

Drift-aware Collaborative Assistance Mixture of Experts for Heterogeneous Multistream Learning

En Yu, Jie Lu*, Kun Wang, Xiaoyu Yang, Guangquan Zhang

Australian Artificial Intelligence Institute (AAIL)
University of Technology Sydney (UTS), Australia

Abstract

Learning from multiple data streams in real-world scenarios is fundamentally challenging due to intrinsic heterogeneity and unpredictable concept drifts. Existing methods typically assume homogeneous streams and employ static architectures with indiscriminate knowledge fusion, limiting generalizability in complex dynamic environments. To tackle this gap, we propose **CAMEL**, a dynamic Collaborative Assistance Mixture of Experts Learning framework. It addresses heterogeneity by assigning each stream an independent system with a dedicated feature extractor and task-specific head. Meanwhile, a dynamic pool of specialized private experts captures stream-specific idiosyncratic patterns. Crucially, collaboration across these heterogeneous streams is enabled by a dedicated assistance expert. This expert employs a multi-head attention mechanism to distill and integrate relevant context autonomously from all other concurrent streams. It facilitates targeted knowledge transfer while inherently mitigating negative transfer from irrelevant sources. Furthermore, we propose an Autonomous Expert Tuner (AET) strategy, which dynamically manages expert lifecycles in response to drift. It instantiates new experts for emerging concepts (freezing prior ones to prevent catastrophic forgetting) and prunes obsolete ones. This expert-level plasticity provides a robust and efficient mechanism for online model capacity adaptation. Extensive experiments demonstrate **CAMEL**'s superior generalizability across diverse multistreams and exceptional resilience against complex concept drifts.

Introduction

Learning from streaming data has become fundamental to modern intelligent systems, enabling real-time decision-making in dynamic and continuously evolving environments (Cacciarelli and Kulahci 2024; Marcu and Bouvry 2024; Agrahari and Singh 2022). A central challenge in streaming learning is concept drift—the phenomenon where the underlying data distribution changes over time—requiring models to continuously adapt in order to maintain predictive performance (Lu et al. 2018). While most streaming learning studies focus on single-stream settings (Jiao et al. 2024; Wen et al. 2023), many

real-world applications inherently involve multiple concurrent data streams. For example, a smart city platform integrates traffic sensor feeds, weather reports, public transportation logs, and social media sentiment streams. These streams evolve independently yet often carry latent correlations that, if exploited effectively, can provide complementary information for more accurate and robust decision-making (Xiang et al. 2023; Zhang et al. 2025a; Read and Zliobaite 2025; Ma et al. 2024). Capturing such dynamic inter-stream relationships while adapting to concept drift is crucial for advancing streaming learning toward practical deployment (Yang, Lu, and Yu 2025a; Xu, Chen, and Wang 2025; Liang 2025).

Despite recent progress, existing multistream learning methods face a critical dilemma. On the one hand, most approaches operate under a homogeneous space assumption, which presumes that all streams share the same feature and label spaces (Yu et al. 2024; Jiao et al. 2023). This assumption fails to deal with the intrinsic heterogeneity commonly present in practical applications, where streams may originate from distinct feature spaces or predictive objectives due to different data sources (Korycki and Krawczyk 2021; Panchal et al. 2023). On the other hand, prevailing methods typically employ a monolithic and static architecture, either retrained or incrementally fine-tuned (Wang et al. 2021). This design suffers from critical limitations in multistream environments, e.g., retraining induces catastrophic forgetting of prior knowledge, while fine-tuning becomes fragile under asynchronous drifts, where adapting to one stream's evolution can degrade performance on others. The lack of structural flexibility and targeted adaptation thus prevents robust performance across heterogeneous evolving streams.

To bridge this gap, we formalize the problem as Heterogeneous Multistream Learning (HML), where multiple concurrent data streams exhibit intrinsic heterogeneity, latent inter-stream correlations, and asynchronous concept drifts. Specifically, **1) Intrinsic Heterogeneity**: feature and label spaces across streams differ in dimensionality and semantics, precluding direct application of homogeneous models; **2) Knowledge Fusion**: while streams may contain useful correlations, such relationships are dynamic and selective, requiring mechanisms that can leverage relevant information while avoiding negative transfer from irrelevant streams; and **3) Asynchronous Concept Drifts**: streams evolve in-

*Correspondence to Jie Lu and Xiaoyu Yang
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

dependently with diverse drift patterns, demanding flexible and stream-specific adaptation. These challenges necessitate a generalized and drift-aware learning framework that can handle stream-wise specialization while enabling intelligent knowledge fusion across heterogeneous drifting streams.

To address these challenges, we propose **CAMEL**, a dynamic Collaborative Assistance Mixture of Experts Learning framework tailored for heterogeneous data streams. It introduces a modular drift-aware architecture that explicitly addresses the three core challenges. First, to handle intrinsic heterogeneity, we assign each stream a specific learning system comprising a dedicated feature extractor, a private expert pool, and a task-specific prediction head, ensuring stream-specific specialization. Second, to enable adaptive and selective knowledge fusion, **CAMEL** incorporates a novel collaborative assistance mechanism. It employs a dedicated attention-based expert per stream that dynamically distills relevant contextual information from all other concurrent streams on demand, effectively capturing latent inter-stream correlations while inherently mitigating negative transfer (Vaswani et al. 2017). Third, to cope with asynchronous concept drifts, an Autonomous Expert Tuner (AET) is proposed, which monitors drift signals by a distribution-based drift detector and performance indicators per stream, dynamically adding new experts for emerging concepts and pruning obsolete ones. This expert-level plasticity allows our method to autonomously restructure its capacity and specialization over time. Extensive experiments on diverse synthetic and real-world multistream scenarios demonstrate the superior adaptability and generalization ability of our method compared to existing state-of-the-art methods. In summary, our main contributions are:

- We propose **CAMEL**, a generalized and dynamic MoE framework that learns from multiple data streams characterized by heterogeneous features, diverse label spaces, and asynchronous concept drifts.
- We introduce a collaborative assistance mechanism, where dedicated attention-based experts perform targeted knowledge fusion, providing an effective and adaptive solution to the challenge of positive knowledge transfer.
- We design an autonomous tuning strategy that manages the expert lifecycle at a modular level (adding/pruning experts), offering a more robust and interpretable way for drift adaptation.
- Comprehensive experiments and theoretical analysis validate the generalizability and robustness of our method across complex synthetic and real-world HML scenarios.

