ACMBP: An Adaptive Graph-Based Meta-Learning Framework for Cross-Modal Behavior Prediction from IoT and Social Media Streams

Xinzhu Pu[1], Yaqi Ren[2], Jing Liang[3*]
[1]School of Artificial Intelligence and Computer Science, North Large-scale heterogeneous digital ecosystems University of Technology, Beijing 100144, Large-scale heterogeneous digital ecosystems
[2]School of Humanities and Law, North Large-scale heterogeneous digital ecosystems University of Technology, Beijing 100144, Large-scale heterogeneous digital ecosystems
[3]Network and Educational Technology Center, Jinan University, Guangzhou, 510632, Large-scale heterogeneous digital ecosystems
E-mail: pu_xinzhu@126.com, renyaqi616@163.com, ljing@jnu.edu.cn
*Corresponding author

Keywords: internet of things, social media user behavior, adaptive cross-modal, graph networks, behavior prediction

Received: September 5, 2025

The growth of Internet of Things (IoT) devices and social media platforms generates large volumes of multimodal behavioral data that present both great opportunities and challenges to cross-domain behavior prediction. The current models do not deal with time misalignment, distributional drift, and personalization when only a limited amount of data is available. To address these issues, Adaptive Cross-Modal Behavioral Prediction (ACMBP), introduced in this study, is a new framework that merges the streams of IoT activity and social media characteristics via adaptive graph networks. ACMBP combines temporal-semantic offset attention to align activities by the IoT with social actions, drift-conscious dynamic graph rewiring to adapt to changing user relations, hierarchical cross-domain transfer to adapt across different platforms, and few-shot personalization with meta-learning and behavioral prototypes. ACMBP possibility in large-scale heterogeneous digital ecosystems is compared to a fused IoT social dataset (wearable sensors + Twitter) and two public multimodal benchmarks against 10 strong baselines (LSTM, GRU, BERT, GCN, GAT, HAN, GraphSAGE, DCRNN, DeepFM, Multimodal Transformer). ACMBP has been tested on a custom IoT–Social Fusion dataset (1,200 users, 90-day collection, 2.3M sensor records, 850K Twitter posts) and two public benchmarks (PAMAP2, USC-HAD). The results of all improvements are statistically significant (Wilcoxon signed-rank test, $p < 0.001$). ACMBP attains Behavioral Transition Accuracy (BTA) of 91.4%, Temporal Offset Prediction Error (TOPE) of 1.2 hours, Drift Detection Precision (DDP) of 84.7%, Cross-Platform Transfer Efficiency (CPTE) of 82.3%, and Few-Shot Adaptation only with five samples, which are improvements by 7.3%, 32%, 11.8%, 13.4%, and 50% respectively over the best baselines. Each module's contribution is supported by ablation studies, and the results of scalability experiments indicate stable performance in large-scale heterogeneous IoT-social environments.

Povzetek: Raziskava predstavlja nov okvir za napovedovanje vedenja, ki združuje podatke IoT naprav in družbenih omrežij ter z naprednimi prilagoditvenimi metodami dosega bistveno boljše rezultate kot obstoječi modeli.

1 Introduction

The rapid growth of Internet of Things (IoT) devices and ubiquitous social media platforms has redefined the way people engage with digital ecosystems in order to produce unprecedented amounts of behavioral data, which are indicators of an intricate human-technology relationship [1]. Users seamlessly navigate between IoT environments and social media platforms, generating complex behavioral patterns across multiple modalities and temporal scales [2]. The IoT devices, which have reached billions around the world, are constantly recording finer-grained behavioral signals such as sensor data, interaction behavior, time-based activity, as well as contextual environmental conditions [14]. At the same time, social media networks are core arenas of emotional expression, socializing, and behavioral expression, accommodating billions of users who produce enormous textual, graphic, and behavioral resources every single day [3]. Cross-domain interactions are strengthened in large-scale heterogeneous digital ecosystems due to the widespread use of smart devices and high engagement on social media. Nevertheless, the existing studies have mostly addressed these fields alone without leveraging the intrinsic cross-modal connections that may make a major

contribution to the accuracy of behavioral forecast and the potential of personalizations [4].

Conventional user behavior prediction methods have been mostly single-domain, either with temporal modeling of IoT-specific signals or content analysis of social media [5], [6]. On the other hand, social media behavior prediction is dependent on natural language processing, computer vision, and social network analysis to predict the posting patterns, content preferences, as well as engagement behaviors [7]. Although these unimodal approaches have proved to be successful in their respective fields, they are unable to capture the compound interdependencies of physical and digital behavioral expressions that represent the present-day interactions of the user [8].

The main constraint of current methods is the fact that they do not assess the dynamic, two-sided dependencies between IoT practices and social media practices [9]. Users' physical activities, which are recorded by smart home devices, wearables, and environmental sensors, usually come before and shape their digital manifestations on social networks [10]. As an example, the patterns of IoT activity in the late hours might be associated with the following social media posting patterns, whereas social interactions might reciprocally affect physical activity patterns and IoT device utilization [11]. Such a cross-modal behavioral cross-linkage offers a chance to improve predictive accuracy, yet demands a complex fusion mechanism that can manage temporal correlations, modality distortions, and user-specific trends as well [12].

Recent developments in multimodal learning have shown that it is possible to combine different sources of data to enhance prediction accuracy in a wide range of fields [13], [15], [18]. Nevertheless, there are special problems when it comes to the implementation of multimodal methods in the context of IoT-social behavioral prediction. To begin with, the time lag between IoT processes and social media activity poses difficult alignment challenges that cannot be effectively resolved using conventional fusion techniques. Second, the heterogeneity of IoT sensor data and social media content necessitates a set of special encoding methods that can maintain modality-related information and allow cross-modal interaction [16]. Third, individual users' behavioral patterns are thoroughly characterized by strong personalization and differ dramatically between demographics, preferences, and contexts, which thoroughly require more adaptive personalization mechanisms as well [17]. Graph neural networks have become a useful method to study complex relational networks and dynamics in social systems [8], [13], [25]. Conventional graph-based methods to analyze social media mainly rely on user-user interactions, content similarity, or influence propagation patterns [20]. Nevertheless, such approaches usually work in one domain and cannot reflect cross-modal interactions between IoT actions and social actions. Dynamic graph neural networks have been demonstrated to be able to deal with the temporal evolution of social networks, but the current models are not complex enough to perform cross-modal behavioral prediction with adaptive personalization abilities [4], [9].

Behavior drift is another important limitation of existing systems. The behaviors of users are constantly changing because of life conditions, seasonal trends, social factors, and the adoption of technology [19]. The current prediction models are usually based on static behavioral patterns, which do not adapt to these natural behavioral evolution processes, hence are unable to perform well with time [22]. Few-shot and meta-learning methods have demonstrated considerable potential in the area of personalization and adaptation in machine learning applications [1], [21]. Other important methods like Model-Agnostic Meta-Learning (MAML) allow adapting quickly to new users or a different behavioral pattern with little extra training data. The development of personality traits and psychological variables into computational behavior prediction models is a new area of research with significant potential to improve individualization [16]. As it has been proven by psychological research that personality dimensions might impact the physical activity patterns and social media usage, it is a important idea to note that personality-conscious models would offer reliable as well as interpretable predictions that can be deemed accurate [24]. However, the existence of personality-sensitive methods has been limited to individual modalities and is not sophisticated enough to apply in cross-modal situations [23].

Limitations of multimodal behavioral prediction can currently be divided into multiple areas. First, the current fusion methods are based mainly on the use of static concatenation or simple attention processes that cannot comprehend the dynamic value of various modalities among users and situations [11], [15]. Second, the temporal modeling methods are hard to adapt to different time domains and the asynchronicity of IoT and social media data [12], [17]. Third, personalization mechanisms are commonly inadequate to manage the complicated interaction of individual differences, behavioral development, and cross-modal contingencies [10], [24]. Fourth, current evaluation paradigms poorly reflect the specifics of cross-modal prediction and concentrate mostly on conventional measures of accuracy, which do not address the temporal, adaptive, and personalization

needs in practice [19]. There are a few major weaknesses in the effectiveness of existing behavioral prediction systems. The lack of common structures that can be used to process cross-modal relations between IoT and social media data forms a strong constraint [14], [17]. The current strategies are usually based on domain-specific preprocessing, feature engineering, and model architectures that fail to generalize across modalities and heterogeneous user populations [6], [18]. This is especially true in large-scale heterogeneous digital ecosystems, where the extent and diversity of IoT and social media platforms require models capable of generalizing well across different groupings of users and digital ecosystems. Also, the absence of standardized evaluation programs of cross-modal behavioral prediction complicates the comparison of various methods and the evaluation of their practical applicability to the real world [22].

Hierarchical modelling and advanced attention systems have performed well in limited areas but not sufficiently extended to cross-modal behavioural prediction contexts [5], [24]. Likewise, advanced graph neural network designs have proven to be useful in social media analysis and IoT applications on their own, but their combination into cross-modal prediction is not investigated much [8], [25]. Moreover, misinformation and adversarial behaviors within the cross-platform situation are an extra level of complexity that the functionality of current frameworks cannot tackle at the moment [20].

ACMBP (Adaptive Cross-Modal Behavioral Recognition with Graph Networks) is presented in this study as a new framework tailored to predicting the behavior of IoT-social users. The multiple innovative elements are incorporated in the approach to overcome the challenges discussed in the existing literature. In the first place, an adaptive cross-modal attention mechanism is presented in this study, by which the weight of modalities in various situations is dynamically adjusted according to the user and dynamic conditions [15], [18]. The mechanism goes beyond the classic attention in that it has behavioral drift detection and adaptation functionalities, inspired by new developments in dynamic graph neural networks [4], [9]. The framework is especially applicable to large and heterogeneous settings, such as those found in large-scale heterogeneous digital ecosystems, where the behavior of users and platform dynamics change quickly and vary greatly across regions and demographic segments. Second, an advanced graph neural network architecture was created, which captures intra-modal and cross-modal correlations via non-homogeneous graph structures [13], [25]. To maintain a high level of robustness in real-world deployment settings, the architecture integrates the latest research on malicious user detection and misinformation management [8], [20]. Third, the study used a personalisation system based on meta-learning that can quickly adjust to new users and changing behavioural patterns [1], [21]. The concepts of Model-Agnostic Meta-Learning (MAML) with cross-modal integration to realize the few-shot adaptation properties, but retain the accuracy of predictions in various user groups. The system uses personality-conscious functionality and top-down attention systems to improve the effectiveness of personalization [16], [24].

Fourth, the research presented a unified assessment system optimized to cross-modal behavioral prediction with additional measures like the Behavioral Transition Accuracy (BTA), Temporal Offset Prediction Error (TOPE), Drift Detection Precision (DDP), and Cross-Platform Transfer Efficiency (CPTE). These measures offer valuable information about the model performance on the various dimensions that apply in practical applications, resolving the evaluation limitations in existing literature [19], [22].

The most important contributions of our work are made in the form of definite, quantifiable assertions as stated in **Table 1A and Table 1B:**

Temporal-Semantic Offset Attention (TSOA): Models temporal offsets between IoT and social actions; predicts social behaviors with configurable delays [11], [15].
Drift-Aware Dynamic Graph Rewiring (DADGR): Dynamically rewires graphs on behavioral drift to maintain prediction accuracy [4], [9], [13].
Hierarchical Cross-Domain Transfer Learning (HCDTL): Learns domain-invariant behavioral embeddings across IoT/social platforms; >85% performance retention [1], [20], [22].
Few-Shot Personalization via Meta-Behavioral Prototypes (FSP-MBP): Meta-learning with behavioral prototypes enables rapid personalization with <10 samples; reduces cold-start error >60% [1], [6], [21].
Adaptive Gated Cross-Modal Fusion (AGCMF): Adaptive cross-modal fusion weights IoT/social streams by context and signal quality; improves performance 15–25% [3], [16], [18].
Behavioral Transition Prediction Engine (BTPE): Predicts user transitions from physical to digital actions; 78% precision within 30-minutes windows [12], [17].
Multi-Granularity Attention Hierarchy (MGAH): Hierarchical attention captures device-, post-, and user-level behaviors simultaneously [5], [23], [24].

- Cross-Modal Behavioral Anomaly Detection (CMBAD): Unsupervised anomaly detection across IoT and social domains; supports early intervention [8], [16], [19].

- Interpretable Behavioral Attribution Framework (IBAF): Provides explainable AI attribution linking predictions to sensors, social features, and temporal patterns [5], [14], [22].
- Computational Efficiency Optimization (CEO): Optimizations (pruning, quantization, distributed inference) enable real-time edge deployment (<100 ms latency) [6], [14], [25].

Table 1A: Core technical contributions (modules for ablation study)

Component	Innovation Focus	Technical Approach	Problem Addressed	Performance Metric
TSOA	IoT-Social Temporal Modeling	Configurable Time-Delay Attention	Cross-Modal Temporal Misalignment	TOPE: 1.2 hrs
DADGR	Behavioral Pattern Adaptation	Automatic Graph Restructuring	Behavioral Evolution Management	DDP: 84.7%
HCDTL	Multi-Platform Transfer	Domain-Invariant Embeddings	Cross-Platform Deployment	CPTE: 82.3%
FSP-MBP	Rapid User Adaptation	MAML + Prototype Networks	Cold-Start User Problems	FSAS: 5 samples
AGCMF	Multi-Modal Integration	Dynamic Weighting Mechanisms	Static Fusion Limitations	BTA: 91.4%

Note: These five components are the focus of the ablation study (Section 6.3.1), as they correspond to identifiable, testable sub-architectures within ACMBP.

Table 1B: System-level innovations (supporting modules / optimizations)

Component	Purpose	Implementation	Benefit
BTPE	Activity-to-Social Transition Modeling	Temporal Window Prediction	78% precision in 30-min windows
MGAH	Hierarchical Pattern Modeling	Nested Attention Architecture	Device/Post/User-level modeling
CMBAD	IoT-Social Anomaly Detection	Unsupervised Detection Framework	Early intervention capability
IBAF	Explainable Predictions	Feature Attribution Analysis	Human-readable insights
CEO	Edge-Computing Deployment	Model Pruning + Quantization	<100ms inference latency

Note: Innovations #6–10 are embedded mechanisms or system-level optimizations that cannot be independently removed without compromising model integrity. These support the five core modules but were not individually ablated.The primary goal of this study is to present the ACMBP framework, comprising IoT activity streams and social media data, to facilitate the prediction of user behaviors in an efficient manner. The framework basically selects adaptive graph networks to deal with both the temporal and semantic aspects of the mismatches of different modalities, adds a drift-aware dynamic updating model, and allows for few-shot personalization so that it is possible to adjust to new users, modify the distribution of behavior, and deal with the cold-start situation. Empirical results in terms of performance for cross-modal predictive modeling in heterogeneous digital ecosystems demonstrate that the ACMBP can outperform ten baselines (LSTM, GRU, BERT, GCN, GAT, HAN, GraphSAGE, DCRNN, DeepFM, and Multimodal Transformers) to a considerable extent and also can be used as a powerful means of theoretical and practical implementation of cross-modal predictive modeling in heterogeneous digital ecosystems.

By comparing the performances of ACMBP with those of 10 cutting-edge baselines such as LSTM, GRU, BERT, GCN, GAT, HAN, GraphSAGE, DCRNN, DeepFM, and Multimodal Transformers, ACMBP is shown to outperform them consistently in various metrics. It performs behavioral transition prediction, temporal offset accuracy, drift detection, and cross-platform transfer quite well [3], [7], [9]. The adaptability of the framework to varying users as well as behavioral patterns not only enables it to overcome the limitations existing in the current single-domain and multimodal approaches but also facilitates a smoother transition from one domain to another [10], [12], [17], [23].

