Drift-Aware Multimodal User Representation Learning via Multi-Scale Temporal Modeling and Sparse Mixture-of-Experts

Ziqing Qian[1,2] **Haohang Chen**[1] **Shengqi Dang**[1,2,3] **Yuhan Xiong**[1]
Canyu Shen[1] **Jiaying Lei**[1,2,3] **Nan Cao**[1,2,3*]

qianziqing0719@gmail.com, nan.cao@tongji.edu.cn
[1]Intelligent Design & InnoVation Lab, Tongji University, Shanghai, China
[2]Shanghai Research Institute for Intelligent Autonomous System, Tongji University, Shanghai, China
[3]Shanghai Innovation Institute, Shanghai, China

Abstract

Understanding user preferences from noisy and temporally evolving social media behaviors is fundamentally challenging due to interest drift, where user preferences shift across time and exhibit both multi-scale temporal patterns and diverse co-existing interests. To address this, we propose **DUMoE**, a unified framework for drift-aware multimodal user representation learning. Our model consists of (i) a temporal dynamics-aware backbone that captures and integrates static profiles, short-term behavioral signals, and long-term dependencies into a coherent representation, and (ii) a sparse mixture-of-experts (MoE) interest adapter that disentangles multiple latent interests via expert specialization and adaptive routing. Each expert models a distinct interest subspace, while a gating network dynamically selects and aggregates a sparse subset of relevant experts for each user. To enable stable and effective optimization, we further introduce a three-stage training strategy that decouples backbone learning, expert specialization, and gating optimization. Extensive experiments on real-world social media datasets show that **DUMoE** consistently outperforms state-of-the-art methods on both user interest prediction and interaction prediction tasks. Code is available at `https://github.com/idvxlab/DUMoE`.

1 Introduction

Modeling user preferences is a fundamental problem in machine learning, underpinning applications such as recommendation [1], personalization [2], and information diffusion [3, 4]. A core challenge arises from interest drift, where user preferences evolve over time, exhibiting both long-term stability and short-term dynamics, while simultaneously spanning multiple co-existing interests. Effectively capturing such multi-scale temporal dynamics and multi-interest structure is essential for learning adaptive and robust user representations.

Social media platforms (e.g., X) provide a continuous stream of multimodal user-generated data, including textual content, visual signals, and interaction behaviors, offering a valuable testbed for modeling fine-grained user preference evolution. However, effectively leveraging such data remains challenging due to (1) heterogeneous multimodal inputs, (2) the need to capture multi-scale temporal dynamics, and (3) data sparsity with irregular user activity. Existing user profiling methods typically rely on static representations [5, 6] or coarse temporal modeling [7, 8, 9], limiting their ability to capture interest drift. Moreover, most approaches focus on single modalities, overlooking the

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

complementary signals across modalities. Critically, prior work lacks a unified framework that jointly models multi-modal information, multi-scale temporal dynamics, and both explicit (interpretable) and implicit (latent) user preferences.

In this work, we propose DUMoE, a **D**rift-aware **U**ser **M**odeling framework with **M**ixture-**O**f-**E**xperts. This technique learns hierarchical user representations by jointly modeling multimodal signals, multi-scale temporal dynamics, and user interest structure within a unified architecture. We first encode heterogeneous user-generated content into a shared semantic space, enabling consistent cross-modal representations. On top of this, we decompose user preferences into two complementary components: (1) explicit preference distributions over predefined interest domains, and (2) latent representations that capture both short-term behavioral dynamics and long-term dependencies. Temporal evolution is modeled via a multi-scale design, combining sequential modeling of recent activities with global aggregation over historical interactions. To further capture diverse user intents, we introduce a multi-interest module based on sparse mixture-of-experts, which disentangles user interests into specialized subspaces through expert specialization and adaptive routing. Extensive experiments on real-world datasets demonstrate that our approach consistently outperforms strong baselines in modeling user preference dynamics and adapting to evolving interests. Our contributions are summarized as follows:

- **Unified drift-aware user representation learning framework.** We propose a unified framework that jointly models **multimodal signals**, **multi-scale temporal dynamics**, and both **explicit and latent user preferences**. The proposed backbone integrates static profiles, short-term behaviors, and long-term dependencies into a hierarchical representation, enabling effective modeling of evolving user interests.
- **Sparse mixture-of-experts for multi-interest disentanglement.** We introduce a **sparse mixture-of-experts (MoE) interest adapter** that decomposes user representations into specialized interest subspaces via expert specialization and adaptive routing. The design promotes **disentangled, interpretable, and personalized** multi-interest modeling.
- **Large-scale temporally annotated multimodal dataset.** We construct a large-scale dataset from X with **chronologically evolving interest annotations**, covering 15 domains with 14K+ users, 7.7M tweets, and 2.9M images, providing a realistic benchmark for modeling user interest drift.

2 Related Work

Multimodal Representation Learning. Modeling user behavior over social media requires representations that effectively integrate heterogeneous signals, particularly textual and visual content [10, 11]. Cross-modal alignment architectures such as VisualBERT [12] and ViLBERT [13] enable fine-grained modality interactions via region-level grounding. Contrastive frameworks including CLIP [14] and ALBEF [15] scale alignment to web-level data, and generative models such as Flamingo [16], BLIP-2 [17], and LLaVA [18] extend multimodal learning to complex reasoning. However, these models are designed for curated, domain-specific corpora rather than the noisy, temporally irregular, and user-attributed streams characteristic of social media—they produce instance-level representations with no mechanism to aggregate signals across a user's evolving behavioral history.

Sequential User Behavior Modeling. Sequential modeling aims to capture temporal dynamics in user behavior, but existing approaches largely operate at the item level with limited semantic expressiveness. Early methods such as GRU4Rec [7] and Caser [8] model short-term dependencies using recurrent and convolutional architectures. Transformer-based models, including SASRec [1] and BERT4Rec [5], improve long-range dependency modeling via self-attention mechanisms. PTUM [9] further pre-trains user representations from unlabeled behavior sequences. Structural and contrastive enhancements are introduced in SR-GNN [19] and DuoRec [20], while UniSRec [21], VQ-Rec [22], and HM4SR [23] incorporate textual or multimodal item features. Yet these methods model behavior at the item level and largely neglect user-level multi-scale temporal dynamics, and open-domain interest drift.

User Interest Disentanglement. Single-vector representations fail to capture preference diversity [2], motivating multi-interest modeling. MIND [24] and ComiRec [25] extract multiple interest vectors via capsule routing and self-attention, while S3-Rec [26], DCCF [27], and SINE [28] improve disentanglement through mutual information, graph contrastive learning, and sparse prototypes,

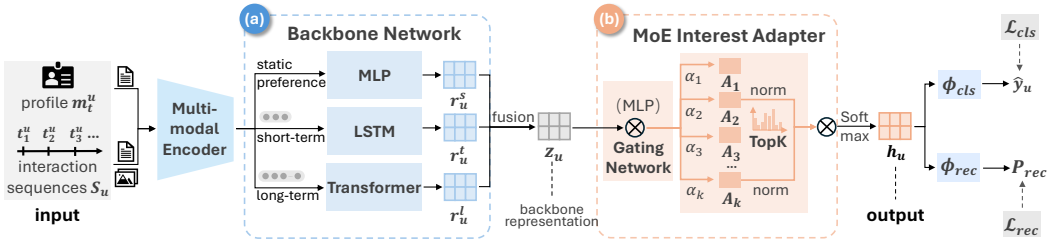


Figure 1: Algorithm Overview.

respectively. DisMIR [29] addresses interest collapse via spectral clustering, and HORAE [30] incorporates relative temporal intervals to track interest shifts over time. Despite this, interest subspaces are learned at a fixed snapshot: how their composition and relative salience drift as user behavior evolves over time remains unmodeled.

Summary. In summary, prior work has made substantial progress along three complementary directions: multimodal representation learning, sequential modeling, and interest disentanglement. However, these lines of research are largely developed in isolation. Existing multimodal methods overlook temporal dynamics, sequential models underutilize multimodal signals, and disentanglement approaches rarely consider both temporal evolution and heterogeneous content. To bridge these gaps, we propose a unified framework that jointly models multimodal signals, temporal dynamics, and multi-faceted user interests, enabling dynamic and expressive user representations for evolving social media environments.

3 Problem Formulation

Given a set of users $\mathcal{U}$, each user $u \in \mathcal{U}$ is associated with a profile $\mathbf{p}_u$ and a chronologically ordered interaction sequence $\mathcal{S}_u$. Our goal is to learn a mapping function

$$f : (\mathcal{S}_u, \mathbf{p}_u) \mapsto \mathbf{h}_u \in \mathbb{R}^d, \tag{1}$$

where $\mathbf{h}_u$ denotes the learned user representation that captures three key aspects of a user's behavior: (1) **interaction-level preference signals**, (2) **multi-scale temporal dependencies** that reflect evolving behavioral dynamics, and (3) **multi-interest structure** across diverse user activities. Here, the interaction sequence $\mathcal{S}_u$ is formally defined as

$$\mathcal{S}_u = \{(m_1^u, t_1^u), (m_2^u, t_2^u), \ldots, (m_n^u, t_n^u)\}, \tag{2}$$

where t_i^u denotes the timestamp of the i-th interaction, with $t_1^u < t_2^u < \cdots < t_n^u$. Each $m_i^u \in \mathcal{M}$ represents the multimodal content associated with the interaction (e.g., posted or retweeted text/image content). The user profile $\mathbf{p}_u$ consists of static attributes, including numerical features (e.g., follower counts) and textual descriptions (e.g., self-declared biography).

