

Continuous Temporal Line Graphs for News Recommendation: A Novel Approach to Modeling User Interest Drift

Zhaozhong Wu

wzz95@mail.nwpu.edu.cn

Northwestern Polytechnical University

Zibo Zhou

Northwestern Polytechnical University

Wei Dai

Northwestern Polytechnical University

Huimin Chen

Northwestern Polytechnical University

Research Article

Keywords: News Recommendation, Graph Neural Networks, Temporal Modeling, Interest Drift, Line Graph

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11300590/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Continuous Temporal Line Graphs for News Recommendation: A Novel Approach to Modeling User Interest Drift

Abstract

In personalized news recommendations, users’ interests frequently shift dynamically, a challenge that conventional node-based graph models find difficult to address. Many existing approaches depend heavily on similarity between static embeddings of users and items, but they do not effectively capture the causal relationships and influence between successive interactions. In this paper, we introduce a new framework, CTLGNN, which effectively captures long-term user preferences using a dynamic self-attention network while also tracking short-term interest drift with a continuous temporal line graph. Our approach converts user-news interactions into edge-nodes, allowing it to clearly capture the changing influence of adjacent interactions on user behavior. Evaluations conducted with real-world datasets demonstrate that CTLGNN surpasses other baseline models in terms of F1, AUC, and Recall metrics.

Keywords: News Recommendation, Graph Neural Networks, Temporal Modeling, Interest Drift, Line Graph

1. Introduction

Due to the rapid increase in online content, personalized news recommendation systems are now crucial for aiding users in efficiently finding information that interests them. User preferences are inherently dynamic and influenced by context, frequently showing temporary drifts in interest that static models struggle to accurately represent. User behavior can shift dramatically, such as when an individual who typically favors political news decides to explore sports content, perhaps due to an overload of political news. Capturing these changes in interest over time accurately continues to pose a significant challenge.

Recent research has identified that user preferences tend to fluctuate over time as a result of contextual changes, mood variations, or external occurrences. This phenomenon, known as *interest drift*, poses a major challenge for recommendation systems.

This behavior is both theoretically assumed and frequently observed in real-world datasets. As an illustration, Fig. 1 presents the distribution of reading categories for a user within the Adressa dataset. There is a noticeable shift from political news to sports and entertainment within a fairly brief period.

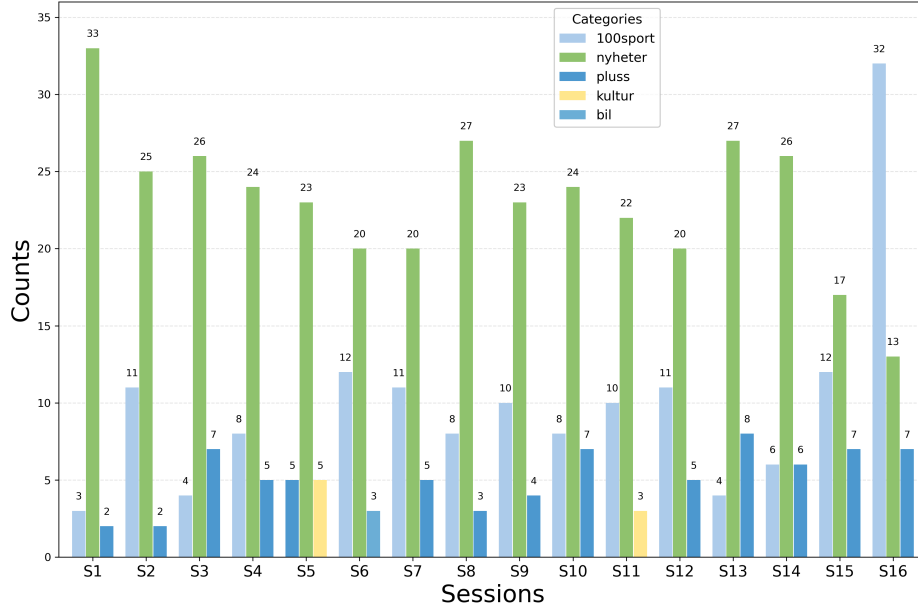


Figure 1: A real user’s reading history from the Adressa1week dataset is divided into 16 sessions, illustrating the shift in their interests from the "nyheter" category to "100sport".

Conventional methods for recommending news often frame the task as finding similarity between nodes, with user and news vectors developed separately and recommendation scores determined through similarity functions. While this assumption holds well under stable user behaviors, it frequently fails in practical situations where recent user interactions might conflict with their long-term preferences. Moreover, these approaches fail to consider the causal effects and structural relationships between one user-news connection and another, which are crucial for comprehending changing behavior trends.

To address these issues, we introduce an innovative framework called the Continuous Temporal Line Graph Neural Network (CTLGNN), which simultaneously captures users’ long-term preferences and short-term interest drift. In order to address long-term interests, we employ a dynamic self-attention graph neural network, inspired by DySAT, to effectively capture both structural and temporal dependencies within users’ past interactions. For capturing short-term dynamics, we present a temporal line graph model where each interaction between a user and a news piece is considered an edge-node. We utilize adjacent interactions to predict future behaviors via localized graph convolution. Our edge-focused approach permits our model to effectively analyze how recent actions impact future behaviors, addressing the constraints of similarity estimation based solely on nodes.

Figure 3 presents an overview of our model architecture. CTLGNN integrates news-level CNN encoders, DySAT-driven historical modeling, and GCN-based temporal line graph reasoning within a framework that supports end-to-end training. Our contributions can be summarized as follows.

- We introduce CTLGNN, an innovative recommendation framework that simultaneously captures both long-term and short-term user interests through an edge-node based approach, effectively balancing users’ historical preferences with their recent interests.
- We present a continuous temporal line graph framework that converts user-news interactions into edge-nodes. This allows the model to focus on events instead of node-similarity heuristics, thereby facilitating efficient modeling of the impacts of recent interactions.
- Instead of presuming similarity between users and news, we focus on interaction edges. Localized edge-wise convolutions are utilized to capture dependencies among clicks that occur close together in time. This approach maintains interaction-level signals and aligns recommendations with immediate behavioral changes, which is particularly useful when there is a sudden shift in interest. In such cases, similarity-based methods require multiple events to identify a reliable “like” signal.
- We conduct extensive evaluations on real-world datasets, and the results demonstrate the remarkable efficacy of our CTLGNN on the tasks of news recommendation.

2. Related Work

2.1. Dynamic Graph Modeling in Recommendation

The modeling of user preferences in recommendation systems has progressed from learning static representations to dynamically modeling behavior sequences. Since user interests inherently evolve over time, dynamic graph-based approaches have become a promising solution for capturing these temporal drifts. Generally, current dynamic graph neural networks (DGNNs) can be categorized into two types depending on their approach to time: *snapshot-based* and *event-based* models.

Snapshot-based models divide the temporal interaction stream into discrete graph snapshots. Traditional models like DySAT Sankar et al. (2020) utilize spatial and temporal self-attention mechanisms on individual slices, allowing for the separate capture of both structure and evolution. EvolveGCN Pareja et al. (2020) focuses on modeling how the weights of GCNs change over time, capturing the temporal evolution patterns in graph parameters. While these methods are commonly used, they create notable redundancy by encoding overlapping structures repeatedly and reduce detail by merging multiple events into broad intervals.

Event-based models avoid discretization and operate directly on asynchronous event streams. TGAT Xu et al. (2020) introduces time encoding into neighborhood aggregation, while CTDNE Nguyen et al. (2018) leverages time-respecting random walks. Recent advancements such as CTGNN Guo et al. (2022) and TCL Wang et al. (2021) employ temporal self-attention mechanisms or neural ODEs to effectively capture the continuous evolution of node embeddings over time. These methods have shown a higher capacity for adapting to fine-grained drift and are well-suited for real-time applications.

Although these advancements have been made, a significant number of DGNNs continue to be primarily **node-centric**, concentrating on updating user or item embeddings while often neglecting the semantics inherent in the interactions. Nevertheless, drifts in short-term preferences tend to be more effectively elucidated by *localized event patterns* rather than by global node states. We tackle this limitation by representing user-news interactions as edge nodes within a temporal line graph, which facilitates event-level reasoning that complements long-term user embedding learning.