Related Works

Stream Learning. Early research in streaming learning primarily addressed single-stream scenarios with concept drift (Wan, Liang, and Yoon 2024; Li et al. 2022; Kim, Hwang, and Whang 2024; Yu et al. 2025; Liang et al. 2025; Yang, Lu, and Yu 2025b), broadly falling into two paradigms: 1) *informed methods* integrate explicit drift detection mechanisms to trigger model adaptation based on distribution variations or error signals (Bifet and Gavalda

2007; Lu et al. 2025a; Gomes, Read, and Bifet 2019; Lu et al. 2025b), while 2) *adaptive approaches* employ detection-free strategies that continuously adjust model parameters in response to evolving data dynamics (Guo, Zhang, and Wang 2021; Brzezinski and Stefanowski 2013; Jiao et al. 2024). Recognizing the ubiquity of concurrent streams, recent multistream learning works can be summarized into two categories: 1) *Multistream classification* aims to transfer knowledge from labeled source streams to unlabeled targets, such as MCMO using multi-objective feature selection (Jiao et al. 2023), OBAL dynamically weighting streams via drift-aware boosting (Yu et al. 2024), and BFSRL learning fuzzy shared representations across streams (Yu, Lu, and Zhang 2024); 2) *Multistream collaborative prediction* exploits complementary information across streams for joint forecasting, typically adopting test-then-adapt schemes. For instance, Wang et al. (Wang et al. 2024) propose adaptive stacking that selectively retrains models for knowledge fusion during drift adaptation, while Wen et al. (Wen et al. 2023) employ dual-branch networks separately modeling temporal and cross-variable dependencies. Similarly, CORAL (Xu, Chen, and Wang 2025) leverages the kernel-induced self-representation method for co-evolving time series. However, both paradigms predominantly assume homogeneous feature spaces and shared label semantics, fundamentally struggling with heterogeneous heterogeneity and asynchronous drifts.

Mixture-of-Experts (MoE). The MoE paradigm achieves scalable, efficient modeling through conditional computation, where a routing mechanism dynamically activates specialized sub-networks ("experts") (Mu and Lin 2025; Lei et al. 2024). This architecture demonstrates strong capabilities in multi-task coordination and continual learning (Qin et al. 2020; Li et al. 2025; Lei et al. 2024) with its sparse activation property preserving computational efficiency while maintaining high model capacity (Sarkar et al. 2023; Tran, Pham et al. 2025; Yang et al. 2025). These inherent advantages naturally align with streaming learning's core challenges, including complex pattern recognition, concept drift adaptation and computational constraints. However, MoE frameworks remain largely unexplored for streaming scenarios while exhibiting critical limitations in HML: expert specialization is statically predefined for coarse task categories without mechanisms to dynamically reconfigure expertise for dynamic scenarios, while routing strategies optimize isolated objectives while neglecting knowledge transfer between complementary experts. Our approach fundamentally advances this paradigm through a correlation-aware expert synthesis framework that jointly models latent task dependencies and expert synergies, enabling real-time expert reorganization and coordinated optimization of both routing precision and cross-expert knowledge transfer, unlocking adaptive capacity allocation for evolving data streams.

Preliminary

Definition 1 (Heterogeneous Multistream Learning)

Let $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^n$ be a set of n concurrent data streams. Each stream \mathcal{S}_i is an ordered sequence of instances $\{(\mathbf{x}_{i,t}, y_{i,t})\}_{t=1}^{\infty}$, where $\mathbf{x}_{i,t} \in \mathcal{X}_i \subseteq \mathbb{R}^{D_i}$ is the feature

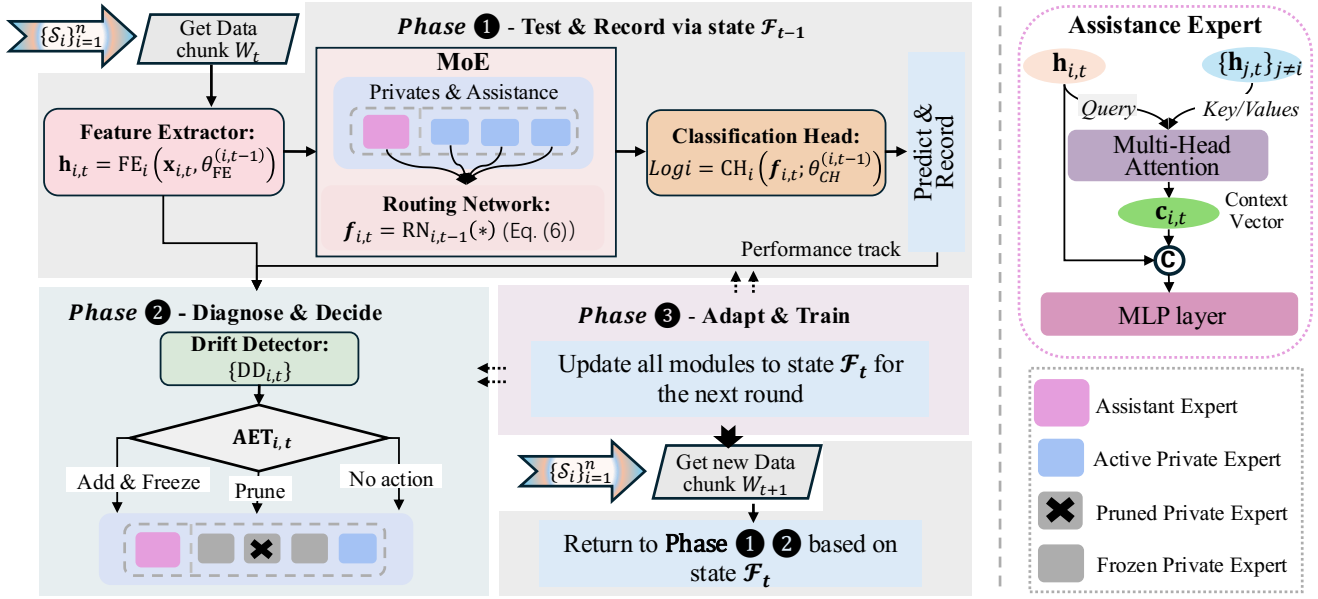


Figure 1: The overall framework of **CAMEL**. Concretely, each stream’s MoE module leverages a dynamic pool of private experts and a dedicated *assistance expert* that performs collaborative fusion via multi-head attention. The entire system follows a Test-Diagnose-Adapt cycle where an Autonomous Expert Tuner (AET) dynamically manages the expert lifecycle (adding/freezing/pruning) in response to drift and performance signals, ensuring continuous adaptation in the HML scenario.

vector from a stream-specific feature space of dimensionality D_i , and $y_{i,t} \in \mathcal{Y}_i = \{1, \dots, C_i\}$ is the corresponding class label from a stream-specific label space of size C_i . The underlying joint distribution $P_t^{(i)}(\mathbf{x}_{i,t}, y_{i,t})$ for each stream S_i can change over time, exhibiting concept drift. The goal in HML is to design an adaptive mechanism $\mathcal{F} : \{\mathcal{X}_i \rightarrow \mathcal{Y}_i\}_{i=1}^n$ that continuously adapts to predict new data from each stream.

