hat{Y}_i = \text{model}\,(\theta_i, X_i)$
- $\mathbf{S}_T \leftarrow \{(X_{i-\tau}, Y_{i-\tau}), \ldots, (X_i, Y_i)\}$
- $g_T = \frac{\partial \mathcal{L}(\mathbf{S}_T)}{\partial \theta_i}$
- $g_j = \frac{\partial \mathcal{L}(\mathbf{R}_j)}{\partial \theta_i},\ \mathrm{j} \in \{1, \ldots, \text{len}(\mathbf{R})\}$
- $\text{sim}_j = \cos(g_T, g_j),\ \mathrm{j} \in \{1, \ldots, \text{len}(\mathbf{R})\}$
- $\mathbf{S}_D \leftarrow \mathbf{S}_D \cup \mathbf{R}_j$ for $\text{sim}_j \geq \lambda$
- $\mathbf{S}_D \leftarrow \mathbf{S}_D \cup (X_i, Y_i)$
- `// Meta-update the model`
- **for** $t \in \{0, \ldots, \tau\}$ **do**
  - $\theta_i^{t+1} \leftarrow \text{Fast-update}(\theta_i^t, (X_{i-\tau+t}, Y_{i-\tau+t}))$
- **end**
- $\theta_{i+1}^0 \leftarrow \text{Update}(\theta_i^{\tau+1}, \mathbf{S}_D)$
- **if** *max (sim)* $< \lambda$ **then**
  - `// Encounter with an unseen domain`
  - $\mathbf{R} \leftarrow \mathbf{R} \cup (X_i, Y_i)$
  - $\mathcal{M}_{I2D}(i) \leftarrow \text{len}(\mathbf{R})$
- **end**
- **else**
  - $\mathcal{M}_{I2D}(i) \leftarrow \text{index}(\max(\text{sim}))$
- **end**
- **if** $\mathcal{M}_{I2D}(i) \neq \mathcal{M}_{I2D}(i-1)$ **then**
  - `// Distribution shift happens`
  - $\theta_{i+1} \leftarrow \text{Update}(\theta_{i+1}, \mathbf{S}_D)$
- **end**

end

allowing the model to recall previously learned patterns from the corresponding domain. By doing so, the model can adjust its parameters appropriately, mitigating the risk of significant performance degradation across domains due to the temporal domain shift.

5 Experiments

5.1 Dataset. We evaluate the proposed method for predicting stream temperature data collected from the Delaware River Basin (DRB), which is an ecologically diverse region and a watershed along the east coast of the United States that provides drinking water to over 15 million people [34]. The dataset [35] used in our evaluation is from the U.S. Geological Survey's National Water Information System [36] and the Water Quality Portal [37]. Observations at a specific latitude and longitude were matched to river segments, which were defined by the geospatial fabric used for the National Hydrologic Model [38]. The river segments are split up to have roughly a 1-day water travel time. The observations were snapped to the nearest stream segment within a tolerance of 250 m. Observations farther than 5,000 m along the river channel to the outlet of a segment were omitted from the dataset. Segments with multiple observation sites were aggregated to a single mean daily water temperature value.

To better evaluate the proposed method, we select eight stream segments from DRB with the least missing observations of water temperature. The temperature observations available per segment vary from 10,105 to 13,000. We use input features at the daily scale from January 01, 1980, to March 31, 2020 (14,701 dates). The input features have 10 dimensions, which include daily mean precipitation, daily mean air temperature, date of the year, solar radiation, shade fraction, potential evapotranspiration and the geometric features of each segment (elevation, length, slope, and width). Air temperature, precipitation, and solar radiation values were derived from the gridMET gridded meteorological dataset [39]. Other input features (e.g., shade fraction, potential evapotranspiration) are difficult to measure frequently, and we use values internally calculated by the physics-based PRMS-SNTemp model [40].

5.2 Baselines. To assess the performance of our proposed DACM method, we conducted extensive experiments on our environmental dataset, comparing it against several baseline models. The baselines include LSTM-based models and Transformer-based models.