2.2. Line Graph-based Interaction Reasoning

Line graph theory provides a systematic method to convert information from the edge level into a node-focused structure, rendering it especially apt for modeling interactions. Given a graph $G = (V, E)$, its line graph $L(G)$ maps each edge $e \in E$ to a node in $L(G)$, and connects two of such nodes if their corresponding edges in G share a vertex. This transformation allows models to capture dependencies between interactions rather than between entities.

SEAL Zhang and Chen (2018) pioneered this direction by treating link prediction as a subgraph classification task. This method involves extracting local enclosing subgraphs and employs GNNs to predict link presence, eliminating the need for manually designed heuristics. Nevertheless, it is computationally intensive and continues to work with static graphs. LineGNN Cai et al. (2021) enhances this idea by transforming subgraphs into line graphs and framing the task as a node classification problem. This approach cuts down on overhead while enabling comprehensive edge-level analysis using conventional GNNs.

Recent research Sun et al. (2023); Zhang et al. (2023b) emphasizes the critical benefits of modeling edge-to-edge dependencies, especially in fields like bioinformatics and social recommendations, where interactions are infrequent or exhibit temporal fluctuations. Our proposed CTLGNN broadens this approach through the introduction of a **event-driven temporal line graph**. Each interaction between a user and news is denoted as an edge node with a timestamp, and we establish temporal connections between them based on shared users or news entities, as well as their temporal closeness.

Demonstrated in Figure 2, our model converts the short-term recommendation problem into a node classification task focused on interaction events, thus reflecting the influence of users’ recent actions on their near-future preferences.

To conclude, this edge-centric modeling provides three primary advantages: i) It facilitates direct reasoning over interaction changes, which are crucial for capturing drifts in interest; ii) It maintains the temporal structure by not condensing behavioral sequences into static user embeddings; iii) It enhances long-term user modeling approaches like DySAT by considering the influence of links.

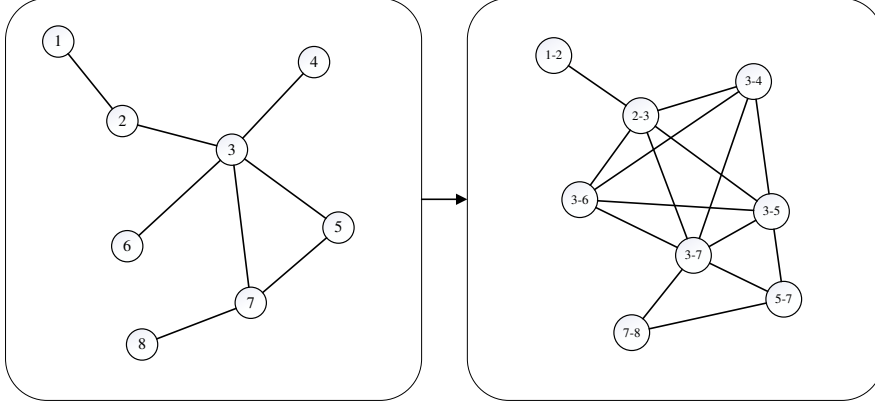


Figure 2: An illustration of transforming a user-news interaction graph into its corresponding line graph. Each user-news click becomes a node in the line graph, with edges between interactions sharing the same user or news entity.

3. The Proposed Framework

3.1. Framework Overview

We introduce an innovative two-view recommendation framework called **CTLGNN** (Continuous Temporal Line Graph Neural Network), designed to effectively capture users’ consistent preferences as well as short-term drifts in their behavior. As shown in Figure 3, the architecture is composed of three primary modules:

1. **News Encoder (Section 3.2):** A CNN-based encoder that transforms the title of each news article into a dense semantic embedding.
2. **Long-term User Modeling via DySAT (Section 3.3):** A snapshot-based dynamic GNN captures historical interactions between users and news, generating a representation of long-term user interests.
3. **CTLG-based Short-term User Modeling (Section 3.4):** A new encoder employing a line graph strategy considers each recent interaction (user-news click) as an edge-node and uses GCN propagation to model its representation within event-level neighborhoods.

Our research facilitates edge-centric reasoning on short-term behavior sequences by representing each interaction as a node within a temporal line graph. Rather than depending exclusively on user embeddings for predictions, we engage in node classification on edge-nodes. This involves assessing

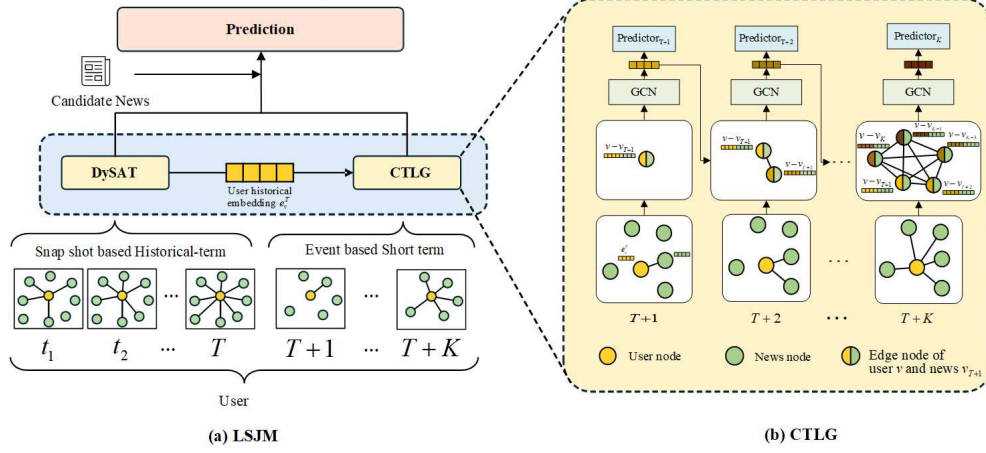


Figure 3: Framework of the proposed our model. It integrates long-term interest modeling via DySAT and short-term drift modeling via CTLG based on a temporal line graph.

the likelihood of a new user-news connection forming based on its temporal neighbors.

Ultimately, we forecast the personalized ranking scores for the news candidates by integrating scores derived from a scoring module, which employs short-term user interest embeddings from CTLG and long-term user interest embeddings from DySAT.

3.2. News Encoder

To obtain expressive representations of news articles, we adopt a CNN-based encoding module inspired by *DAN* (Zhu et al., 2019). Specifically, we focus on the textual content in the *title* of each news article, as it offers semantics appropriate for upcoming recommendation tasks.

Consider a news title being tokenized into a sequence of n words, with each word transformed into a d -dimensional vector using pre-trained word embeddings such as GloVe or Word2Vec. The matrix representing the input title is referred to as

$$\mathbf{T} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n] \in \mathbb{R}^{n \times d}.$$

We utilize convolutional filters $\mathbf{C} \in \mathbb{R}^{k \times d}$ with a window size of k on the input matrix, then apply a ReLU activation:

$$\mathbf{m}_i = \text{ReLU}(\mathbf{C} \cdot \mathbf{T}_{i:i+k-1} + b),$$

where $b \in \mathbb{R}$ represents a bias term, and $\mathbf{T}_{i:i+k-1}$ signifies the i -th sliding window.

We utilize max-pooling on the derived feature maps to highlight the most prominent features:

$$\mathbf{p} = \max_i \mathbf{m}_i.$$

Various filters with distinct window sizes are utilized to detect different n -gram patterns. Consider $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(r)}$ as the pooled outputs from r different filters. The ultimate news embedding is formed by combining all the pooled features together:

$$\mathbf{e}_{\text{news}} = \text{concat}(\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(r)}) \in \mathbb{R}^{r \cdot d'},$$

where d' represents the size of the output for each filter.

The embedding $\mathbf{e}_{\text{news}}$ functions as the initial node feature for every news node. Additionally, it is utilized to create edge-nodes when combined with user representations during the short-term modeling stage (Section 3.4).

The CNN-based encoder provides several advantages: (1) it provides rapid, parallel processing for extensive collections of articles; (2) it eliminates the need for the overhead associated with pre-training large-scale language models; (3) it generates compact and transferable embeddings that are applicable in both historical and short-term modules.