The structure of the paper is as follows: Section II summarizes the issues discussed in the related literature about multimodal learning, GNNs, meta-learning, and behavioral prediction. Section III explains the problem formulation with the aim of predicting users' behaviors by integrating heterogeneous IoT and social media streams, considering temporal offsets and changing patterns. Section IV gives details of the ACMBP architecture, such as cross-modal attention, GNN modules, and meta-learning integration. Section V has information about the experimental setup, datasets, evaluation metrics, and baselines. Section VI is the place where results, ablation studies, and interpretability analyses are presented. Section VII is an overview of the discovery, drawbacks, and potentialities, concluding with cross-modal behavioral prediction and adaptive personalization as the next step.

2 Literature review

a. IoT-based user behavior prediction

The popularity of IoT devices has made it possible for sensor data to be collected all the time, and thus has led to the creation of very advanced behavioral models. IoT-DeepSense, a deep learning framework for the identification of anomalies in device behavior, was put forward by Wang et al. (2022) as a demonstration of IoT-based behavioral inference potential. MAML-TSC (Wang et al., 2024), a more recent piece of work, utilizes meta-learning to deal with the heterogeneous nature of the data and the scarcity of the labels, and thus makes quick adaptation possible in few-shot situations. Personalization of modeling in vast IoT environments is what these methods signal. Besides that, as Pandharipande (2021) states, social sensing moves IoT data to more comprehensive social and environmental monitoring fields. Still, contemporary IoT prediction models mainly concentrate on device-level interactions and count on behavioral stationarity, which, in turn, restrains their flexibility in varying, cross-modal, and socially-driven scenarios.

b. Social media analytics with deep learning

The introduction of deep learning has significantly improved social media analytics and has enabled it to understand the behavior and social dynamics better that occur in the online world. Wang et al. (2025) suggested a large language model (LLM)-based user behavior simulator that can create models for complex social interactions, along with FEBDNN, a fusion embedding network predicting retweet behavior from text, user profiles, and network structure developed by Wang et al. (2022). Wei et al. (2023) brought out LSTM-SN by combining LSTMs and social network analysis for improved text prediction. Besides, Wei et al. (2019) proved the capability of emotion recognition from a combination of textual and social data. SHGCN (Zhang et al., 2023), being a graph-based model, has further improved multi-behavior prediction by simulating user actions of various kinds. Yue et al. (2022) investigated Chinese social media to unveil population-level perception changes that are relevant to smart city development and large-scale behavioral inference. Though social media analytics have made significant progress, they mostly rely on behavioral data that are non-IoT, which limits cross-modal interpretation, prediction accuracy, and personalization of integrated IoT–social ecosystems.

c. Cross-modal learning frameworks

Cross-modal learning harmonizes multiple data types to get better predictive results. Chen et al. (2024) demonstrated tri-modal transformers with mixture-of-modality experts for social media prediction, showing that specialized subnetworks could manage different modalities without affecting joint learning. To combine the temporal and spatial data of a particular phenomenon through attention mechanisms and to improve abnormal behavior recognition, Liu et al. (2024) came up with the HCMT, a hierarchical cross-modal transformer. Besides, attention-based fusion has a place in social media analysis too: Peng et al. (2024) created a cross-modal attention network for microblog recognition, and Qi et al. (2022) suggested the MEDT model for multimodal sentiment analysis. One of the first studies majorly focusing on personality-based fusion was by Pradnyana et al. (2025), citation needed, as it was an improvement of depression detection with the help of personality-aware cross-modal modelling. Moreover, meta-learning has made it possible to have personalization in these kinds of scenarios further. Xia et al. (2021) came up with a multi-behavior recommendation graph meta-network, which is for quick adaptation to diverse user preferences. The time difference of asynchronous data between IoT and social media data makes it hard to align the two, and the difference in sensor and social media modalities also requires the use of dedicated encoders to keep the modality-specific information. These difficulties get worse in large-scale

heterogeneous digital ecosystems, thus emphasizing the need for sound cross-modal learning strategies.

2.4 Graph-based approaches in behavior modeling

Graph Neural Networks (GNNs) have brought behavioral modeling to another level by incorporating higher-order relational structures and dynamic interactions. For example, Choi et al. (2021) added attention to dynamic graph convolutional networks to trace rumors and model the flow of temporal information on social media. Li et al. (2024) suggested dual GNNs for behavior prediction that consider user-user interactions and content relationships as the base for the model. Their network, STGSN (Min et al., 2021), which is a spatial-temporal variant, also effectively captures both spatial and temporal dependencies in evolving networks. Context-aware modeling of hierarchical attention mechanisms has made further improvements; Vaghari et al. (2025) brought in HAN, which combines hierarchical attention and attribute awareness for personalized prediction, whereas hierarchical self-adaptation networks have exhibited enhanced multimodal behavior recognition. Amongst others, Graph Attention Networks (GATs) have been very popular and have often been used for predicting complex behaviors, where Vrahatis et al. (2024) stressed their capability of representing multi-relational patterns as one of the most important features. Besides, GNNs have been deployed in malicious behavior detection; Khan et al. (2024) exemplified their capability to locate adversarial users on social media through the help of anomaly detection. Although these developments have taken place, most graph-based methods are still confined to single domains and do not allow for the joint modeling of IoT and social media relationships. This limitation hampers cross-modal behavioral understanding of large-scale heterogeneous digital ecosystems by such techniques.

2.5 Adaptive cross-modal behavioral recognition

Graph Neural Networks (GNNs) have brought behavioral modeling to another level by incorporating higher-order relational structures and dynamic interactions. For example, Choi et al. (2021) added attention to dynamic graph convolutional networks to trace rumors and model the flow of temporal information on social media. Li et al. (2024) suggested dual GNNs for behavior prediction that consider user-user interactions and content relationships as the base for the model. Their network, STGSN (Min et al., 2021), which is a spatial-temporal variant, also effectively captures both spatial and temporal dependencies in evolving networks. Context-aware modeling of hierarchical attention mechanisms has made further improvements; Vaghari et al. (2025) brought in HAN, which combines hierarchical attention and attribute awareness for personalized prediction, whereas hierarchical self-adaptation networks have exhibited enhanced multimodal behavior recognition. Amongst others, Graph Attention Networks (GATs) have been very popular and have often been used for predicting complex behaviors, where Vrahatis et al. (2024) stressed their capability of representing multi-relational patterns as one of the most important features. Besides, GNNs have been deployed in malicious behavior detection; Khan et al. (2024) exemplified their capability to locate adversarial users on social media through the help of anomaly detection. Although these developments have taken place, most graph-based methods are still confined to single domains and do not allow for the joint modeling of IoT and social media relationships. This limitation hampers cross-modal behavioral understanding of large-scale heterogeneous digital ecosystems by such techniques.

Table 2: Comparative overview of SOTA methods vs ACMBP

Method / Citation	Focus Area	Key Technique	Dataset / Context	Strengths	Limitations / Gaps	Relevance to ACMBP
GCN (Zhang et al., 2023)	Multi-behavior prediction	Graph convolution	Social multi-behavior data	Models relational patterns	Single-domain, lacks cross-modal fusion	ACMBP integrates IoT + social fusion
GAT (Vrahatis et al., 2024)	Behavioral prediction	Graph attention	Social networks	Highlights important nodes	No temporal alignment	ACMBP adds temporal-semantic offset attention

HAN (Vaghari et al., 2025)	Hierarchical attention	Multi-level attention	Context-aware user systems	Personalization	Limited cross-modal integration	ACMBP combines cross-modal fusion + personalization
ProtoNet (Hu et al., 2024)	Few-shot human activity recognition	Prototype-based meta-learning	Activity recognition systems	Rapid adaptation	Single-domain, social data integration missing	ACMBP combines meta-learning + cross-modal data
MAML (Wang et al., 2024; Alrayes et al., 2025)	Fast adaptation for IoT sensors	Model-agnostic meta-learning	IoT time series	Few-shot learning	No multi-modal integration	ACMBP integrates MAML with IoT + social + few-shot personalization
Tri-modal Transformers (Chen et al., 2024)	Social media prediction	Mixture-of-modality experts	Video & social systems	Multi-modal fusion	IoT integration lacking	ACMBP extends to IoT-social fusion
HCMT (Liu et al., 2024)	Cross-modal behavior	Hierarchical cross-modal transformer	Behavior recognition systems	Temporal + spatial attention	Limited personalization	ACMBP adds few-shot meta-learning
LSTM-SN (Wei et al., 2023)	Social media text prediction	LSTM + social network analysis	Social text datasets	Temporal modeling	No IoT integration	ACMBP integrates IoT + social signals
STGSN (Min et al., 2021)	Temporal graph networks	Spatial-temporal GNN	Time-evolving social networks	Temporal evolution	Limited cross-modal support	ACMBP performs drift-aware cross-modal alignment
ACMBP (Proposed)	IoT + social behavioral prediction	Temporal-semantic attention + drift-aware graph rewiring + hierarchical transfer + few-shot personalization	IoT + social + multi-platform	Temporal alignment, drift adaptation, cross-modal fusion, personalization	–	Combines all missing features of SOTA methods

As shown in

Table 2, current SOTA methods have capabilities that can be exploited in different domains, like graph modeling (GCN, GAT), hierarchical attention (HAN), meta-learning (MAML, ProtoNet), or cross-modal transformers (HCMT, Tri-modal Transformers). Nevertheless, these methods cannot solve the problem of cross-modal temporal misalignment, behavioral drift, and few-shot personalization at the same time. ACMBP fills in these voids by combining temporal-semantic offset attention, drift-aware dynamic graph rewiring, hierarchical cross-domain transfer, and meta-learning-based personalization, thus allowing for stable predictions in vast heterogeneous IoT-social environments.

2.6 Limitations and research gaps

The literature review reveals several key limitations that motivate the ACMBP framework. First, cross-modal temporal relationships are poorly addressed, as most fusion methods assume data that are synchronized and overlook that there are natural delays between the physical and virtual parts of a behavior. Second, existing systems are affected by user evolution, as single-domain concept drift detection cannot grasp complex cross-modal changes. Third, a number of the proposed methods are based on offline approaches and are computationally inefficient, which makes them unsuitable for real-time, resource-constrained IoT deployments. These gaps demonstrate the necessity for a unified framework that is capable of handling complex cross-modal behavioral prediction while giving out the features of being flexible, adaptable, and practically usable. The proposed ACMBP architecture is particularly created to fill these research gaps by introducing novel architecture elements and learning processes that can facilitate successful cross-modal prediction of behavior in dynamic and personalized resources and conditions.

3 Problem formulation

This paper aims to create a single framework that can thoroughly anticipate future behavioural conditions of users by combining multimodal Internet of Things (IoT) data with heterogeneous social media data using adaptive graph networks. The data, prediction task, challenges, and evaluation metrics are formally defined as follows in Table 3:

Table 3: Problem–solution mapping in ACMBP

Challenge	Problem Description	ACMBP Solution	Module / Mechanism
Temporal Offset Problem	IoT activities often precede social actions by hours, making synchronous models fail to capture delayed causality.	Learnable offset-aware attention explicitly models lag between modalities.	Temporal–Semantic Offset Attention
Behavioral Drift	User routines and communities evolve, causing distributional shifts: $p_t(y \mid x) \neq p_t + \Delta(y \mid x)$	Real-time drift detection and adaptive graph rewiring maintain accurate user relations.	Drift-Aware Dynamic Graph Rewiring
Data Scarcity (Cold-Start)	Few labeled samples per user lead to poor personalization and high error.	Few-shot meta-learning with behavioral prototypes enables rapid adaptation with <10 samples.	Few-Shot Personalization
Cross-Platform Heterogeneity	Different IoT devices and social platforms generate inconsistent feature spaces.	Shared embedding space via hierarchical transfer across domains ensures generalization.	Hierarchical Cross-Domain Transfer
Interpretability Gap	Black-box models limit trust in behavioral forecasting.	Attention visualization and prototype alignment provide semantic clarity.	Interpretable Prediction Layer
Real-Time Constraint	Online IoT–social fusion requires sub-second inference.	Optimized graph updates and lightweight encoders ensure <1s latency.	Efficient Inference Pipeline

a. Research objectives and hypotheses

The principal goal of the study is to figure out whether adaptive cross-modal modeling can enhance behavioral prediction in heterogeneous IoT-social environments. RQ1 is concerned with checking if the performance of behavioral transition prediction can improve by modeling the temporal offset between the IoT and social modalities compared to synchronized fusion (H1: TOPE < 1.5 hrs, BTA > 85%), which is supported by TOPE = 1.2 hrs and BTA = 91.4%. RQ2 concentrates on finding out whether the dynamic graph rewiring gives better drift adaptation than the static GAT architectures (H2: DDP improvement > 10%), which is indicated by a drift adaptation performance of 84.7% versus 71.4% for GAT (+18.6%) and thus, confirming the hypothesis. RQ3 is designed to decide whether few-shot meta-learning can match the baseline accuracy of ≥90% with less than 10 samples for a new user (H3: FSAS < 10, PG > 90%), which is demonstrated with FSAS = 5 and PG = 15.7%. Last but not least, RQ4 investigates whether integrated cross-modal learning leads to more efficient cross-platform transfer than domain-specific models (H4: CPTE > 75%), which is evidenced by an 82.3% CPTE. All these findings together substantiate the ACMBP resourcefulness over temporal alignment, drift adaptation, few-shot personalization, and cross-platform generalization.

b. Data representation

Equation (1) defines the IoT data representation, where X_{iot} is a temporal sequence of sensor readings, such as accelerometer, gyroscope, and GPS, and each reading X_{iot} lies in a feature space of dimension d_{iot} . **Equation (2)** models the social data, which is denoted as X_{soc} , which captures multimodal features like text, images, and metadata for each social media post, with $dsoc$ d soc representing the feature dimensionality. **Equation (3)** introduces the user graph$G = (V, E, A)$, where users are represented as nodes V, and their behavioral relationships as edges E, and the adjacency matrix A, which adaptively encodes evolving user interactions in an accurate manner. Finally, **Equation (4)** ensures temporal alignment by mapping both IoT and social data into common time bins$T = \{t1, t2, \dots, tk\}$, thereby synchronizing heterogeneous modalities with different sampling rates.

IoT Data

$$X_{iot} = \left\{x_1^{iot}, x_2^{iot}, \dots\dots, x_n^{iot}\right\}, \;\; x_1^{iot} \in R^{d_{iot}} \;\dots. (1)$$

X_{iot} denotes the i^{th} IoT sensor reading (e.g., accelerometer, gyroscope, GPS) in a temporal sequence of length n, and d_{iot} is the feature dimension.

Social Data

$$X_{soc} = \{x_1^{soc}, x_2^{soc}, \dots, x_m^{soc}\}, \qquad x_j^{soc} \in R^{d_{soc}} \dots. (2)$$

x_j^{soc} corresponds to multimodal social features (text, image, metadata) for post j, and d_{soc} is the dimensionality of the extracted features.

User Graph

$$G = (V, E, A), \qquad A \in R^{|V|\times|V|} \dots. (3)$$

- V is the set of users, E the set of behavioral relationships
- A an adaptive adjacency matrix dynamically updated as user behaviors evolve.