4 Methodology

In this section, we first present the proposed representation learning framework and then introduce the loss function and the stage-wise optimization strategy. The framework (Fig. 1) consists of two key components: a **temporal dynamics-aware backbone network** and a **sparse mixture-of-experts (MoE) interest adapter**. The backbone network (Fig. 1(a)) learns hierarchical user representations by capturing both short-term behavioral dynamics and long-term interaction dependencies. Built upon this representation, the MoE interest adapter (Fig. 1(b)) performs **multi-interest disentanglement** through expert specialization and adaptive routing, enabling dynamic preference modeling.

4.1 Temporal Dynamics-Aware Backbone Network

As shown in Fig. 1(a), the backbone network is designed to capture user preferences by jointly leveraging static profile information and the user's temporally evolving interaction behaviors. The architecture consists of two main components:

Post and Profile Encoding. We first encode both user posts and profile information into a unified embedding space based on a multimodal encoder. Here, we adopt a pre-trained UNITE encoder [31] due to its strong capability in learning aligned text-image representations. For each user post m_t^u consisting of textual and visual content, we apply the encoder $\mathcal{E}$ to jointly encode both modalities into a post representation $\mathbf{e}_t^u$:

$$\mathbf{e}_t^u = \mathcal{E}(m_t^u) \tag{3}$$

In addition, we also encode the heterogeneous attributes of a user profile $\mathbf{p}_u$ separately: numerical features (e.g., follower count, age) are normalized via min–max scaling and projected into embeddings, while textual attributes (e.g., self-declared biography) are encoded using $\mathcal{E}$. The final profile representation $\mathbf{a}_u$ is constructed by concatenating all attribute embeddings.

Temporal Dynamic Preference Modeling. Next, to capture the diverse behavioral patterns exhibited by users in real-world social media environments, we model user preferences from three complementary temporal perspectives: (1) static preference signals, (2) short-term dynamic patterns, and (3) long-term interests. Accordingly, we extract three corresponding feature representations, denoted by $\mathbf{r}_u^s$, $\mathbf{r}_u^t$, and $\mathbf{r}_u^l$, and integrate them into a unified user representation. These features are defined as

$$\mathbf{r}_u^s = \phi_s(\mathbf{a}_u), \tag{4}$$

$$\mathbf{r}_u^t = \phi_t(\mathbf{S}_u^t), \tag{5}$$

$$\mathbf{r}_u^l = \phi_l(\mathbf{S}_u^l), \tag{6}$$

where $\phi_s(\cdot)$ is a two-layer MLP that encodes the static profile feature $\mathbf{a}_u$; $\phi_t(\cdot)$ is an LSTM that processes the short-term interaction sequence $\mathbf{S}_u^t$ to capture recent behavioral dynamics; and $\phi_l(\cdot)$ is a Transformer encoder that models long-range dependency patterns over the long-term interaction sequence $\mathbf{S}_u^l$. The short-term and long-term sequences are defined as

$$\mathbf{S}_u^t = \{\mathbf{e}_i^u \mid i = T_u - L_t + 1, \ldots, T_u\}, \tag{7}$$

$$\mathbf{S}_u^l = \{\mathbf{e}_i^u \mid i = T_u - L_l + 1, \ldots, T_u\}, \tag{8}$$

where L_t and L_l denote the lengths of the short-term and long-term windows, respectively, with $L_t << L_l$, and T_u is the current time step for user u.

Finally, we fuse the three features through direct sum followed by nonlinear projection:

$$\mathbf{z}_u = \phi_f\left(\mathbf{r}_u^s \oplus \mathbf{r}_u^t \oplus \mathbf{r}_u^l\right), \tag{9}$$

where $\oplus$ denotes the direct sum (concatenation) operation, and $\phi_f(\cdot)$ is an MLP that integrates the three components into a unified backbone representation $\mathbf{z}_u$.

4.2 Mixture-of-Experts Interest Adapter

To capture multiple latent interests underlying diverse user behaviors, we introduce a **sparse Mixture-of-Experts (MoE) interest adapter**. The module consists of a set of lightweight expert networks and an adaptive gating mechanism. Each expert is implemented as an adapter module that specializes in modeling a distinct latent interest subspace, enabling **multi-interest disentanglement**. The gating network learns a user-specific routing distribution over experts, dynamically selecting a sparse subset of relevant experts for each user. The final user representation is obtained via a gated aggregation of expert outputs:

$$\mathbf{h}_u = \sum_{k=1}^{K} \alpha_{u,k}\, \mathcal{A}_k(\mathbf{z}_u), \tag{10}$$

where K denotes the number of experts, $\mathcal{A}_k(\cdot)$ is the k-th interest expert, and $\alpha_{u,k}$ is the gating weight for user u. The gating weights satisfy $\sum_{k=1}^{K} \alpha_{u,k} = 1$. In practice, we employ sparse routing, where only a subset of experts receives non-zero weights. Next, we detail the expert design and gating mechanism below.

Interest Expert. Each expert is implemented as a lightweight adapter that transforms the shared backbone representation $\mathbf{z}_u$. Specifically, the k-th expert is defined as

$$\mathcal{A}_k(\mathbf{z}_u) = \mathbf{z}_u + \phi_k(\mathbf{z}_u), \tag{11}$$

where $\phi_k(\cdot)$ is a multi-layer perceptron (MLP) that models interest-specific transformations. The residual formulation allows each expert to capture an **interest-specific deviation** from the shared representation, ensuring that all experts remain anchored in a common semantic space while enabling specialization across distinct interest dimensions.

Gating Weight. To mitigate interference across interest subspaces and encourage expert specialization, we adopt a **sparse top-K' routing** mechanism. The gating weights are defined as

$$\alpha_{u,k} = \begin{cases} \dfrac{\tilde{\alpha}_{u,k}}{\sum_{k' \in \mathcal{K}_u} \tilde{\alpha}_{u,k'}} & k \in \mathcal{K}_u \\ 0 & k \notin \mathcal{K}_u \end{cases} \tag{12}$$

where $\mathcal{K}_u$ contains the indices of the top-K' experts selected from the routing scores $\tilde{\boldsymbol{\alpha}}_u$. This design enforces sparse activation, allowing each user to be represented by a small subset of dominant interests, thereby improving both efficiency and interpretability. The routing scores are computed as

$$\tilde{\boldsymbol{\alpha}}_u = \mathrm{softmax}(\mathbf{W}_g \mathbf{z}_u + \mathbf{b}_g), \tag{13}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable parameters.

4.3 Learning Objective and Training Procedure

To enable the learned representation to understand user preferences and behaviors, we employ a supervised learning approach to learn the representation. We use two learning objectives: (1) interest classification and (2) interaction prediction. The overall training objective is a sum of the two loss terms for each task, corresponding to the two tasks respectively:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \, \mathcal{L}_{\text{rec}} \tag{14}$$

where $\mathcal{L}_{\text{cls}}$ is the interest classification loss, $\mathcal{L}_{\text{rec}}$ is the interaction prediction loss, and λ is a weight balances the two parts.

Interest Classification. The learning objective for interest classification minimizes a class-weighted focal loss between the ground-truth interest label y_u and the prediction $\phi_{\text{cls}}(\mathbf{h}_u)$:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{B_{\text{cls}}} \sum_{i=1}^{B_{\text{cls}}} w_{y_i} (1 - p_{y_i})^{\gamma} \log p_{y_i} \tag{15}$$

Here, $p_{y_i} = \phi_{\text{cls}}(\mathbf{h}_u)_{y_i}$ denotes the predicted probability of the ground-truth class, where i indexes users in a mini-batch. B_{cls} is the batch size of users for the classification task. w_{y_i} is the inverse-frequency class weight for handling class imbalance, and $\gamma = 2$ is the focusing parameter. $\phi_{\text{cls}}(\cdot)$ is implemented as $\mathrm{softmax}(\mathbf{W}_{\text{out}}\mathbf{h}_u + \mathbf{b}_{\text{out}})$ with learnable parameters $\mathbf{W}_{\text{out}}, \mathbf{b}_{\text{out}}$, and y_u is assigned from the C predefined interest categories based on the user's interaction history.