- **LSTM-based models**: LSTM has been widely used for sequential data modeling. Based on LSTM model, we implement several continual learning methods: Experience Replay (ER) stores past experiences and replays them during training to prevent catastrophic forgetting. Meta Experience Replay (MER) [21] extends traditional ER by incorporating meta-learning to learn how to better generalize across tasks by using a meta-optimization process. Gradient Episodic Memory (GEM) [19] ensures that the gradient updates made for new tasks do not interfere with the gradients of previously learned tasks by constraining the dot product between the gradients for the new and old tasks to remain non-negative. Averaged Gradient Episodic Memory (A-GEM) [20] approximates the GEM process by ensuring that the gradient direction for the new task does not interfere, on average, with the past tasks. For stream temperature prediction, Heterogeneous Recurrent Graph Networks (HRGN) [41] is a specialized model, designed to capture the diverse and distinct behaviors of interconnected systems. It extends LSTM by modeling the relationships among different river segments. Moreover, HRGN introduces a data assimilation

Table 1: Predictive RMSE for water temperature using DACM and baseline methods, with best results bolded. The gray lines represent the methods that continuously update their parameters given new observations.

Method	A	B	C	D	E	F	G	H
LSTM	1.6729	2.0050	1.6355	1.6793	1.6143	1.8958	1.1050	1.7049
ER	1.5607	2.3885	1.5701	1.8482	1.6880	1.5765	1.3235	1.4520
MER	1.5664	1.7826	1.4073	1.7743	1.4331	1.3123	1.0148	1.0269
GEM	1.6979	2.1084	2.3709	1.5374	2.1063	2.2310	1.4578	1.6435
A-GEM	1.6097	1.8788	1.7955	1.5221	1.8650	1.7154	1.5142	1.5820
HRGN	1.6819	1.8971	1.5193	1.4785	1.4063	1.2591	1.0323	1.1946
Transformer	1.7555	2.3227	1.6957	1.6957	1.7043	1.7287	1.3537	1.8052
Informer	1.8953	2.9826	2.1455	2.5286	2.5672	3.0769	2.0790	2.2417
Autoformer	2.3198	2.5177	2.7918	2.2023	2.8069	2.8854	2.4399	2.8286
DACM	**1.5170**	**1.5337**	**1.3603**	**1.4518**	**1.3759**	**1.1722**	**0.8421**	**0.9915**

mechanism, which efficiently adjusts the model's internal state in response to incoming observations.

- **Transformer-based models**: Transformer is a model based on self-attention mechanisms, suitable for capturing long-range dependencies in time series data. Informer [42] is a Transformer-based model specifically designed for efficient long-term time series forecasting. It introduces a generative style decoder to predict long-time series sequences using only one forward. Autoformer [43] is another Transformer variant that introduces a decomposition-based architecture to separate time series data into trend and seasonal components and uses auto-correlation mechanisms to effectively model the periodic dependencies over time.

5.3 Experimental Settings. Since DACM operates as an online learning method, continuously testing and then training on incoming sequences of intervals, there is no pre-defined split between the training and testing datasets. For the evaluation presented in the following tables and figures, we use data from January 01, 2015 to March 31, 2020 as the evaluation period. The basic LSTM and all the Transformer-based models use a fixed set of parameters in testing, and they are trained only on the data from January 01, 1980, to December 31, 2014. To provide a pre-defined segmentation of tasks for continual learning, we utilize the natural division of seasons and define each season as a separate task. For the backbone model, we utilize a two-layer LSTM for all the LSTM-based models. Both the LSTM-based and Transformer-based models have hidden layer dimensions set to 10. The learning rate is set to 0.05. We use a cosine similarity threshold $\lambda = 0.8$ and set the length of exploration window $\tau = 2$. The root mean squared error (RMSE) between the observed and predicted values is used as the performance metric to assess the model's accuracy. The evaluations are conducted only on the eight selected stream segments with the most water temperature observations.

5.4 Results and Analyses

5.4.1 Comparative performance with baselines. As shown in Table 1, our DACM method outperforms all of the baseline models. The primary limitation of basic LSTM and Transformer models lies in their assumption of static data distributions. Both of these models learn a fixed set of parameters across the entire training dataset, which prevents them from effectively adapting to the temporal data with non-stationary and evolving distributions. By ignoring the temporal concept drift in underlying patterns, they fail to capture the dynamic behavior inherent in environmental processes, leading to suboptimal performance.

