Tsinghua Science and Technology
Doi: 10.26599/TST.2026.90100036

Research Article

Enhanced Adversarial Sequential Recommendation Fused with Siamese Networks for Edge Environments

Junsan Zhang, Fengmei Ding, Jia Luo (✉), Yunpeng Si, Lei Shi, and Peiying Zhang

Abstract With the rapid development of the Internet of Things (IoT), massive user behavior data are continuously generated at the network edge, making intelligent service optimization under constrained computational resources and dynamic environments a critical challenge. As a representative AI-driven service, sequential recommendation predicts users' future preferences by modeling historical behavior sequences. Existing Generative Adversarial Network (GAN)-based sequential recommendation methods suffer from the discriminator's hard-boundary classification, leading to feature misjudgment. Moreover, Transformer-based dot-product attention is sensitive to uncertain and noisy data, while window truncation limits long-range dependency modeling. Therefore, we propose an **E**nhanced **A**dversarial Sequential **Rec**ommendation fused with **S**iamese **N**etworks for Edge Environments (EASNRec). With a lightweight architecture and robust feature modeling, the method is suitable for edge deployment under resource constraints and noisy conditions. First, a Siamese network is introduced into the GAN framework to project real and generated sequences into a unified feature space, where a similarity loss is constructed to jointly optimize the adversarial learning process of the generator and discriminator. Second, EASNRec integrates dynamic random augmentation with contrastive learning to explore the latent structure of user behavior sequences from multiple views, thereby enhancing model robustness. Furthermore, we design an **U**ncertainty **A**dversarial **M**emory **M**echanism (UAMM) that enables the model to suppress high-uncertainty and noisy data while capturing long-term dependencies in user behavior sequences. Experimental results show that EASNRec significantly outperforms state-of-the-art sequential recommendation models on three public datasets.

Keywords Edge Computing, Sequential Recommendation, GAN, Siamese Network, Contrastive Learning

1 Introduction

With the rapid proliferation of IoT technologies and the widespread deployment of smart terminals, user behavior data at the network edge have been growing explosively and are highly distributed. Edge computing provides a new paradigm for intelligent services with low latency and high responsiveness [1]. However, the limited computational resources and dynamic conditions of edge environments still pose critical challenges for service deployment. Against

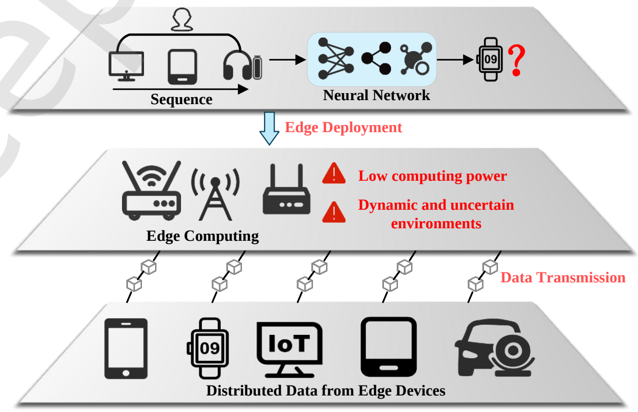

Fig. 1 The architecture of intelligent sequential service optimization in edge computing environments. The bottom layer collects user behavior data, the middle layer performs edge computing, and the top layer deploys the sequential recommendation model at the edge nodes.

• Junsan Zhang, Fengmei Ding, Yunpeng Si, and Peiying Zhang are with the Qingdao Institute of Software, College of Computer Science and Technology, Shandong Key Laboratory of Intelligent Oil and Gas Industrial Software, China University of Petroleum (East China), Qingdao 266580, China. E-mail: zhangjunsan@upc.edu.cn, z23070113@s.upc.edu.cn, z24070122@s.upc.edu.cn, zhangpeiying@upc.edu.cn
• Jia Luo is with the College of Economics and Management, Beijing University of Technology, Beijing 100124, China, is also with the Chongqing Research Institute, Beijing University of Technology, Chongqing 401121, China, and is also with the Interdisciplinary Faculty of Science and Engineering, Shimane University, Shimane 6908504, Japan. E-mail: jialuo@riko.shimane-u.ac.jp.
• Lei Shi is with the State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China, is also with the Key Laboratory of Public Opinion Governance and Computational Communication, Sichuan Police College, Luzhou 646000, China. E-mail: leiky_shi@cuc.edu.cn.
Manuscript received: 2026-01-12; revised: 2026-02-17; accepted: 2026-03-23

this backdrop, leveraging artificial intelligence to optimize intelligent services for edge environments has become a research focus [2]. Fig. 1 illustrates a closed-loop workflow that spans from bottom-layer massive behavioral data collection to intermediate-layer edge node transmission, and finally to top-layer sequential recommendation model deployment. As a core technology for information filtering and personalized services, recommender systems have gradually become an essential component of intelligent service frameworks. Traditional recommendation methods typically rely on users' static preference information, making it difficult to capture the dynamic evolution of user interests [3, 4]. To address this, sequential recommendation explicitly models the temporal dependencies in user behaviors and has emerged as a key approach for capturing dynamic user preferences in personalized recommendation.