Interaction Prediction. For interaction prediction, we aim to learn user representations such that the representation of a user is similar to the representation of the next post they actually interact with (e.g., a post or repost), and dissimilar to randomly sampled negative posts. To achieve this, we adopt a negative sampling approach with binary cross-entropy (BCE) loss, which encourages high inner products for positive pairs and low inner products for negative pairs, defined as:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{B_{\text{rec}}} \sum_{i=1}^{B_{\text{rec}}} \left[\log \sigma(s_i^+) + \frac{1}{K} \sum_{k=1}^{K} \log \sigma(-s_{ik}^-) \right] \tag{16}$$

where $\sigma(\cdot)$ is the sigmoid function, B_{rec} is the mini-batch size consisting of users who have interaction samples, and K denotes the number of negative samples per user. For each user i in this batch, $\mathbf{h}_u$ denotes the representation of user i, and $s_i^+ = \phi_{\text{rec}}(\mathbf{h}_u)^\top \mathbf{v}_i^+$ is the inner product between the user's projected representation $\phi_{\text{rec}}(\mathbf{h}_u)$ and the positive post embedding $\mathbf{v}_i^+$ (the ground-truth next post). Here, ϕ_{rec} is implemented as an MLP. Similarly, $s_{ik}^- = \phi_{\text{rec}}(\mathbf{h}_u)^\top \mathbf{v}_{ik}^-$ is the inner product with the k-th negative post embedding, where $\{\mathbf{v}_{ik}^-\}_{k=1}^K$ are sampled negative posts for user i.

Training Strategy. We adopt a three-stage optimization procedure to stabilize training and promote expert specialization: (1) **backbone pre-training**, (2) **expert specialization**, and (3) **gating optimization**. The prediction heads ϕ_{cls} and ϕ_{rec} are jointly optimized across all stages. **Stage 1: Backbone Pre-training.** We train the backbone network together with the prediction heads, while removing the MoE adapter. The user representation is directly taken as $\mathbf{h}_u = \mathbf{z}_u$. This stage learns a strong and discriminative shared representation. **Stage 2: Expert Specialization.** We freeze the backbone and train each expert independently using users whose dominant interest label matches the expert index. This encourages each expert to capture a distinct interest subspace. After this stage, the expert pool $\{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$ consists of specialized and fixed experts. **Stage 3: Gating Optimization.** We freeze both the backbone and the experts, and train only the gating network parameters ($\mathbf{W}_g$, $\mathbf{b}_g$), allowing it to learn a stable user-specific routing distribution over the specialized experts.

Inference. At inference time, the full model is applied end-to-end. The backbone and adapter representations are further combined via a KL-based gating signal (see Appendix D.3 for details).

This stage-wise design decouples representation learning, expert specialization, and routing optimization, resulting in improved stability, clearer expert roles, and more reliable personalized routing.

5 Experiments

To evaluate the effectiveness of the proposed technique, we first introduce a large-scale multimodal dataset collected from X (detailed in Section 5.1). Based on this dataset, we detail the experimental settings and conduct comprehensive experiments on two downstream tasks: (1) user interest classification and (2) social interaction prediction, to assess the quality and utility of the learned representations. Finally, we present ablation studies to analyze the contribution of each key component.

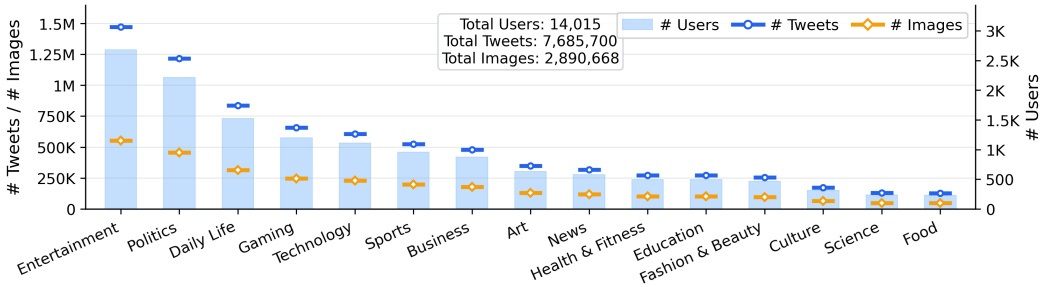


Figure 2: Data distribution across different interest domains.

5.1 Dataset Construction

We construct a large-scale multimodal social interaction dataset from X (formerly twitter) to support drift-aware user modeling. As shown in Fig. 2, the dataset includes user profiles, interaction behaviors (posts and reposts), multimodal content, and timestamps, covering 15 high-level interest domains with diverse topical distributions. For each domain, active users are first identified followed by collecting their historical interactions, including textual posts, associated images, and profile metadata. Overall, the dataset comprises 14,015 users, 7,685,700 posts, and 2,890,668 images.

Based on the collected dataset, we construct training samples via a three-step procedure: (1) Temporal segmentation. For each user, interactions are chronologically ordered and partitioned into non-overlapping collections, each containing 30 posts. (2) Pseudo-label annotation. For each target collection, we assign an interest label from 15 predefined domains using GPT-5.1. The annotation is conditioned on the content of the target collection and up to 2 preceding collections as historical context. This design incorporates temporal dependencies to better capture interest drift and improve label consistency. (3) Quality verification. To assess annotation quality, we manually inspect a randomly sampled subset of labeled collections. The results indicate that the generated labels are largely consistent with human judgments.

5.2 Experimental settings

Baselines. We compare our method with representative baselines from three categories, covering key paradigms in user modeling: sequential recommendation, multi-interest modeling, and content-aware representation learning.

- *Sequential recommendation.* We include three representative baselines for modeling user behavior from interaction sequences. SASRec [1] uses causal self-attention to model sequential dependencies. BERT4Rec [5] employs a masked language modeling objective for bidirectional sequence modeling. PTUM [9] incorporates self-supervised learning signals, including masked behavior and next-item prediction.
- *Multi-interest modeling.* We include two representative baselines that explicitly model multiple user interests. MIND [24] employs capsule-based dynamic routing to extract latent interest representations from behavior sequences. HORAE [30] combines temporal-aware pre-training with explicit interest decomposition to capture temporal dynamics and multi-interest structures. HM4SR [23] integrates hierarchical temporal modeling with a mixture-of-experts architecture to capture evolving user interests from multimodal sequential behaviors.
- *Content-aware representation learning.* We include two representative baselines that incorporate auxiliary content information into user modeling. PeterRec [6] adopts a pre-training and fine-tuning paradigm to learn transferable user representations from behavioral data. UniSRec [21] integrates textual content with Mixture-of-Experts adapters to learn unified sequential representations across domains.

Metrics. For user interest classification, we evaluate performance using macro Recall, Hit@1, NDCG@3, and KL divergence. Hit@1 reflects top-1 prediction accuracy, while NDCG@3 captures the quality of ranked predictions with higher emphasis on top positions. KL divergence measures the alignment between the predicted distribution and the label-smoothed ground-truth distribution. For interaction prediction, we report Accuracy (Acc), Recall, F1 score, and binary cross-entropy (BCE), providing both classification-based and probabilistic evaluation of model performance.

Implementation details. The backbone network is trained using the proposed three-stage strategy with a batch size of 128. After the pre-training stage, its parameters are frozen for subsequent optimization. The dataset is split chronologically at the user-collection level, with each user's 12 collections allocated as 8 for training, 2 for validation, and 2 for testing. All experiments use three random seeds. Mean results are reported in main text, with full statistics provided in Appendix D.6.

5.3 Experiment I: User Interest Classification

This experiment aims to evaluate the quality of learned user representations by measuring how accurately they predict users' interest categories. We implement all baselines using their publicly available code or official reproductions, following the same data splits and evaluation protocol for fair comparison.

Table 1: Interest classification performance. ↑ / ↓ indicates higher/lower is better.

Method	Hit@1 ↑	NDCG@3 ↑	Recall ↑	KL Divergence ↓
SASRec	0.767	0.872	0.695	0.609
BERT4Rec	0.769	0.874	0.699	0.586
PTUM	0.808	0.899	0.751	0.487
MIND	0.816	0.903	0.759	0.478
HORAE	0.801	0.896	0.740	0.512
HM4SR	0.759	0.866	0.688	0.615
PeterRec	0.785	0.886	0.717	0.537
UniSRec	0.763	0.869	0.681	0.609
DUMoE	**0.874**	**0.941**	**0.839**	**0.318**

As shown in Table 1, DUMoE consistently outperforms all baselines across all evaluation metrics. It achieves a Hit@1 of **0.874**, a Recall of **0.839**, and an NDCG@3 of **0.941**, demonstrating its

effectiveness in accurately identifying the most relevant target while maintaining strong ranking quality. The substantial improvement in Recall further indicates that our method captures a broader set of relevant candidates, which is critical in recommendation scenarios with diverse user preferences. Compared to the strongest baselines MIND and PTUM, our method improves Hit@1 by approximately **7.1%** and **8.1%**, respectively, and yields a notable gain in Recall (**+10.5%** over MIND). Relative to HORAE, which explicitly models multi-interest representations, our approach achieves an **9.1%** improvement in Hit@1, highlighting its advantage in learning more precise and comprehensive user representations. In contrast, purely sequential models such as SASRec and BERT4Rec lag significantly behind (e.g., **12.2%** lower Hit@1 for SASRec), suggesting that sequence-only modeling without explicit disentanglement struggles to capture complex user behaviors.