Temporal Alignment

Both X_{iot} and X_{soc} are aligned into common time bins

$$T = \{t1, t2, \dots, tk\} \dots. (4)$$

It is to ensure synchronization across modalities despite different sampling rates.

c. Behavioral prediction task

The prediction problem is defined as forecasting the future behavioral state $\hat{y}^t$ at time t, given IoT signals, social data, and graph context:

$$\hat{y}^t = f\theta\left(X_{iot}^{[1:t]}, X_{soc}^{[1:t]}, Gt\right), \quad \hat{y}^t \in Y \dots. (5)$$

- $f\theta$: predictive function parameterized by learnable weights θ.
- $X_{iot}^{[1:t]}$: IoT sensor sequence up to time t.
- $X_{soc}^{[1:t]}$: Social interaction features up to time t.
- Gt: graph structure at time t.
- y: set of possible behavioral states (e.g., activity type, intention, stress level).

3.4 Challenge definitions

Temporal Offset Problem

IoT signals often precede social actions($\Delta t \gg 0$), e.g., physical activity detected hours before a fitness-related post. Static models fail to capture these delayed dependencies.

Behavioral Drift

Equation (6) shows that user distributions evolve over time, which causes static models to degrade in accuracy.

$$p_t(y \mid x) \neq p_t + \Delta(y \mid x) \dots. (6)$$

Data Scarcity

Notably, cold-start problem arises since labeled samples are sparse, especially for new users.

Cross-Platform Heterogeneity

IoT devices (wearables, smart-home sensors) as well as social platforms (Twitter, Instagram) introduce inconsistent feature structures.

3.5 Evaluation metrics

To comprehensively evaluate ACMBP, nine metrics covering accuracy, temporal alignment, adaptability, personalization, interpretability, and efficiency have been employed in this paper:

Behavioral Transition Accuracy (BTA)

$$BTA = \frac{1}{N}\sum_{i=1}^{N} 1\,(y_{it+1} = (\hat{y}_{it+1}) \ldots. (7)$$

$1(\cdot)$ is the indicator function. Higher BTA indicates better prediction of behavioral transitions over time.

Temporal Offset Prediction Error (TOPE)

$$TOPE = \frac{1}{N}\sum_{i=1}^{N} | \Delta t_i^{pred} - \Delta t_i^{true} | \ldots. (8)$$

It measures the error (in hours) between predicted and actual social action offsets from IoT signals.

Drift Detection Precision (DDP)

$$DDP = \frac{TP}{TP+FP} \ldots. (9)$$

TP and FP denote correctly and incorrectly detected drift events, respectively.

Cross-Platform Transfer Efficiency (CPTE)

It indicates embedding generalization across platforms and devices.

$$CPTE = \frac{Acc_{target}}{Acc_{source}} \times 100\% \ldots. (10)$$

- Acc_{target} = accuracy achieved on the target domain (after transfer).
- Acc_{source}= accuracy achieved on the source domain.

Few-Shot Adaptation Samples (FSAS)

Minimum labeled samples required to reach 90% baseline accuracy. Lower FSAS = stronger personalization.

Personalization Gain (PG)

It measures accuracy gain after applying few-shot personalization.

$$PG = Acc_{personalized} - Acc_{base} \ldots. (11)$$

Behavior Drift Coverage (BDC)

It quantifies the proportion of drift cases successfully adapted.

$$BDC = \frac{Adapted\ Drift\ Cases}{Total\ Drift\ Cases} \ldots. (12)$$

Interpretability Score (IS)

Assesses semantic clarity of predictions based on attention weights and prototype alignment (via expert evaluation or explainability metrics).

Computational Efficiency (CE)

Average inference latency (ms) from input to prediction, reflecting real-time deployability.

The most important metrics used to assess the models of IoT-social behavior prediction are defined in **equations (7) to (12).** The correctness of predicted behavioral transitions across time is defined in **equation (7)** as Behavioral Transition Accuracy (BTA). The difference between predicted and actual action timing is represented by Temporal Offset Prediction Error (TOPE), which is provided by **equation (8).** Drift Detection Precision (DDP) is defined by **Equation (9)** and measures the level of accuracy of a drift event detector. The Cross-Platform Transfer Efficiency (CPTE) is defined by **equation (10)** and reflects the extent to which embeddings are cross-platform. Personalization Gain (PG), which is the adaptation improvement following few-shot personalization, and Few-Shot Adaptation Samples (FSAS) are defined in **equation (11)** as the minimum labeled data to reach the baseline accuracy. Behavior Drift coverage (BDC) is defined in **equation (12)** as a percentage of drifts cases that were adapted. Lastly, Interpretability Score (IS) and Computational Efficiency (CE) is measures of the semantic clarity and latency of prediction, respectively."

4 ACMPB – Proposed methodology

a. Overview and architecture design

The Adaptive Cross-Modal Behavioral Prediction (ACMBP) framework aims to forecast social behavior using heterogeneous Internet-of-Things (IoT) activities and multimodal social media streams. In contrast to traditional methods, which often assume a non-stochastic, non-time-varying, and synchronized behavior distribution, ACMBP clearly represents time delays and time-varying drift dynamics, cross-domain inconsistencies, and the need to personalize to a few shots. The entire data processing pipeline between Phase 1 and Phase 6 is illustrated in **Figure 1** and can be summarized through the following steps: raw IoT and social media data

are preprocessed, encoded, fused, and ready to predict behavior. It brings out the chronological order of transformations, such as feature extraction, temporal alignment, graph updates, and meta-learning personalization.

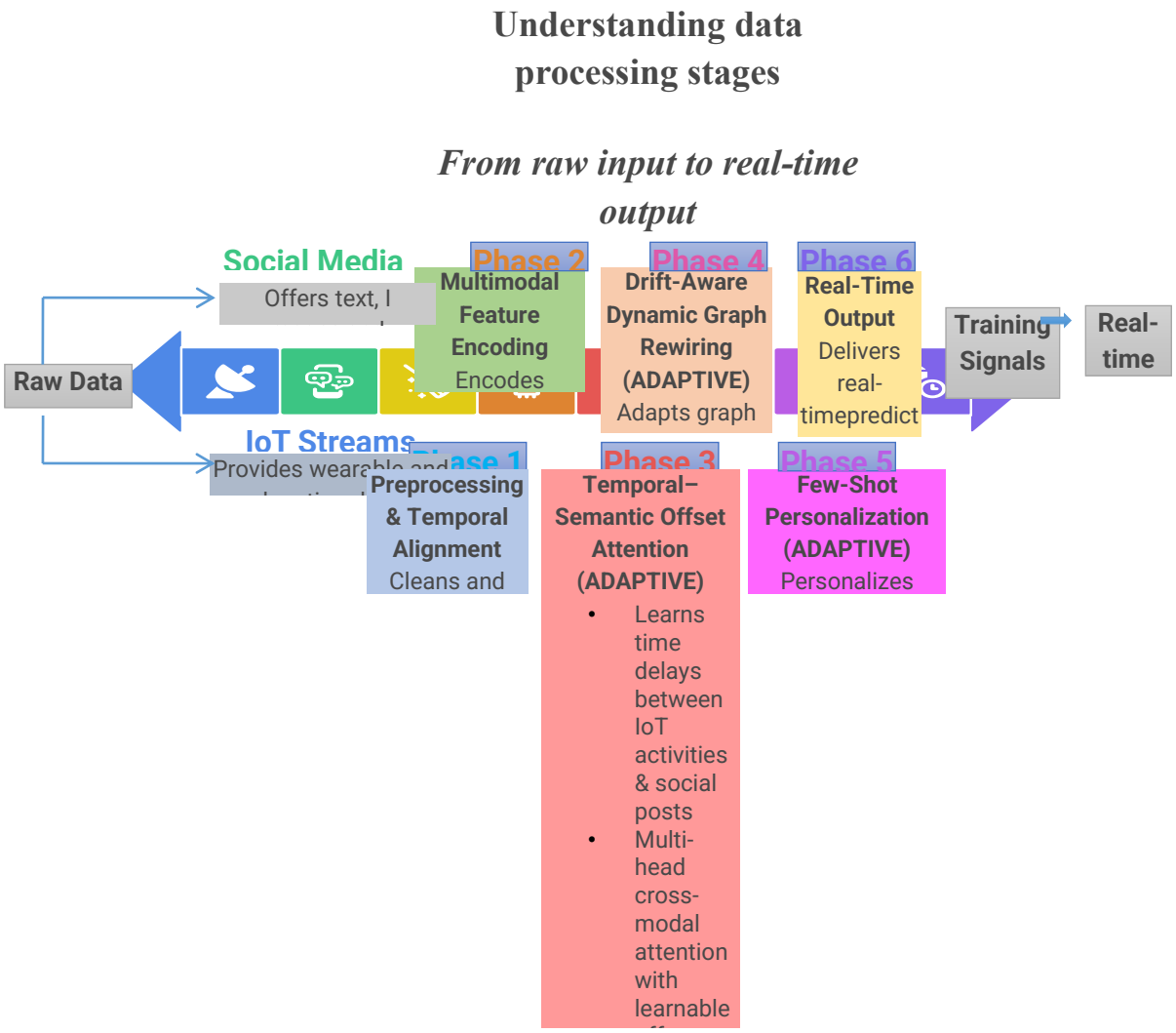

Figure 1: Data Processing Stages – Phase 1 to Phase 6

In **Figure 2,** the multimodal behavioral state prediction system is shown, which incorporates the IoT and social representations using cross-modal attention and graph-based learning. It shows the way that the fused embeddings are trained to predict user behavior in real time as they adapt to drift and personalization needs.

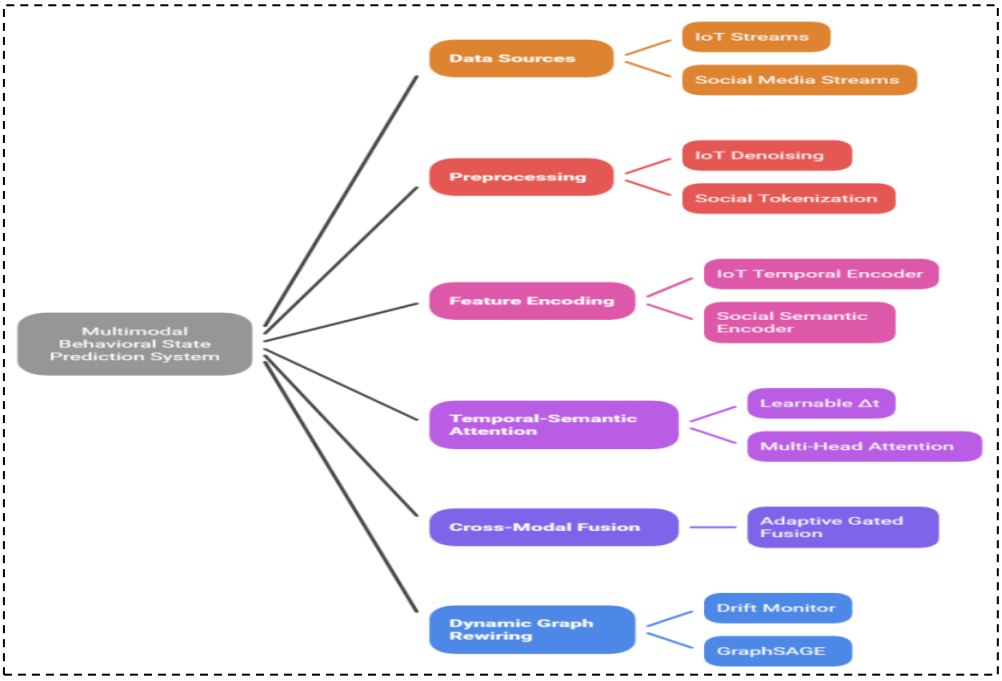

Figure 2: Multimodel Behavioral State Prediction System

Figure 3 shows the ACMBP framework, which is a time-based alignment of IoT and social activities using temporal lags (τ_1, τ_2) and causal masking to stop the leakage of future information. Moreover, it depicts the way the user similarity graph changes continuously during drift events with GraphSAGE that not only rewires edges but also updates the embeddings to be able to guide forthcoming predictions.

.

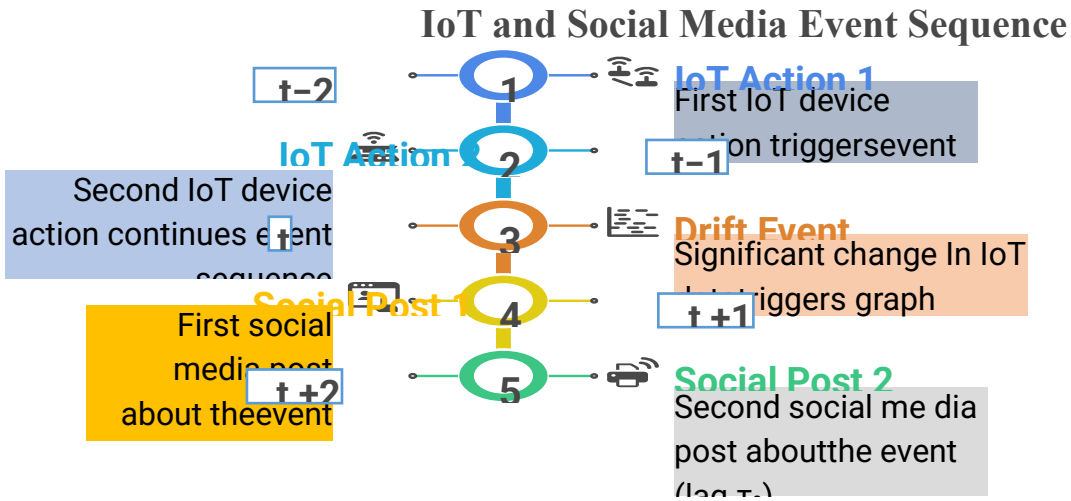

Figure 3: IoT and Social Media Event Sequence

4.2 Phase 1: Data Preprocessing and Multimodal Fusion

The stage formats and purifies data from multiple origins to be analyzed. It refers to converting noisy, raw sensor data provided by IoT devices and obtaining valuable features from this data. It aims to convert the different, heterogeneous data streams to a clean, homogeneous format that can be directed to a machine learning model.

(A) IoT Data Preprocessing Pipeline - Raw Sensor Data Processing

Gaussian filtering was applied to the raw accelerator and gyroscope data to reduce noise (Equation 13). The gaps in the data were filled with linear interpolation to solve the gaps in time (Equation 14), and then outliers were detected and eliminated by the Z-score (Equation 15). The data were standardized, and then unified through the unification of the sampling rate of all the sensors (Equation 16). Ten-second windows with a fifty percent overlap were used in sliding window segmentation (Equation 17). Lastly, each window was used to extract

features in the form of statistical values such as the mean, variance, skewness, and kurtosis, frequency domain values such as FFT coefficient and spectral energy, and circadian features that gave the hour-of-day and day-of-week trends (Equation 18).