In the early stages of sequential recommendation, researchers primarily relied on traditional methods such as Markov chain models [5] and matrix factorization [6]. With the advent of deep learning technologies, neural network-based sequential recommendation models have gradually become mainstream. For instance, models based on Convolutional Neural Networks (CNNs) [7] and Recurrent Neural Networks (RNNs) [8, 9] can fully leverage the powerful feature extraction capabilities of deep learning to uncover latent patterns and dependencies in user behavior sequences. In recent years, Transformer-based models [10–12] employ self-attention mechanisms to learn the importance of items and the correlations among behaviors, further enhancing the sequential recommendation performance. However, existing methods still face critical challenges when applying Transformers, which are particularly pronounced in edge environments: (1) User implicit feedback often contains a large amount of ambiguous behavior. For example, a user may generally prefer bread but occasionally browse baking tools. Such short-term interests or latent preferences are difficult for dot-product attention to evaluate properly, and traditional mechanisms tend to compress all historical behaviors into a deterministic vector. (2) Dot-product attention is overly sensitive to noisy interactions [13], which can amplify spurious correlations among low-quality features. For instance, if a user who loves bread accidentally clicks on a refrigerator, self-attention may incorrectly associate the refrigerator with bread, leading to irrelevant subsequent recommendations. (3) To balance computational efficiency, current methods often adopt window truncation strategies for long sequences, but this engineering compromise results in the loss of distant behavioral information.

The aforementioned limitations of Transformer models are significantly amplified under the inherent constraints of edge computing. On the one hand, the high-noise characteristics of edge data make the dot-product attention mechanism of Transformers more prone to amplifying spurious correlations, while the lack of sufficient contextual information at the edge side prevents effective filtering, leading to performance degradation. On the other hand, the strict computational constraints of edge nodes force the adoption of window truncation strategies, thus sacrificing the ability to model long-range dependencies.

In recent years, Generative Adversarial Networks (GANs) and their variants have demonstrated tremendous potential in recommendation systems. By engaging the generator and discriminator in adversarial training, GANs can generate high-quality user interaction sequences, thereby improving recommendation performance. ELECRec [14] for the first time directly reconstructs the sequential recommendation model as a discriminator, optimizes the negative sample generation process through end-to-end adversarial training, and establishes a new benchmark for sequence modeling. Despite their strong generative capabilities [15, 16], GANs have certain shortcomings that may persist and affect overall performance, and these limitations are particularly pronounced in edge environments: (1) Traditional discriminators typically employ a "hard-boundary" classification strategy, assigning a label of 1 to real samples and 0 to generated samples. However, this binary classification strategy forces the discriminator to construct inflexible decision boundaries in the feature space, leading to misclassification of samples near the boundary due to insufficient separability. Fig. 2 illustrates the concept of the "hard-boundary" classification strategy. In edge environments, data collected at the edge side originate from a large number of distributed terminals and commonly suffer from issues such as uneven sampling and severe noise interference, which inherently leads to greater overlap between real and generated samples in the feature space. Under such circumstances, if the discriminator adopts a binary 0/1 hard classification strategy, it is forced to construct rigid decision boundaries over intrinsically ambiguous samples, thereby significantly increasing the misclassification probability of samples near the boundary. Such misleading gradients arising from these misjudgments cannot be effectively corrected under the limited resources at the edge, thereby exacerbating the instability of adversarial training. (2) The generator relies on feedback from the discriminator for parameter updates. However, the "hard-boundary" classification can only measure the degree of matching between generated and real samples. It cannot provide fine-grained supervision on local patterns, such as item correlations and temporal dependencies within sequences, making it difficult for the generator to produce high-quality behavior sequences. In edge environments, the binary decision signals provided by hard-boundary classification are overly coarse and insufficient to reveal which specific structural aspects of the generated sequences should be improved, leading to inefficient learning of the generator and consequently limiting further improvements in overall recommendation performance.

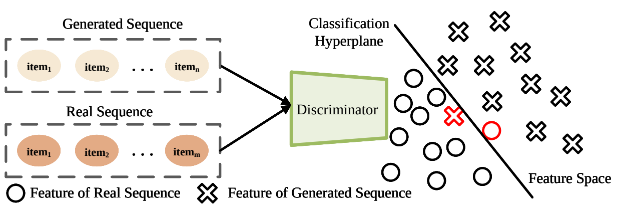

Fig. 2 Illustration of hard-boundary classification in GAN. The binary classifier forces a rigid decision boundary between real and generated sequence features. Samples marked in red are highly prone to misclassification due to the inherent ambiguity of their features.