Moreover, DUMoE achieves the lowest KL divergence of **0.318**, substantially below all baselines and corresponding to a **33.5%** reduction over the second-best method MIND (0.478), indicating significantly better calibration of the predicted interest distribution. Most competing methods exhibit KL values in the range of 0.478–0.615, reflecting notable distributional mismatch even when ranking performance is relatively strong. The superior NDCG@3 further confirms that our method not only identifies the correct target but also ranks relevant items more effectively at top positions. Collectively, these results demonstrate that our framework achieves both strong ranking performance and improved distributional alignment, validating the effectiveness of the proposed representation learning strategy.

5.4 Experiment II: User Interaction Prediction

This experiment evaluates the model's ability to predict future user interactions. Given a chronological sequence of behavioral windows for a user, we use the representation derived from our method to predict items in the subsequent window. We adopt the same lightweight prediction head and training protocol for all baselines to ensure fair comparison. Further details are provided in Appendix D.5.

Table 2 summarizes the results. DUMoE achieves the best performance in Recall, obtaining a score of **0.922**, clearly outperforming all baselines. In particular, it surpasses the strongest baseline HORAE (Recall: 0.900) by approximately **2.4%**, demonstrating its stronger capability in retrieving future positive interactions. In terms of F1, DUMoE achieves **0.888**, ranking second among all methods and remaining highly competitive with the best-performing baseline PTUM (0.889). For Accuracy, DUMoE achieves **0.883**, which is slightly lower than PTUM (0.888), MIND (0.886), and PeterRec (0.884), but remains competitive with these methods. For BCE (lower is better), our method achieves **0.276**, matching PeterRec (0.276), although it is slightly higher than the best-performing baseline PTUM (0.270). These results demonstrate that our temporal dynamics-aware backbone combined with the MoE interest adapter produces effective user representations for predicting future interactions, with particularly strong advantages in retrieving future positive interactions while maintaining competitive classification performance.

Table 2: Interaction prediction performance. ↑ / ↓ indicates higher/lower is better.

Method	**Acc** ↑	**Recall** ↑	**F1** ↑	**BCE** ↓
SASRec	0.875	0.889	0.876	0.295
BERT4Rec	0.878	0.892	0.880	0.292
PTUM	0.888	0.898	0.889	0.270
MIND	0.886	0.897	0.887	0.273
HORAE	0.884	0.900	0.886	0.276
HM4SR	0.867	0.878	0.869	0.307
PeterRec	0.884	0.900	0.886	0.276
UniSRec	0.875	0.894	0.878	0.295
DUMoE	**0.883**	**0.922**	**0.888**	**0.276**

5.5 Ablation Study

We perform ablation experiments on downstream tasks using the same evaluation protocol (Table 3). We study the effect of key components, including (1) backbone network, (2) mixture-of-experts interest adapter, and (3) the training strategy and objective.

Backbone Network. The backbone ablations (Static Only, Short-term Only, and Long-term Only) reveal that each component contributes meaningfully to the final performance. Among single-branch variants, Long-term Only achieves the highest Hit@1 (0.863), but still trails the full model by 1.1%, while Short-term Only (0.809) and Static Only (0.781) lag further behind. To further evaluate the contribution of temporal modeling, we also compare against a content-only baseline (UNITE Mean-Pool), which achieves lower Hit@1 (0.828) and interaction Recall (0.877) than the full model (0.874 and 0.922), with higher KL (0.436) and BCE (0.333). Overall, these results demonstrate the importance of jointly integrating complementary static and multi-scale temporal information with pretrained content representations to learn more comprehensive and effective user representations.

Table 3: Ablation study on downstream tasks. ↑/↓ indicates higher/lower is better.

	Interest Classification				**Interaction Prediction**			
Variant	Hit@1 ↑	NDCG@3 ↑	Recall ↑	KL ↓	Acc ↑	Recall ↑	F1 ↑	BCE ↓
Static Only	0.781	0.880	0.741	0.560	0.809	0.839	0.814	0.409
Short-term Only	0.809	0.898	0.743	0.464	0.846	0.852	0.847	0.350
Long-term Only	0.863	0.936	0.838	0.340	0.862	0.892	0.866	0.319
UNITE Mean-Pool	0.828	0.915	0.770	0.436	0.856	0.877	0.859	0.333
w/o Adapter	0.872	0.940	0.837	0.321	0.862	0.885	0.865	0.319
w/o Gating	0.872	0.940	0.839	0.316	0.864	0.883	0.867	0.316
w/o Stage-wise Training	0.863	0.935	0.848	0.343	0.867	0.887	0.870	0.309
$\mathcal{L}_{\text{cls}}$ Only	0.868	0.937	0.844	0.316	0.795	0.847	0.805	0.441
$\mathcal{L}_{\text{rec}}$ Only	0.805	0.901	0.736	0.469	0.867	0.882	0.869	0.310
Full model	**0.874**	**0.941**	**0.839**	**0.318**	**0.883**	**0.922**	**0.888**	**0.276**

Mixture-of-Experts Interest Adapter. Removing the adapter pool (w/o Adapter) and replacing sparse gating with uniform averaging (w/o Gating) both degrade performance compared with the full model. Specifically, w/o Adapter achieves lower Hit@1 (0.872) and higher BCE (0.319), while w/o Gating yields Hit@1 of 0.872 and BCE of 0.316. Both variants also underperform the full model on interaction prediction (Acc: 0.862 and 0.864 vs. 0.883), validating that both the adapter pool and adaptive routing are important for capturing diverse user interests.

Training Strategy and Objective. Disabling the stage-wise training strategy (w/o Stage-wise Training) leads to lower Hit@1 (0.863) and higher KL (0.343) than the full model, although it achieves a higher classification Recall (0.848 vs. 0.839). It also degrades interaction prediction (Acc: 0.867, Recall: 0.887, F1: 0.870, BCE: 0.309), confirming the effectiveness of our three-stage approach in coordinating the learning dynamics and improving overall performance. Regarding the training objective, $\mathcal{L}_{\text{cls}}$ alone drives discriminative interest identification (Hit@1: 0.868, KL: 0.316) but substantially degrades interaction prediction (Acc: 0.795, BCE: 0.441); conversely, $\mathcal{L}_{\text{rec}}$ alone yields competitive interaction prediction (Acc: 0.867, BCE: 0.310) but underperforms on interest classification (Hit@1: 0.805, KL: 0.469), as reconstruction supervision alone lacks the discriminative signal for accurate interest identification.

5.6 Expert Behavior Analysis

To further investigate how the MoE interest adapter captures diverse user interests, we analyze expert specialization and adaptive routing patterns across different interest domains. We provide detailed expert-category correspondence matrices and domain-wise routing statistics in the Appendix E.

Expert Specialization. We first examine expert utilization and its correspondence with users' dominant interest categories. The results reveal a non-uniform but well-distributed utilization pattern, with expert usage generally reflecting the distribution of interest categories. Entertainment and Politics exhibit the highest mean gating values (MG), at 0.197 and 0.194, respectively. The Top Expert column further shows that most categories are primarily associated with their corresponding experts, with Sports and Fashion & Beauty being primarily associated with the Food and Science experts, respectively. Regarding expert specialization, the Politics expert achieves the highest recall and precision (**88.80%** and **93.50%**), followed by the Entertainment expert (**82.20%** and **85.10%**).

By comparison, the Science expert exhibits a lower recall of 36.50%, highlighting differences in specialization across domains. These results indicate that the MoE adapter learns meaningful domain-related representations, while the non-uniform routing patterns also reflect varying degrees of overlap among users' interests.

Adaptive Routing. We further analyze the routing entropy across different interest domains to examine how the gating network distributes users across experts. The average routing entropy is **0.584**, with substantial variations across domains. Politics exhibits the lowest entropy of **0.269**, indicating more concentrated expert routing, whereas Science and Fashion & Beauty exhibit higher entropy values of **1.066** and **1.049**, respectively, suggesting more distributed expert activation. These differences indicate that the gating network learns domain-dependent routing patterns rather than applying uniform expert assignments to all users. In particular, the contrast between concentrated routing in Politics and more distributed routing in Science and Fashion & Beauty reflects the adaptive nature of expert selection across different user interest domains.

5.7 Discussion

While our framework demonstrates strong performance on user interest classification and interaction prediction, several limitations warrant discussion. First, pseudo-label generation relies on GPT-5.1 with contextual signals derived from adjacent behavioral windows. Although this design improves temporal consistency, label quality may still be affected by annotation noise or domain-specific biases inherent to the language model. Future work could explore semi-supervised or self-supervised alternatives to reduce dependence on external supervision. Second, the multi-scale backbone adopts fixed window lengths for short-term (LSTM) and long-term (Transformer) modeling. In practice, optimal temporal granularity likely varies across users depending on their activity patterns and engagement frequency. Adaptive window selection mechanisms could provide finer-grained personalization in such heterogeneous settings. Third, the MoE interest adapter employs a fixed number of experts with static top-K' routing. While effective, the optimal number of activated experts may differ substantially across users with varying behavioral diversity. Dynamic expert allocation conditioned on user-level complexity is a promising direction for further improving adaptability. Finally, the current framework operates in an offline training regime. Although our three-stage training strategy promotes stable optimization, online continual learning with efficient incremental updates remains an open challenge. Incorporating lightweight adaptation mechanisms to handle real-time interest drift would be valuable for production-scale deployment. Addressing these limitations in future work can further enhance the robustness, efficiency, and generalizability of dynamic user profiling.