Continual learning methods, while designed to mitigate catastrophic forgetting by maintaining knowledge across multiple tasks, struggle in temporal datasets due to the inherent challenge of generalizing across evolving tasks. These methods attempt to find a balance between retaining past knowledge and learning new patterns, but this trade-off becomes problematic in temporal settings where the data distribution is constantly shifting. Moreover, they follow the pre-defined temporal task separation but cannot adaptively segment time periods with homogeneous patterns. As a result, continual learning methods demonstrate worse performance when applied to non-stationary datasets, as they cannot fully capture to the unique characteristics of temporal domain shifts. In the comparison between episodic-memory-based and experience-replay-based methods, episodic-memory-based methods (GEM and A-GEM) have relatively worse performance. They aim to ensure that gradient updates for new tasks do not interfere with gradients from previously learned tasks, however, this assumption does not hold well in our dataset. In

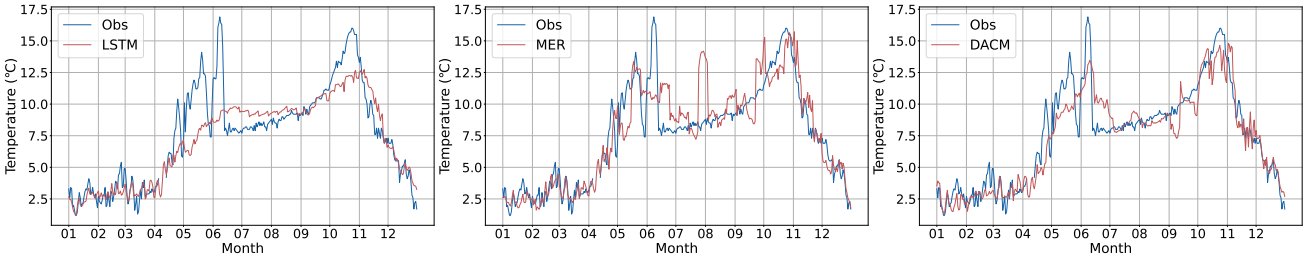


Figure 2: Illustrative cases demonstrating baseline limitations in predictive performance and adaptiveness.

particular, for comparison between episodic-memory-based methods, A-GEM not only enhances the efficiency of GEM but also slightly improves performance. For experience-replay methods, MER outperforms ER, highlighting the effectiveness of meta-learning.

Transformer-based models perform worse than LSTM in our dataset because they rely heavily on previous context, neglecting the critical point-to-point relationships required for accurate predictions. On top of that, models like Informer and Autoformer, while specifically designed for time series forecasting, rely even more heavily on previous observations. However, in our dataset, water temperature observations are frequently missing, requiring a greater reliance on related features. As a result, Informer and Autoformer struggle to capture the necessary one-to-one mappings, leading to unsatisfactory performance.

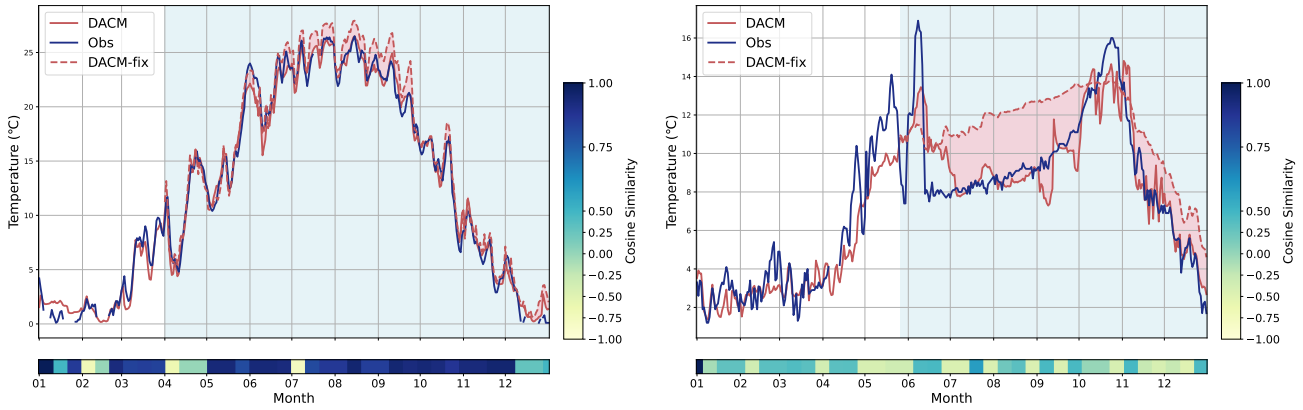
HRGN, specifically designed for this application scenario, performs the best among all the baselines. However, its accuracy falls short compared to DACM, particularly when faced with drastic and unpredictable changes, further highlighting the effectiveness of DACM in handling such dynamic conditions.