To address the aforementioned issues, this paper proposes an **E**nhanced **A**dversarial Sequential **Rec**ommendation with **S**iamese **N**etworks for Edge Environments (EASNRec). In this method, the generator takes masked user interaction sequences collected by edge terminals as input to generate candidate sequences, while the discriminator performs accurate classification. This work innovatively introduces a Siamese network, reconstructing the traditional binary adversarial paradigm into a semantic metric learning process within a unified latent space. Specifically, the Siamese network maps both real and generated sequences into the same feature space, and by optimizing a similarity-based loss, it reduces the distance between the generated and real distributions, thereby encouraging the discriminator to learn a smoother feature space distribution. Meanwhile, real sequences are encoded and combined with dynamic random augmentation and contrastive learning techniques to model sequence features from multiple perspectives, further enhancing feature representation capabilities. Furthermore, this work incorporates an Uncertainty Adversarial Memory Mechanism (UAMM) into the encoder, generator, and discriminator at the edge computing layer to enhance the traditional Transformer module.

The main contributions of this work are summarized below.

- We propose an adversarial network method integrating a Siamese network. Real and generated sequences are projected into a unified feature space, and a similarity-based loss is constructed to guide adversarial optimization, enabling a smooth transition from "hard classification" to "soft classification".
- We design a novel mechanism called UAMM, which integrates penalty correction terms and adversarial Gaussian noise into the attention mechanism to dynamically suppress the interference of high-uncertainty and noisy features. Additionally, it leverages the memory slots of the External Memory Mechanism (EMM) to model the long-term dependencies of user behaviors.
- We conduct experiments on three public datasets and demonstrate the effectiveness of our model.

The structure of this paper is organized as follows: Section 2 reviews related work; Section 3 presents the problem formulation and data augmentation methods; Section 4 details the EASNRec model framework; Section 5 reports the experimental evaluation and result analysis; and Section 6 concludes the paper and outlines future work.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation models learn user preferences and predict future behaviors by uncovering the intrinsic order and dependencies within user-item interaction sequences. Traditional sequential recommendation models can be broadly categorized into three main types: frequent pattern mining, Markov models, and latent factor models. Frequent pattern mining extracts various patterns from interaction sequences. Although it offers strong interpretability, it is difficult to handle complex and large-scale datasets. Early studies by Rendle et al. [5] combined first-order Markov chains with matrix factorization, effectively improving user behavior prediction. Latent factor models estimate sequential transition relationships by learning implicit representations. However, most of these methods are based on linear or shallow assumptions, which limits their ability to capture complex nonlinear patterns and long-range dependencies. To address the limitations of traditional methods, researchers have progressively incorporated deep learning techniques. CNN-based approaches reconstruct behavior sequences into two-dimensional matrices. For instance, Caser [7] captures sequential patterns and user profile features through horizontal and vertical convolutions. RNNs [17] are an early dominant approach due to their sequence modeling capability. GRU4Rec [8] pioneers the integration of Gated Recurrent Units (GRU) in recommendation systems, effectively capturing short-term behavioral dependencies.

The introduction of the Transformer architecture marks a new era in sequential recommendation [18]. SASRec [11] employs unidirectional self-attention for dynamic interest modeling, while BERT4Rec [12] utilizes bidirectional attention to capture global dependencies among user behaviors. However, although Transformers exhibit significant advantages in sequential modeling, their attention mechanisms are sensitive to uncertainty and noise within sequences, and the window truncation strategy commonly adopted in practical implementations limits the model 's ability to capture long-range dependencies, thereby affecting the accuracy and robustness of sequential recommendation. To address these issues, this work designs UAMM that effectively enhances robustness and improves long-range dependency modeling in sequential recommendation.

As a powerful generative deep learning model, GANs have been gradually introduced into the field of sequential recommendation through adversarial training between a generator and a discriminator. For example, SparseEnNet [19] proposed a robust adversarial augmentation method that enhances recommendation performance by applying strengthened augmentation to generated sequences; ECL-SR [20] combined contrastive learning with a generative adversarial framework, using contrastive constraints in the feature space to improve the similarity between generated and real sequences and thus enhance sequential representations. However, traditional GAN architectures are limited by the discriminator 's "hard-boundary" classification strategy, which can easily lead to sample confusion in the feature space. To address this, we propose an adversarial method integrating a Siamese network, where a similarity-based loss guides optimization, enabling a smooth transition from "hard" to "soft" classification.