6 Conclusion

In this paper, we proposed a dynamic user profiling framework that jointly models explicit and implicit preferences across multiple temporal scales. Our approach features a multi-scale backbone fusing static, short-term, and long-term features, a Mixture-of-Experts interest adapter with sparse routing for multi-interest decomposition, and a three-stage training strategy that ensures stable specialization. We also contributed a large-scale multimodal dataset from X with temporally evolving interest labels. Extensive experiments on user interest classification and interaction prediction demonstrated that our method consistently achieves strong performance across both tasks. Ablation studies validated the contribution of each key component. Future work includes extending the framework to more fine-grained temporal dynamics and exploring online adaptation scenarios. Our code and dataset will be made publicly available under the privacy policy to facilitate reproducibility.

Acknowledgments

Nan Cao is the corresponding author. This work was supported by the National Key Research and Development Program of China (2023YFB3107100).

References

[1] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[2] Minyar Sassi Hidri. Learning-based models for building user profiles for personalized information access. *arXiv preprint arXiv:2405.15791*, 2024.

[3] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[4] Ziqing Qian, Jiaying Lei, Shengqi Dang, and Nan Cao. Designed to spread: A generative approach to enhance information diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 944–952, 2026.

[5] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[6] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1469–1478, 2020.

[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[8] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[9] Chuhan Wu, Fangzhao Wu, Tao Qi, Jianxun Lian, Yongfeng Huang, and Xing Xie. Ptum: Pre-training user model from unlabeled user behaviors via self-supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1939–1944, 2020.

[10] Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. User modeling and user profiling: A comprehensive survey. *arXiv preprint arXiv:2402.09660*, 2024.

[11] Christopher Ifeanyi Eke, Azah Anir Norman, Liyana Shuib, and Henry Friday Nweke. A survey of user profiling: State-of-the-art, challenges, and solutions. *IEEE Access*, 7:144907–144924, 2019.

[12] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[13] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[15] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[16] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[19] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353, 2019.

[20] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 813–823, 2022.

[21] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 585–593, 2022.

[22] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pages 1162–1171, 2023.

[23] Shengzhe Zhang, Liyi Chen, Dazhong Shen, Chao Wang, and Hui Xiong. Hierarchical time-aware mixture of experts for multi-modal sequential recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 3672–3682, 2025.

[24] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2615–2623, 2019.

[25] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2942–2951, 2020.

[26] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020.

[27] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1137–1146, 2023.

[28] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 598–606, 2021.

[29] Yingpeng Du, Ziyan Wang, Zhu Sun, Yining Ma, Hongzhi Liu, and Jie Zhang. Disentangled multi-interest representation learning for sequential recommendation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 677–688, 2024.

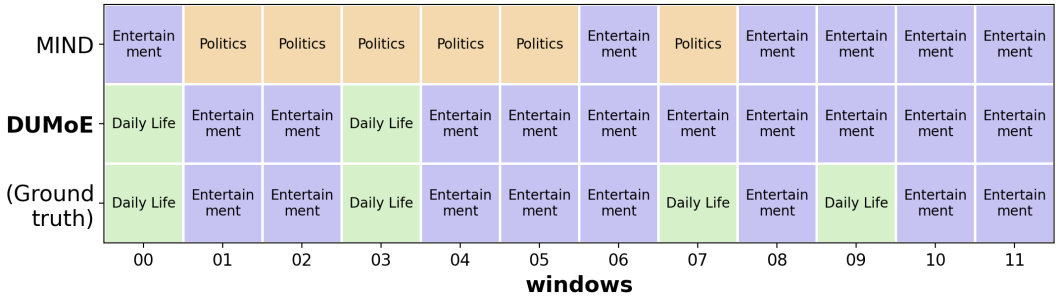
[30] Shirui Hu, Weichang Wu, Zuoli Tang, Zhaoxin Huan, Lin Wang, Xiaolu Zhang, Jun Zhou, Lixin Zou, and Chenliang Li. Horae: Temporal multi-interest pre-training for sequential recommendation. *ACM Transactions on Information Systems*, 43(4):1–29, 2025.

[31] Fanheng Kong, Jingyuan Zhang, Yahui Liu, Hongzhi Zhang, Shi Feng, Xiaocui Yang, Daling Wang, Yu Tian, Fuzheng Zhang, Guorui Zhou, et al. Modality curation: Building universal embeddings for advanced multimodal information retrieval. *arXiv preprint arXiv:2505.19650*, 2025.

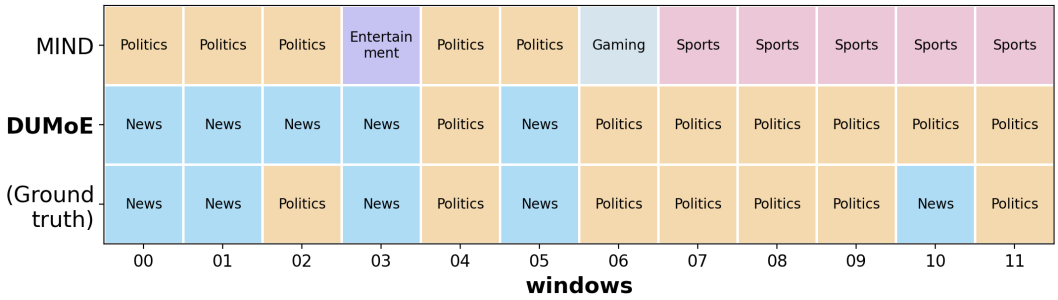
A Qualitative Analysis

A.1 Interest Drift Case Study

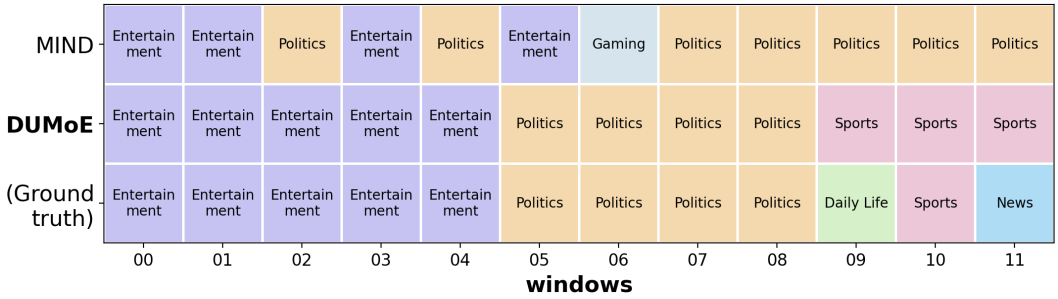
Figure 3 visualizes window-level interest predictions from MIND and DUMoE alongside ground-truth labels for three representative users. We highlight User B as an illustrative case. The ground truth shows an alternating *News–Politics* pattern across 12 windows, reflecting fine-grained interest fluctuation. MIND fails to track this, instead drifting into unrelated categories such as *Gaming* and *Sports* in later windows. DUMoE consistently captures the *News–Politics* alternation, with only minor deviations, demonstrating its ability to model co-existing interests and short-term drift. The other two users exhibit similar trends: DUMoE more faithfully follows ground-truth transitions while MIND tends to collapse to a dominant category or produce inconsistent predictions.



(a) User A: Entertainment-dominant interest with occasional drift to Daily Life.



(b) User B: Alternating News–Politics interest.



(c) User C: Entertainment-to-Politics shift followed by late-stage drift.

Figure 3: Window-level interest trajectories for three representative users.

B Broader Impact

This work advances the ability of machine learning systems to model temporally evolving user interests from noisy social media data, with potential benefits for recommendation, content filtering, and personalized information access. The sparse Mixture-of-Experts design further promotes interpretability by routing users to disentangled interest subspaces rather than opaque monolithic

representations. We also release a large-scale multimodal dataset to support future research in user modeling and social computing; to protect privacy, only anonymized user IDs and tweet IDs are released, with content filtering applied prior to distribution.

We acknowledge that fine-grained user interest modeling carries inherent dual-use risks. Temporal user representations could be misused for surveillance, behavioral profiling, or the targeted delivery of manipulative content timed to moments of heightened susceptibility. Closely tracking interest drift may also deepen filter bubbles by reinforcing existing preferences at the expense of viewpoint diversity; we recommend that systems built on our representations incorporate diversity constraints as a complementary objective. Additionally, our pseudo-label pipeline relies on an LLM, whose training biases may produce less accurate annotations for underrepresented or non-English-speaking communities, potentially leading to uneven model performance across demographic groups. To mitigate these risks, we distribute the dataset under terms of use that prohibit surveillance and targeted manipulation, and encourage practitioners to engage with ethics review processes when deploying user behavior modeling in high-stakes contexts.