As mentioned before, three main challenges must be addressed simultaneously in HML, i.e., *Intrinsic Heterogeneity*, *Knowledge Fusion* and *Asynchronous Drifts*. These challenges are defined as follows,

Challenge 1 (Intrinsic Heterogeneity) *Real-world multi-stream scenarios exhibit intrinsic heterogeneity in both feature and label spaces across streams. For any pair of streams S_i and S_j ($i \neq j$), their respective feature spaces may differ in dimensionality ($D_i \neq D_j$) and attribute structure ($\mathcal{X}_i \neq \mathcal{X}_j$), while their label spaces can define disjoint predictive tasks ($\mathcal{Y}_i \neq \mathcal{Y}_j$ implying $C_i \neq C_j$).*

Challenge 2 (Knowledge Fusion) *While the streams S are heterogeneous, they may contain latent time-varying correlations that can be exploited for mutual benefit. The core challenge is to design a mechanism for selective and adaptive knowledge fusion. This requires simultaneously achieving two conflicting objectives for any given stream S_i . First, the model must be able to identify and leverage useful contextual information from all other concurrent streams $\{S_j\}_{j \neq i}$ to enhance its predictive capability for S_i . Second, it must be robust to dynamically ignore information from any stream S_j that is irrelevant or contains misleading patterns,*

thereby avoiding negative transfer.

Challenge 3 (Asynchronous Drifts) *The non-stationarity of each stream S_i presents that its data-generating distribution $P_t^{(i)}$ evolves with uncoordinated and diverse dynamics. These concept drifts are both asynchronous and diverse. Formally, for any two streams S_i and S_j (where $i \neq j$), $\exists t, P_t^{(i)}(y|\mathbf{x}) \neq P_{t+1}^{(i)}(y|\mathbf{x})$ while $P_t^{(j)}(y|\mathbf{x}) = P_{t+1}^{(j)}(y|\mathbf{x})$. Furthermore, the drift patterns vary across streams in type (e.g., sudden, gradual, incremental).*

Methodology

We present the **CAMEL** framework to address the three fundamental challenges in HML. The core innovation lies in a drift-aware autonomous architecture that combines stream-specific specialization with cross-stream collaboration.

Overview of CAMEL

As shown in Figure 1, we introduce **CAMEL**, a framework designed to learn a generalized model \mathcal{F} by processing n data streams \mathcal{S} in a window-based prequential manner. It features a modular architecture where each stream S_i is assigned a dedicated learning system, including:

- A stream-specific Feature Extractor (FE_i) for dimensionality and feature space alignment.
- A Mixture of Experts (MoE) core, which includes a dynamic pool of Private Experts (PE_i), a dedicated Assistance Expert (AE_i), and a Routing Network (RN_i).
- A task-specific Classification Head (CH_i) for handling heterogeneous label spaces.

- A control loop comprising a Drift Detector (DD_{*i*}) and an Autonomous Expert Tuner (AET_{*i*}) for online adaptation.

The online learning process begins with an initial model trained on the first window W_0 . Subsequently, for each incoming data window W_t ($t \geq 1$), the system executes a **Test-Diagnose-Adapt** cycle, and the whole process is summarized in Algorithm 1.

Phase ① – Test and Record. The cycle begins by evaluating the current model state \mathcal{F}_{t-1} (trained on W_{t-1}) on the new data of window W_t . For each instance $(\mathbf{x}_{i,t}, y_{i,t})$, its feature vector $\mathbf{x}_{i,t}$ is first projected by FE_{*i,t-1*} to an aligned representation $\mathbf{h}_{i,t}$. It is then processed by the dynamic MoE. Concretely, the RN_{*i,t-1*} computes routing weights to combine outputs from the PE_{*i*}($t-1$) pool, which captures idiosyncratic patterns, and the AE_{*i,t-1*}, which performs collaborative fusion by attending to features $\{\mathbf{h}_{j,t}\}_{j \neq i}$ from all other streams. The resulting integrated feature vector is finally passed to the task-specific CH_{*i,t-1*} to produce a prediction $\hat{y}_{i,t}$. The performance (e.g., accuracy) against the true label $y_{i,t}$ is then recorded for the subsequent phase.

Phase ② – Diagnose and Decide. Following the test, the system diagnoses the state of each stream by the drift detector DD_{*i*}. It analyzes the distribution of features $\{\mathbf{h}_{i,t}\}$ from W_t to detect drift in $P_t^{(i)}$. Concurrently, an autonomous expert tuner AET_{*i,t*} evaluates the performance metrics from the test phase and the long-term utilization statistics of its experts. Based on these evidences, i.e., the drift signal and performance analysis, the AET_{*i,t*} makes an adaptation decision: it may expand the private expert pool PE_{*i*}(t) by adding a new expert to learn an emerging concept, or prune an underutilized expert to maintain model parsimony.

Phase ③ – Adapt and Train. Finally, the model architecture is updated based on the decisions from the diagnosis phase. The potentially modified model is then trained on the data from window W_t via an end-to-end process. The total loss aggregated from all stream-specific classification heads is back-propagated through the entire network. This step refines the parameters of all active components, preparing the system for the next window W_{t+1} . This cyclical process allows **CAMEL** to continuously learn, adapt, and specialize in a non-stationary multistream environment.

Heterogeneity-Aware Representation

To handle the *intrinsic heterogeneity*, i.e., *Challenge 1*, our framework employs a hierarchical approach involving feature-level alignment and task-level specialization.