2.2 Siamese Networks

Siamese networks are a deep learning architecture based on parameter sharing, specifically designed for similarity measurement tasks. They have been widely applied in various fields, including facial recognition [21], object tracking [22], and one-shot learning [23]. Originally introduced by Bromley et al. in 1994 [24] for signature verification, the core idea of Siamese networks is to learn the semantic similarity between input pairs through a dual-branch network with shared weights.

Compared with traditional similarity measurement methods, the advantage of Siamese networks lies in their ability to automatically extract and learn nonlinear relationships between features through deep networks. The architecture of a Siamese network, with its parameter-sharing mechanism across twin subnetworks, ensures consistency in the feature mapping space for input pairs. Research on Siamese networks continues to advance; for example, by introducing modules that integrate channel and spatial attention [25], the network can adaptively focus on more discriminative regions, thereby enhancing tracking robustness. As a cornerstone of metric learning, Siamese networks have inspired a series of few-shot learning models, such as prototypical networks and matching networks [26]. Some studies have explicitly combined Siamese networks with meta-learning frameworks, constructing models like Meta-SN, which improve few-shot text classification performance [27]. These studies demonstrate that Siamese networks possess strong generalizability in feature alignment and similarity constraint modeling, providing technical support for their application in this work.

3 Preliminaries

3.1 Problem Definition

We denote the user set as U and the item set as V. For any user $u \in U$, the interaction behavior sequence is expressed as $S_u = \{v_1^u, v_2^u, \ldots, v_t^u\}$, where v_j^u denotes the item that the user u interacted with at the j-th time step. The goal of sequential recommendation is to predict the item v^* that the user is most likely to interact with at the next time step $|S_u| + 1$ based on the current sequence S_u, which can be expressed as:

$$v^* = \arg\max_{v \in V} P(v_{|S_u|+1}^u = v \mid S_u) \tag{1}$$

The model is optimized by minimizing the negative log-likelihood loss function L_{rec}.

$$L_{\text{rec}} = \sum_{u=1}^{|U|} \sum_{t=1}^{|S_u|-1} -\log p_\theta(v_{t+1}^u \mid v_1^u, v_2^u, \ldots, v_t^u) \tag{2}$$

where θ represents the model parameters, and $|U|$ is the total number of users.

3.2 Data Augmentation Strategies

To improve the model's robustness against interaction noise, this section adopts a representation-level data augmentation strategy that acts directly on user embedding vectors. Let h_u denote the embedding sequence of user u. The specific methods are described as follows:

- **Dropout:** This method randomly masks the dimensions of embedding vectors with a given probability p, preventing the model from over-relying on specific features during training. Formally, let $m \in \{0,1\}^{L\times d}$ be a binary mask matrix sampled from a Bernoulli distribution:

$$m_{i,j} \sim \text{Bernoulli}(1-p), \quad \begin{matrix} i = 1,\dots,L, \\ j = 1,\dots,d \end{matrix} \tag{3}$$

The augmented embedding sequence is expressed as:

$$h_u^D = h_u \odot m \tag{4}$$

where $\odot$ denotes element-wise multiplication.

- **Perturbation:**This method improves model robustness by adding random noise to embedding vectors. Specifically, a controllable noise vector Δ_u is added to each embedding vector:

$$h_u^P = h_u + \Delta_u \tag{5}$$

where

$$\Delta_u = \hat{\Delta} \odot h_u, \quad \hat{\Delta} \sim U(0,\epsilon) \tag{6}$$

Here, ϵ is used to control the magnitude of the perturbation.

- **Mask:**This method replaces some embedding vectors with zero vectors to force the model to infer missing information from the remaining context. A subset of positions $T_u \subseteq \{1,\dots,L\}$ is sampled uniformly at random with a masking ratio γ, and the embedding vectors at the masked positions are set to zero:

$$h_u^M[t] = \begin{cases} 0, & t \in T_u \\ h_u[t], & t \notin T_u \end{cases}, \quad |T_u| = \lfloor \gamma L \rfloor \tag{7}$$

4 Methodology

Fig. 3 illustrates the overall framework of EASNRec, where all core components are deployed at the edge computing layer. In this section, we present a detailed description of each component of EASNRec.

4.1 Data Augmentation and Contrastive Learning

To enhance the robustness of sequence representations, this section proposes a joint optimization framework that combines dynamic data augmentation with contrastive learning. Given a user's behavior sequence S_u, it is mapped to the latent space via an encoder to obtain the latent representation at each time step:

$$h_t^u = f_\theta(v_1^u, v_2^u, \dots, v_t^u) \tag{8}$$

where f_θ is the Transformer encoder function, and θ represents the model parameters.