C Dataset Details

C.1 Dataset Statistics

Table 4 reports per-domain statistics of our dataset. The data were collected from X (formerly Twitter), and the interest taxonomy is derived by consolidating platform-specific community classifications into **15 high-level interest domains**. Overall, the dataset comprises **14,015** users, **7,685,700** tweets, and **2,890,668** associated images. The pronounced skew across domains reflects the organic distribution of user interests on the platform and motivates the class-weighted focal loss adopted in our learning objective (Eq. 15 of the main paper).

Table 4: Per-domain dataset statistics.

Domain	**# Users**	**# Tweets**	**# Images**
Entertainment	2,681	1,470,236	552,971
Politics	2,217	1,215,783	457,268
Daily Life	1,524	835,748	314,333
Gaming	1,201	658,618	247,713
Technology	1,108	607,617	228,531
Sports	957	524,810	197,386
Business	872	478,197	179,855
News	580	318,067	119,628
Health & Fitness	499	273,647	102,921
Education	494	270,905	101,890
Fashion & Beauty	466	255,550	96,115
Art	634	347,680	130,766
Culture	314	172,195	64,764
Science	236	129,420	48,676
Food	232	127,227	47,851
Total	**14,015**	**7,685,700**	**2,890,668**

C.2 Pseudo-Label Annotation Protocol

To obtain large-scale temporally-evolving interest labels, we use GPT-5.1 to automatically annotate each user's tweet sequence. For each user, tweets are first sorted in descending chronological order and partitioned into non-overlapping groups of 30, yielding G_u ordered snapshots. We then construct labeling units using a sliding-window scheme: each window t (with $t = 0, 1, \ldots, G_u - 1$) consists of up to three consecutive groups $(t, t{+}1, t{+}2)$, forming a short-term context of at most 90 tweets. This design allows each group to appear in multiple overlapping windows, enriching temporal coverage. Due to API constraints, each window contains at most 50 images.

Each window is converted into a multimodal prompt with three sub-windows (A, B, C), interleaving tweet text and associated images. The prompt used in our implementation is as follows:

Multimodal Interest Classification Prompt

You are a precise multimodal content classification assistant. You analyze both text and images jointly to infer user interest. Follow category definitions strictly and avoid introducing external assumptions.

Please analyze the user's tweets across three temporal windows and determine the single most relevant interest category.

1. Window A (Group t)
2. Window B (Group $t+1$)
3. Window C (Group $t+2$)

Available categories (15 in total): Entertainment, Sports, Technology, Business, Education, Gaming, Politics, Art, Culture, Science, Food, Daily Life, Fashion & Beauty, Health & Fitness, News.

Each window contains interleaved tweet text and associated images.

Output Requirement: Return ONLY the single best-matching category name. Do not output explanations, punctuation, or additional text.

We evaluated multiple large language model families, including GPT, Claude, and Gemini, on a pilot set of 8 users in terms of accuracy, stability, and cost. GPT-based models consistently demonstrated the best overall trade-off, combining high classification accuracy with stable outputs and low refusal rates on multimodal inputs. Within the GPT family, we further compared GPT-5.1, GPT-5.2, and GPT-5.4. While GPT-5.4 achieves slightly higher accuracy (93.2%), it is significantly slower and more expensive; GPT-5.1 achieves comparable performance (92.1%) with better stability and fewer refusals, and is thus selected (temperature $= 0.3$, max tokens $= 20$). Finally, we randomly sampled approximately 5% of users ($\approx$700) for manual verification. Three independent annotators with domain expertise in social media content analysis labeled the same sampled windows using the 15-category taxonomy. The average pairwise inter-annotator agreement was **85.45%**, with a Fleiss' kappa of **0.838**, indicating strong consistency among human annotators. The agreement between GPT-5.1 annotations and human judgments reached **92.3%**, further supporting the reliability of our pseudo-label annotation pipeline.

D Implementation and Reproducibility

All experiments are conducted on a server equipped with 2$\times$ NVIDIA A800 80GB PCIe GPUs (each with 80 GB memory). The software environment consists of PyTorch 2.10.0+cu128 and CUDA 12.8. Mixed-precision training (AMP, `float16`) is enabled throughout to reduce memory usage and accelerate training.

D.1 Input Encoding and Preprocessing

Interaction Sequence Construction. Following the dataset construction protocol (Section 5.1), each user's interaction history is partitioned into non-overlapping groups of 30 posts in chronological order. The short-term sequence $\mathcal{S}_u^t$ covers the most recent $L_t = 3$ groups (90 posts), while the long-term sequence $\mathcal{S}_u^l$ spans the most recent $L_l = 10$ groups (300 posts), such that $\mathcal{S}_u^t \subset \mathcal{S}_u^l$. Within each window, post embeddings are first mean-pooled within each group to obtain one group-level embedding, yielding sequences of length L_t and L_l respectively. These group-level sequences are then fed into $\phi_t(\cdot)$ (LSTM) and $\phi_l(\cdot)$ (Transformer) as described in Section 4.1, with $\mathbf{r}_u^t$ taken as the final hidden state of the LSTM and $\mathbf{r}_u^l$ taken as the `[CLS]` token output of the Transformer. Users with fewer than L_l groups in their history use all available groups, with zero-padding applied as needed. The window lengths $L_t = 3$ and $L_l = 10$ are chosen to maintain a meaningful separation between temporal scales: the short-term window captures recent behavioral shifts, while the long-term window provides broader historical context sufficient to model stable interest patterns.

Post and Profile Encoding. We use UNITE-Base-Qwen2-VL-2B [31] (1536-d output) as the frozen multimodal encoder for all text and image inputs. Each post embedding is obtained by jointly

encoding the tweet text and associated image. All embeddings are pre-extracted offline prior to training. We construct a 1542-d representation $\mathbf{a}_u$ by concatenating textual and numerical features as detailed in Table 5, which is then fed into the static MLP $\phi_s(\cdot)$.

Table 5: User profile attribute encoding.

Field	Processing	Dim
`screen_name`, `name`, `description`, `location`	UNITE encode $\rightarrow$ mean	1536
`verified`	binarize	1
`media_count`	binarize + log-scale	2
`followers_count`	log-scale to $[0, 1]$	1
`friends_count`	log-scale to $[0, 1]$	1
`statuses_count`	log-scale to $[0, 1]$	1
Concatenation		1542

D.2 Three-Stage Training Details

We optimize the backbone and MoE adapter with AdamW (weight decay 10^{-2}, gradient clipping 1.0) using a three-stage schedule, with all stages using a batch size of 128. In Stage 1 (backbone pre-training, lr 1×10^{-4}, 15 epochs), the backbone is trained together with prediction heads ϕ_{cls} and ϕ_{rec} under the two-task objective balanced by $\lambda = 1.5$, while all adapter and gating parameters are frozen. In Stage 2 (category-guided expert specialization, lr 5×10^{-4}, 5 epochs), the backbone is frozen, and each of the $K = 15$ experts is trained using only samples whose dominant interest label matches its corresponding category. Specifically, expert k is optimized on samples with label $y = k$, while the recommendation head is jointly optimized alongside each expert. This category-guided training strategy encourages each expert to capture category-related user representations. In Stage 3 (gating optimization, lr 1×10^{-3}, 5 epochs), both the backbone and experts are frozen, and only the gating parameters $(W_g, \mathbf{b}_g)$ are updated with sparse top-$K' = 3$ routing.

The model architecture is configured as follows: the short-term encoder is a 1-layer LSTM with hidden size 256; the long-term encoder is a 2-layer Transformer with 8 attention heads and FFN dimension 1024; each expert adapter is a bottleneck MLP with 64-d hidden size and GELU activation; the fusion head ϕ_f projects to 256-d with dropout 0.4.

D.3 KL-Based Gating Mechanism

At inference, the backbone representation $\mathbf{z}_u$ and the MoE-adapted representation $\mathbf{h}_u$ are fused via a KL-based gate that uses interest drift intensity as an adaptive signal. To estimate drift, we compute a reference backbone representation $\mathbf{z}_u^-$ by removing the most recent L_t posts from both branches, retaining only the distal portion $[L_t{+}1,\ L_l]$ of the long-term sequence $\mathcal{S}_u^l$. Both $\mathbf{z}_u$ and $\mathbf{z}_u^-$ are produced by the same backbone defined in Eq. (9) with shared parameters. The KL divergence between their projected interest distributions measures how much the most recent behavioral context shifts the user's interest prediction — a large divergence indicates strong short-term drift, whereas a small divergence suggests that recent posts carry little new information. The gate then adaptively weights $\mathbf{z}_u$ and $\mathbf{h}_u$ based on this signal, as summarized in Algorithm 1.