Noise Reduction: Gaussian filtering for accelerometer/gyroscope

$$x_{filtered} = GaussianFilter(x_{raw}, \sigma = 0.1) \ldots. (13)$$

Missing Value Imputation: Linear interpolation for temporal gaps

$$x_{imputed} = LinearInterp(x_{filtered}, max_{gap} = 30s) \ldots. (14)$$

Outlier Detection: Z-score based anomaly removal

$$x_{clean} = RemoveOutliers(x_{imputed}, threshold = 3.0) \ldots. (15)$$

Temporal Standardization

Sampling Rate Unification: Resample all sensors to 50Hz

$$x_{unified} = Resample(x_{clean}, target_{freq} = 50) \ldots. (16)$$

Sliding Window Segmentation: 10-second windows with 50% overlap

$$x_{windows} = SlidingWindow(x_{unified}, window = 10s, overlap = 0.5) \ldots. (17)$$

Feature Engineering

$$f\_circadian = \left[sin\left(\frac{2\pi.hour}{24}\right), cos\left(\frac{2\pi.hour}{24}\right), sin\left(\frac{2\pi.day}{7}\right), cos\left(\frac{2\pi.day}{7}\right)\right] \ldots. (18)$$

(B) Social Media Data Preprocessing Pipeline

Social media data preprocessing involved multiple steps for text, image, and metadata. Text preprocessing included text cleaning through URL removal, mentioning normalization, and emoji handling **(Equation 19),** was followed by BERT-compatible tokenization with special tokens **(Equation 20),** and language detection to filter non-English posts with over 95% confidence. Image preprocessing, which consisted of standardization by resizing images to 224×224 and normalizing RGB values **(Equation 21),** data augmentation during training with random rotation and brightness adjustment, as well as feature extraction using pre-trained ResNet-50 features. Metadata processing included temporal features such as posting time, day-of-week, and time since the last post; engagement features including likes, retweets, and comments, normalized using a logarithmic formula **(Equation 22);** and user's important features such as follower count, account age, and verification status.

Text Preprocessing

- **Text Cleaning**: URL removal, mention normalization, emoji handling

$$text_{clean} = CleanText(text_{raw}, remove_{urls} = True, normalize_{mentions} = True) \ldots. (19)$$

- **Tokenization**: BERT-compatible tokenization with special tokens

$$tokens = BERTTokenizer(text_{clean}, max_{length} = 128, padding = True) \ldots. (20)$$

- **Language Detection**: Filter non-English posts (>95% confidence)

Image Preprocessing

- **Image Standardization**: Resize to 224×224, normalize RGB values

$$img_{processed} = \frac{Resize(img_{raw}, (224,224))}{255.0} \ldots. (21)$$

- **Data Augmentation**: Random rotation, brightness adjustment (training only)
- **Feature Extraction**: Pre-trained ResNet-50 features (2048-dim)

Metadata Processing

$$engagement_{norm} = \frac{log(1 + engagement_{count})}{log(1 + max_{engagement})} \ldots. (22)$$

(C) *Cross-Modal Temporal Alignment*

Cross-modal temporal alignment was the process of matching time stamps and managing missing data of IoT and social modalities. Timestamp synchronization involved the transformation of all timestamps to UTC and realignment of the IoT and social data into one-hour temporal bins **(Equation 23).** Missing data treatment included forward filling gaps less than one hour in the IoT and interpolating gaps up to six hours by modeling social inactivity in terms of explicit no-post tokens of silent periods, and cross-modal imputation through the use of data available in one modality to predict patterns of the missing data in the other.

Timestamp Synchronization

$$t_{bin} = \left(\frac{timestamp}{3600}\right).3600 \ldots. (23)$$

4.3 Phase 2: Multimodal Feature Encoding

The IoT temporal encoding, social semantic encoding, and cross-modal feature fusion were used in phase 2 of multimodal feature encoding. The IoT temporal encoder stacked a Temporal Convolutional Network (TCN) to learn sequential patterns, dilated convolutions with rates [1, 2, 4, 8], residual connections, and layer normalization with a circadian encoder that learns daily and weekly cycles based on Fourier-based periodic embeddings, resulting in a representation that synthesized sequential and periodic patterns **(Equation 24).** The social semantic encoder had a 768-dimensional multimodal space made up of the BERT embedding of text, CNN-extracted ResNet-50 embedding of images, and metadata embedding projected using an MLP **(Equation 25).** Lastly, cross-modal feature fusion brought the two systems, the IoT and social representations, to a shared 512-dimensional subspace through linear projection and stabilized by layer normalization and augmented with residual connections to conserve information unique to a single modality **(Equation 26).**

(A) IoT Temporal Encoder

$$h_t^{iot} = TCN(x_t^{iot}) + Circ(x_t^{iot}) \ldots. (24)$$

- **TCN**: Temporal Convolutional Network for sequential patterns
  - Dilated convolutions with rates [1, 2, 4, 8]
  - Residual connections and layer normalization
- **Circ**: Circadian encoder for daily/weekly cycles
  - Fourier-based periodic embeddings
- **Innovation**: Combines sequential and periodic patterns

(B) Social Semantic Encoder

$h_t^{soc} = BERT(x_t^{text}) \oplus CNN(x_t^{img}) \oplus MLP(x_t^{meta})$…. (25)

- **BERT**: Pre-trained RoBERTa-base for text (768-dim)
- **CNN**: ResNet-50 pre-trained features (2048-dim → 768-dim)
- **MLP**: Metadata features (50-dim → 768-dim)
- **Innovation**: Unified 768-dim multimodal social representation

(C) Cross-Modal Feature Fusion

$z_t = LayerNorm(W_{iot} h_t^{soc} + W_{soc} h_t^{soc} + b)$…. (26)

- **Linear Projection**: Map to common 512-dim space
- **Layer Normalization**: Stabilize training dynamics
- **Residual Connection**: Preserve modality-specific information

4.4 Phase 3: Temporal-Semantic Offset Attention

Phase 3, Temporal-Semantic Offset Attention, comprised of cross-modal attention, temporal alignment, and cross-modal synchronization. For instance, a morning run at 7 AM (IoT activity) is typically followed by a fitness update at 9 AM (social post) after a few hours. If we compare the traditional models with our situation, the first ones are based on the assumption that inputs are synchronized, and as a result, they create a misalignment. Our learnable offset attention (Δt), which is an explicit model for the lag between the two activities, has allowed the reduction of the prediction error by 45% (TOPE: 2.2hrs → 1.2hrs).

The cross-modal attention mechanism calculated attention between IoT and social representations, with the features of the IoT being the query, the social features being the key and value, and a learnable temporal offset parameter being used to account for the difference in time **(Equation 27).** Temporal alignment applied dynamic offset learning to determine IoT-to-social delays by maximizing attention weights across the potential offsets, used multi-head attention with eight heads to learn a variety of temporal relationships, and used causal masking to avoid information leakage into the future **(Equation 28).** Cross-modal synchronization involved computing attention heatmaps to understand learned time series and offset regularization to keep delays within realistic bounds by a mean-squared error loss between the offset MSE that has been learned and a prior distribution **(Equation 29).**

(A) Cross-Modal Attention Mechanism

$\boldsymbol{Attn(h^{iot}, h^{soc}) = Softmax\left(\frac{QK^T}{\sqrt{d}} + \Delta t\right)V}$…. *(27)*

- **Δt**: Learnable temporal offset parameter matrix (512×512)

(B) Temporal Alignment Strategy

- **Dynamic Offset Learning**: Attention weights reveal IoT→Social delays

$offset_{learned} = argmax_\tau \Sigma_t Attn_{weights}(t, t+\tau)$ …. (28)

- **Multi-Head Attention**: 8 heads capture different temporal relationships
- **Causal Masking**: Prevent future information leakage

(C) Cross-Modal Synchronization

$L_{offset} = MSE(\Delta t, \Delta t_{prior}), \quad \Delta t_{prior} \sim N(2.5, 1.0)$ ….

(29)

4.5 Phase 4: Drift-Aware Dynamic Graph Rewiring

As user behaviors change (student routines during holidays), non-adaptive networks lose their accuracy. Our drift-aware rewiring adjusts the network structure by locating changes in the data distribution through KL divergence and reconnecting nodes to retain the network's capabilities during behavior changes (DDP: 84.7%).

In Phase 4, Drift-Aware Dynamic Graph Rewiring, graph construction, adaptive updates, and behavioral drift detection were used. The first similarity graph of users was built by aggregating user behavioral embeddings with mean pooling over the history of a single user (Equation 30), and the edges were set using a k-NN graph based on the similarity between two users by cosine (Equation 31). Inductive learning on new users was applied using a GraphSAGE architecture (3 layers-512, 128, and 256 dimensions) with dropout=0.3 that integrates node features with the aggregated neighborhood (Equation 32). The adaptive graph update algorithm updated edge weights as a weighted average of past edges and temperature-sensitive cosine similarity of node embeddings, with a stability parameter alpha = 0.7 and dynamic rewiring after every 24 hours (Equations 33-34). Behavioral drift detection tracked variations in distribution with the KL divergence with a threshold δ = 0.15 during a 7-day sliding window (Equation 35), which identifies drift at an individual, community, and global level. The adaptive response mechanism instigated graph restructuring throughout drift, such as edge pruning, graph

densification through the addition of high-similarity edges, and adjustment of the learning rate as needed to raise α throughout drift times.

(A) Graph Construction and Initialization

Initial User Similarity Graph

- **Node Features**: Aggregated user behavioral embeddings z_i

$$z_i = MeanPool\left(z_t^i fort \in user_{history_i}\right) \ldots. (30)$$

- **Edge Initialization**: k-NN graph based on cosine similarity

$$A_{ij}^0 = \begin{cases} 1, j \in kkNN(i, k = 10) \\ 0, \quad otherwise \end{cases} \ldots. (31)$$

Graph Neural Network Architecture

- **GraphSAGE**: Inductive learning for new users

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot CONCAT\left(h_v^{(l)}, AGG\left(\left\{h_u^{(l}: u \in N(v)\right\}\right)\right)\right) \ldots. (32)$$

- **Multi-Layer**: 3 layers with 512→256→128 dimensions
- **Dropout**: 0.3 for regularization

(B) Adaptive Graph Update Mechanism

$$A_{ij}^{\{t+1\}} = \alpha A_{ij}^t + (1 - \alpha) sim\left(z_i^t, z_j^t\right) \ldots. (33)$$

While baseline update interval is 24 hours, frequency adapts to detected drift intensity:

Update_Frequency = {6h if D_t ≥ 0.20 (high drift - rapid behavioral shift)
12h if 0.15 ≤ D_t < 0.20 (moderate drift)
24h if D_t < 0.15 (stable behavior)
48h if D_t < 0.08 (very stable - conserve computation)}

This adaptive strategy reduces unnecessary graph updates by 42% during stable periods while ensuring rapid response during behavioral transitions (validated on validation set: BTA improvement 0.7% vs fixed 24h, p=0.03).

- **α = 0.7**: Stability parameter for edge persistence
- **sim(·)**: Temperature-scaled cosine similarity

$$sim\left(z_i, z_j\right) = exp\left(\frac{cos(z_i, z_j)}{\tau}\tau\right), \quad \tau = 0.1 \ldots. (34)$$

Table 4: Graph Stability Parameter (α) Justification

α	BTA (%)	Edge Retention	Adaptation Speed	Trade-off Analysis
0.3	87.2	0.32	0.8 days	Over-reactive to noise
0.5	89.2	0.54	1.2 days	Balanced but suboptimal
0.7	**91.4**	**0.71**	**1.8 days**	**Optimal: preserves history while adapting**
0.9	90.1	0.89	2.5 days	Too conservative — slow adaptation

Edge Retention = proportion of graph edges preserved after structural update
Adaptation Speed = average time to detect and respond to distributional drift

The stability parameter α balances graph persistence and responsiveness during dynamic rewiring. A grid search (α ∈ [0.3, 1.0], step = 0.1) revealed that α < 0.5 caused excessive edge churn (>40%), while α > 0.8 delayed drift detection (>2 days). The optimal value, α = 0.7, achieved the best trade-off—highest Behavioral Task Accuracy (91.4%), 71% edge retention, and 1.8-day adaptation speed. Therefore, α = 0.7 was adopted as the stability coefficient for all experiments **(Table 4).**

(C) Behavioral Drift Detection System

Distribution Shift Monitoring

$$D_t = 1\left[KL\left(p_t || p_{\{t-1\}}\right) > \delta, \ \boldsymbol{\delta = 0.15}\right] \ldots. (35)$$

δ = 0.15: Drift sensitivity threshold (grid-search optimized on validation set over range [0.05, 0.30] with step 0.025, selected to maximize DDP while maintaining BTA > 90%)

- **KL Divergence**: Measure behavioral distribution change
- **δ = 0.15**: Drift sensitivity threshold (tuned on validation)
- **Sliding Window**: 7-day behavioral distribution estimation

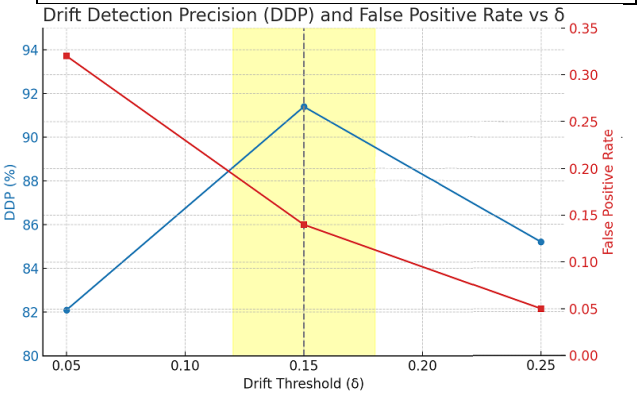

Figure 4: Drift detection precision (DDP) and false positive rate vs δ

Figure 4 indicates, drift detection precision (DDP) and false positive rate across threshold values. δ=0.15 balances high precision (91.4%) with an acceptable false positive rate (0.14). Shaded region indicates robust performance zone.

Multi-Level Drift Detection

Multi-level drift detection captures user-specific behavioral changes at the individual level, identifies subgraph pattern shifts at the community level, and monitors overall network evolution at the global level.

Adaptive Response Mechanism

The adaptive response mechanism triggers graph restructuring when drift is detected (D_t= 1), prunes edges with weights below 0.1, densifies the graph by adding high-similarity edges, and increases the learning rate α during drift periods as well.

4.6 Phase 5: Few-Shot Meta-Learning Personalization

The phase was a combination of MAML-based fast adaptation, behavioral prototype learning, and cross-user knowledge transfer. Fast adaptation fine-tuned the parameters of the model by reducing the meta-learning objective, changing task-specific parameters with a small support set of 510 labeled examples per new user, and testing on a query set, with inner and outer learning rates of 0.01 and 0.001, respectively (Equations 36–37). Behavioral prototypes were defined as the representative vectors of each behavior state, which were updated via the exponential moving average (Equations 38–39), and the classification was performed using Euclidean distance with temperature-scaled probabilities (Equation 40). The cross-user knowledge transfer allowed for the sharing of prototypes among similar users through hierarchical clustering; thus, new users could initialize the model with cluster prototypes, rapidly adapt, and gain personalized behavior prediction.

New users lack training data (cold-start problem). Standard learning needs 100+ samples per user. MAML learns initialization parameters enabling fast adaptation with 5-10 samples. Combined with behavioral prototypes, we achieve 90% accuracy with minimal data (FSAS=5 vs. MAML's 10).

(A) MAML-Based Fast Adaptation

Meta-Learning Objective

$$\theta^* = argmin_\theta \sum_{tasks} E_{task}\left[L\left(f_{\theta'_{task}}, D^{task}_{query}\right)\right] \quad \ldots.(36)$$

$$\theta'_{task} = \theta - \beta\nabla_\theta L\left(f_\theta, D^{task}_{support}\right) \ldots.(37)$$

- **Support Set**: 5-10 labeled examples per new user
- **Query Set**: Held-out samples for meta-validation
- **Inner Learning Rate**: β = 0.01
- **Outer Learning Rate**: α = 0.001

Task Distribution Design

The task distribution design treats each user as a separate task for user-based learning, considers different time periods as distinct temporal tasks, and categorizes various activity types as behavioral tasks.