After obtaining the latent representation h_t^u at each time step, the overall latent representation of the user's behavior sequence h_u can be derived by aggregating these representations. Here, the average of the last k time steps is taken, as follows:

$$h_u = \text{Average}(h_{L-k+1}^u, h_{L-k+2}^u, \dots, h_L^u) \tag{9}$$

Unlike traditional single data augmentation methods, we adopt a dynamic random data augmentation strategy for the latent representations. Specifically, in each training iteration, the model independently and randomly selects one augmentation for each input sequence. Concretely, one augmentation is randomly chosen with equal probability from a candidate pool $\mathcal{A} = \{\text{Dropout}, \text{Perturbation}, \text{Mask}\}$, and this selection is performed independently for every forward pass. This dynamic mechanism ensures that the model is exposed to a diverse set of augmented variants throughout training, mitigating the bias that may arise from a single augmentation and thereby more comprehensively enhancing the robustness of the representations.

During each training epoch, this mechanism generates augmented samples h_u^+, which are then used alongside the original samples h_u for contrastive learning. The goal of contrastive learning is to make the representations of the augmented and original samples as similar as possible while ensuring they are distinct from other negative samples. To this end, we introduce the following contrastive loss function:

$$L_{cl} = \sum_{u=1}^{|U|} -\log \frac{e^{\text{sim}(h_u,h_u^+)/\tau}}{e^{\text{sim}(h_u,h_u^+)/\tau} + \sum_{i\neq u, i\in M(u)} e^{\text{sim}(h_u,h_i)/\tau}} \tag{10}$$

where $\text{sim}(\cdot,\cdot)$ is the cosine similarity function, τ is the temperature coefficient, and $M(u)$ denotes the users in the mini batch that contains u.

4.2 GAN Fused with Siamese Network

This study proposes incorporating a Siamese network into the GAN framework. This design leverages the Siamese network to construct a deep feature contrast space between generated and real sequences, thereby effectively facilitating the synergistic optimization of the generator and discriminator. This section provides a detailed explanation of the core modules of this architecture and its optimization mechanism.

4.2.1 Siamese Network

Recent studies have demonstrated the significant potential of Siamese networks in optimizing recommendation systems [28]. This paper innovatively integrates the Siamese network into the GAN framework, enabling joint optimization of generation and discrimination by measuring the feature similarity between generated and real sequences in the