Besides the overall performance comparison between DACM and the baselines, we showcase two key drawbacks of the baseline models using a drastically changing example, as shown in Figure 2. First, in the LSTM baseline, the model struggles to adapt to varying distributions, causing it to lag behind and fail to capture current patterns. Second, in the MER baseline, while the model is capable of detecting distribution shifts, it tends to overfit on the most recent data, leading to unstable performance. In contrast, DACM is able to quickly and accurately adapt to new domains and maintain high accuracy even with distribution shifts.

5.4.2 Visualization of domain segmentation. One key outcome of our DACM method is its ability to produce data-driven domain segmentation, which is particularly valuable in handling continuous temporal data where clear boundaries between domains are often non-existent. A typical way to create domain segmentation in environmental studies is to follow seasonal patterns. For example, a common segmentation approach is to divide the year into four seasons: spring (March, April, May), summer (June, July, August), fall (September, October, November), and winter (December, January, February). However, this seasonal segmentation is overly simplistic, as it fails to capture fine-grained patterns in a dynamic and continuously evolving environment.

In contrast, our data-driven approach dynamically segments the temporal data based on real-time changes in data distribution, grouping intervals into domains according to calculated similarities. This allows for a more nuanced understanding of domain shifts. To visualize these segmentation outcomes, we present domain segmentation aligned with the predictions generated by the DACM method shown in Figure 3. The domain information is represented by calculating the gradient similarity between the domain of the first interval in the year and the domain of the current interval.

We showcase two examples: one with relatively stable behavior (Figure 3(a)) and another with significant variability due to an upstream temperature-regulating reservoir (Figure 3(b)). In Figure 3(a), the conditional distribution remains relatively consistent throughout the year, and the predictions closely match the observed data. When we fix the domain as of April 01 and prevent domain updates, the predictions degrade slightly, though not significantly. However, in Figure 3(b), there is a sharp drop in water temperature around mid-June due to the cold water inflow from the upstream reservoir. The predictions made by DACM demonstrate its ability to quickly adapt to these changes, maintaining a high level of accuracy. On the other hand, if we fix the domain as of May 26, the predictions deviate significantly from the observations, underscoring the model's inability to keep up with evolving patterns. This comparison clearly illustrates the necessity of allowing for shifts between domains. Without this flexibility, models are unable to adapt to sudden distributional changes



(a) One stationary example.

(b) One non-stationary example.

Figure 3: Comparing predictions made by DACM, observations and domain segmentation. DACM adaptively adjusts the domains, whereas DACM-fix keeps the domain fixed in the blue shaded area. The pink shaded area highlights the difference before and after introducing the domain-adaptive mechanism.

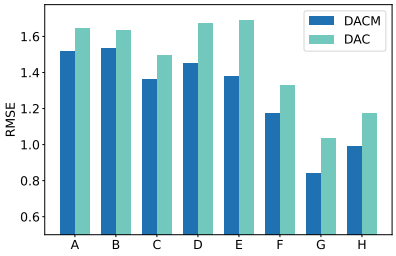


Figure 4: Comparison between DACM and DAC across different stream segments.

in the data, leading to significant gaps between predictions and observations.