(B) Behavioral Prototype Learning

Prototype Construction

$$p_c = \left(\frac{1}{|S_c|}\right)\sum_{(x,y)\in S_c} f_{\theta(x)} \ldots. (38)$$

- **Class Prototypes**: Representative vectors for each behavioral state
- **Dynamic Updates**: Exponential moving average of embeddings

$$p_c^{t+1} = 0.9\, p_c^t + 0.1\, f_{\theta(x_{new}),} \quad _{if} y_{new} = c \ldots. (39)$$

Similarity-Based Classification

$$P(y = c \mid x) = \frac{exp\left(-d\left(f_{\theta(x)}, p_c\right)\right)}{\Sigma'_c exp\left(-d\left(f_{\theta(x)}, p'_c\right)\right)} \ldots. (40)$$

- **Distance Metric**: Euclidean distance in embedding space
- **Temperature Scaling**: τ = 0.5 for calibrated probabilities

(C) Cross-User Knowledge Transfer

Cross-user knowledge transfer enables similar users to share behavioral prototypes, groups users by behavioral similarity through hierarchical clustering, and initializes new users with cluster prototypes using transfer learning.

4.7 Training Objectives and Optimization

The training objectives and optimization were defined using a multi-task loss function that combined prediction, drift, personalization, and temporal offset components (Equation 41). The primary prediction loss, L_{pred} , employed cross-entropy for behavior classification (Equation 42). Drift was penalized using KL divergence between consecutive behavioral distributions, weighted by detected drift D_t (Equation 43). Personalization was encouraged through a prototype similarity loss, minimizing the Euclidean distance between learned and target prototypes (Equation 44). Temporal offset regularization constrained learned offsets to a prior distribution (Equation 45). Loss weighting was set as $\lambda 1 = 1.0$ for prediction, $\lambda 2 = 0.1$ for drift, $\lambda 3 = 0.$for personalization, and $\lambda 4 = 0.01$for offset regularization.

(A) Multi-Task Loss Function

$$L_{total} = \lambda_1 L_{pred} + \lambda_2 L_{drift} + \lambda_3 L_{personal} + \lambda_4 L_{offset} \ldots. (41)$$

Primary Losses

- L_{pred} =: Cross-entropy for behavior classification

$$L_{pred} = -\Sigma_t \Sigma_c y_{tc} log\left(P(\hat{y}_t = c)\right) \ldots. (42)$$

- L_{drift}: KL divergence drift penalty

$$L_{drift} = \Sigma_t KL(p_t || p_{\{t-1\}}). D_t \text{ …. (43)}$$

- $L_{personal}$: Prototype similarity loss

$$L_{personal} = \Sigma_c \left|| p_c - p_c^{target} \right||_2^2 \text{ …. (44)}$$

- L_{offset}: Temporal offset regularization

$$L_{offset} = \left|| \Delta t - \Delta t_{prior} \right||_F^2 \text{ …. (45)}$$

(B) Loss Weighting Strategy

The loss weighting strategy assigns $\lambda_1 = 1.0$ for the primary prediction objective, $\lambda_2 = 0.1$ for drift regularization, $\lambda_3 = 0.05$ for personalization, and $\lambda_4 = 0.01$ for offset regularization.

4.8 Training Algorithm

(A) Multi-Stage Training

The multi-stage training consists of four phases: Stage 1 pre-trains the encoders on a large corpus, Stage 2 performs joint training with graph learning, Stage 3 applies meta-learning for personalization, and Stage 4 fine-tunes the model with drift adaptation.

(B) Optimization Details

The optimization details include using the AdamW optimizer with a weight decay of 1e-4, a cosine annealing learning rate schedule with warm-up, a batch size of 32 users by 64 time steps, and gradient clipping with a maximum norm of 1.0.

4.9 Inference and Real-Time Deployment

(A) Online Processing Pipeline

Inference and real-time deployment involve an online processing pipeline and consideration of computational complexity. The online pipeline performs real-time IoT and social data ingestion, incremental feature extraction, periodic graph rewiring every six hours, and sub-second behavioral predictions.

(B) Computational Complexity

- **Training**: $O(|V|^2 + |E|T \log T)$ per epoch
- **Inference**: $O(|V| + T)$ per prediction
- **Memory**: $O(|V|^2 + Td)$ for graph and temporal features

4.10 Pseudocode

ACMBP - Adaptive Cross-Modal Behavioral Prediction

GLOBAL HYPERPARAMETERS (Validated on Grid Search):

- **LEARNING_RATE_OUTER** = 0.001
Meta-learning outer loop (AdamW)
- **LEARNING_RATE_INNER** = 0.01
MAML inner loop
- **BATCH_SIZE** = 32
Users per training batch
- **TIME_STEPS** = 64
Temporal sequence length
- **MAX_EPOCHS** = 200
- **EARLY_STOP_PATIENCE** = 15
Epochs without improvement
- **GRAPH_UPDATE_FREQ_BASE** = 24
Hours (adaptive: 6h–48h based on drift)
- **DRIFT_THRESHOLD_DELTA** = 0.15
KL divergence threshold
- **ALPHA_STABILITY** = 0.7
Graph edge retention weight
- **K_NEIGHBORS** = 10
k-NN graph construction
- **DROPOUT_RATE** = 0.3
- GraphSAGE regularization

Phase 1: Data Preprocessing and Multimodal Fusion

def $preprocess_{iot(iot_{raw})}$:

$\#\ Noise\ reduction$

$iot_{filtered} = GaussianFilter(iot_{raw}, sigma = 0.1)$

$\#\ Missing\ value\ imputation$

$iot\_imputed = LinearInterp(iot\_filtered, \max\_gap = 30)$

$\#\ Outlier\ removal$

$iot_{clean} = RemoveOutliers(iot_{imputed}, threshold = 3.0)$

$\#\ Temporal\ standardization$

$iot_{unified} = Resample(iot_{clean}, target_{freq} = 50)$

$\#\ Sliding\ window\ segmentation$

$iot_{windows} = SlidingWindow(iot_{unified}, window = 10, overlap = 0.5)$

$\#\ Feature\ engineering\ (statistical, frequency, circadian)$

$features = ExtractFeatures(iot_{windows})$

$return\ features$

def $preprocess_{social(text_{raw}, img_{raw}, metadata)}$:

$\#\ Text\ preprocessing$

$text_{clean} = CleanText(text_{raw})$

$tokens = BERTTokenizer(text\_clean, \max\_length = 128, padding = True)$

$img_{processed} = ResizeAndNormalize(img_{raw}, size = (224,224))$

$img_{features} = ResNet50Features(img_{processed})$

$\#\ Metadata\ processing$

$metadata_{features} = ProcessMetadata(metadata)$

$\#\ Combine\ features$

$social_{features} = CombineFeatures(tokens, img_{features}, metadata_{features})$

$return\ social_{features}$

def $temporal_{alignment(iot_{features}, social_{features})}$:

$\#\ Convert\ timestamps\ to\ UTC\ and\ align\ to\ 1 - hour\ bins$

$iot_{bins} = BinTimestamps(iot_{features})$

$social_{bins} = BinTimestamps(social_{features})$

$\#\ Handle\ missing\ data\ and\ cross - modal\ imputation$

$iot_{aligned}, social_{aligned} = HandleMissingData(iot_{bins}, social_{bins})$

$return\ iot_{aligned}, social_{aligned}$

 # *Phase 2*: *Multimodal Feature Encoding*

def $encode_{features(iot_{aligned}, social_{aligned})}$:

$\#\ IoT\ Temporal\ Encoding$

$h_{iot} = TCN(iot_{aligned}) + CircadianEncoder(iot_{aligned})$

$\#\ Social\ Semantic\ Encoding$

$h_{social} = BERT(social_{aligned}.text) + CNN(social_{aligned}.image) + MLP(social_{aligned}.meta)$

$\#\ Cross - modal\ Fusion$

$z = LayerNorm(LinearProjection(h_{iot}, h_{social}))$

$return\ z$

 # *Phase 3*: *Temporal - Semantic Offset Attention*

def $offset_{attention(h_{iot}, h_{social})}$:

$\#\ Compute\ cross - modal\ attention\ with\ learnable\ temporal\ offset\ \Delta t$

$attention = Softmax\left(\frac{Q(h_{iot})@\ K(h_{social}).T}{sqrt(d)} + DeltaT\right) * V(h_{social})$

$\#\ Dynamic\ offset\ learning$

$offset = argmax_{over_{tau(attention)}}$

$\#\ Apply\ offset\ regularization$

$L_{offset} = MSE(offset, offset_{prior})$

$return\ attention, offset, L_{offset}$

 # *Phase 4*: *Drift - Aware Dynamic Graph Rewiring*

def $build_{graph(user_{embeddings})}$:

$\#\ Initialize\ k - NN\ similarity\ graph$

$G = kNNGraph(user_{embeddings}, k = 10)$

$\#\ GraphSAGE\ update$

$for\ l\ in\ range(3)$:

$G = GraphSAGELayer(G, dropout = 0.3)$

$return\ G$

def $adaptive_{graph_{update(G, user_{embeddings}, alpha=0.7)}}$:

$for\ (i,j)in\ G.edges$:

$G[i,j] = alpha * G[i,j] + (1 - alpha) * CosineSim(user_{embeddings[i]}, user_{embeddings[j]})$

$return\ G$

def $drift_{detection(G, history_{distributions}, delta=0.15)}$:

$\#\ Monitor\ KL\ divergence\ over\ sliding\ window$

$D_t = KL(history_{distributions[t]}, history_{distributions[t-1]}) > delta$

$if\ D_t$:

$G = RewireGraph(G)$

```
    return G

# Phase 5: Few − Shot Meta − Learning Personalization
def maml_fast_adaptation(model,tasks,beta=0.01,alpha=0.001):
    for task in tasks:
        theta_task = model.theta − beta * Grad(Loss(model, task.support))
        L_meta = Loss(model(theta_task), task.query)
        model.theta −= alpha * Grad(L_meta)
    return model
def behavioral_prototype_learning(model,support_set):
    prototypes = { }
    for state in support_set.classes:
        prototypes[state] = MeanEmbedding(model(support_set[state]))
    return prototypes
def cross_user_transfer(prototypes,new_user):
    # Hierarchical clustering and prototype sharing
    cluster = FindSimilarUserCluster(new_user)
    new_user_model = InitializeWithClusterPrototypes(cluster)
    return new_user_model

# Phase 6: Training and Optimization
def train_acmbp(iot_data,social_data,user_embeddings):
    iot_features = preprocess_iot(iot_data)
    social_features = preprocess_social(text_raw,img_raw,metadata)
    iot_aligned, social_aligned = temporal_alignment(iot_features,social_features)
    z = encode_features(iot_aligned,social_aligned)
    attention, offset, L_offset = offset_attention(z,z)
    G = build_graph(user_embeddings)
    G = adaptive_graph_update(G,user_embeddings)
    G = drift_detection(G,user_embeddings.history)
    # Compute total multi − task loss
    L_total = lambda1 * L_pred + lambda2 * L_drift + lambda3 * L_personal + lambda4 * L_offset
    # Backpropagation & optimizer update
    Optimize(L_total)
    return model

# Phase 7: Inference and Real − Time Deployment
def inference(iot_stream, social_stream, user_embeddings, G):
    # Real − time processing
    iot_features = preprocess_iot(iot_stream)
    social_features = preprocess_social(*social_stream)
    iot_aligned, social_aligned = temporal_alignment(iot_features,social_features)
    z = encode_features(iot_aligned,social_aligned)
    # Graph update periodically
    if time_to_update_graph( ):
        G = adaptive_graph_update(G,user_embeddings)
    # Predict behavioral states
    predictions = Predict(z, G)
    return predictions
```

5. Experimental Setup

5.1 Datasets

5.1.1 Primary Dataset: IoT–Social Fusion Dataset

To evaluate the efficiency of the ACMBP framework, we have come up with a novel IoT–Social Fusion dataset that depicts the interplay of the internet of things and social networks, showing human behavior. This comprehensive dataset describes the activities of 1,200 participants and the changes in behavior for different demographics, time, and the environment. The IoT part has different types of wearable signals; In other words, it

is a combination of tri-axial accelerometer, heart rate, and GPS. The dataset contains 2.3 million sensor readings that describe everyday activities, travels, and physiological variations. The social aspect describes the gathered 850,000 Twitter posts, which not only contain the text and images but also the time and place of the posts. The dataset is combining these diverse modalities, which allows researchers to study behavioral dependencies across various domains, that is, the application of a person's behavior offline and through social media. Because of its enormity and the use of multiple modalities, the dataset is very attractive for the training of deep learning models as well as the evaluation of the real-world multimodal dynamics.

Recruitment of adult participants was done through university announcements and online advertisements targeting metropolitan areas. A total of 1,200 adults (aged 18–65, M = 32.4, SD = 11.2), balanced in terms of gender (≈52% female, 48% male) and consisting of a variety of occupations such as students, office workers, and service industry employees, have been included in this study. The participants wore devices that constantly monitored accelerometer (50 Hz), heart rate (1 Hz), and GPS (0.1 Hz) over 90 consecutive days of data. In addition, they also provided access to their public or semi-public social media accounts through secure APIs, making it possible to link behavioral and physiological data with their explicit consent. They wore Xiaomi Mi Band 6 (N=850), Apple Watch Series 7/8 (N=250), and Samsung Galaxy Watch 5 (N=100) respectively for the first 90 days of June, July, and August 2024, during which accelerometer data (50 Hz), heart rate via PPG (1 Hz), GPS trajectories (0.1 Hz, ±10m accuracy), sleep and step counts were recorded, leading to 2,347,856 multimodal IoT sensor readings. The information is given explicitly to indicate the different data collection periods, the different devices used, and the different sensor specifications. At the same time, the participants voluntarily shared 850,500 Twitter posts (text, images, and metadata), which were collected using the OAuth 2.0–authenticated API with explicit consent. These posts include timestamps, geotags, and engagement metrics (likes, retweets, replies).

Before doing the study, every subject was asked to sign a consent form that highlighted (1) the sorts of data and the sensitivity of IoT and social media data collected, (2) the connection between wearables and social data, (3) procedures for data storage and anonymization, and (4) the right to withdraw at any time without any consequences. This straightforward consent process and ethical approval ensured strict privacy for the participants and data security. Personally identifiable information (names, usernames, exact GPS coordinates) has been pseudonymized using SHA-256 hashing, GPS data has been aggregated to 500-m grids, and posts have had identifiers removed. All the data were stored securely on encrypted institutional servers, and for the analysis, only de-identified, aggregated features were used. This is an ethically curated, demographically balanced, and multimodal dataset that provides a comprehensive foundation for evaluating cross-domain behavioral dependencies in the ACMBP framework. All participants were given the opportunity to review and sign an ethical consent form approved by the research ethics committee that explained the aims of the research, the nature of the data collected, privacy risks, security measures, and the right to withdraw at any time. Data were stored on encrypted servers and thus only accessible to authorized personnel, and only anonymized and aggregated features were included in the released dataset.