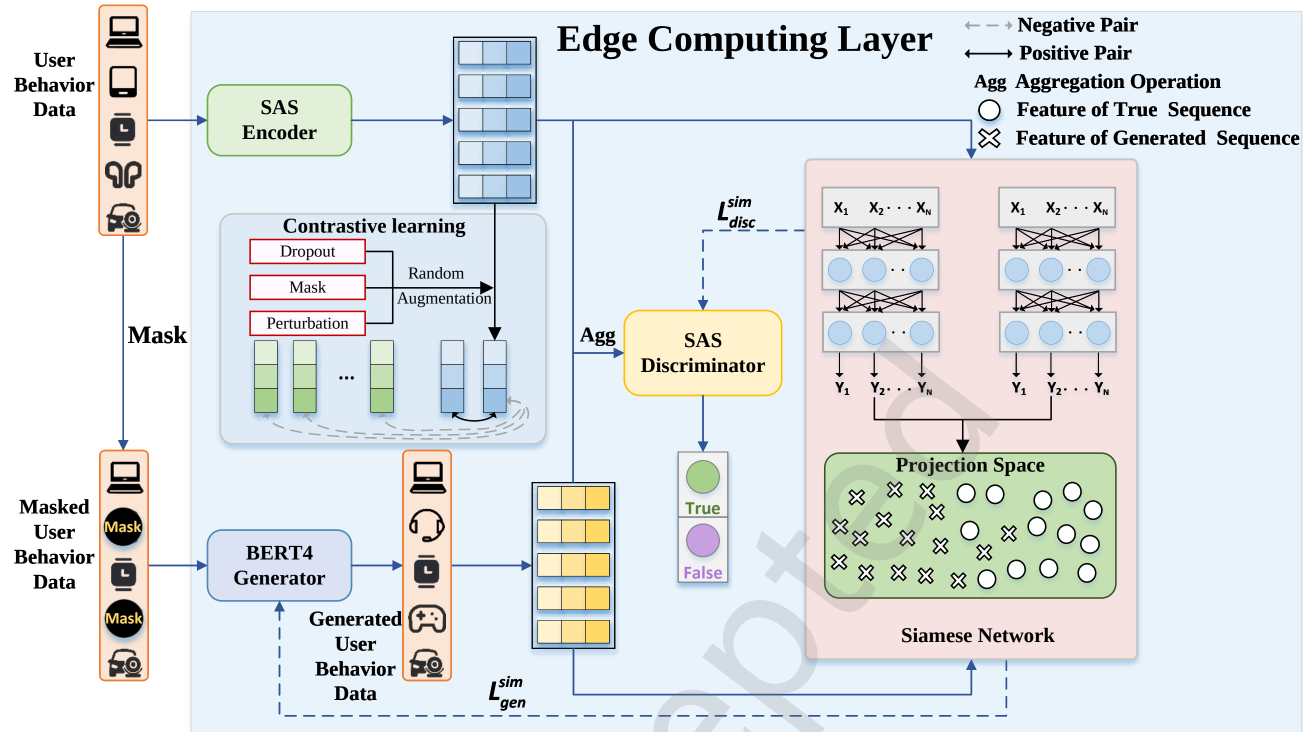

Fig. 3 Overview of the EASNRec framework for edge environments. The SAS encoder encodes user behavior data into embedding vectors, the BERT4 generator and SAS discriminator perform adversarial training, and the Siamese network maps real and generated sequences into a shared projection space to achieve soft classification.

projection space. The Siamese network employs a two-layer fully connected structure to achieve dimensionality reduction in the feature space, as illustrated in Fig. 4. Specifically, the generated and real sequences are fed into a dual-branch network with shared weights to obtain their respective feature representations:

$$f_{\text{gen}} = \text{Sn}(G(S_u)) \tag{11}$$

$$f_{\text{real}} = \text{Sn}(S_u) \tag{12}$$

where $G(\cdot)$ denotes the generator, and $\text{Sn}(\cdot)$ denotes the Siamese network.

To quantify the feature similarity between the two types of sequences, we employ the following cosine similarity function:

$$Sim\_p = \cos(f_{\text{real}}, f_{\text{gen}}) \tag{13}$$

4.2.2 *Generator and Discriminator*

In the GAN architecture, the generator and the discriminator are jointly optimized through adversarial training. EASNRec integrates the advantages of sequential recommendation models with the adversarial learning mechanism.

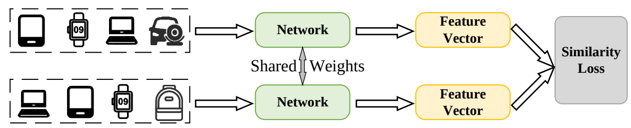

Fig. 4 The dual-branch shared-weight architecture of the Siamese network.

Generator: A BERT4Rec model based on a bidirectional Transformer is used as the generator G. Given a user interaction sequence S_u, we first apply a masking operation to hide some items in the sequence. Specifically, a predefined masking ratio γ is used to randomly mask a subset of items, resulting in a partially masked interaction sequence S'_u. The generator G reconstructs the original sequence based on S'_u, producing a generated sequence $S''_u = G(S'_u)$. The generated and real sequences are then fed into the Siamese network to compute the similarity loss. To train the generator G, we optimize its performance by minimizing the generator's loss function, which consists of both the standard cross-entropy loss and the similarity loss. The loss function of the generator is as follows:

$$\min L_{gen} = \sum_{u=1}^{|U|} \sum_{v_m \in S_u^m} -\frac{1}{|S_u^m|} \log p(v_m = v_m^* | S'_u) + \lambda_1 L_{gen}^{\text{sim}} \tag{14}$$

$$L_{gen}^{\text{sim}} = -Sim\_p \tag{15}$$

where S_u^m is the set of masked items for user u, v_m^* is the ground truth item at the masked position, and $p(v_m = v_m^*|S_u')$ is the probability that the generator predicts the masked position v_m as the true item v_m^* given the masked sequence S_u'. L_{gen}^{sim} is the generator similarity loss. The parameter λ_1 is introduced to balance the generative loss and the similarity measure.

Discriminator: A SASRec model based on a self-attention mechanism is used as the discriminator D. The primary task of the discriminator D is to predict the authenticity of each item in the user interaction sequence. The discriminator takes the real and the generated sequences processed through an aggregation function as input and distinguishes between them. Its loss function consists of both the cross-entropy loss and the similarity loss:

$$\begin{aligned}\min L_{\text{disc}} = &\sum_{u=1}^{|U|}\sum_{t=2}^{T} -\mathbb{I}(v_t^u = v_t^{u''})\log\sigma(w^T D(S_u'', h_t^u, t)) \\ &- \mathbb{I}(v_t^u \neq v_t^{u''})\log(1-\sigma(w^T D(S_u'', h_t^u, t))) \\ &+ \lambda_2 L_{\text{disc}}^{\text{sim}}\end{aligned} \tag{16}$$

$$L_{\text{disc}}^{\text{sim}} = -(1 - Sim\_p) \tag{17}$$

where T is the maximum sequence length, $v_t^{u''}$ is the t^{th} item in the generated sequence, $\mathbb{I}$ is an indicator function, σ is the Sigmoid function, and w is a learnable parameter matrix. $L_{\text{disc}}^{\text{sim}}$ is the discriminator similarity loss. The parameter λ_2 is introduced to balance the discriminatory loss and the similarity measure.