3.2.1. Embedding Integration and Edge-node Construction

Within our framework, the modules for both long-term and short-term modeling depend on meticulously crafted input embeddings, which are formed by combining user and news representations in a way that maintains temporal consistency.

News Representation. Each news article is encoded using the CNN-based encoder described in Section 3.2. These embeddings are precomputed during a preprocessing phase and remain constant during both the training and inference phases. That is, the representation $\mathbf{e}_n \in \mathbb{R}^{128}$ for each news article n remains constant and is utilized in both historical modeling and the creation of edge-nodes.

User Representation. In contrast, user embeddings are dynamic. In long-term modeling, DySAT encodes a user’s historical behavior across various time segments to generate a user across multiple time slices is encoded through DySAT, producing a stable long-term representation $\mathbf{e}_u \in \mathbb{R}^{128}$. In

short-term modeling, we keep a user representation that evolves over time and is revised following every interaction between the user and news.

In particular, when a user u engages with a news article n at time t , an edge-node (u, n) is created and added to the temporal line graph. The input feature is formed by combining the latest user embedding, $\mathbf{e}_u^{(t)}$, with the constant news embedding, $\mathbf{e}_n$.

$$\mathbf{z}_{(u,n)} = \text{concat}(\mathbf{e}_u^{(t)}, \mathbf{e}_n) \in \mathbb{R}^{256}.$$

This feature acts as the preliminary input for the GCN layers within the CTLG module. Once it has been propagated through the temporal line graph, the updated edge-node representation is employed to generate a new user embedding $\mathbf{e}_u^{(t+1)} \in \mathbb{R}^{128}$, fulfilling two roles:

- It is employed to predict the probability of future user-news interactions.
- It serves as the updated input for the subsequent interaction event involving user u .

Dimensional Consistency. Each news embedding is set to 128 dimensions. Consistent dimensionality is preserved across the pipeline as the output from the GCN in CTLG is also re-projected to 128 dimensions, ensuring it remains compatible with the fusion and prediction modules.

This mechanism creates a dynamic flow of representation in which the user’s interest state is progressively updated through a series of interaction events. The integration of static news semantics with the dynamically changing preferences of users allows for detailed modeling of short-term interest drifts in a continuous and adaptive way.

3.3. Long-term User Modeling via DySAT

We utilize the Dynamic Self-Attention Network (DySAT) Sankar et al. (2020) as the foundation for long-term modeling to capture users’ stable preferences developed through prolonged historical interactions. DySAT is a graph encoder that operates on snapshots, integrating both structural and temporal self-attention to represent how nodes evolve in dynamic graphs.

Snapshot Construction. We take a sequence of user-news interactions and divide the interaction stream into T temporal snapshots $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T$. Each snapshot $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ represents the graph at time t , featuring a shared set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}_t$ that are specific to that time.

Structural Self-Attention. In each snapshot $\mathcal{G}_t$, we derive a node embedding $\mathbf{h}_v^{(t)}$ by consolidating features from its neighbors employing a graph attention mechanism. For a node v with neighbors $\mathcal{N}_v^{(t)}$, the calculation is performed as follows:

$$e_{uv}^{(t)} = \text{LeakyReLU} \left(\mathbf{a}^\top \left[\mathbf{W}_s \mathbf{x}_u^{(t)} \parallel \mathbf{W}_s \mathbf{x}_v^{(t)} \right] \right),$$

$$\alpha_{uv}^{(t)} = \frac{\exp(e_{uv}^{(t)})}{\sum_{w \in \mathcal{N}_v^{(t)}} \exp(e_{uw}^{(t)})}, \quad \mathbf{h}_v^{(t)} = \sigma \left(\sum_{u \in \mathcal{N}_v^{(t)}} \alpha_{uv}^{(t)} \cdot \mathbf{W}_s \mathbf{x}_u^{(t)} \right).$$

Here, $\mathbf{x}_u^{(t)}$ represents the input feature for node u at time t , $\mathbf{W}_s$ refers to a trainable transformation matrix, $\sigma(\cdot)$ is a non-linear activation function (e.g., ELU), and $\parallel \cdot \parallel$ signifies the concatenation of vectors.

Temporal Self-Attention. In order to track the development of a node over various snapshots, DySAT employs temporal self-attention on its structural embeddings $\{\mathbf{h}_v^{(1)}, \dots, \mathbf{h}_v^{(T)}\}$. Let $\mathbf{H}_v \in \mathbb{R}^{T \times F}$ represent the sequence of embeddings for node v , which is then projected into the matrices for query, key, and value.

$$\mathbf{Q}_v = \mathbf{H}_v \mathbf{W}_q, \quad \mathbf{K}_v = \mathbf{H}_v \mathbf{W}_k, \quad \mathbf{V}_v = \mathbf{H}_v \mathbf{W}_v,$$

$$\beta_{ij} = \frac{\exp \left(\frac{(\mathbf{Q}_v \mathbf{K}_v^\top)_{ij}}{\sqrt{F}} + M_{ij} \right)}{\sum_{k=1}^T \exp \left(\frac{(\mathbf{Q}_v \mathbf{K}_v^\top)_{ik}}{\sqrt{F}} + M_{ik} \right)},$$

$$\mathbf{z}_v = \sum_{j=1}^T \beta_{ij} \cdot \mathbf{V}_v[j], \quad \text{where } M_{ij} = \begin{cases} 0 & \text{if } i \leq j, \\ -\infty & \text{otherwise.} \end{cases}$$

This attention mechanism is masked to maintain temporal causality, ensuring that a node does not attend to future snapshots. The ultimate long-term user embedding is represented as $\mathbf{z}_v \in \mathbb{R}^F$.

Output and Downstream Usage. The generated user embedding $\mathbf{z}_u$ is utilized in two manners:

- It serves as the user representation within the historical prediction head, where it is compared with candidate news embeddings for guidance through contrastive loss (Section 3.5).

- It sets up the user segment of the edge-node input within the CTLG module (Section 3.4), thereby connecting the long-term and short-term modeling components.

Discussion. Unlike RNN-based sequential models, DySAT is adept at capturing both neighborhood structures and temporal patterns, which makes it particularly proficient in modeling preference consistency and gradual shifts. Crucially, by maintaining per-snapshot embedding outputs, DySAT facilitates alignment with recent interactions in the event stream and supports fine-grained integration with edge-node construction within the CTLG pipeline.

3.4. CTLG-based Short-term User Modeling

DySAT is proficient in representing users’ long-term preferences but falls short in dealing with quick and temporary changes in interest. To counter this limitation, we present **CTLG-based Short-term User Modeling**, an innovative module designed to model short-term user behavior at the granularity of interaction events.

Line Graph Construction. In the initial bipartite graph connecting users and news, each edge (u, n) stands for a click event. In the line graph, we convert this edge into a node, represented as $v_{(u,n)}$. Two edge-nodes, $v_{(u_1,n_1)}$ and $v_{(u_2,n_2)}$, are linked in the line graph if their corresponding edges share either the same user or the same news article.

$$(v_{(u_1,n_1)}, v_{(u_2,n_2)}) \in \mathcal{E}_{\text{line}} \iff u_1 = u_2 \text{ or } n_1 = n_2.$$

This shift enables us to represent how recent interactions impact future user actions by concentrating on the relational dynamics between interaction events.

Edge-node Feature Construction. Every edge-node is set up by combining the embeddings of both the user and the news participating in the interaction:

$$\mathbf{z}_{(u,n)}^{(0)} = \text{concat}(\mathbf{e}_u^{(t)}, \mathbf{e}_n) \in \mathbb{R}^{256}.$$

where $\mathbf{e}_u^{(t)}$ is the user embedding at the present time step, initialized by DySAT, while $\mathbf{e}_n$ denotes the fixed CNN-based news embedding.

Graph Propagation. We utilize a two-layer GCN Kipf and Welling (2016) on the constructed temporal line graph to derive context-aware edge-

node embeddings:

$$\mathbf{z}_{(u,n)}^{(k+1)} = \sigma \left(\mathbf{z}_{(u,n)}^{(k)} + \beta \cdot \sum_{d \in \mathcal{N}_{(u,n)}} \mathbf{z}_d^{(k)} \right) \mathbf{W}^{(k)}.$$

Here, $\mathcal{N}_{(u,n)}$ denotes the neighboring edge-nodes to (u, n) , β serves as a normalization factor, and $\mathbf{W}^{(k)}$ are trainable weights. The activation function $\sigma(\cdot)$ is defined as ReLU.