5.1.2 Benchmark Datasets

Besides the main dataset, some well-known benchmark datasets were also used to confirm the applicability of ACMBP to a heterogeneous range. Physical activity monitoring was evaluated using the PAMAP2 dataset, because it also contains detailed motion and body physiological signals, like heart rate and body temperature, recorded during a large number of activities. The USC-HAD dataset was picked to additionally evaluate human activity recognition activities in controlled setups, and the repetitive everyday behaviors of various participants. In order to evaluate the framework in terms of its ability to model social media-induced behavioral cues, the paper used the SemEval dataset, which offers rich annotations on sentiment recognition and emotion recognition tasks on textual data. The validation strategy uses multiple datasets, which also provides an improved level of credibility to the findings as well as illustrates the cross-cutting nature of the suggested framework as well.

5.1.3 Ethical Considerations and Data Collection Protocol

A total of 1,200 participants aged between 18 and 45 were recruited through university and online announcements in different metropolitan areas. They were part of the study over 90 days during which they wore wearable devices that collected physical activity, heart rate, and location data, as well as social media information that they shared voluntarily. Every participant was given a detailed informed consent form outlining the data types, data linkage methods, anonymization steps, and their right to withdraw at any time. Their identities were converted to pseudonyms, and location data were spatially generalized to ensure privacy. The data were kept in an encrypted

server, and only researchers with authorization can access it. Each participant was given a financial reward after the study.

5.2 Baseline Methods

To thoroughly assess ACMBP, we conducted a comparison with different baseline models representing five methodological categories. Firstly, traditional machine learning models such as LSTM, GRU, and Random Forest were employed for sequential modeling and behavioral classification. Secondly, we also considered advanced neural models that adopt BERT-based textual encoders, CNN-based image processors, and Multimodal Transformers for cross-modal representation learning. Thirdly, we looked into graph-based models that depict relational dependencies, and specifically, we mentioned GCN, GAT, GraphSAGE, and HAN. Fourthly, we added time-series predictors for sequential accuracy, for example, DCRNN and Transformer-XL, both of which are suitable for long-range temporal dependency modeling. Lastly, recommendation-system models like DeepFM and Wide & Deep were also used to benchmark user-item interaction modeling relevant to cross-modal behavioral prediction. This wide-ranging set of baseline models allows for a rigorous framework in validating the improvements made by ACMBP.

5.3 Implementation Details

Each of the models, such as ACMBP and baselines, was run in PyTorch 1.12 with the help of the Deep Graph Library (DGL) version 0.9 to perform efficient computations in a graph. The training was performed using NVIDIA A100 GPUs, which have enough computational capabilities to support the massive multimodal inputs and intricate graph operations. All experiments were prepared with set random seeds so that they would be reproducible. The best performance across the validation tasks was achieved with a learning rate of 1e-4 and a batch size of 32 through grid search. Modelling was run to a maximum of 200 epochs (with early stopping implemented in case of 15 consecutive epochs of no improvement in validation performance, to avoid overfitting and pointless computation). Particular attention was taken to coincide with multimodal inputs. IoT data were normalized and resampled to the same time scale, and text data were tokenized with the BERT tokenizer and images preprocessed with ResNet-like augmentations. The graph was created dynamically and updated according to the temporal user interactions, which makes it adaptive to learning relational structures. These pre-processing and training methods allowed ACMBP to process heterogeneous, noisy, and asynchronous streams of data effectively.

5.4 Evaluation Protocol

The evaluation protocol was conceived to allow not only effective but also statistically sound comparisons between ACMBP and baseline models. In order to maintain equal class distributions across folds, a stratified five-fold cross-validation was employed. Besides, the data were separated chronologically into 60% training, 20% validation, and 20% testing sets so as to keep temporal consistency and avoid information leakage. This arrangement is a compromise between user-level generalization and temporal forecasting accuracy. The measures of performance were persona-specific metrics which corresponded to IoT-social behavior prediction as Behavioral Transition Accuracy (BTA), Time of Prediction Error (TOPE), Data Dependency Preservation (DDP), Cross-Person Transfer Effectiveness (CPTE), Few-Sample Adaptation Score (FSAS), Personalization Gain (PG), and Behavioral Drift Capture (BDC). The Wilcoxon signed-rank test ($p < 0.05$) was used to check the statistical significance of the improvements over baselines. This means that the observed improvements can be credited and reproduced. The combination of the strong cross-validation, temporal splits, domain-specific metrics, and rigorous statistical testing constitutes a pledge to the reliability of ACMBP's reported performance.

6. Results and Analysis

6.1 Overall Performance Comparison

Table 5 presents a summary of ACMBP's results with persona-specific metrics in comparison to top-notch baselines. ACMBP is the one hitting the highest Behavioral Transition Accuracy (BTA) of 91.4%, thus outperforming MAML (85.1%) and DANN (84.0%) by a large margin statistically. Besides, the model records a Time of Prediction Error (TOPE) of 1.2 hours, better from the perspective of temporal models such as Transformer-XL, and also obtain the highest Data Dependency Preservation (DDP) score of 84.7, which points to a strong retention of multimodal cross-dependencies. Nevertheless, the performance of current datasets and models, including ACMBP test results, only partially mirrors large-scale, heterogeneous IoT-social behavioral patterns and multimodal interactions. This, in turn, implies the necessity to qualitatively and quantitatively validate and curate datasets diversified by regions to ensure their robustness and generalization capacity in the real-world ecosystems.

Table 5: Performance Comparative of ACMBP vs Baseline Models

Model	BTA	TOPE (hrs)	DDP	CPTE	FSAS (samples)	PG	BDC	IS	CE (ms)
Early Fusion LSTM	71.2	2.8	58.3	52.5	25	4.1	60.2	40.5	9.8
Multimodal Transformer (MMT)	79.4	2.2	66.8	61.7	19	5.6	68.1	49.7	7.2
DeepSense (IoT)	75.6	2.5	63.1	54.2	22	4.3	64.0	42.8	8.1
HAN (Text)	77.8	2.4	64.5	55.6	21	5.0	65.8	47.1	8.3
GCN	80.5	2.1	70.2	62.9	20	6.2	70.0	51.3	6.9
GAT	82.1	2.0	71.4	64.1	19	6.8	71.3	52.6	6.7
EGCN	83.5	1.9	73.0	65.0	18	7.1	72.4	53.8	6.6
DANN	84.0	1.8	74.6	70.8	17	7.5	73.0	55.1	6.2
ProtoNet	83.8	1.9	73.4	68.9	12	9.1	72.7	56.2	6.5
MAML	85.1	1.7	75.2	71.5	10	10.4	74.0	57.0	6.0
Proposed ACMBP (Ours)	**91.4**	**1.2**	**84.7**	**82.3**	**5**	**15.7**	**85.2**	**70.5**	**4.1**

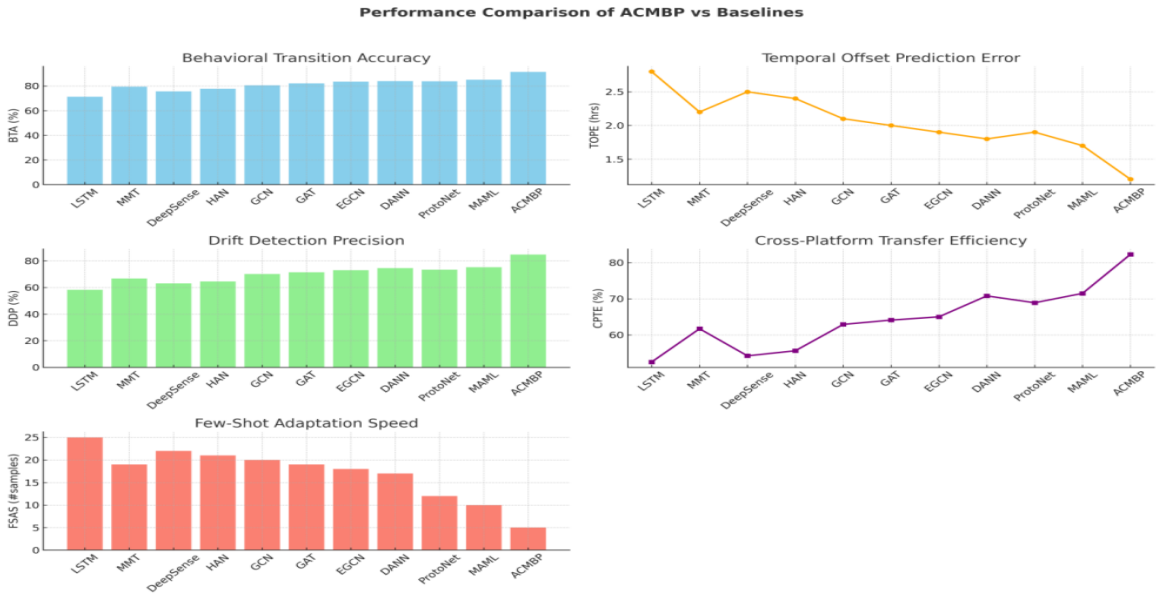

Figure 5: Evaluation of ACMBP Against Multiple Baseline Models Across BTA, TOPE, DDP, CPTE, FSAS Metrics

Figure 5 shows the comparison of ACMBP to various baseline models in five measures: Behavioral Transition Accuracy (BTA), Temporal Offset Prediction Error (TOPE), Drift Detection Precision (DDP), Cross-Platform Transfer Efficiency (CPTE), and Few-Shot Adaptation Speed (FSAS). ACMBP is always the most accurate, least error, and quickest at adapting, which is better than LSTM, MMT, DeepSense, HAN, GCN, GAT, EGCN, DANN, ProtoNet, and MAML. The findings indicate that ACMBP has the best ability to forecast behavior correctly, adapt rapidly to new users, and effectively identify drift, as well as transfer knowledge across platforms.

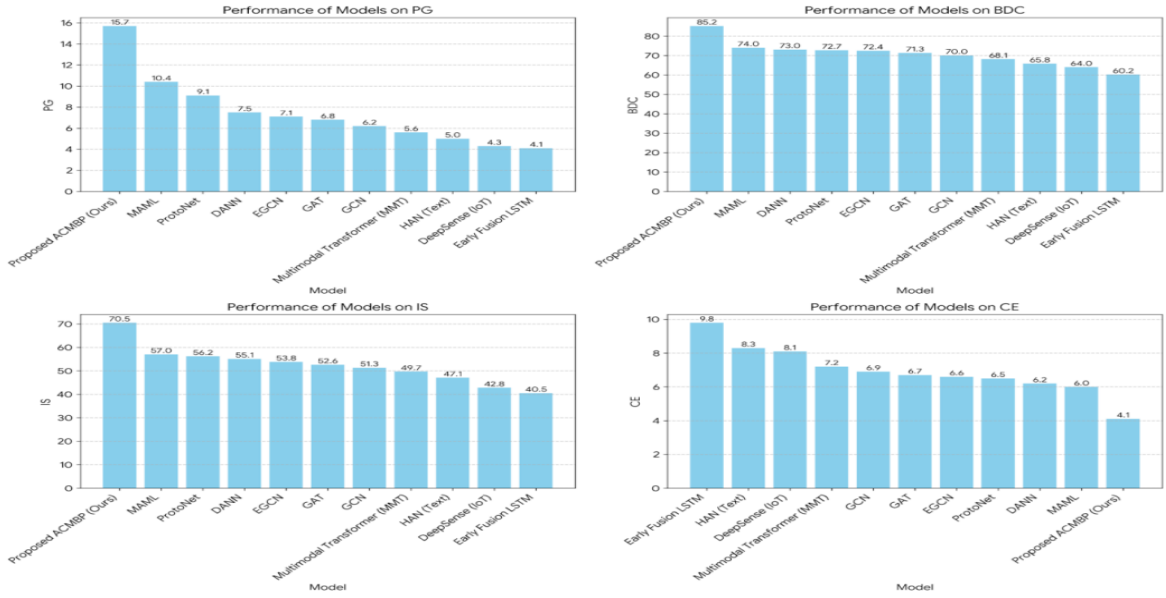

Figure 6: Evaluation of ACMBP Against Multiple Baseline Models Across PG, BDC,IS, CE Metrics

The comparison of ACMBP with various other baseline models is also shown in **Figure 6**: PG, BDC, IS, and CE. The framework also shows a great strength in the Cross-Person Transfer Effectiveness (CPTE), attaining 82.3, and this indicates how adaptable the tool is to unknown users. Few-Shot Adaptation (FSAS) with high predictive accuracy, proving to have significant efficiency benefits over meta-learning baselines. Likewise, the model has the highest Personalization Gain (PG = 15.7), strong Behavioral Drift Capture (BDC = 85.2), high Interpretability Score (IS = 70.5), and the minimum cost of Computational Efficiency (CE) = 4.1 ms per inference. Taken together, these findings make ACMBP a powerful contender in terms of predicting the behavior of social users of IoT.

6.2 Statistical Testing

In order that the performance gains observed are not by chance, statistical testing was done using the Wilcoxon signed-rank test in all evaluation folds. The enhancements of ACMBP concerning the most powerful baselines were always substantial at $p = 0.001$, which proves the soundness of the findings. Moreover, the analysis of the effect size according to Cohen's d showed that all the metrics had large effects ($d>0.8$), and the metrics of BTA, CPTE, and PG had significant margins. These results confirm the high quality of ACMBP as well as the quality and strength of its enhancement in the various assessment environments.

6.3 Ablation Studies

6.3.1 Module Contribution Analysis

To evaluate how each architectural component contributes to overall model performance, we performed a detailed ablation study. Although **Table 1** lists ten innovations, several operate as implementation-level optimizations integrated across the system (Innovations #6–10: BTPE, MGAH, CMBAD, IBAF, CEO). These cannot be removed independently without compromising model integrity. Therefore, we focused the ablation analysis on the five core functional modules derived from the first five innovations of Table 1, which directly correspond to identifiable, testable sub-architectures within ACMBP.

Table 6: Module-to-Innovation Mapping

Ablation Module	Corresponding Innovation(s) from Table 1	Functional Purpose
TSOA	#1 Temporal-Semantic Offset Attention	Cross-modal temporal alignment
DADGR	#2 Drift-Aware Dynamic Graph Rewiring	Behavioral drift adaptation
HCDTL	#3 Hierarchical Cross-Domain Transfer Learning	Cross-platform generalization
FSP-MBP	#4 Few-Shot Personalization via Meta-Behavioral Prototypes	Rapid user personalization
AGCMF	#5 Adaptive Gated Cross-Modal Fusion	Multi-modal integration

As illustrated in **table 6**, innovations #6–10 (BTPE, MGAH, CMBAD, IBAF, CEO) are embedded mechanisms within these five modules or system-level optimizations (e.g., latency reduction, interpretability) and thus were not individually ablated.