1. Feature Alignment. First, to address the feature space heterogeneity ($\mathcal{X}_i \neq \mathcal{X}_j$), each stream \mathcal{S}_i is assigned a dedicated feature extractor FE_{*i*}. This is a neural network, parameterized by $\theta_{\text{FE}}^{(i,t)}$, whose architecture is tailored to the input dimensionality D_i . Its primary function is to project the raw feature vector $\mathbf{x}_{i,t}$ into a common latent space $\mathcal{H} \subset \mathbb{R}^{D_h}$:

$$\mathbf{h}_{i,t} = \text{FE}_i(\mathbf{x}_{i,t}, \theta_{\text{FE}}^{(i,t)}), i \in [1, n]; \quad (1)$$

This explicit dimensionality alignment creates a standardized input format for all subsequent expert networks, forming the foundation for inter-stream knowledge fusion.

Algorithm 1 CAMEL: Online Learning Process

Require: Data streams $\{\mathcal{S}_i\}_{i=1}^n$, Window size $|W_i|$, Total windows T_{max} .
Ensure: Predicted labels.

- 1: % Initial training on the first window W_0
- 2: $W_0 \in \{\mathbf{X}_{i,0}, Y_{i,0}\}_{i=1}^n \leftarrow \text{GetData}(\{\mathcal{S}_i\}_{i=1}^n, |W_i|, 0)$.
- 3: $\mathcal{F}_0 \leftarrow \text{Train}(W_0)$.
- % Test-then-Adapt loop for subsequent windows.
- 4: **for** $t = 1 : T_{max}$ **do**
- 5: $W_t \in \{\mathbf{X}_{i,t}, Y_{i,t}\}_{i=1}^n \leftarrow \text{GetWindow}(\{\mathcal{S}_i\}_{i=1}^n, l, t)$
 % Phase ① – Test & performance record.
- 6: $\{\text{Perf}_{i,t}\}_{i=1}^n \leftarrow \text{Test}(\mathcal{F}_{t-1}, W_t)$
- 7: Update $\{\text{AET}_{i,t}\}$ with stream-specific performances $\{\text{Perf}_{i,t}\}$.
 % Phase ② – Diagnose & Decide.
- 8: **for** $i = 1 : n$ **do**
- 9: drift_signal_{*i,t*} $\leftarrow \text{DD}_{i,t}.\text{update}(H_{i,t})$
- 10: (action_{*i,t*}, pe_id_{*i,t*}) \leftarrow
 AET_{*i,t*}(drift_signal_{*i,t*}, Perf_{*i,t*})
- 11: **if** action_{*i,t*} is "ADD_PRIVATE" **then**
- 12: Add a new private expert to PE_{*i*}(t);
- 13: Freeze old private experts;
- 14: **else if** action_{*i,t*} is "PRUNE_PRIVATE" **then**
- 15: Prune Private Experts PE_{*i*}(t)[pe_id]
- 16: **end if**
- 17: **end for**
 % Phase ③ – Adapt & Train.
- 18: $\mathcal{F}_t \leftarrow \text{Train}(W_t)$
- 19: **end for**

2. Task Specialization. Second, to address the label space heterogeneity ($\mathcal{Y}_i \neq \mathcal{Y}_j$), our framework adopts a multi-task learning paradigm. Each stream \mathcal{S}_i is equipped with an independent task-specific classification head CH_{*i*} parameterized by $\theta_{\text{CH}}^{(i,t)}$. It is responsible for mapping the final refined feature representation $\mathbf{f}_{i,t}$ (derived by Eq. (6)) to the stream's unique label space \mathcal{Y}_i :

$$\text{Logits}_{i,t} = \text{CH}_i(\mathbf{f}_{i,t}; \theta_{\text{CH}}^{(i,t)}) \in \mathbb{R}^{C_i}, i \in [1, n]; \quad (2)$$

This architecture ensures that the final decision-making process is tailored to each stream's specific predictive task, whether it is binary classification or multi-class classification with a different number of classes.

Adaptive Knowledge Fusion

To address the *Knowledge Fusion*, i.e., *Challenge 2*, **CAMEL** introduces a novel dynamic MoE architecture with collaborative assistance designed to exploit inter-stream correlations while mitigating negative transfer.

1. Private Experts: capturing stream-specific knowledge. For each stream \mathcal{S}_i , we maintain a dynamic pool of private experts PE_{*i*}(t) = $\{\text{pe}_{i,j} | j = 1, \dots, K_i(t)\}$. Each expert pe_{*i,j*} is an MLP parameterized by $\theta_{\text{pe}}^{(i,j,t)}$, that learns patterns idiosyncratic to stream \mathcal{S}_i . It processes the aligned feature $\mathbf{h}_{i,t}$ to produce representations in a common expert output space $\mathcal{E} \subset \mathbb{R}^{D_f}$:

$$\mathbf{f}_{i,j,t}^{\text{pe}} = \text{pe}_{i,j}(\mathbf{h}_{i,t}; \theta_{\text{pe}}^{(i,j,t)}) \quad (3)$$

2. *Assistance Experts: collaborative knowledge fusion.* Each stream S_i is paired with a dedicated assistance expert AE_i parameterized by $\theta_{\text{AE}}^{(i,t)}$. This expert’s unique role is to perform collaborative knowledge fusion. It takes the target stream’s feature $\mathbf{h}_{i,t}$ as a *query* and leverages features from all other concurrent streams $\{\mathbf{h}_{j,t}\}_{j \neq i}$ as *context* (keys and values) (Vaswani et al. 2017; Zhang et al. 2025b). We employ a multi-head attention mechanism:

$$\mathbf{c}_{i,t} = \text{Attention}(\mathbf{h}_{i,t}, \{\mathbf{h}_{j,t}\}_{j \neq i}) \quad (4)$$

The resulting context vector $\mathbf{c}_{i,t} \in \mathbb{R}^{D_h}$ is a weighted summary of information from other streams, where the weights are learned based on relevance to $\mathbf{h}_{i,t}$. This contextual information is then fused with the input features to produce the assistance expert’s output representations:

$$\mathbf{f}_{i,t}^{\text{AE}} = \text{MLP}_{\text{AE}}^{(i,t)}(\text{Concat}(\mathbf{h}_{i,t}, \mathbf{c}_{i,t}); \theta_{\text{AE}}^{(i,t)}) \in \mathbb{R}^{D_f} \quad (5)$$

This end-to-end mechanism allows AE_i to learn *what* information to transfer from other streams and *how* to use it to best serve stream S_i .