Temporal User Update. Following every interaction (u, n) , we refresh the user embedding by taking the output from the GCN at the corresponding edge-node.

$$\mathbf{e}_u^{(t+1)} = \text{proj} \left(\mathbf{z}_{(u,n)}^{(K)} \right), \quad \mathbf{e}_u^{(t+1)} \in \mathbb{R}^{128}.$$

This updated embedding captures the immediate impact of the latest interaction and serves the purpose of: i) Creating the input for the subsequent edge-node associated with u ; ii) Computing prediction scores with candidate news articles; iii) Providing real-time modeling of interest drift.

Event-level Prediction. For each user’s current state $\mathbf{e}_u^{(t)}$ following GCN propagation, we calculate scores with each candidate news article $\mathbf{e}_n^{\text{cand}}$ using a score fusion module, as detailed in Section 3.5. Predicting the next clicked article is conceptualized as a ranking problem based on these scores.

Advantages of CTLG. Our approach to short-term modeling provides several advantages: i) It explicitly models the temporal dependency among interaction events rather than collapsing them into user summaries; ii) It adapts user representations in a continuous, event-driven manner; iii) It supports fine-grained link reasoning via edge-centric GCN propagation.

Visualization. Figure 2 demonstrates how user-news clicks are transformed into edge-nodes, leading to the creation of the temporal line graph. The resulting structure enables localized and temporal-aware reasoning over recent behaviors.

3.5. Score Fusion and Training Objective

We develop a **score fusion module** to combine user preferences from the long-term and short-term modeling branches, integrating the outputs from DySAT and CTLGNN for the final prediction. This design enables the system to take into account both long-term and short-term user interests when evaluating and ranking potential news articles.

Scoring from DySAT. Let $\mathbf{e}_u^{\text{long}}$ denote the user embedding obtained from DySAT, and let $\mathbf{e}_n^{\text{cand}}$ denote the embedding of a candidate news article. A similarity score is calculated utilizing a bilinear interaction function:

$$s^{\text{long}}(u, n) = (\mathbf{e}_u^{\text{long}})^\top \mathbf{W}_{\text{long}} \mathbf{e}_n^{\text{cand}},$$

where $\mathbf{W}_{\text{long}} \in \mathbb{R}^{128 \times 128}$ represents a learnable parameter matrix.

Scoring from CTLGNN. Let $\mathbf{e}_u^{\text{short}}$ represent the updated user embedding following CTLGNN propagation. We calculate a similar bilinear score:

$$s^{\text{short}}(u, n) = (\mathbf{e}_u^{\text{short}})^\top \mathbf{W}_{\text{short}} \mathbf{e}_n^{\text{cand}}.$$

Score Fusion. To combine the two scores, we apply a learnable weighted sum:

$$s(u, n) = \lambda \cdot s^{\text{short}}(u, n) + (1 - \lambda) \cdot s^{\text{long}}(u, n),$$

where $\lambda \in [0, 1]$ is a fixed hyperparameter or is dynamically predicted by a gating network:

$$\lambda = \sigma(\mathbf{w}^\top \cdot \text{concat}(\mathbf{e}_u^{\text{short}}, \mathbf{e}_u^{\text{long}}) + b).$$

Here, $\mathbf{w}$ and b are learnable parameters, and $\sigma(\cdot)$ is the sigmoid activation.

Training Objective. For user u , with a collection of candidate news articles $\mathcal{N}_u$ and the ground-truth clicked news n^+ , we perform a softmax operation across all possible candidates:

$$\hat{y}_n = \frac{\exp(s(u, n))}{\sum_{n' \in \mathcal{N}_u} \exp(s(u, n'))}.$$

Loss is characterized as the standard cross-entropy applied to the ground-truth positive sample:

$$\mathcal{L}_{\text{pred}} = -\log \hat{y}_{n^+}.$$

Total Objective. To ensure consistent learning between long- and short-term modules, we optionally apply a representation consistency regularization

$$\mathcal{L}_{\text{consist}} = \|\mathbf{e}_u^{\text{short}} - \mathbf{e}_u^{\text{long}}\|_2^2.$$

The ultimate goal of the training is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pred}} + \gamma \cdot \mathcal{L}_{\text{consist}}.$$

where γ is a hyperparameter that controls the strength of consistency regularization.

Discussion. Our fusion mechanism offers adaptable trade-offs between stable and rapidly evolving preferences. Unlike conventional late-fusion or fixed-weighting approaches, our learnable fusion adjusts to individual user behavior, making it versatile for various recommendation scenarios.

4. Experiments

4.1. Dataset and Experimental Setup

We perform comprehensive experiments on popular news recommendation datasets to assess how well our proposed model captures both stable and evolving user preferences.

Dataset. In our experiments, we utilize the MIND-SMALL Wu et al. (2020b) subset, which comprises around 100,000 users, 160,000 news articles, and 6 million impression logs. Every news article is associated with a title, abstract, category, and subcategory. Consistent with established practice Wu et al. (2020b), we combine the title and abstract to represent the article.

The Adressa dataset was released in 2017, a collaborative effort by the Norwegian University of Science and Technology (NTNU) and the Trondheim-based local newspaper, Adressavisen, as part of the RecTech project focused on recommendation technology Zhu et al. (2019). This dataset is structured around events, containing user IDs and their anonymous interactions with news. Each event comprises 18 distinct attributes of the news. In our model, we do not utilize all these attributes. Instead, we choose user ID, news ID, timestamp, and title to form our dataset.

The original dataset is available in two versions: one containing data for a single week (from January 1 to January 7, 2017) and the other spanning 10 weeks (from January 1 to March 31, 2017). Additionally, we modified the dataset’s contents by excluding users with fewer than 10 records and those with 4 or fewer sessions, ensuring there is sufficient information to extract historical interest. The data is divided sequentially into two sections, ensuring no more than five news pieces for the short-term model, while maintaining at least one for the historical model. The specifics of the data are displayed in Tab.1.

News Representation. We employ a pre-trained CNN encoder to obtain semantic representations of news articles, akin to the approach in (Zhu et al., 2019). The encoder takes the title and abstract as input and produces

Table 1: Statistics of the dataset

Number	Adressa1week	Adressa10week	MIND-small
users	5050	20907	50000
events	129076	1075133	230117
news	20428	81018	93698

a 128-dimensional fixed-length embedding that stays unchanged throughout the training process. These embeddings function as both node features in DySAT and components of edge-node features in CTLGNN.

User Behavior Slicing. The interaction history of each user is organized chronologically and segmented into temporal slices for DySAT over the same prolonged period. In CTLGNN, we take the same number click events to build the continuous temporal line graph.

The candidate set. In line with common practice, each session’s candidate set includes the ground-truth clicked news and four randomly selected negative articles from the same impression.

Implementation Details. In our experiments, all competitive models are developed using PyTorch and are trained on a server that features a NVIDIA RTX 3080 Ti GPU along with 64 GB of RAM. The Adam optimizer is utilized with a starting learning rate of 0.001. We implement early stopping depending on validation performance.

In our news encoder module, the configuration for the CNN filter is $[c_1, c_2] = [2, 50]$ with a stride of $[2, 2]$, and the number of pooling filters is configured to be 8. Following the recommendation in Zhu et al. (2019), we have chosen the word embedding dimension to be $d = 100$, where n represents the maximum length of the title. All embeddings, such as those for users, news, and edge-node representations, are defined with 128 dimensions. The CTLG module employs a 2-layer GCN to capture local structures, while DySAT is configured with 2 structural and 2 temporal self-attention heads. In historical modeling, we use 8 multi-heads as per DySAT’s default setting. The batch size is set at 512, training spans 200 epochs, and we maintain a constant 16 temporal snapshots. The final loss incorporates a trade-off hyperparameter λ set at 0.8, while the consistency loss weight γ is configured at 0.1. The remaining crucial parameter settings for the baseline models adhere to the configurations detailed in DAN (Zhu et al., 2019).