During adversarial training, the generator and discriminator update their parameters by minimizing their respective loss functions. The similarity losses for both components ($-Sim\_p$ and $-(1-Sim\_p)$) are minimized simultaneously, gradually enhancing the feature similarity between the generated and real sequences. Eventually, the model converges to an equilibrium state. EASNRec improves the authenticity of generated sequences by designing similarity loss functions in the feature projection space, which encourages the features of generated sequences to approximate the distribution of real sequences.

4.2.3 *Effect of Siamese Network on Adversarial Optimization*

The reason the Siamese network can effectively enhance the stability of adversarial training and improve generation quality is that it reconstructs the traditional hard-boundary discrimination into a continuous similarity measurement in the feature space. Specifically, the generator's similarity loss encourages the generated sequences to gradually approach the real samples in the feature space, enabling the model to retain effective optimization directions even when it becomes close to the real data distribution. Meanwhile, the discriminator's similarity loss promotes a smoother decision boundary in the feature space, thereby preventing misclassification and gradient oscillation caused by hard boundaries. By jointly optimizing the adversarial loss and the similarity loss (Eqs. (14) and (16)), the model achieves a dynamic equilibrium during training in which the generated features gradually align with the real data distribution and the discriminator's decision boundary becomes progressively softened. This fundamentally alleviates the optimization inefficiency of traditional GANs caused by sparse or unstable gradients when the generated and real sequences become highly similar.

4.3 Uncertainty-Adversarial Memory Mechanism

This section proposes the UAMM (see Fig. 5), which integrates the Uncertainty-Aware Adversarial Attention (UAA-Attention) with the EMM.

4.3.1 *Uncertainty-Aware Adversarial Attention*

To reduce the sensitivity of conventional attention mechanisms to uncertain and noisy feature data, this section proposes a UAA-Attention mechanism, as illustrated in Fig. 5. This mechanism employs two parallel branches: the upper branch computes the original attention, while the lower branch computes attention after injecting adversarial noise into the query and key. The core idea is to introduce a penalization term for the queries and keys to quantify feature uncertainty, and to inject controllable noise into the base features to simulate stochastic perturbations in the data, thereby enhancing the model's robustness to input disturbances.

Traditional attention mechanisms solely compute the similarity between the base queries and keys, whereas in this paper, penalty correction terms are introduced into the attention computation process to reduce the weights of features with high uncertainty. Specifically, given an input sequence tensor $X \in \mathbb{R}^{B\times L\times d}$ (where B denotes the batch size, L denotes the sequence length, and d denotes the hidden layer dimension), the original attention score is computed as the semantic relevance minus an uncertainty penalty term:

$$A_{\text{ori}} = \frac{Q\cdot K^T}{\sqrt{d_h}} - \frac{Q_p\cdot K_p^T}{\sqrt{d_h}} \tag{18}$$