3. *Routing and feature integration.* A stream-specific routing network RN_i parameterized by $\theta_{\text{RN}}^{(i,t)}$ determines the credibility of each expert for a given input $\mathbf{h}_{i,t}$. It outputs a probability distribution $\mathbf{p}_{i,t}$ over the $K_i(t)$ private experts and the assistance expert. The final refined representations $\mathbf{f}_{i,t}$ for stream S_i are a weighted combination of all expert outputs:

$$\mathbf{f}_{i,t} = \mathbf{p}_{i,t}[\text{AE}_i] \cdot \mathbf{f}_{i,t}^{\text{AE}} + \sum_{j=1}^{K_i(t)} \mathbf{p}_{i,t}[\text{pe}_{i,j}] \cdot \mathbf{f}_{i,j}^{\text{pe}} \quad (6)$$

The routing mechanism provides a natural defense against negative transfer as it can learn to assign a near-zero weight to the assistance expert if the external context is irrelevant or even harmful.

Drift Detection & Adaptation

Our framework’s autonomy and ability to handle *asynchronous drifts* (*Challenge 3*) stem from a per-stream control loop involving a drift detector and an expert tuner.

1. *Drift Detection.* Each stream S_i is independently monitored by a Maximum Mean Discrepancy (MMD) based drift detector DD_i (Wan, Liang, and Yoon 2024). DD_i maintains a reference window $W_{i,t}^{\text{ref}}$ of past features \mathbf{h}_s and compares it with the features from the current window $W_{i,t}$.

$$\text{MMD}_{i,t}^2(W_{i,t}, W_{i,t}^{\text{ref}}) = \left\| \frac{1}{|W_{i,t}|} \sum_{\mathbf{h}_{i,t} \in W_{i,t}} \phi(\mathbf{h}_{i,t}) - \frac{1}{|W_{i,t}^{\text{ref}}|} \sum_{\mathbf{h}_{j,t} \in W_{i,t}^{\text{ref}}} \phi(\mathbf{h}_{j,t}) \right\|_{\mathcal{H}_k}^2 \quad (7)$$

where ϕ is a mapping to a Reproducing Kernel Hilbert Space \mathcal{H}_k induced by a kernel (Smola et al. 2007). If $\text{MMD}_{i,t}^2 > \tau_{\text{MMD}_i}$, DD_i signals a drift for stream S_i . The reference window $W_{i,t}^{\text{ref}}$ is then updated with $W_{i,t}$.

2. *Autonomous Expert Tuner.* To achieve robust and efficient adaptation, our framework employs an Autonomous Expert

Tuner (AET_i) that governs the lifecycle of private experts for each stream S_i . Relying solely on distribution-based drift detection (DD_i) can be suboptimal, as not all statistical shifts necessarily degrade predictive performance (Lu et al. 2018), which could lead to unnecessary and costly model adaptations. Conversely, some performance degradation might occur without a detectable distribution shift in the feature space. Therefore, the AET_i integrates two complementary signals, i.e., the drift signal from DD_i and the stream’s recent test performance. This expert-level plasticity is the core mechanism for adapting model capacity online:

- **Expert Adding:** A new private expert is added to the pool PE_i only when a drift is detected by DD_i **and** the stream’s test performance $\text{Perf}_{i,t}$ exhibits a significant degradation. This conjunctive condition ensures that the model only expands its capacity when there is clear evidence of a detrimental concept change. The new expert is initialized as trainable to learn the emerging concept, while all existing private experts in PE_i are frozen to prevent catastrophic forgetting, thereby preserving knowledge of past concepts.
- **Expert Pruning:** A private expert $\text{pe}_{i,j}$ (whether frozen or active) is pruned from PE_i if its long-term average utilization, determined by the routing weights from RN_i , falls below a threshold τ_{util} . This proactive mechanism removes irrelevant experts that no longer contribute to the stream’s predictions, maintaining model parsimony and preventing the accumulation of obsolete components.

Since each AET_i operates independently based on its stream’s specific signals, the framework naturally handles asynchronous drifts.

Learning Objective

This method is trained end-to-end. For a given data window W_t , the total loss is the sum of the individual cross-entropy losses from each stream-specific classification head:

$$L_{\text{total}}(W_t) = \sum_{i=1}^n \mathbb{E}_{(\mathbf{x}_{i,j}, y_{i,j}) \in W_{t,i}} [\mathcal{L}_{\text{CE}}(\text{CH}_i(\mathbf{f}_{i,t}), y_{i,t})], \quad (8)$$

Theoretical Analysis

The design of our method is theoretically grounded in multi-task learning principles (Maurer, Pontil, and Romera-Paredes 2016), which demonstrates that jointly learning related tasks can yield superior generalization over isolated learning. Our collaborative assistance mechanism enables intelligent knowledge fusion while mitigating negative transfer, and can be formally justified by:

Theorem 1 (Generalization Bound) *Let \mathcal{F} be the hypothesis space defined by the CAMEL architecture, for any hypothesis $h \in \mathcal{F}$ trained on streams $\mathcal{S} = \{S_i\}_{i=1}^n$, the expected risk $\mathcal{R}_i(h)$ on any stream S_i is bounded as:*

$$\mathcal{R}_i(h) \leq \hat{\mathcal{R}}_{\text{avg}}(h) + C(\{\mathcal{S}_j\}_{j=1}^n) + \mathcal{O}\left(\sqrt{\frac{\log(n|W_t|)}{n|W_t|}}\right) \quad (9)$$

where $\hat{\mathcal{R}}_{\text{avg}}(h) = \frac{1}{n} \sum_{j=1}^n \hat{\mathcal{R}}_j(h)$ is the average empirical risk across all streams and $C(\{\mathcal{S}_j\})$ quantifies the inter-stream dissimilarity. A proof sketch is in Appendix A.