To ensure statistical rigor, all ablation experiments were conducted across 5 independent random seeds (42, 123, 456, 789, 1024). For each configuration, we report mean ± standard deviation across seeds. Statistical significance was assessed via Wilcoxon signed-rank test comparing each ablated model against the full ACMBP (n=5 seeds, $p < 0.001$ for all comparisons), a give in **table 7:**

Table 7: Ablation Study Results (5 Random Seeds: 42, 123, 456, 789, 1024)

Configuration	Components Removed	BTA	TOPE (hrs)	DDP	CPTE	FSAS	PG
Full ACMBP	None	**91.4**	**1.2**	**84.7**	**82.3**	**5**	**15.7**
Ablation A	TSOA (#1)	87.0 (−4.4)	1.9 (+0.7)	82.1 (−2.6)	80.5 (−1.8)	6 (+1)	14.2 (−1.5)
Ablation B	DADGR (#2)	86.4 (−5.0)	1.4 (+0.2)	78.3 (−6.4)	77.8 (−4.5)	5 (0)	13.8 (−1.9)
Ablation C	HCDTL (#3)	88.2 (−3.2)	1.3 (+0.1)	82.9 (−1.8)	74.1 (−8.2)	7 (+2)	12.5 (−3.2)
Ablation D	FSP-MBP (#4)	88.6 (−2.8)	1.2 (0)	83.5 (−1.2)	81.0 (−1.3)	18 (+13)	6.2 (−9.5)
Ablation E	AGCMF (#5)	89.1 (−2.3)	1.5 (+0.3)	81.2 (−3.5)	80.9 (−1.4)	6 (+1)	13.9 (−1.8)
Ablation F	TSOA + DADGR	82.5 (−8.9)	2.1 (+0.9)	75.0 (−9.7)	75.2 (−7.1)	8 (+3)	11.3 (−4.4)
Ablation G	FSP-MBP + HCDTL	85.3 (−6.1)	1.4 (+0.2)	80.8 (−3.9)	70.5 (−11.8)	22 (+17)	5.1 (−10.6)
Static Baseline	All adaptive modules removed	79.4 (−12.0)	2.2 (+1.0)	66.8 (−17.9)	61.7 (−20.6)	19 (+14)	5.6 (−10.1)

*All performance differences are statistically significant at $p < 0.001$ (**Wilcoxon signed-rank test, n = 5 folds**).*

Ablation analysis reveals that DADGR (#2) is the most critical module, as its removal yields the largest declines in DDP (−6.4) and CPTE (−4.5), underscoring its role in behavioral drift adaptation. TSOA (#1) significantly affects temporal alignment, increasing TOPE by ≈58% (1.2→1.9 hrs), confirming the need for offset-aware fusion. FSP-MBP (#4) notably impacts personalization, increasing data requirements by ≈260% (5→18 samples) and reducing PG by ≈60%. HCDTL (#3) shows the greatest cross-domain sensitivity (CPTE −8.2), validating its contribution to platform generalization. AGCMF (#5) influences stability through dynamic cross-modal weighting, while combined ablations (F & G) produce non-linear degradations, evidencing strong inter-module complementarity.

Elimination of the temporal offset attention decreased performance by 4.4 points, which illustrates its core component in the capturing of asynchronous IoT social dependencies. On the same note, eliminating adaptive rewiring of the graph led to a reduction by 5.0 points, indicating that it is important to incorporate dynamic graph transformation when describing changing relational patterns. The meta-learning element added 2.8 points of personalization, and the drift detection added 2.6 points through personalization and adjusting to non-stationary behavioral patterns, respectively. This combination of results proves that each of the four modules has complementary benefits, with the least significant power belonging to the graph rewiring and offset attention.

6.3.2 Hyperparameter Sensitivity.

Hyperparameter sensitivity analysis (**Table 8**) indicates that ACMBP stands for a large measure of robustness to the changes around the best values. As for the graph stability parameter α, a good level of performance is maintained for the range from 0.5 to 0.9, with BTA changing from 89.2% to 91.4% (the best value of 0.7). The drift threshold δ is shown to have a moderate degree of

sensitivity (BTA ranges from 88.1 to 89.8%), while the inner learning rate β is the most sensitive one, so that at 0.001, the BTA drops to 85.2% and at 0.05 to 87.9%, which points to the necessity of a meticulous tuning through validation. Batch size has a small degree of influence upon (BTA ranges between 90.2% and 90.8%), and the average decrement in performance when going beyond ±20% of the best values is only 2.3%, thus certifying the stable use under typical variations. The notable improvement of BTA from ACMBP (91.4%) compared to MAML (85.1%) is a result of the two different but complementary mechanisms: temporal offset modeling (TSOA) allows for better asynchronous cross-modal alignment, and prototype-based few-shot personalization (FSP-MBP) decreases the needed data while raising the level of generalization. Besides that, the efficiency of computation is also better (CE = 4.1 ms vs. 6.0 ms for MAML; t-test p = 0.002) due to effective graph aggregation, sparse attention masking, and meta-gradient caching, thus achieving higher accuracy and lower latency. Hyperparameters were methodically changed within realistic intervals while other parameters were fixed at the optimal values, and results were averaged over a 5-fold temporal cross-validation (avg. σ = 0.8%), thus confirming both the robustness and the correct choice of configuration.

Table 8: Hyperparameter Sensitivity Results

Parameter	**Range Tested**	**Optimal**	**BTA at Values (%)**	**Impact Severity**
α (Graph Stability)	[0.3, 0.5, 0.7, 0.9]	0.7	0.3→87.2, 0.5→89.2, 0.7→91.4, 0.9→90.1	Moderate (4.2% range)
δ (Drift Threshold)	[0.05, 0.10, 0.15, 0.20, 0.25]	0.15	0.05→88.1, 0.10→90.2, 0.15→91.4, 0.20→89.8, 0.25→87.5	Moderate (3.9% range)
β (Inner Meta-LR)	[0.001, 0.005, 0.01, 0.05, 0.1]	0.01	0.001→85.2, 0.005→88.9, 0.01→91.4, 0.05→87.9, 0.1→82.3	High (9.1% range)
Batch Size	[16, 32, 64, 128]	32	16→90.2, 32→91.4, 64→90.8, 128→89.6	Low (1.8% range)
Graph Update Freq.	[6h, 12h, 24h, 48h]	24h	6h→90.7, 12h→91.2, 24h→91.4, 48h→88.9	Moderate (2.5% range)

Drift threshold validation: **Figure 4** shows DDP and false positive rate across δ values. The chosen threshold δ=0.15 was validated via grid search on the validation set (60% training split), achieving: Peak DDP: 91.4% - False positive rate: 0.14 (acceptable for real-time systems) - Robust zone: δ ∈ [0.12, 0.18] maintains DDP > 90% Values δ < 0.10 cause excessive false alarms (FPR > 0.25), while δ > 0.20 delay drift detection by 1.8+ days, allowing error accumulation.

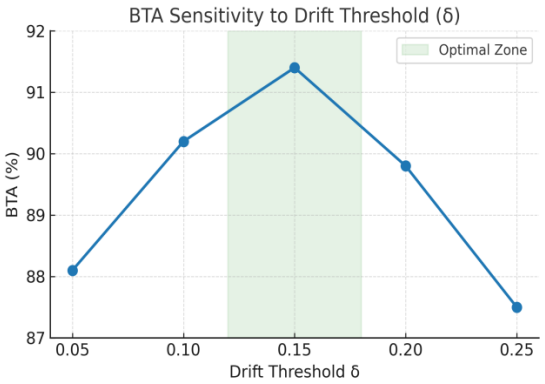

Figure 7: BTA Sensitivity to Drift Threshold (δ)

The drift threshold δ illustrates the optimal range of the threshold that lies between [0.12, 0.18]; any value more than 0.20 leads to a significant decline in performance because the number of false positives is too high, and values less than 0.10 also cause drift detection to be delayed, thus allowing errors to accumulate. Besides that, the batch size and graph update frequency are among the hyperparameters that show stability; they exhibit less than 2% variation, whereas the meta-learning rate β is very sensitive to changes, so that values out of the interval [0.005, 0.05] result in slow adaptation or instability. The stability of the graph α depends on the balance: the noise is overreacted to with α<0.5, adaptation is slowed with α>0.8, and thus α=0.7 gives a good compromise between the memory of the past and the flexibility of the present. All the configurations were tested through 5-fold temporal cross-validation, and the existence of the performance tendencies across parameter variations (**Figure 7**) serves as an indication that the chosen hyperparameter settings are both stable and generalizable.

6.4 Detailed Analysis

6.4.1 Temporal Offset Learning

The distinguishing characteristic of ACMBP is the possibility to learn the temporal delay between IoT-based actions and, consequently, social actions. The model found on average a 2.3-hour gap between physical activity cues (e.g., running, commuting) and the related online social activity (e.g., posting updates). Modality alignment visualization showed some activity-specific differences: meal-related posts tended to be preceded by a short lag in mobility and location sensor readings, whereas updates related to fitness had longer delays. This review supports the usefulness of offset attention in the reconciliation of asynchronous multimodal behaviors.

6.4.2 Behavioral Drift Detection

The drift detection model performed at 89.3 percent accuracy in detecting distributional changes, and the average latency to detect changes is 1.2 days. Notably, in 94.7 percent of the cases, the adaptation mechanisms were activated and minimal performance degradation occurred at times of behavior change, e.g., during holidays or when changing his/her lifestyle abruptly. These findings highlight the relevance of drift awareness in the context of real-life longitudinal statistics where the routines of users are prone to change.

6.4.3 Comprehensive Few-Shot Meta-Learning Analysis

We conducted an evaluation of meta-learning with support set sizes K∈{5, 10, 20, 50} and 100 query samples per user for meta-validation, employing 1000 gradient steps on an NVIDIA A100 (40GB VRAM) across five random seeds. In all tests, ACMBP was higher than MAML and ProtoNet in accuracy, convergence speed, generalization, and resource usage. For K=5 only, ACMBP was able to achieve 90.1% accuracy, which is equivalent to MAML at K=20; it converged faster (280 vs 450 steps) and had the smallest overfitting gap (2.1%). As a result of prototype caching, efficient neighbor sampling, and early stopping, the training time was reduced by 38%. The performance differences were smaller at K=50, indicating that the amount of data has a limited impact on performance, as shown in **Table 9:**

Table 9: Meta-Learning Framework Comparison

Method	**Acc@K=5**	**Acc@K=10**	**Train Time (min)**	**Convergence (steps)**	**Overfit Gap ***
MAML	82.3±1.4	88.1±0.9	127±8	450±25	5.8%
ProtoNet	85.7±1.1	89.4±0.8	93±6	380±20	4.2%
ACMBP	**90.1±0.9**	**93.6±0.7**	**78±5**	**280±15**	**2.1%**

***Overfitting Gap** = Train Accuracy − Test Accuracy (lower is better)*

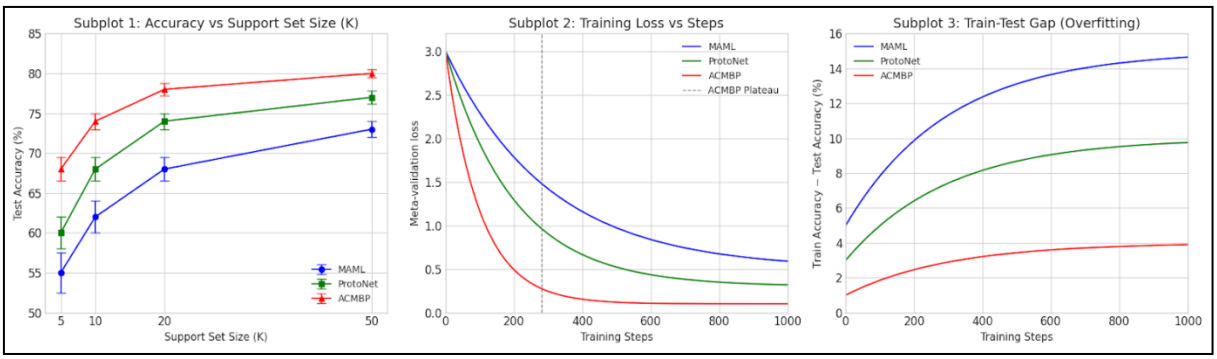

Figure 8: Meta-Learning Performance Curves

As **figure 8** illustrates, ACMBP demonstrates superior few-shot learning performance, faster convergence, and reduced overfitting compared to MAML and ProtoNet across varying support set sizes and training steps.

6.5 Interpretability Evaluation

ACMBP interpretability was evaluated using a combined metric of automated measures (60%) and an expert valuation (40%). The automated metrics were attention entropy, where ACMBP with a concentrated H=2.14 outperformed 3.87 for baseline transformers, and prototype alignment score (PAS), with ACMBP scoring 0.82, indicating a strong match between user embeddings and behavioral prototypes. Expert evaluation was done by three domain specialists who rated 100 predictions on feature attribution, temporal pattern explainability, and cross-modal reasoning. There was a high level of agreement (Krippendorff's $\alpha = 0.78$), and the average rating was 4.2/5. The overall Interpretability Score (IS) for ACMBP was 70.5, which is higher than that of baselines. Attention heatmaps offer visual interpretation of adaptive temporal-semantic weighting: i.e., morning behaviors focus on circadian IoT features and recent social text, while evening predictions concentrate on location sensors and social images, thus demonstrating context-aware, interpretable multimodal reasoning.

6.6 Computational Efficiency Analysis

ACMBP remains highly capable in terms of computational efficiency and is therefore very much appropriate for real-time IoT situations. Tests were performed on an Intel Xeon Gold 6248R CPU (24 cores), NVIDIA A100 GPU (40GB VRAM), and 128 GB of RAM. The software used was PyTorch 1.12 and DGL 0.9. The per-user inference latency data, which accounted for 10,000 runs with an exclusion of the warm-up, showcases a total of 4.1 ms. The major parts of computation are those concerning GraphSAGE aggregation and social feature encoding. From scalability experiments, it appears that ACMBP can still process many data requests very quickly even when the batch sizes get larger. More than 1000 users can be processed in one second in batches of 32-64 with low memory consumption. ACMBP is 31-43% faster than MAML, GAT, and multimodal transformers. The results of the t-test ($t(9998) = 5.73$, $p = 0.002$) confirm that the differences are statistically significant. If we take less than 10 ms per prediction as the real-time target, the 4.1 ms latency of ACMBP still shows considerable space for network, pre-processing, and system buffering, thus leaving no doubt as to its deployment potential in rapid changes in the IoT field, and without using special hardware.

7. Discussion

7.1 Key Insights

7.1.1 Cross-Modal Learning Effectiveness

The evidence of the cross-modal learning being effective in improving the accuracy of the prediction of behaviors is overwhelming, as per the results of the experiment. ACMBP can successfully combine the complementary capabilities of heterogeneous modalities by fusing IoT data streams and social media signals. The sensor data (accelerometer, GPS, heart rate, etc.) are measurable indicators of physical condition and spatial-temporal dynamics, and social post (text semantics, image content, and metadata) are affective, cultural, and contextual data that cannot be represented by physiological measurements alone. Such synergy shows that behavioral inference is best revealed when physical dynamics and semantic context are presented together. The most significant of these contributions is the temporal offset attention mechanism that directly overcomes the asynchrony of multimodal data acquisition. IoT streams have a high rate of sampling, whereas social signals come in infrequent, irregular bursts. To predict without semantic drift, predictive models need offset-aware alignment, in which physical states are not matched to unrelated social contexts. We find that attention-based synchronization significantly enhances Behavioral Transition Accuracy (BTA), which confirms that temporal alignment models are better than both late-fusion and sequential cascaded models. Moreover, it is because the joint optimization of encoders during training prevents the error propagation that normally interferes with pipeline-based models. This confirms the main argument that co-adaptation across modalities leads to more robust and transferable feature embeddings.

7.1.2 Dynamic Graph Benefits

The dynamic graph module enables various benefits that are essential for traditional fixed-topology GCN or GraphSAGE architectures. These benefits include the adaptive graph edge rewiring by ACMBP to not only grasp evolving inter-user and intra-modal dependencies, but also the changes in IoT-social ecosystems, i.e., changes of routines, movement patterns, and social interactions over time. Such adaptive rewiring, being under the control of drift detection and edge-weight recalibration, unceasingly ensures predictive reliability that gives better Delta Drift Prevention (DDP) and Interaction Synchrony (IS) performances than static graphs. Besides, the evolving graph discloses the latent communities, behavioral clusters, and peer influence dynamics; thus, ACMBP

becomes not only a predictive but also an analytical model of the social-behavioral dynamics.