Table 2: Hyperparameter Settings

Hyperparameter	Value
Word embedding dimension d	100
News/User/Edge-node embedding	128
CNN filters (size, stride)	(2,50), (2,2)
CNN pooling filters	8
GCN layers in CTLGNN	2
Self-attention heads in DySAT	2 (structural), 2 (temporal)
Multi-heads for historical modeling	8
Temporal snapshots	16
Batch size	512
Epochs	200
Learning rate	0.001
Dropout rate	0.2
Negative samples	4
Loss weight λ	0.8
Consistency weight γ	0.1

4.2. Baselines

We assess our proposed framework by comparing it to a wide array of baseline models, which include semantic-based, sequential, knowledge-enhanced, and graph-structured models. Each model is trained using uniform data splits and candidate sampling methods.

DKN (Deep Knowledge-Aware Network) (Wang et al., 2018): DKN incorporates entities from knowledge graphs into news representation through a multi-channel CNN. This method captures both the word-level semantics and the entity-level signals, yet it lacks the ability to model user sequential behavior. *As a static approach enhanced by knowledge, its performance is typically surpassed by sequential models like DAN.*

DAN (Deep Attention Network) (Zhu et al., 2019): DAN models user interests via a CNN-based attentive representation of news and an RNN-based encoder of sequential clicks. *Its integration of personalized temporal dynamics enables it to surpass static methods like DKN.*

DySAT (Dynamic Self-Attention Network) (Sankar et al., 2020): DySAT develops dynamic node embeddings by simultaneously utilizing structural and temporal self-attention on graphs that change over time. *It effectively captures the evolution of user-news interactions over time, yet it falls*

short in modeling fine-grained behavior at the edge level.

GRU (Gated Recurrent Unit) (Chung et al., 2014): The GRU is an effective temporal baseline that converts user click sequences into hidden states, showcasing changing preferences. *Its performance underscores the significance of maintaining temporal continuity in user modeling.*

CPRS (Contextualized Personalized Recommender System) (Wu et al., 2020a): CPRS integrates personalized and contextual elements through deep content-aware encoders. *While it is designed to allow for personalized adaptation, it does not incorporate graph-based structural signals.*

CNE-SUE (Collaborative News Encoding and Structural User Encoding) (Mao et al., 2021): CNE-SUE utilizes collaborative filtering for encoding news and employs graph-based modeling for encoding users. Its dual-branch architecture offers moderate enhancements compared to traditional methods that focus solely on content.

TLAM-NR (News recommendation method for session-based neural neural network with two-layer attention mechanism) (Wang and Hu, 2021): TLAM-NR utilizes multi-head attention to concurrently capture both long-term and short-term user interests with temporal awareness. *Its capability to integrate time dependencies across multiple scales gives it a competitive edge over temporal models.*

NUG (Neural User Graph) (Manh Nguyen et al., 2022): NUG constructs a graph encoder focused on users by integrating user profiles and click contexts alongside GNN propagation. *This collaborative user modeling aids in capturing inter-user consistency, despite the absence of fine-grained edge updates.*

CupMar (Contextual User Profiling with Multi-aspect News Representation) (Tran et al., 2023): CupMar integrates user context with various aspects of news content, including the title, abstract, and metadata, to create personalized representations. *While the combination of diverse features enhances the quality of these representations, it does not incorporate structural modeling.*

NAML (Neural Attentive Multi-view Learning) (Wu et al., 2019b): NAML utilizes hierarchical attention across title, abstract, and entity features, incorporating a user-specific encoder. *This hierarchical structure facilitates efficient content aggregation, but it does not support dynamic user adaptation.*

LSTUR (Long- and Short-term User Representations) (An et al., 2019): LSTUR integrates long-term user embeddings with short-term se-

quential clicks using GRU. *This combined method offers better personalization than models that rely only on content.*

NRMS (Neural News Recommendation with Multi-head Self-Attention) (Wu et al., 2019a): NRMS employs multi-head self-attention to effectively and flexibly capture news content and user behaviors. *Although it achieves remarkable outcomes through parallel attention modeling, it neglects the structural graph information.*

GNewsRec (Zhou et al., 2022): GNewsRec utilizes Graph Neural Networks (GNNs) to encode user-news graphs and incorporates textual representations of news. *Although it effectively captures static relational patterns, it falls short in modeling the evolution of user preferences.*

Hi-Fi Ark (Hierarchical Fusion for Attention-based Recommendation Knowledge) (Zhang et al., 2023a): Hi-Fi Ark applies multi-level user interest fusion using hierarchical attention to capture fine-grained preferences. *The deep fusion design results in competitive performance, establishing it a strong baseline.*

Ours (Full Model): Our framework integrates DySAT-based long-term structural modeling with CTLG-based short-term edge-level reasoning, augmented by a dynamic score fusion module. *It effectively captures both global preference evolution and local interest drift, leading to consistent performance gains across datasets.*

4.3. Evaluation Metrics and Protocol

We assess the performance of our proposed model by employing a comprehensive array of evaluation metrics that are widely utilized in personalized news recommendation tasks, emphasizing both the quality of ranking and the effectiveness of classification.

1) **F1 Score (F1)**: it is the harmonic mean of precision and recall, which can be formulated by

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

It reflects the balance between false positives and false negatives, particularly important in binary detection tasks and in imbalanced scenarios.

2) **Area Under the ROC Curve (AUC)**: it measures the probability that a randomly chosen positive sample is ranked above a randomly chosen

negative sample. It is defined by

$$AUC = \int_0^1 TPR(FPR) dFPR \quad (2)$$

A higher AUC indicates better model discrimination capability.

3) **Recall**: it calculates the ratio of relevant items accurately suggested out of all relevant items:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

4) **Average Precision (AP)**: it calculates the area under the precision-recall curve, as defined by

$$AP = \sum_n (\text{Recall}_n - \text{Recall}_{n-1}) \cdot \text{Precision}_n \quad (4)$$

This metric offers a single numerical summary of precision-recall trade-offs across various thresholds.

Our objective is to prioritize candidate news articles for each user session. During each session, one true positive and four negative samples are utilized. Both F1 and AUC metrics are applied in these tasks. In a multi-class setting, we present micro-averaged and macro-averaged F1 and AP metrics to demonstrate performance across both high- and low-frequency classes.

4.4. Main Results

We compare the performance of our proposed approach with 14 leading methods across three datasets: Adressa1week, Adressa10week, and MIND-small. The evaluation is performed using four metrics: F1, AUC, Recall, and Average Precision (AP). Table 3 provides a detailed summary of the results, while Fig. 4 and Fig. 5 illustrate the core performance patterns.

We additionally present the comparisons between our model and other competing baselines with regard to each metric. As illustrated in Fig. 4, our method attains the top F1 score on all three datasets. The improvements are particularly remarkable on Adressa1week and MIND-small datasets, achieving F1 scores of 84.15% and 72.91% respectively, which exceed the next best baselines by over 2.7%. These results emphasize the success of our temporal user modeling as well as the incorporation of short-term interest fusion.

As shown in Fig. 5, our model not only attains competitive AUCs, with a peak of 99.18% on Adressa1week, but also provides impressive recall and AP

Table 3: Performance comparison of different models across three datasets. Our method achieves consistent superiority under all metrics.