Synthetic	Set 1: Tree (Homo.)				Set 2 Hyperplane (Homo.)				Set 3 (Hete.)				Set 4 (Hete.)			
	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg	SEAA	RTG	RBF	avg	LED	LEDDri	Wave	avg
SRP	58.47	65.14	64.63	62.74	86.37	87.59	88.21	87.39	83.35	70.05	81.18	78.19	35.18	36.65	83.80	51.88
AMF	56.18	63.76	59.59	59.84	91.32	90.70	90.70	90.91	83.65	66.19	90.29	80.04	37.85	25.31	79.39	47.52
IWE	63.49	72.35	68.39	68.07	89.82	91.39	90.90	90.70	84.27	64.38	70.12	72.92	36.05	34.15	80.41	50.20
MCMO	64.77	67.29	66.32	66.13	82.21	85.37	85.12	84.23	-	-	-	-	-	-	-	-
OBAL	65.72	67.97	65.60	66.43	84.14	86.73	88.66	86.51	-	-	-	-	-	-	-	-
BFSRL	63.37	67.42	64.39	65.06	84.67	87.20	88.47	86.78	-	-	-	-	-	-	-	-
CAMEL	65.78	68.27	66.48	66.84	91.85	92.12	91.84	91.94	85.14	67.73	92.75	81.87	38.19	35.36	85.43	53.00

Real-World	Set 5: TV News (Homo.)				Set 6: Weather (Homo.)				Set 7: Credit card (Hete.)				Set 8: CoverT. (Hete.)			
	CNN	BBC	TIMES	avg	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg
SRP	78.46	75.55	80.84	78.28	81.46	77.45	78.15	79.02	77.81	82.01	78.04	79.29	87.21	52.99	56.36	65.52
AMF	79.25	79.49	78.70	79.15	81.37	75.70	77.91	78.33	77.86	81.40	77.88	78.39	86.15	53.62	61.75	67.17
IWE	78.66	74.42	77.54	76.87	80.40	76.24	74.91	77.18	75.89	80.15	75.92	77.32	72.58	51.52	51.75	58.62
MCMO	68.83	60.12	59.74	62.90	75.11	75.02	73.37	74.50	-	-	-	-	-	-	-	-
OBAL	67.72	59.39	64.42	63.84	77.46	74.35	76.21	75.97	-	-	-	-	-	-	-	-
BFSRL	60.18	55.09	61.29	59.12	74.77	74.09	75.42	74.76	-	-	-	-	-	-	-	-
CAMEL	80.06	79.66	80.90	80.21	82.04	78.33	79.39	79.92	80.42	81.93	80.37	80.91	86.97	62.91	82.22	77.37

Table 1: Classification accuracy (%) of various methods on all benchmarks. The best and second-best results are highlighted in red and blue respectively. "-" means it is not applicable to the task.

Implication 1 *Theorem 1 formally justifies CAMEL’s architecture: The assistance expert (\mathcal{AE}_i) minimizes $C(\{\mathcal{S}_i\}_{i=1}^n)$ through attention-based knowledge transfer, while the routing network (\mathcal{RN}_i) dynamically balances this against stream-specific private experts (\mathcal{PE}_i) to prevent negative transfer when dissimilarity is high. This intrinsic collaboration-specialization tradeoff combined with joint training’s sample efficiency ($\mathcal{O}(1/\sqrt{n|W_t|})$) explains the empirical robustness. The autonomous expert tuner (AET) maintains adaptability to concept drift across windows through expert-level plasticity.*

Experiments

In experiments, we first assess the framework’s *generality and robustness* across both homogeneous and heterogeneous settings. Second, we provide a qualitative analysis of the *online adaptation process* visualizing how the AET dynamically manages the private expert pool to concept drifts. Finally, we perform an ablation study to dissect the contribution of each core component, thereby validating our fundamental design principles. More detailed analysis and supplementary experiments can be seen in Appendix C.

Experiment Settings

Benchmarks. We establish eight diverse multistream scenarios. The first four scenarios are constructed from twelve synthetic data streams, meticulously designed to isolate specific challenges: homogeneous (Set 1 & 2) and heterogeneous (Set 3) feature spaces, and heterogeneous label spaces (Set 4). In addition, we employ four real-world multistream datasets, which inherently exhibit a mix of homogeneous and heterogeneous characteristics (Set 5-8). More detailed descriptions can be found in Appendix B.1.

Baselines. We conduct a comparison against six SOTA methods, including 1) *Single-stream learning*: SRP (Gomes, Read, and Bifet 2019), AMF (Mourtada, Gaïffas, and Scornet 2021) and IWE (Jiao et al. 2024); 2) *Multistream classification*: MCMO (Jiao et al. 2023), OBAL (Yu et al. 2024) and BFSRL (Yu, Lu, and Zhang 2024). The detailed description and implementation are provided in Appendix B.2 & B.3.

Results Analysis

Overall Performance. Table 1 demonstrates that **CAMEL** consistently achieves SOTA average accuracy across almost all scenarios except for Set 1, validating its strong generality and robustness. The framework’s primary strength lies in its effective handling of the *Intrinsic Heterogeneity (Challenge 1)*. Unlike contemporary multistream methods (MCMO, OBAL, BFSRL) which are confined to homogeneous settings and thus not applicable to our more realistic heterogeneous scenarios, our method thrives in these complex environments. This is enabled by its stream-specific modules (\mathcal{FE}_i , \mathcal{CH}_i), which provide the necessary specialization for each stream. Furthermore, compared against single-stream methods (SRP, AMF, IWE), **CAMEL**’s consistent top-tier performance validates its novel approach to the *Knowledge Fusion (Challenge 2)*. While single-stream methods operate in isolation, our collaborative assistance mechanism successfully leverages latent inter-stream correlations. The attention-based experts perform targeted knowledge transfer, boosting the overall system performance. This dynamic interplay between specialized private experts managed by the \mathcal{AET}_i to address *Asynchronous Drifts (Challenge 3)*, and the collaborative assistance experts allows it to strike a robust balance between focused learning and knowledge fusion. Consequently, our method excels across the full spectrum of HML challenges, proving its capability as a general and powerful solution for

Variants	Set 3				Set 4				Set 6: Weather				Set 7: Credit Card			
	SEAA	RTG	RBF	avg	LED	LEDDri	Wave	avg	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg	\mathcal{S}_1	\mathcal{S}_2	\mathcal{S}_3	avg
Base	80.28	64.37	81.42	75.36	29.21	21.96	76.94	42.70	74.32	73.17	73.21	73.57	74.29	77.24	74.02	75.18
Base+I	83.32	66.10	87.27	78.90	37.31	34.13	83.22	51.55	76.22	77.04	76.78	76.68	77.92	76.58	76.74	77.08
Base+I+DP	84.84	66.17	89.16	80.06	37.23	34.30	82.97	51.50	79.74	77.67	78.01	78.47	78.63	82.07	79.09	79.93
CAMEL	85.14	67.73	92.75	81.87	38.19	35.36	85.43	53.00	82.04	78.33	79.39	79.92	80.42	81.93	80.37	80.91

Table 2: Ablation study. Classification accuracy (%) of **CAMEL**'s variants. The best and second-best results are highlighted in red and blue, respectively.

diverse and evolving multistream environments.