5.4.3 Effectiveness of using meta-learning. In the DACM method, we leverage two buffers: T-buffer for temporal exploration aspect and D-buffer for distributional exploitation aspect. Using the meta-learning approach, we update the model with optimized initial parameters, allowing DACM to quickly adapt to new patterns with a small amount of data. This essentially pursues a balance between T-buffer and D-buffer, which is key to handling dynamic and evolving data.

To validate the effectiveness of this meta-learning component, we conducted an ablation study where we replaced the meta-loss with a naive supervised loss on the combination data from the T-buffer and D-buffer. We refer to this version of the model as Domain-Adaptive Continual Learning (DAC). As shown in Figure 4, DACM consistently outperforms DAC across all data points, highlighting the importance of incorporating meta-learning to achieve superior performance in dynamic environments.

6 Conclusion

In this work, we proposed DACM, an innovative machine learning method that adaptively adjusts to non-stationary patterns in sequential temporal data. Without relying on pre-defined domain segmentation, DACM employed a data-driven approach to group data into domains based on the similarity of the conditional distribution. As we sequentially explored incoming temporal data, our method selectively chose similar domains from previously encountered ones. By utilizing a meta-learning strategy, DACM stroke a balance between temporal exploration and distributional exploitation using a limited amount of data. Tested on a real-world water temperature dataset, DACM demonstrated superior accuracy and robustness compared to several commonly used baselines. Furthermore, the application of DACM was not limited to water temperature predictions; it could also be extended to a wide range of environmental applications that commonly exhibit temporal data shifts due to changes in environmental conditions.

Acknowledgment

This work was supported by the National Science Foundation (NSF) under grants 2239175, 2316305, 2147195, 2425844, 2425845, 2430978, and 2126474, the USGS grants G21AC10564 and G22AC00266, the NASA grants 80NSSC22K1164 and 80NSSC24K1061, the Google's AI for Social Good Impact Scholars program, and the National Institutes of Health (NIH) through the National Library of Medicine under grant R00LM013383. This research was also supported in part by the University of Pittsburgh Center for Research Computing through the resources provided.

References

[1] Wang Zhou, Kaiyu Guan, Bin Peng, Jinyun Tang, Zhenong Jin, Chongya Jiang, Robert Grant, and Symon Mezbahuddin. Quantifying carbon budget, crop yields and their responses to environmental variability using the ecosys model for us midwestern agroecosystems. *Agricultural and Forest Meteorology*, 307:108521, 2021.

[2] Jeffrey G Arnold, Daniel N Moriasi, Philip W Gassman, Karim C Abbaspour, Michael J White, Raghavan Srinivasan, Chinnasamy Santhi, RD Harmel, Ann Van Griensven, Michael W Van Liew, et al. Swat: Model use, calibration, and validation. *Transactions of the ASABE*, 55(4):1491–1508, 2012.

[3] Steven L Markstrom. *P2S–coupled Simulation with the Precipitation-Runoff Modeling System (PRMS) and the Stream Temperature Network (SNTemp) Models*. US Department of the Interior, US Geological Survey, 2012.

[4] Matthew R Hipsey, Louise C Bruce, Casper Boon, Brendan Busch, Cayelan C Carey, David P Hamilton, Paul C Hanson, Jordan S Read, Eduardo de Sousa, Michael Weber, et al. 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, 2019.

[5] Licheng Liu, Wang Zhou, Kaiyu Guan, Bin Peng, Shaoming Xu, Jinyun Tang, Qing Zhu, Jessica Till, Xiaowei Jia, Chongya Jiang, et al. Knowledge-guided machine learning can improve carbon cycle quantification in agroecosystems. *Nature Communications*, 15(1):357, 2024.

[6] 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. *ACM/IMS Transactions on Data Science*, 2(3):1–26, 2021.

[7] Shifa Zhong, Kai Zhang, Majid Bagheri, Joel G Burken, April Gu, Baikun Li, Xingmao Ma, Babetta L Marrone, Zhiyong Jason Ren, Joshua Schrier, et al. Machine learning: new ideas and tools in environmental science and engineering. *Environmental science & technology*, 55(19):12741–12754, 2021.