Model	Adressa1week				Adressa10week				MIND-small			
	F1	AUC	Recall	AP	F1	AUC	Recall	AP	F1	AUC	Recall	AP
DKN	74.56	67.89	75.83	61.78	62.63	57.02	64.70	51.98	61.89	57.07	60.02	50.12
DAN	76.44	70.78	77.56	65.82	63.88	60.87	68.25	56.61	63.67	59.43	65.41	56.37
DySAT	75.65	78.96	82.67	73.34	66.29	68.19	72.37	63.33	64.68	65.44	70.71	59.50
GRU	78.32	74.79	79.77	69.18	69.98	66.45	71.80	61.47	67.09	60.48	67.86	57.72
CPRS	79.84	75.66	80.45	70.59	71.65	66.93	69.19	62.45	67.86	61.43	68.38	58.80
CNE-SUE	71.54	69.01	73.01	64.39	64.31	62.08	64.95	57.95	59.67	61.50	60.86	55.09
TLAM-NR	80.43	75.64	79.64	69.28	70.78	68.83	74.07	63.04	68.36	60.45	67.24	56.67
NUG	75.88	71.67	74.53	65.65	65.26	60.92	66.33	55.80	67.73	67.56	63.35	56.46
CupMar	74.87	78.43	81.52	72.67	64.85	66.74	70.96	62.46	64.01	64.15	69.78	58.33
NAML	77.01	72.33	78.80	67.44	67.20	64.45	69.11	60.45	66.83	63.12	67.20	58.66
LSTUR	78.65	74.80	79.02	68.90	68.30	65.01	70.32	61.34	68.22	64.45	68.85	59.20
NRMS	80.22	76.31	80.85	71.22	70.01	66.89	72.01	62.23	69.34	65.78	70.45	60.44
GNewsRec	79.02	75.50	79.90	70.10	68.91	65.73	70.45	61.78	68.98	64.91	69.12	59.87
Hi-Fi Ark	81.44	78.10	81.63	73.56	71.40	68.11	72.30	63.10	71.55	68.10	70.88	61.33
Ours	84.15	99.18	85.49	89.05	72.88	69.91	71.92	64.95	72.91	69.47	71.03	62.34

scores. While certain baseline models, such as DySAT and NRMS, demonstrate strong performance on particular datasets, their effectiveness varies between datasets, suggesting a lack of consistent robustness. In contrast, our model consistently demonstrates superior performance, indicating its ability to effectively encompass both long-term and short-term user interests under various temporal and structural scenarios.

Our model demonstrates a clear advantage, especially with datasets characterized by shorter user histories, such as Adressa1week, underscoring the success of our event-driven short-term modeling approach. This indicates that our model is appropriate for real-world situations where user behavior shifts quickly and historical data is limited or obsolete.

In contrast to content/knowledge-static baselines (DKN, CPRS, NAML, CupMar), our model is dynamic—combining long-term graph dynamics (DySAT) with edge-level short-term reasoning (CTLG) and fusing them through a learnable scoring module. This approach reduces false correlations and detects recent drifts in interest, resulting in an improved F1 score.

Compared to sequence-only temporal models like GRU, DAN, LSTUR, NRMS, and TLAM-NR, our graph-aware modules enhance explicit message passing through the interaction graph, effectively capturing global dependencies and minimizing popularity bias. Static-graph methods such as GNewsRec, NUG, and CNE-SUE face limitations in dynamic scenarios because stale edges fail to capture changes in preferences. Hi-Fi Ark secures the second-

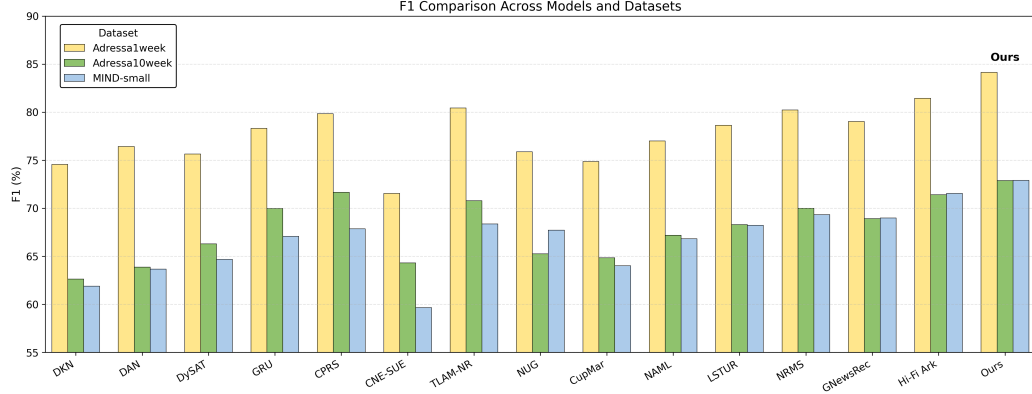


Figure 4: F1 score comparison across all models and datasets. Our model consistently outperforms all baselines.

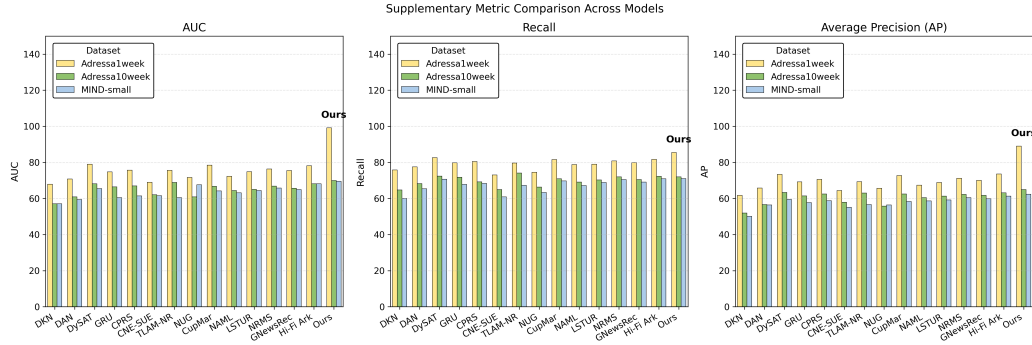


Figure 5: Performance comparison on AUC, Recall, and Average Precision (AP) metrics. Our model remains consistently competitive across all measures.

highest performance due to powerful hierarchical content aggregation. However, it gains the most from comprehensive fields (title+abstract+entities); when limited to title-only settings, its lead diminishes.

Finally, although DySAT effectively captures long-term evolution, it falls short in handling event-based edge reasoning and is less responsive to sudden short-term changes. Our CTLGNN addresses these shortcomings, which accounts for the overall improvements in F1 scores. The empirical findings demonstrate that our approach attains state-of-the-art performance and shows robust generalization capabilities across various datasets and evaluation standards.

4.5. Ablation Study

In order to assess the role of each key component in our framework, we perform a set of ablation studies using the Adressalweek and MIND-small datasets. The subsequent variants are taken into account.

- **Ours w/o Short-term Interest Graph:** This variant removes the short-term line graph module and inputs historical user representations directly into the final prediction layer.
- **Ours w/o DySAT:** It is obtained by removing the DySAT module and keeping just the event-wise short-term modeling based on line-graph GCN, thus restricting the model to focus on local temporal dependencies.
- **Ours w/o Interest Fusion:** This variant removes the interest fusion module and relies solely on the most recent short-term representation as the final embedding, disregarding temporal consistency.
- **Ours w/o Consistency Loss:** In this variant, the consistency regularization term linking long-term and short-term representations is turned off to evaluate its impact on overall stability and generalization.

Table 4: Ablation results on Adressalweek and MIND-small datasets.

Model Variant	Adressalweek		MIND-small	
	F1	AUC	F1	AUC
Full Model (Ours)	84.15	99.18	72.91	69.47
w/o Short-term Graph	79.04	94.61	68.73	65.12
w/o DySAT	80.58	96.35	69.25	66.10
w/o Interest Fusion	81.77	97.88	70.19	67.14
w/o Consistency Loss	82.24	98.04	70.91	67.89

According to Table 4, every variant results in significant declines in performance, emphasizing the critical role of each component. The most significant decrease occurs when the short-term interest graph is removed, resulting in F1 scores decreasing by 5.1% and 4.2% for the two datasets, respectively.

This confirms the importance of incorporating fine-grained event-level user behavior by using line graph modeling.

Turning off the DySAT module results in a significant drop, especially in tracking the long-term evolution of users. Eliminating the interest fusion module also leads to a decline in performance, indicating that the temporal integration of user interests is essential for effective representation learning.

Finally, excluding the consistency loss results in a moderate but steady decline, suggesting that the alignment of long-term and short-term embeddings enhances model stability and reduces representational drift over time. These results collectively support the reasoning behind our model architecture’s design, as each component contributes by complementing others in understanding the dynamic and evolving nature of user preferences.

4.5.1. Study of Short-term News Count

Subsequently, we assess the impact of the quantity of short-term interactions depicted in the temporal line graph on the performance of our model.