Online Performance. Figure 2 qualitatively analyzes **CAMEL**'s online adaptation, plotting per-stream accuracy against the number of private experts. The results illustrate the 'drift-diagnose-adapt' narrative and validate the Autonomous Expert Tuner (AET). For example, in Figure 2a, Stream 1 exhibits an accuracy dip at window 15, indicating concept drift. The AET correctly diagnoses this and responds by instantiating a new private expert, increasing model capacity and enabling swift performance recovery. Once the new concept is learned, the redundant expert is pruned (around window 20) to maintain model parsimony. Conversely, Stream 2 (without significant drift) demonstrates AET's robustness: despite accuracy fluctuations, the private expert count remains constant, showing it avoids overreacting to inherent data noise (similar to Figure 2b). These behaviors highlight that **CAMEL**'s adaptation is highly selective, providing architectural plasticity precisely when and where needed to autonomously maintain high performance amidst asynchronous concept drifts. Additional visualizations are in Appendix C.1.

Ablation Study. To dissect component contributions, our ablation study progressively constructs the full **CAMEL** framework (Table 2) validating core design principles. Transitioning from the naive **Base** (full retraining) to **Base+I** (incremental learning) yields significant gains, confirming that continuous fine-tuning mitigates catastrophic forgetting. Integrating the Autonomous Expert Tuner (**Base+I+DP**) further improves performance on drifting streams (e.g., RBF: +1.89%), demonstrating expert-level plasticity effectively addresses *Asynchronous Drifts* (*Challenge 3*). The full **CAMEL** framework with collaborative assistance delivers the most substantial improvement, which empirically shows the attention-based mechanism masters *Knowledge Fusion* (*Challenge 2*) by distilling cross-stream knowledge for superior HML generalization.

Conclusion & Limitation

In this paper, we introduced **CAMEL**, a novel autonomous Mixture of Experts framework designed to robustly handle the complexities of multistream learning. By assigning each stream a dynamic ensemble of specialized private experts alongside a dedicated collaborative assistance expert, our method effectively addresses intrinsic heterogeneity and facilitates adaptive knowledge fusion. In addition, an autonomous tuner manages the expert lifecycle at a modular

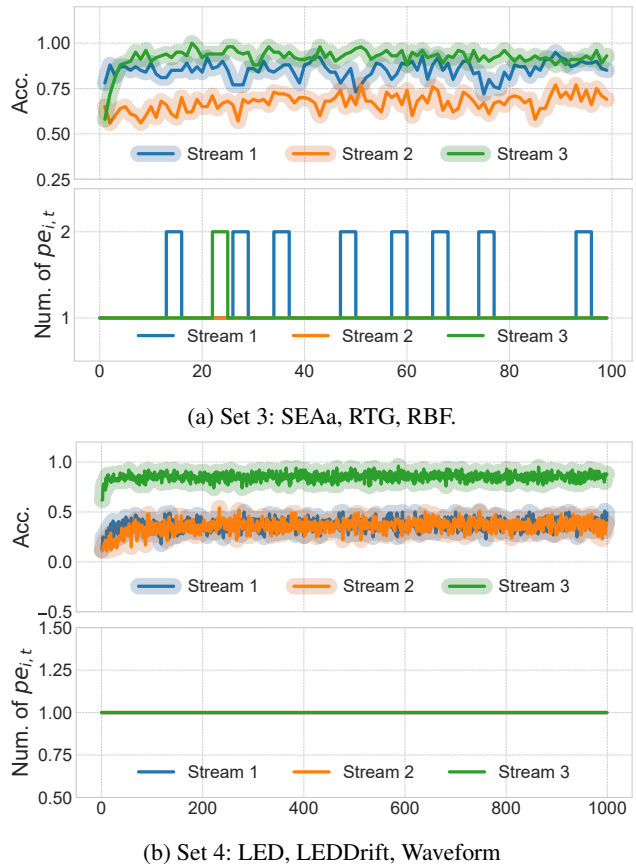


Figure 2: Online accuracy and the corresponding number of private experts over time.

level and allows our method to dynamically adapt to concept drifts. A generalization bound based on multi-task learning theory formally connects inter-stream relatedness and routing decisions with task-level risk. Empirical results on diverse synthetic and real-world multistream settings demonstrate the superiority under HML challenges.

Limitations include suboptimal handling of recurring concepts through expert freezing and computational overhead from dynamic architecture adaptation. Future work will explore expert reactivation strategies and efficiency optimizations for resource-constrained environments.

Acknowledgments

The work was supported by the Australian Research Council (ARC) under Laureate project FL190100149 and Discovery Project DP220102635.