D.4 Baseline Implementation Details

All baselines are pretrained on the same behavioral sequences following their respective original training objectives: next-item prediction for SASRec, PTUM, MIND, PeterRec, UniSRec, HORAE, and HM4SR, and masked item modeling for BERT4Rec. Following the content-aware setting of UniSRec [21], we initialize the item embedding table from precomputed UNITE vectors and keep it frozen throughout pretraining, replacing the original id-based embeddings with semantic representations; all other architectural and optimization details follow the original papers. For each window, the input sequence contains up to 300 temporally ordered posts; windows with fewer than 2 valid items are discarded. All baselines use a hidden dimension of 256 and export user representations of the same dimensionality, ensuring compatibility with the downstream prediction heads. Baselines are trained for 25 epochs with AdamW (lr 10^{-3}, weight decay 10^{-2}, batch size 128) with mixed-precision enabled, with checkpoint selection by minimum validation loss.

Algorithm 1 KL-Based Inference Gating

Require: Backbone repr $\mathbf{z}_u$, short-term-removed backbone repr $\mathbf{z}_u^-$, MoE repr $\mathbf{h}_u$, classifier ϕ_{cls}
Require: $s = 0.01$, $\alpha_{\min} = 0.05$, $\alpha_{\max} = 0.80$, $\tau = 0.012$
Ensure: Fused representation $\tilde{\mathbf{h}}_u$

$\mathbf{p}_u \leftarrow \phi_{\text{cls}}(\mathbf{z}_u), \quad \mathbf{p}_u^- \leftarrow \phi_{\text{cls}}(\mathbf{z}_u^-)$ ▷ project to interest simplex
$D_u \leftarrow D_{\text{KL}}(\mathbf{p}_u \| \mathbf{p}_u^-)$ ▷ drift intensity: effect of recent posts on interest dist.
if $D_u \le \tau$ **then** ▷ hard switch: negligible drift
    $\tilde{\mathbf{h}}_u \leftarrow \mathbf{z}_u$
else ▷ soft fusion: larger drift ⇒ larger α_u ⇒ more MoE
    $H_u \leftarrow -\mathbf{p}_u^\top \log \mathbf{p}_u \;/\; \log C$ ▷ normalized entropy of $\mathbf{p}_u$
    $\alpha_u \leftarrow \sigma\left(\dfrac{D_u - \tau}{s}\right) \cdot (0.5 + 0.5\, H_u)$
    $\alpha_u \leftarrow \text{clip}(\alpha_u,\, \alpha_{\min},\, \alpha_{\max})$
    $\tilde{\mathbf{h}}_u \leftarrow (1 - \alpha_u)\, \mathbf{z}_u + \alpha_u\, \mathbf{h}_u$
end if
return $\tilde{\mathbf{h}}_u$

D.5 Downstream Task Settings

All baselines and DUMoE are evaluated on the same train/val/test split and user behavioral windows, using the same frozen UNITE tweet embeddings and training configuration for fair comparison.

User interest classification. We adopt a frozen-encoder probing protocol: all model parameters are fixed after training, and a lightweight two-layer MLP probe (hidden size 256, ReLU, dropout 0.2) is trained on top of the exported representations using cross-entropy loss with Adam (lr 10^{-3}, batch size 512) for 20 epochs. Checkpoint selection is based on validation NDCG@3, and test-set metrics are computed in a single forward pass with no further tuning. KL divergence is reported as a calibration diagnostic rather than a training objective, computed against the label-smoothed target:

$$\mathbf{t} = 0.9 \cdot \text{onehot}(y) + \frac{0.1}{C} \cdot \mathbf{1}, \tag{17}$$

where C is the number of interest categories.

User interaction prediction. For each user u and target window t, training samples are constructed as triples $(\mathbf{h}_{u,t-1},\ \mathbf{v}^+,\ \mathbf{v}^-)$, where $\mathbf{h}_{u,t-1}$ is the frozen user representation derived from window $t-1$, ensuring no future information leakage. For each (u, t) pair, we sample 10 **positive posts** $\mathbf{v}^+$ from those the user actually interacted with within window t. **Negative posts** $\mathbf{v}^-$ are sampled at a 1:1 ratio from posts that satisfy: (1) falling within the same time range as window t, and (2) not belonging to user u. This time-range constraint encourages the model to capture genuine user–post affinity rather than temporal co-occurrence patterns. The prediction head first projects the post embedding $\mathbf{f} \in \mathbb{R}^{1536}$ into a 256-d space via a linear layer with ReLU and dropout ($p = 0.3$), then concatenates it with the user representation $\mathbf{z}_{u,t-1} \in \mathbb{R}^{256}$ and passes the result through a two-layer MLP to produce a scalar score s. The head is trained for 5 epochs with the BCE-based negative sampling loss defined in section 4.3, with only the projection and scoring parameters updated while the user encoder remains frozen.

D.6 Full Statistical Results

We report the full results of user interest classification and interaction prediction over three random seeds (42, 52, and 62). Each entry is presented as the mean and standard deviation across the three runs. For the classification task(Table 6), DUMoE significantly outperforms all baselines across all metrics (mean p-value < 0.01). For the interaction prediction task(Table 7), DUMoE achieves comparable results with the strongest baselines and significantly outperforms SASRec, BERT4Rec, UniSRec, and HM4SR (mean p-value < 0.05).

We also provide the full statistical results of the ablation study in Tables 8 and 9. The results further detail the contributions of individual architectural components and training objectives to both tasks.

Table 6: Full results of user interest classification. Values are mean ± standard deviation. The last column reports the mean p-value of each baseline compared with DUMoE.

Model	Hit@1 ↑	NDCG@3 ↑	Recall ↑	KL ↓	Mean p
SASRec	0.7669±0.0147	0.8717±0.0081	0.6950±0.0196	0.6092±0.0180	2.87×10^{-3}
BERT4Rec	0.7687±0.0055	0.8744±0.0038	0.6991±0.0058	0.5855±0.0128	3.36×10^{-4}
PTUM	0.8082±0.0022	0.8990±0.0007	0.7513±0.0040	0.4872±0.0041	5.35×10^{-4}
MIND	0.8158±0.0035	0.9034±0.0005	0.7594±0.0022	0.4780±0.0053	4.66×10^{-4}
HORAE	0.8006±0.0047	0.8963±0.0008	0.7403±0.0042	0.5121±0.0059	5.45×10^{-4}
HM4SR	0.7587±0.0066	0.8658±0.0032	0.6884±0.0091	0.6145±0.0124	1.41×10^{-3}
PeterRec	0.7850±0.0008	0.8855±0.0003	0.7172±0.0037	0.5371±0.0059	2.90×10^{-4}
UniSRec	0.7627±0.0029	0.8694±0.0025	0.6808±0.0057	0.6091±0.0117	3.65×10^{-4}
DUMoE	**0.8735±0.0025**	**0.9410±0.0019**	**0.8389±0.0045**	**0.3177±0.0071**	–

Table 7: Full results of user interaction prediction. Values are mean ± standard deviation. The last column reports the mean p-value of each baseline compared with DUMoE.

Model	Acc ↑	Recall ↑	F1 ↑	BCE ↓	Mean p
SASRec	0.8746±0.0016	0.8890±0.0093	0.8764±0.0019	0.2953±0.0044	0.0216
BERT4Rec	0.8781±0.0008	0.8920±0.0104	0.8798±0.0018	0.2917±0.0015	0.0176
PTUM	0.8877±0.0017	0.8979±0.0098	0.8889±0.0022	0.2696±0.0028	0.6610
MIND	0.8857±0.0012	0.8967±0.0104	0.8869±0.0022	0.2730±0.0030	0.5440
HORAE	0.8842±0.0014	0.9001±0.0109	0.8860±0.0024	0.2759±0.0041	0.3910
HM4SR	0.8670±0.0017	0.8783±0.0057	0.8685±0.0022	0.3071±0.0025	0.0023
PeterRec	0.8844±0.0011	0.8996±0.0056	0.8862±0.0011	0.2755±0.0023	0.3780
UniSRec	0.8752±0.0005	0.8939±0.0103	0.8775±0.0014	0.2950±0.0017	0.0158
DUMoE	**0.8831±0.0011**	**0.9220±0.0030**	**0.8875±0.0009**	**0.2755±0.0033**	–

Table 8: Full ablation results of user interest classification. Values are mean ± standard deviation.

Variant	Hit@1 ↑	NDCG@3 ↑	Recall ↑	KL ↓
Static Only	0.7808±0.0050	0.8802±0.0045	0.7409±0.0099	0.5595±0.0133
Short-term Only	0.8093±0.0033	0.8980±0.0030	0.7430±0.0039	0.4636±0.0153
Long-term Only	0.8632±0.0042	0.9359±0.0032	0.8378±0.0080	0.3402±0.0116
UNITE Mean-Pool	0.8283±0.0042	0.9150±0.0016	0.7698±0.0096	0.4358±0.0078
w/o Adapter	0.8719±0.0024	0.9397±0.0006	0.8365±0.0054	0.3206±0.0050
w/o Gating	0.8719±0.0063	0.9395±0.0043	0.8392±0.0089	0.3156±0.0132
w/o Stage-wise Training	0.8632±0.0062	0.9349±0.0034	0.8476±0.0059	0.3432±0.0145
$\mathcal{L}_{cls}$ Only	0.8681±0.0027	0.9373±0.0012	0.8443±0.0057	0.3160±0.0070
$\mathcal{L}_{rec}$ Only	0.8045±0.0118	0.9007±0.0050	0.7359±0.0160	0.4694±0.0109
Full model	**0.8735±0.0025**	**0.9410±0.0019**	**0.8389±0.0045**	**0.3177±0.0071**

Table 9: Full ablation results of user interaction prediction. Values are mean ± standard deviation.