According to Table 5, incorporating five short-term news interactions enables our model to achieve the highest performance. Raising the number above five does not consistently result in benefits and can even cause a decrease in performance, probably due to the inclusion of redundant or outdated user preferences. On the other hand, having fewer than five interactions might not offer enough contextual hints to create informative temporal line graphs, which can lead to less than optimal F1 and AUC scores.

This result demonstrates that judiciously choosing the window size for short-term interest modeling is essential for maintaining a balance between expressiveness and noise reduction. Our pre-filtered dataset is designed to match the five-news configuration, removing users with fewer than five interactions to maintain consistency in modeling. Figure 6a demonstrates how performance varies with the quantity of short-term news.

4.5.2. Study of GCN Layer Depth

This study examines the effect of varying the number of GCN layers on our module designed for modeling short-term interest. According to Table 6, the performance reaches its highest point when two GCN layers are utilized. The trends are visually validated in Figure 6b. Specifically, a one-layer GCN is inadequate for capturing higher-level connectivity, resulting in less than optimal performance. When the depth exceeds two layers, performance diminishes as a result of over-smoothing, a phenomenon where node

Table 5: Effect of Short-term News Count on Model Performance

News Count	F1	AUC	Recall	AP
1	78.12	73.45	75.89	69.22
3	81.56	76.21	78.63	71.45
5	83.67	78.56	80.13	72.98
7	82.33	77.10	79.56	71.87
9	80.15	75.42	77.34	70.11

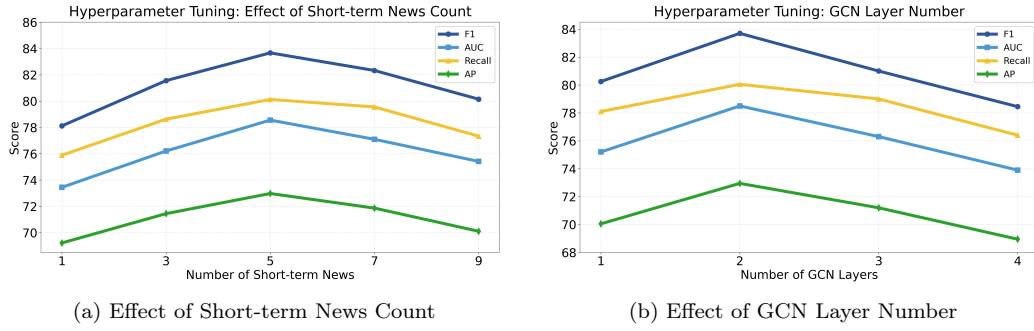


Figure 6: Comparison of model performance under different settings.

representations become indistinguishable and noise builds up.

This outcome is consistent with earlier research in graph learning, demonstrating that GCNs of moderate depth frequently strike a better balance between expressiveness and noise. As a result, our final model uses a two-layer GCN as the standard setup.

Table 6: Effect of GCN Layer Number on Model Performance

GCN Layers	F1	AUC	Recall	AP
1	80.33	75.22	78.12	70.11
2	83.67	78.56	80.13	72.98
3	81.12	76.34	79.01	71.23
4	78.45	73.89	76.45	69.04

4.6. Case Study

To showcase both the effectiveness and clarity of our model, we offer a case study that includes a randomly selected sample user. Figure 7 illustrates

the predicted score and ultimate ranking of the target news article among 50 candidates, with NRMS positioned at 17/50, DySAT at 6/50, and our method at 1/50, obtaining the highest score. It also shows the distribution of importance weights over the user’s recent interactions, where t-0 represents the most recent click.

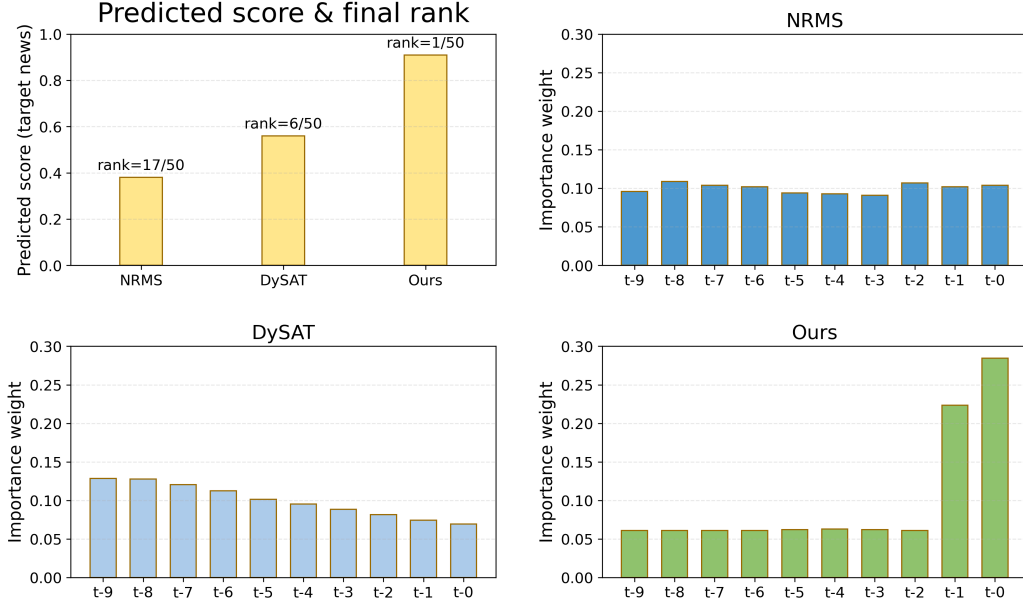


Figure 7: A case study on a sample user is presented. In the top-left, the target news’ predicted score (displayed as bars) and its final rank (shown as labels) among 50 candidates are illustrated. The top-right, bottom-left, and bottom-right sections show the importance-weight distributions over recent interactions, where t-0 represents the most recent interaction, for NRMS, DySAT, and our approach, respectively.

NRMS assigns a low ranking to the target news and displays a nearly uniform distribution of importance, suggesting it fails to adequately capture the user’s recent change in interest. DySAT achieves superior performance by incorporating temporal dynamics within a global temporal interaction graph. However, it tends to overemphasize older items, which results in the oversight of fleeting short-term signals.

In contrast, our model integrates short-term event indicators alongside long-term preferences simultaneously. The short-term representation highlights a rise in interest during recent sessions, resulting in a rank-1 prediction. In the importance plot, our model assigns the highest weights to the two lat-

est clicks (t-1, t-0), aligning with the target article.

4.7. Effect of User Interaction Volume

In order to explore the impact of user activity on model performance, we classify users by the quantity of their past interactions. As indicated in Table 7 and Figure 8, the model’s performance consistently enhances as user engagement rises.

Notably, users with 5–9 interactions exhibit the lowest performance on all metrics, which is anticipated due to the scarcity of behavioral signals. With a rise in the number of interactions, the model gains from a more comprehensive context, which improves the learning of user representation. After 100 interactions, the benefits level off, suggesting that improvements in representation have reached a saturation point.

These findings confirm the success of our design, which excludes users with less than five interactions and adjusts user embeddings in response to recent behavior. It further illustrates the model’s robustness in scenarios with both sparse and dense activity.

Table 7: Performance Across Different User Activity Levels

User Group	F1	AUC	Recall	AP
5–9 interactions	72.50	80.85	75.70	61.20
10–29 interactions	77.30	84.35	78.90	65.45
30–49 interactions	80.90	87.05	82.00	68.40
50–99 interactions	83.15	88.55	84.20	71.20
100–199 interactions	84.00	89.00	85.10	72.40
200+ interactions	84.70	89.35	85.80	73.05

5. Future Work

While our proposed framework shows robust results using dynamic user modeling and temporal line graphs, there are still numerous promising avenues for future research.