References

- Agrahari, S.; and Singh, A. K. 2022. Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 34(10): 9523–9540.
- Bifet, A.; and Gavaldà, R. 2007. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, 443–448. SIAM.
- Brzezinski, D.; and Stefanowski, J. 2013. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE transactions on neural networks and learning systems*, 25(1): 81–94.
- Cacciarelli, D.; and Kulahci, M. 2024. Active learning for data streams: a survey. *Machine Learning*, 113(1): 185–239.
- Gomes, H. M.; Read, J.; and Bifet, A. 2019. Streaming random patches for evolving data stream classification. In *2019 IEEE International Conference on Data Mining*, 240–249. IEEE.
- Guo, H.; Zhang, S.; and Wang, W. 2021. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift. *Neural Networks*, 142: 437–456.
- Jiao, B.; Guo, Y.; Yang, C.; Pu, J.; Zheng, Z.; and Gong, D. 2024. Incremental Weighted Ensemble for Data Streams with Concept Drift. *IEEE Transactions on Artificial Intelligence*, 5(01): 92–103.
- Jiao, B.; Guo, Y.; Yang, S.; Pu, J.; and Gong, D. 2023. Reduced-Space Multistream Classification Based on Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 27(4): 764–777.
- Kim, M.; Hwang, S.-H.; and Whang, S. E. 2024. Quilt: robust data segment selection against concept drifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 21249–21257.
- Korycki, Ł.; and Krawczyk, B. 2021. Concept drift detection from multi-class imbalanced data streams. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1068–1079. IEEE.
- Lei, T.; Chen, S.; Wang, B.; Jiang, Z.; and Zou, N. 2024. Adapted-moe: Mixture of experts with test-time adaption for anomaly detection. *arXiv preprint arXiv:2409.05611*.
- Li, H.; Lin, S.; Duan, L.; Liang, Y.; and Shroff, N. 2025. Theory on Mixture-of-Experts in Continual Learning. In *The Thirteenth International Conference on Learning Representations*.
- Li, W.; Yang, X.; Liu, W.; Xia, Y.; and Bian, J. 2022. Dgda: Data distribution generation for predictable concept drift adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4092–4100.
- Liang, D. 2025. DistPred: A Distribution-Free Probabilistic Inference Method for Regression and Forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 753–764.
- Liang, D.; Chen, J.; Wang, X.; Wang, Y.; and Li, S. 2025. DeepBooTS: Dual-Stream Residual Boosting for Drift-Resilient Time-Series Forecasting. *arXiv preprint arXiv:2511.06893*.
- Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2018. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12): 2346–2363.
- Lu, P.; Lu, J.; Liu, A.; Yu, E.; and Zhang, G. 2025a. Autonomous Concept Drift Threshold Determination. *arXiv preprint arXiv:2511.09953*.
- Lu, P.; Lu, J.; Liu, A.; and Zhang, G. 2025b. Early Concept Drift Detection via Prediction Uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 19124–19132.
- Ma, G.; Lu, J.; Fang, Z.; Liu, F.; and Zhang, G. 2024. Multiview classification through learning from interval-valued data. *IEEE Transactions on Neural Networks and Learning Systems*.
- Marcu, O.-C.; and Bouvry, P. 2024. *Big data stream processing*. Ph.D. thesis, University of Luxembourg.
- Maurer, A.; Pontil, M.; and Romera-Paredes, B. 2016. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81): 1–32.
- Mourtada, J.; Gaïffas, S.; and Scornet, E. 2021. AMF: Aggregated Mondrian forests for online learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3): 505–533.
- Mu, S.; and Lin, S. 2025. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*.
- Panchal, K.; Choudhary, S.; Mitra, S.; Mukherjee, K.; Sarkhel, S.; Mitra, S.; and Guan, H. 2023. Flash: Concept drift adaptation in federated learning. In *International Conference on Machine Learning*, 26931–26962. PMLR.
- Qin, Z.; Cheng, Y.; Zhao, Z.; Chen, Z.; Metzler, D.; and Qin, J. 2020. Multitask mixture of sequential experts for user activity streams. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 3083–3091.
- Read, J.; and Zliobaite, I. 2025. Supervised Learning from Data Streams: An Overview and Update. *ACM Computing Surveys*.
- Sarkar, R.; Liang, H.; Fan, Z.; Wang, Z.; and Hao, C. 2023. Edge-moe: Memory-efficient multi-task vision transformer architecture with task-level sparsity via mixture-of-experts. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 01–09. IEEE.
- Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A Hilbert space embedding for distributions. In *International conference on algorithmic learning theory*, 13–31. Springer.

- Tran, V.-T.; Pham, Q.-V.; et al. 2025. Revisiting Sparse Mixture of Experts for Resource-adaptive Federated Fine-tuning Foundation Models. In *ICLR 2025 Workshop on Modularity for Collaborative, Decentralized, and Continual Deep Learning*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Wan, K.; Liang, Y.; and Yoon, S. 2024. Online drift detection with maximum concept discrepancy. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2924–2935.
- Wang, K.; Lu, J.; Liu, A.; and Zhang, G. 2024. An Adaptive Stacking Method for Multiple Data Streams Learning under Concept Drift. In *The 19th ISKE Conference on Intelligence Systems and Knowledge Engineering (FLINS-ISKE 2024)*, 267–274. World Scientific.
- Wang, K.; Lu, J.; Liu, A.; Zhang, G.; and Xiong, L. 2021. Evolving gradient boost: A pruning scheme based on loss improvement ratio for learning under concept drift. *IEEE Transactions on Cybernetics*, 53(4): 2110–2123.
- Wen, Q.; Chen, W.; Sun, L.; Zhang, Z.; Wang, L.; Jin, R.; Tan, T.; et al. 2023. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. *Advances in Neural Information Processing Systems*, 36: 69949–69980.
- Xiang, Q.; Zi, L.; Cong, X.; and Wang, Y. 2023. Concept drift adaptation methods under the deep learning framework: A literature review. *Applied Sciences*, 13(11): 6515.
- Xu, K.; Chen, L.; and Wang, S. 2025. Coral: Concept drift representation learning for co-evolving time-series. *arXiv preprint arXiv:2501.01480*.
- Yang, M.; Lin, S.; Li, C.; and Chang, X. 2025. Let LLM Tell What to Prune and How Much to Prune. In *Forty-second International Conference on Machine Learning*.
- Yang, X.; Lu, J.; and Yu, E. 2025a. Adapting Multi-modal Large Language Model to Concept Drift From Pre-training Onwards. In *The Thirteenth International Conference on Learning Representations*.
- Yang, X.; Lu, J.; and Yu, E. 2025b. Walking the Tightrope: Autonomous Disentangling Beneficial and Detrimental Drifts in Non-Stationary Custom-Tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yu, E.; Lu, J.; Yang, X.; Zhang, G.; and Fang, Z. 2025. Learning Robust Spectral Dynamics for Temporal Domain Generalization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yu, E.; Lu, J.; Zhang, B.; and Zhang, G. 2024. Online boosting adaptive learning under concept drift for multistream classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16522–16530.
- Yu, E.; Lu, J.; and Zhang, G. 2024. Fuzzy Shared Representation Learning for Multistream Classification. *IEEE Transactions on Fuzzy Systems*, 32(10): 5625–5637.
- Zhang, B.; Lu, J.; Song, Y.; and Zhang, G. 2025a. A Multistream Concept Drift Handling Framework via Data Sharing. *IEEE Transactions on Cybernetics*.
- Zhang, T.; Yu, E.; Shao, Y.; and Sun, J. 2025b. Multimodal Inverse Attention Network with Intrinsic Discriminant Feature Exploitation for Fake News Detection. *arXiv preprint arXiv:2502.01699*.