Variant	Acc ↑	Recall ↑	F1 ↑	BCE ↓
Static Only	0.8088±0.0024	0.8393±0.0211	0.8144±0.0031	0.4085±0.0045
Short-term Only	0.8464±0.0006	0.8519±0.0051	0.8472±0.0002	0.3499±0.0010
Long-term Only	0.8622±0.0047	0.8921±0.0128	0.8662±0.0024	0.3185±0.0082
UNITE Mean-Pool	0.8561±0.0011	0.8770±0.0116	0.8590±0.0013	0.3329±0.0020
w/o Adapter	0.8624±0.0030	0.8849±0.0182	0.8654±0.0023	0.3187±0.0072
w/o Gating	0.8641±0.0044	0.8826±0.0173	0.8666±0.0019	0.3155±0.0077
w/o Stage-wise Training	0.8672±0.0030	0.8867±0.0079	0.8698±0.0021	0.3086±0.0048
$\mathcal{L}_{cls}$ Only	0.7948±0.0037	0.8469±0.0046	0.8050±0.0022	0.4409±0.0040
$\mathcal{L}_{rec}$ Only	0.8666±0.0029	0.8824±0.0078	0.8687±0.0018	0.3098±0.0077
Full model	**0.8831±0.0011**	**0.9220±0.0030**	**0.8875±0.0009**	**0.2755±0.0033**

E Expert Analysis

The MoE adapter is designed to capture category-related patterns in user representations while accommodating temporal changes in user interests. We further examine its behavior through category-level expert usage statistics and representative cases, focusing on the correspondence between interest categories and expert assignments, routing uncertainty, and adapter activation.

Metrics. We use three metrics to characterize expert behavior. First, the mean gate weight (MG) measures the average routing weight assigned to the expert corresponding to category c:

$$\mathrm{MG}_c = \frac{1}{N_c} \sum_{u:y_u=c} g_{u,k(c)}, \tag{18}$$

where N_c is the number of windows with ground-truth label c, and $k(c)$ denotes its corresponding expert. Unselected experts are assigned zero gate weight.

Second, mean routing entropy ($\overline{H}_c$) measures the uncertainty of expert selection. For each window u, we calculate the entropy of its normalized Top-K' gating weights:

$$H_u = -\sum_{j \in \mathcal{T}_u} p_{u,j} \log p_{u,j}, \qquad \overline{H}_c = \frac{1}{N_c} \sum_{u:y_u=c} H_u, \tag{19}$$

where $\mathcal{T}_u$ denotes the selected experts and $p_{u,j}$ is obtained by normalizing their non-negative gate weights. The logarithm is natural; thus, under Top-3 routing, the maximum entropy is $\ln 3 \approx 1.099$.

Finally, the KL-based adapter activation rate (AR) measures the proportion of windows for which the adapter is activated:

$$\mathrm{AR} = \frac{1}{N_c} \sum_{u:y_u=c} \mathbb{I}(D_u > \tau), \tag{20}$$

where D_u is the KL divergence between the current and reference interest distributions, τ is the activation threshold, and N_c is the number of windows with ground-truth label c. The adapter is activated when $D_u > \tau$; otherwise, the backbone representation is used directly.

Expert Usage and Label–Expert Correspondence. Table 10 summarizes expert usage and activation statistics across the 15 interest categories. Politics and Entertainment exhibit particularly strong category–expert correspondence, with F1 scores of 91.10% and 83.63%, respectively. Technology and Gaming also achieve relatively high recall (76.40% and 73.10%). In contrast, Sports and Fashion & Beauty are most frequently assigned to the Food and Science experts, respectively. Overall, the label–expert agreement is 72.0%, suggesting that the routing mechanism captures category-related patterns while allowing cross-category associations rather than enforcing a strict one-to-one mapping.

Table 10: Expert routing and adapter activation statistics across interest categories.

Category	Count	Top Expert	MG	Recall	Precision	H	AR (%)
Entertainment	4646	Entertainment	0.197	0.822	0.851	0.392	86.00%
Politics	4587	Politics	0.194	0.888	0.935	0.269	96.40%
Daily Life	2279	Daily Life	0.088	0.623	0.657	0.497	59.30%
Gaming	2157	Gaming	0.076	0.731	0.762	0.567	33.20%
Sports	1797	Food	0.000	0.521	0.478	0.948	90.20%
Technology	1783	Technology	0.101	0.764	0.798	0.471	35.10%
Business	1469	Business	0.079	0.512	0.539	0.667	23.80%
Art	985	Art	0.073	0.548	0.582	0.573	36.20%
Health & Fitness	939	Health & Fitness	0.017	0.447	0.473	0.996	28.40%
News	924	News	0.046	0.498	0.521	0.915	30.50%
Fashion & Beauty	805	Science	0.000	0.443	0.412	1.049	25.20%
Education	787	Education	0.025	0.413	0.438	0.967	24.00%
Culture	525	Culture	0.013	0.382	0.405	1.009	31.80%
Food	447	Food	0.058	0.402	0.420	1.053	14.50%
Science	410	Science	0.032	0.365	0.381	1.066	19.50%

Routing Entropy and Adapter Activation. The overall mean routing entropy is 0.584 across 24,540 test windows, below the maximum of $\ln 3 \approx 1.099$. Politics (0.269) and Entertainment (0.392) exhibit relatively concentrated routing distributions, whereas Science (1.066) and Fashion & Beauty (1.049) show more evenly distributed weights among the selected experts.

The overall adapter activation rate is 59.8%, with substantial variation across categories. For example, Politics and Sports have relatively high activation rates of 96.40% and 90.20%, respectively, while Food and Science exhibit lower rates of 14.50% and 19.50%. These differences indicate varying levels of estimated temporal drift across interest categories.

Table 11: Representative cases illustrating expert routing and adapter activation.

Pattern	User Window	Label	Top Expert	Entropy	KL
Stable Routing	U_001_w011	Entertainment	Entertainment	0.023	0.104
Stable Routing	U_002_w010	Entertainment	Entertainment	0.026	0.017
High Uncertainty	U_003_w010	Health & Fitness	Health & Fitness	1.099	0.005
High Uncertainty	U_004_w010	Health & Fitness	Art	1.099	0.003
Drift Triggered	U_005_w010	Politics	Politics	0.213	8.04
Drift Triggered	U_006_w010	Health & Fitness	Health & Fitness	0.948	5.77

Qualitative Case Studies. Table 11 presents representative windows illustrating three routing and activation patterns. The stable-routing cases exhibit low routing entropy, with the corresponding category expert receiving the highest weight. Although their KL divergence is relatively small, both values exceed the activation threshold, and the adapter is activated. The high-uncertainty cases exhibit maximum routing entropy (1.099), indicating nearly uniform weights across the three selected experts. However, their KL divergence remains below the threshold, so the adapter is not activated. In contrast, the drift-triggered cases exhibit substantially higher KL divergence, leading to adapter activation. These examples highlight the complementary roles of routing entropy and KL-based activation: the former characterizes the distribution of expert weights, whereas the latter determines whether adapter fusion is applied.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes],

   Justification: We accurately describe our contributions in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 5.7, including annotation noise, fixed window lengths, static expert count, and offline training.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not contain theoretical result.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the model architecture, training procedure, dataset construction, and evaluation settings (Sections 4 and 5). We will release anonymized data and code to support reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release anonymized data and full code upon acceptance, including preprocessing scripts, model implementation, and training/evaluation pipelines. Detailed instructions for data preparation, environment setup, and running experiments will be provided in the supplemental material to ensure faithful reproduction of the main results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify data splits, optimizer, batch size, and hyperparameter selection in Section 5 and Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the mean and standard deviation over three random seeds in Appendix D.6, together with statistical significance tests comparing DUMoE against the baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report GPU experiments compute resources in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed and complied with the NeurIPS Code of Ethics. Data is collected from X's public API in accordance with its terms of service.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss potential benefits and risks in Appendix B, along with mitigation strategies for data release.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We will release only anonymized user IDs and tweet IDs, and will apply content filtering to remove harmful posts prior to public release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all baseline models and the encoder with their original references. X data is collected under the platform's academic access terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We document the dataset construction, domain taxonomy, and annotation protocol in Section 5.1 and Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The dataset is constructed from publicly available social media data and primarily annotated automatically using GPT-5.1. Manual inspection by research team members is conducted on a small subset for annotation quality verification. No crowdsourcing experiments or human subject studies are conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We only use publicly available social media data for human annotation quality checks, which does not require IRB approval under our institutional guidelines.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: GPT-5.1 is used in the pseudo-label annotation pipeline (Section 5.1), directly affecting dataset construction and experimental results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.