[8] Changlu Chen, Yanbin Liu, Ling Chen, and Chengqi Zhang. Test-time training for spatial-temporal forecasting. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 463–471. SIAM, 2024.

[9] 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*, pages 266–275, 2023.

[10] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[11] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[12] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[14] Runxue Bao, Yiming Sun, Yuhe Gao, Jindong Wang, Qiang Yang, Zhi-Hong Mao, and Ye Ye. A recent survey of heterogeneous transfer learning. *arXiv preprint arXiv:2310.08459*, 2024.

[15] Nasrin Kalanat, Yiqun Xie, Yanhua Li, and Xiaowei Jia. Spatial-temporal augmented adaptation via cycle-consistent adversarial network: An application in streamflow prediction. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 598–606. SIAM, 2024.

[16] Yiming Sun, Yuhe Gao, Runxue Bao, Gregory F Cooper, Jessi Espino, Harry Hochheiser, Marian G Michaels, John M Aronis, Chenxi Song, and Ye Ye. Online transfer learning for rsv case detection. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, pages 512–521. IEEE, 2024.

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[18] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[19] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[20] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[21] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[22] Khurram Javed and Martha White. Meta-learning representations for continual learning. *Advances in neural information processing systems*, 32, 2019.

[23] Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. *Advances in Neural Information Processing Systems*, 33:11588–11598, 2020.

[24] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[25] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[26] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *International conference on learning representations*, 2018.

[27] 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)*, pages 217–225. SIAM, 2023.

[28] Shengyu Chen, Jacob A Zwart, and Xiaowei Jia. Physics-guided graph meta learning for predicting water temperature and streamflow in stream networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2752–2761, 2022.

[29] Chaopeng Shen and Kathryn Lawson. Applications of deep learning in hydrology. *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, pages 283–297, 2021.

[30] 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.

[31] Kyeungwoo Cho and Yeonjoo Kim. Improving streamflow prediction in the wrf-hydro model with lstm networks. *Journal of Hydrology*, 605:127297, 2022.

[32] 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)*, pages 580–589. IEEE, 2024.

[33] Runlong Yu, Robert Ladwig, Xiang Xu, Peijun Zhu, Paul C Hanson, Yiqun Xie, and Xiaowei Jia. Evolution-based feature selection for predicting dissolved oxygen concentrations in lakes. In *International Conference on Parallel Problem Solving from Nature*, pages 398–415. Springer, 2024.

[34] Tanja N Williamson et al. Summary of hydrologic modeling for the delaware river basin using the water availability tool for environmental resources (water). Technical Report 2015-5143, U.S. Geological Survey Scientific Investigations Report, 2015.

[35] Samantha K Oliver et al. Predicting water temperature in the delaware river basin. U.S. Geological Survey Data Release, 2021.

[36] US Geological Survey. National water information system (usgs water data for the nation). 2016.

[37] Emily K Read et al. Water quality data for national-scale aquatic research: The water quality portal. *Water Resources Research*, 2017.

[38] R Steven Regan, Steven L Markstrom, Lauren E Hay, Roland J Viger, Parker A Norton, Jessica M Driscoll, and Jacob H LaFontaine. Description of the national hydrologic model for use with the precipitation-runoff modeling system (prms). Technical Report 6-B9, U.S. Geological Survey Techniques and Methods, 2018.

[39] John T Abatzoglou. Development of gridded surface meteorological data for ecological applications and modelling. *International journal of climatology*, 33(1):121–131, 2013.

[40] Michael J Sanders, Steven L Markstrom, R Steven Regan, and R Dwight Atkinson. Documentation of a daily mean stream temperature module—an enhancement to the precipitation-runoff modeling system. Technical Report 6-D4, US Geological Survey, 2017.

[41] Shengyu Chen, Alison Appling, Samantha Oliver, Hayley Corson-Dosch, Jordan Read, Jeffrey Sadler, Jacob Zwart, and Xiaowei Jia. Heterogeneous stream-reservoir graph networks with data assimilation. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1024–1029. IEEE, 2021.

[42] 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, pages 11106–11115, 2021.

[43] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.