- **Cold-start Enhancing.** While our model is capable of generalizing fairly well to users who have limited interaction histories, it is not specifically crafted to tackle the cold-start issue. In future studies,

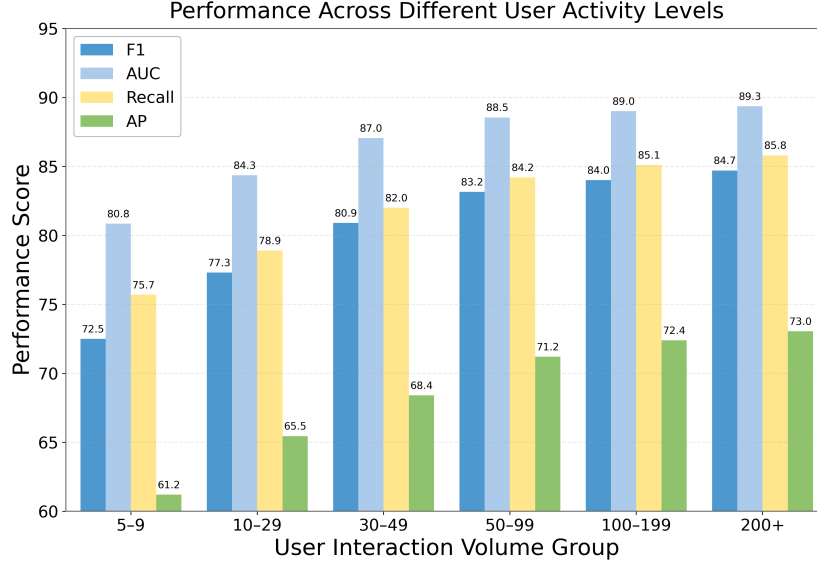


Figure 8: Model performance under different user interaction volumes.

incorporating user demographic data, external profiles, or initializing prompts from large pre-trained models could enhance personalization for new users.

- **Model compression and efficiency.** Since our framework incorporates the dynamic creation of graphs and iterative convolution processes on time-sensitive line graphs, maintaining computational efficiency is crucial for real-time applications. Future work includes developing lightweight versions of temporal GCNs, implementing pruning techniques for short-term interest graphs, or utilizing self-supervised pretraining to lower inference costs.
- **Multi-modal and contextual information.** Integrating more diverse modalities like images, videos, or user feedback signals (e.g., dwell time, scroll depth) could improve the modeling of user intent. Additionally, incorporating contextual features such as the time of day or indicators of breaking news can augment temporal sensitivity and responsiveness.
- **Fine-grained interest disentanglement.** At present, our model integrates user interests in a comprehensive manner at every time step.

A new research avenue is focused on separating various elements of user interests—such as political, entertainment, and regional preferences—by applying methods like contrastive learning or interest factorization, which aim to enhance interpretability and controllability.

- **Interactive and continual recommendation.** In practical scenarios, recommendation systems typically function in online environments where user preferences change over time. Future studies could investigate reinforcement learning or continual learning frameworks to facilitate adaptive user modeling through dynamic feedback mechanisms.

These extensions have the potential to enhance both the efficiency and applicability of our framework in more realistic and large-scale recommendation settings.

6. Conclusion

In this study, we introduce a new framework for news recommendation that is aware of changes in user interest. This framework combines both long-term and short-term preference modeling within a unified framework that combines a snapshot-based dynamic self-attention network with an event-level temporal line graph.. The design utilizes a dynamic self-attention network to capture long-term user behavior and constructs event-level line graphs to explicitly model fine-grained short-term interest changes over time. A customized score fusion mechanism enhances the balance between the contributions of the two components, allowing the system to flexibly adapt to changes in user behavior dynamics.

Comprehensive tests using real-world news datasets show that our method consistently surpasses robust baseline models in various evaluation metrics, such as F1, AUC, Recall, and Average Precision. Ablation studies confirm the importance of each essential module, and further analyses demonstrate that our framework consistently performs well even with different degrees of user activity and interaction sparsity. The attention visualizations and case studies offer additional understanding of how the model interprets textual signals and adjusts user representations to accommodate interest drift.

This study presents an adaptable and understandable approach for capturing changing user preferences in real-time recommendation environments. Looking ahead, we intend to investigate more efficient methods for constructing line graphs in order to lower computational demands and examine how

to integrate multi-modal content features, including images and videos. Additionally, broadening the framework to include other time-sensitive recommendation areas, like short videos and personalized feeds, offers a promising avenue for wider applicability.

7. Declarations

7.1. Ethics approval and consent to participate

Not applicable.

7.2. Availability of data and material

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

7.3. Competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

- An, J., Wu, F., Wu, C., Liu, Q., Huang, Y., Xie, X., 2019. Neural news recommendation with long-and short-term user representations, in: ACL, pp. 336–345.
- Cai, L., Li, J., Wang, J., Ji, S., 2021. Line graph neural networks for link prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 5103–5113.
- Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 .
- Guo, J., Han, Z., Su, Z., Li, J., Tresp, V., Wang, Y., 2022. Continuous temporal graph networks for event-based graph data. arXiv preprint arXiv:2205.15924 .
- Kipf, T.N., Welling, M., 2016. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 .

- Manh Nguyen, C., Xuan Bach, N., Minh Phuong, T., 2022. Combining user specific and global news features for neural news recommendation, in: Asian Conference on Intelligent Information and Database Systems, Springer. pp. 648–660.
- Mao, Z., Zeng, X., Wong, K.F., 2021. Neural news recommendation with collaborative news encoding and structural user encoding. arXiv preprint arXiv:2109.00750 .
- Nguyen, G.H., Lee, J.B., Rossi, R.A., Ahmed, N.K., Koh, E., Kim, S., 2018. Continuous-time dynamic network embeddings, in: Companion proceedings of the the web conference 2018, pp. 969–976.
- Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C., 2020. Evolvecn: Evolving graph convolutional networks for dynamic graphs, in: Proceedings of the AAAI conference on artificial intelligence, pp. 5363–5370.
- Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H., 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks, in: Proceedings of the 13th international conference on web search and data mining, pp. 519–527.
- Sun, S., Tian, H., Wang, R., Zhang, Z., 2023. Biomedical interaction prediction with adaptive line graph contrastive learning. *Mathematics* 11, 732.
- Tran, D.H., Sheng, Q.Z., Zhang, W.E., Tran, N.H., Khoa, N.L.D., 2023. Cupmar: A deep learning model for personalized news recommendation based on contextual user-profile and multi-aspect article representation. *World Wide Web* 26, 713–732.
- Wang, H., Hu, Y., 2021. News recommendation method for session-based neural neural network with two-layer attention mechanism. *Journal of Nanjing University of Posts and Telecommunications(Natural Science Edition)* 41, 99–106.
- Wang, H., Zhang, F., Xie, X., Guo, M., 2018. Dkn: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 world wide web conference, pp. 1835–1844.

- Wang, L., Chang, X., Li, S., Chu, Y., Li, H., Zhang, W., He, X., Song, L., Zhou, J., Yang, H., 2021. Tcl: Transformer-based dynamic graph modelling via contrastive learning. *arXiv preprint arXiv:2105.07944* .
- Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X., 2019a. Neural news recommendation with multi-head self-attention, in: *EMNLP*, pp. 6389–6394.
- Wu, C., Wu, F., Qi, T., Huang, Y., 2020a. User modeling with click preference and reading satisfaction for news recommendation., in: *IJCAI*, pp. 3023–3029.
- Wu, C., Wu, F., Qi, T., Huang, Y., Xie, X., 2019b. Neural news recommendation with attentive multi-view learning, in: *IJCAI*, pp. 3863–3869.
- Wu, F., Qiao, Y., Chen, J.H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al., 2020b. Mind: A large-scale dataset for news recommendation, in: *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 3597–3606.
- Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K., 2020. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962* .
- Zhang, M., Chen, Y., 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems* 31.
- Zhang, W., Liu, Y., Chen, L., Pan, L., 2023a. Hi-fi ark: Hierarchical fusion architecture for news recommendation. *Information Processing & Management* 60, 103206.
- Zhang, Z., Sun, S., Ma, G., Zhong, C., 2023b. Line graph contrastive learning for link prediction. *Pattern Recognition* 140, 109537.
- Zhou, L., Zhang, W., Xie, Y., Liu, Y., Pan, L., Guo, H., 2022. Gnewsrec: Graph-enhanced multi-level memory network for news recommendation. *World Wide Web* 25, 95–119.
- Zhu, Q., Zhou, X., Song, Z., Tan, J., Guo, L., 2019. Dan: Deep attention neural network for news recommendation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5973–5980.