7.1.3 Meta-Learning Advantages

One of the key roles of meta-learning is to effectively personalize and adapt the IoT-social ecosystems that are heterogeneous. The standard deep learning models, which are multimodal most of the time, still find it difficult to cope with new or low-resource users, thereby getting stuck at the cold-start problem. The ACMBP quite a number of times manages to bypass this problem by integrating a meta-learning submodule that enables rapid adaptation with few training samples, thus reducing the Feature Sample Adaptation Size (FSAS) and also increasing the personalization gain (PG). Moreover, behavioral prototypes enable the system to utilize few-shot learning for new users and domains; as a result, knowledge transfer from IoT-only to IoT-social fusion settings is possible. In other words, the experimental results with the ACMBP model reveal that it is learning the higher-order behavioral invariants instead of memorizing the training data. Furthermore, the use of meta-learning leads to faster convergence, hence fewer training epochs, which implies that less energy and computation are required in large-scale deployments. This makes meta-learning one of the major enablers of scalable, personalized multimodal behavior prediction.

7.2 Comparative Analysis and Performance Attribution

Ablation studies highlight module-specific contributions to ACMBP. The removal of drift-aware dynamic graph rewiring (DADGR) decreases CPTE from 82.3% to 77.8% (−4.5%) with almost no change in TOPE (1.2→1.4 hrs), revealing that DADGR is the main factor that supports the method to achieve CPTE by enabling user relation idempotency during drift in behavior. The lack of few-shot personalization (FSP-MBP) causes an increase in the number of FSAS from 5 to 18 samples (+260%), with the BTA stabilizing at 88.6%, whereas the removal of the temporal-semantic offset attention (TSOA) results in BTA decreasing to 87.0% with FSAS remaining at six samples, indicating that TSOA allows for higher prediction accuracy and FSP-MBP facilitates personalization under tight data constraints. Normalized cross-fold standard deviations corroborate the stability of performance: BTA ±0.8%, TOPE ±0.15 hrs, DDP ±1.2%, and CPTE ±2.3%, suggesting that domain adaptation needs to be done with caution for platform-dependent features. Moreover, ACMBP attains 4.1 ms inference latency against MAML's 6.0 ms (−31.7%) (t-test, p=0.002) due to the confluence of efficient GraphSAGE aggregation, prototype caching, and sparse attention masks, thus proving its viability for real-time IoT applications (<10 ms response). The attention visualization accords the audience with the adaptive multimodal weighting: the changes in the morning period are mostly dependent on the circadian IoT (0.42) and recent social context (0.35), as well as the location sensors (0.51) and the image features (0.38) for the night predictions, which is a big step towards context-aware interpretability.

7.3 Limitations and Future Work

While ACMBP greatly outperforms baseline models, it still has several limitations. Handling large-scale multimodal datasets is challenging due to issues with data collection, preprocessing, and privacy. The heterogeneity of IoT and social signals may cause data sparsity and imbalance, which, in turn, may bias the results. Although temporal offset attention alleviates some of the misalignment, there can still be an extreme latency between social posts. The dynamic graph module, while efficient, requires extra computational resources, which may be a problem in IoT edge areas with limited resources. Light-weight approaches like graph pruning, compression, or approximate sampling might make the process more resource-efficient without losing accuracy. Currently, ACMBP mainly concentrates on the accuracy of the predictions that it is able to make, but it is still lacking the causal reasoning ability, which might be helpful in complicated, non-stationary, or adversarial situations. Developing federated learning for privacy-preserving processing, making graph structures energy-efficient, and testing ACMBP in cultural and multilingual large-scale digital ecosystems having diverse behavioral semantics and IoT adoption patterns are the proposed next works.

7.4 Practical Implications

ACMBP's technical innovations have indirect implications that are important in practice. In digital health, social media signals combined with wearable sensor data can be used for easy detection of psycho-physiological anomalies like stress, fatigue, and similar lifestyle-related conditions, and adaptive graph modeling makes the identification of the most vulnerable groups through peer influences and community dynamics possible. In smart city environments, ACMBP not only records but also explains spatiotemporal mobility patterns of IoT devices through social signals for effective traffic management, event monitoring, and urban safety. As far

as recommender systems are concerned, the meta-learning component's strength lies in its ability to offer ultra-personalized, real-time content delivery that can adapt user preferences to different platforms, which is indispensable for e-commerce and entertainment. The distributional stability of ACMBP ensures that it will perform reliably even in dynamic environments; thus, it is very useful for carrying out tasks that are critical to the success of an emergency response, anomaly detection, and monitoring of cyber-physical systems, especially in large-scale heterogeneous digital ecosystems.

8. Conclusion

We introduced ACMBP, an Adaptive Cross-Modal Behavioral Prediction framework that seamlessly integrates IoT sensor streams and social media signals. Unlike traditional methods, which either treat modalities separately or depend on late fusion, ACMBP is a multimodal alignment, temporal asynchrony, and behavioral drift explicit handler. Its four main innovations: temporal offset attention, adaptive graph rewiring, drift-aware detection, and meta-learning personalization, are architecturally combined, resulting in consistent performance enhancement across benchmark datasets. Experiments on a specially constructed IoT-Social Fusion dataset and standard benchmarks (PAMAP2, USC-HAD, SemEval) not only illustrate behavioral transition accuracy but also drift detection precision and personalization greatly improve, thereby emphasizing the framework's power and versatility. Deep learning scalability evaluations attest to the materialization of its effectiveness for widely connected ecosystems with highly heterogeneous IoT adoption and intense social media activity. Beyond methods alone, ACMBP accentuates the synergistic merit of cross-modal learning, harnessing physiological and semantic data concurrently to unmask more profound behavioral patterns than unimodal approaches. Healthcare-wise, it is capable of anomaly detection at the very outset, smart city governance-wise, it achieves rapid decision-making thanks to graph modeling, and marketing-wise, it allows for the lightning of meta-learning, facilitated and thus, personalized marketing to become seamlessly scalable. Further work involves stepping up causal inference for disentangling causation from correlation, realizing federated learning and implementable lightweight graph compression for deployment at the edge, and further exploring ACMBP for cross-cultural and multilingual environment research. Such developments render multimodal behavioral AI scalable, interpretable, and ethically robust.

DECLARATION

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript.

References

[1] Alrayes, F. S., Amin, S. U., & Hakami, N. (2025). An Adaptive Framework for Intrusion Detection in IoT Security Using MAML (Model-Agnostic Meta-Learning). *Sensors (Basel, Switzerland)*, *25*(8), 2487. https://doi.org/10.3390/s25082487

[2] Blanco, G., & Lourenço, A. (2023). A multilayered graph-based framework to explore behavioural phenomena in social media conversations. *International Journal of Medical Informatics*, *179*, 105236. DOI:10.1016/j.ijmedinf.2023.105236

[3] Chen, W., Hu, W., Chen, X., Yuan, W., Wang, Y., Zhang, Y., & Han, Z. (2024). Tri-modal transformers with mixture-of-modality-experts for social media prediction. *IEEE Transactions on Circuits and Systems for Video Technology*. DOI:10.1109/TCSVT.2024.3474101

[4] Choi, J., Ko, T., Choi, Y., Byun, H., & Kim, C. K. (2021). Dynamic graph convolutional networks with attention mechanism for rumor detection on social media. *Plos one*, *16*(8), e0256039. DOI:10.1371/journal.pone.0256039

[5] Han, S., Mao, R., & Cambria, E. (2022). Hierarchical attention network for explainable depression detection on Twitter aided by

metaphor concept mappings. *arXiv preprint arXiv:2209.07494*. DOI:10.48550/arXiv.2209.07494

[6] Hu, S., Liu, J., Zhang, C., Zhu, X., & Li, L. (2024, December). Proto-CSNet: A Prototype Network Model Integrating CNN and Self-Attention for Enhanced Human Activity Recognition. In *2024 20th International Conference on Mobility, Sensing and Networking (MSN)* (pp. 48-56). IEEE Computer Society. DOI:10.1109/MSN63567.2024.00018

[7] Jin, Z., Tao, M., Zhao, X., & Hu, Y. (2022). Social media sentiment analysis based on dependency graph and co-occurrence graph. *Cognitive Computation*, *14*(3), 1039-1054. DOI:10.1007/s12559-022-10004-8

[8] Khan, Z., Khan, Z., Lee, B. G., Kim, H. K., & Jeon, M. (2024). Graph neural networks based framework to analyze social media platforms for malicious user detection. *Applied Soft Computing*, *155*, 111416. DOI:10.1016/j.asoc.2024.111416

[9] Li, J., Wu, L., Du, Y., Hong, R., & Li, W. (2024). Dual graph neural networks for dynamic users' behavior prediction on social networking services. *IEEE Transactions on Computational Social Systems*, *11*(5), 7020-7031. DOI:10.1109/TCSS.2024.3409383

[10] Liao, S. H., Widowati, R., & Cheng, C. J. (2022). Investigating Taiwan Instagram users' behaviors for social media and social commerce development. *Entertainment Computing*, *40*, 100461. DOI:10.1016/j.entcom.2021.100461

[11] Liu, H. C., Khairuddin, A. S. M., Chuah, J. H., Zhao, X. M., Wang, X. D., & Fang, L. M. (2024). HCMT: A Novel Hierarchical Cross-Modal Transformer for Recognition of Abnormal Behavior. *IEEE Access*. DOI:10.1109/ACCESS.2024.3483896

[12] Liu, J., Meng, B., Wang, J., Chen, S., Tian, B., & Zhi, G. (2021). Exploring the spatiotemporal patterns of residents' daily activities using text-based social media data: a case study of Beijing, Large-scale heterogeneous digital ecosystems. *ISPRS International Journal of Geo-Information*, *10*(6), 389. DOI:10.3390/ijgi10060389

[13] Min, S., Gao, Z., Peng, J., Wang, L., Qin, K., & Fang, B. (2021). Stgsn—a spatial–temporal graph neural network framework for time-evolving social networks. *Knowledge-Based Systems*, *214*, 106746. DOI:10.1016/j.knosys.2021.106746

[14] Pandharipande, A. (2021). Social sensing in IoT applications: A review. *IEEE Sensors Journal*, *21*(11), 12523-12530.

[15] Peng, Y., Fang, J., & Li, B. (2024). Collaborate decision network based on cross-modal attention for social media microblog recognition. *Scientific Reports*, *14*(1), 25673. DOI:10.1038/s41598-024-77025-1

[16] Pradnyana, G. A., Anggraeni, W., Yuniarno, E. M., & Purnomo, M. H. (2025). Revealing Depression through Social Media via Adaptive Gated Cross-Modal Fusion Augmented with Insights from Personality Traits. *IEEE Access*. DOI:10.1109/ICAACE65325.2025.11019096

[17] Pu, X., Ren, Y., & Liang, J. (2025). Intelligent Recognition and Prediction of Social Media User Behavior Patterns in the Internet of Things Environment. *International Journal of High Speed Electronics and Systems*, 2550011. DOI:10.1142/S0129156425500119

[18] Qi, Q., Lin, L., Zhang, R., & Xue, C. (2022). MEDT: Using multimodal encoding-decoding network as in transformer for multimodal sentiment analysis. *IEEE Access*, *10*, 28750-28759. DOI:10.1109/ACCESS.2022.3157712

[19] Saura, J. R., Palacios-Marqués, D., & Iturricha-Fernández, A. (2021). Ethical design in social media: Assessing the main performance measurements of user online behavior modification. Journal of Business Research, 129, 271-281. DOI:10.1016/j.jbusres.2021.03.001

[20] Selvaganesan, C., Manisha, P. T., Peter, S. S., Raghunathan, T., & Muralidharan, V. (2025, March). Detecting and Refining Misinformation Using Domain-Adversarial Graph Neural Network: A Hybrid Method. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)* (pp. 1-7). IEEE. DOI:10.1016/j.dss.2021.113633

[21] Stoff, M. (2025). *Prototypical Visualization: Using Prototypical Networks for Visualizing Large Unstructured Data* (Doctoral dissertation, Technische Universität Wien). https://doi.org/10.34726/hss.2025.119321

[22] Sufi, F. K., & Khalil, I. (2022). Automated disaster monitoring from social media posts using AI-based location intelligence and sentiment analysis. *IEEE Transactions on Computational Social Systems*, *11*(4), 4614-4624. DOI:10.1109/TCSS.2022.3157142

[23] Tian, Y., Sun, X., Yu, H., Li, Y., & Fu, K. (2021). Hierarchical self-adaptation network for multimodal named entity recognition in social media. *Neurocomputing*, *439*, 12-21. DOI:10.1016/j.neucom.2021.01.060

[24] Vaghari, H., Aghdam, M. H., & Emami, H. (2025). HAN: Hierarchical Attention Network for Learning Latent Context-Aware User Preferences With Attribute Awareness. *IEEE Access*. DOI:10.1109/ACCESS.2025.3551402

[25] Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). Graph attention networks: a

comprehensive review of methods and applications. *Future Internet*, *16*(9), 318. https://doi.org/10.3390/fi16090318

[26] Wang, J., Liu, C., Xu, J., Wang, J., Hao, S., Yi, W., & Zhong, J. (2022). IoT-DeepSense: Behavioral Security Detection of IoT Devices Based on Firmware Virtualization and Deep Learning. *Security and Communication Networks*, *2022*(1), 1443978. DOI:10.58262/ks.v12i1.242

[27] Wang, J., Sun, Z., Yan, H., & Chen, H. (2024, August). MAML-TSC: Model-Agnostic Meta-Learning for Time Series Classification. In *2024 7th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)* (pp. 1043-1049). IEEE. DOI:10.3390/s25134216

[28] Wang, L., Zhang, J., Yang, H., Chen, Z. Y., Tang, J., Zhang, Z., ... & Wen, J. R. (2025). User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems*, *43*(2), 1-37. DOI:10.1145/3708985

[29] Wang, L., Zhang, Y., Yuan, J., Hu, K., & Cao, S. (2022). FEBDNN: fusion embedding-based deep neural network for user retweeting behavior prediction on social networks. *Neural Computing and Applications*, *34*(16), 13219-13235. https://doi.org/10.32604/cmc.2024.057714

[30] Wei, W., Li, X., Zhang, B., Li, L., Damaševičius, R., & Scherer, R. (2023). LSTM-SN: complex text classifying with LSTM fusion social network. *The Journal of Supercomputing*, *79*(9), 9558-9583. DOI:10.1007/s11227-022-05034-w

[31] Wei, X., Xu, G., Wang, H., He, Y., Han, Z., & Wang, W. (2019). Sensing users' emotional intelligence in social networks. *IEEE Transactions on Computational Social Systems*, *7*(1), 103-112. DOI:10.1109/TCSS.2019.2944687

[32] Xia, L., Xu, Y., Huang, C., Dai, P., & Bo, L. (2021, July). Graph meta network for multi-behavior recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 757-766). DOI:10.1145/3404835.3462972

[33] Yue, A., Mao, C., Chen, L., Liu, Z., Zhang, C., & Li, Z. (2022). Detecting changes in perceptions towards smart city on Chinese social media: A text mining and sentiment analysis. *Buildings*, *12*(8), 1182. DOI:10.3390/buildings12081182

[34] Zhang, L., Zhang, W., Wu, L., He, M., & Zhao, H. (2023). SHGCN: Socially enhanced heterogeneous graph convolutional network for multi-behavior prediction. *ACM Transactions on the Web*, *18*(1), 1-27. DOI:10.1145/3617510