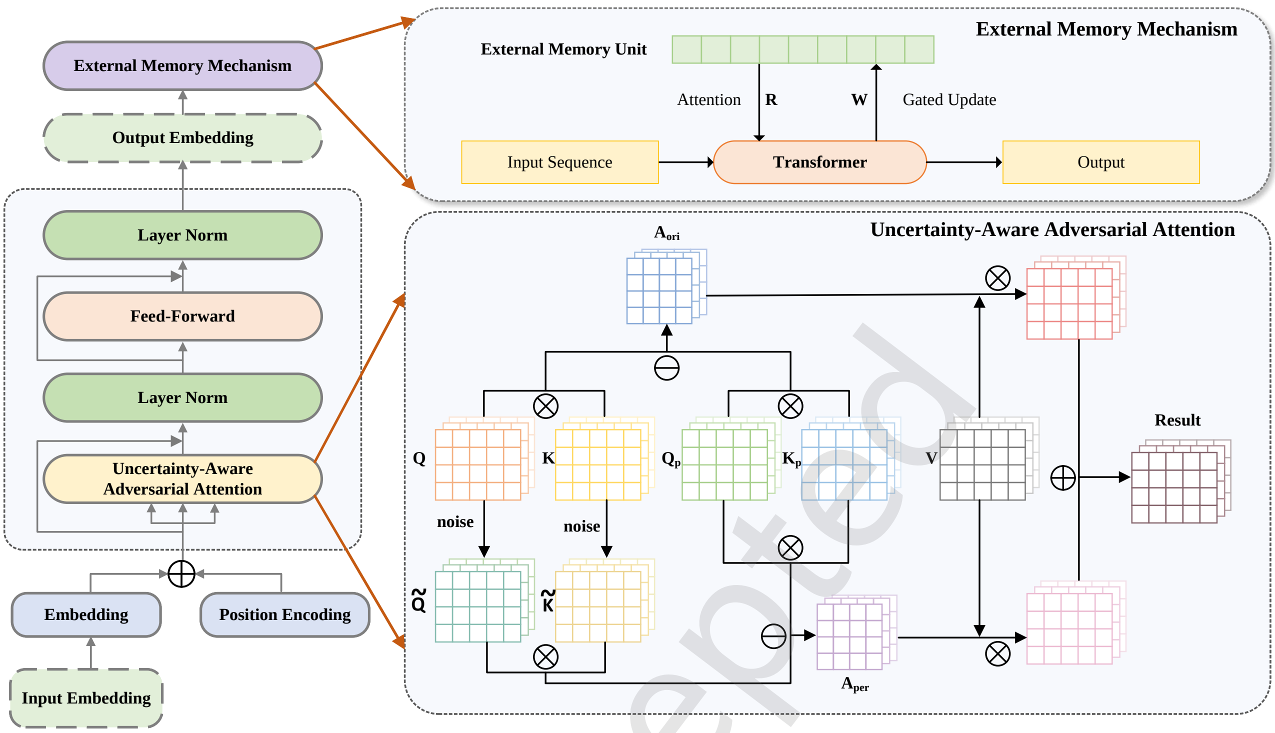

Fig. 5 Uncertainty-aware adversarial memory mechanism, integrating uncertainty-aware adversarial attention mechanism and external memory mechanism.

where $Q = X \cdot W_q$, $K = X \cdot W_k$, $Q_p = X \cdot W_q^p$, $K_p = X \cdot W_k^p$, and $W_q, W_k, W_q^p, W_k^p \in \mathbb{R}^{d \times d_h}$, where d_h is the dimension of the attention head. In Eq.(18), the first term represents the standard semantic relevance score, while the second term corresponds to the introduced uncertainty penalty. The learnable projection weights W_q^p and W_k^p are driven and supervised by the end-to-end recommendation task loss, and are specifically responsible for extracting and quantifying uncertainty from the input features. This design enables the attention allocation to explicitly subtract a measure of feature uncertainty. For feature combinations with high uncertainty (i.e., large values of $Q_p \cdot K_p^\top$), the resulting attention weights are significantly suppressed; conversely, for features with high certainty, the penalty term becomes small and the semantic relevance score dominates the attention distribution.

To further enhance the robustness of the model, Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is added to the query and key, generating perturbed attention scores:

$$A_{\text{per}} = \frac{\widetilde{Q} \cdot \widetilde{K}^T}{\sqrt{d_h}} - \frac{Q_p \cdot K_p^T}{\sqrt{d_h}} \tag{19}$$

where $\widetilde{Q}$ and $\widetilde{K}$ represent the query and the key after the application of random perturbations, respectively.

After computing the original attention scores A_{ori} and the perturbed attention scores A_{per}, the model transforms them into context vectors, and the final context representation is obtained by summing the two vectors.

$$C_{\text{final}} = \text{Softmax}(A_{\text{ori}})V + \text{Softmax}(A_{\text{per}})V \tag{20}$$

4.3.2 External Memory Mechanism

Due to the enormous computational cost of directly performing global modeling on long sequences, existing methods typically employ a window truncation strategy. While this approach reduces the computational burden to some extent, it also tends to overlook important long-term information in the sequence that reflects the periodicity and gradual evolution of user behavior. The EMM module proposed in this section (as shown in Fig. 5) is designed with a two-stage cooperative mechanism consisting of memory reading and writing.

Memory Reading Stage: The model extracts long-term interest representations relevant to the current behavior through attention interactions between the query vector and the memory matrix. Specifically, given the Transformer output at time step t, h_t^{trm}, as the query vector, the model computes its matching scores with each memory slot via a learnable matrix W_a and obtains attention weights through a Softmax:

$$\alpha = \text{Softmax}\left(h_t^{\text{trm}} \cdot W_a\right) \tag{21}$$

Here, W_a maps the query vector into the same semantic space as the memory slots to measure their relevance. The attention weights are then used to perform a weighted sum over the memory matrix $M \in \mathbb{R}^{N \times d_m}$ (where N is the number of memory slots and d_m is the dimension of each memory vector) to obtain the long-term interest representation:

$$m_{\text{read}} = \sum_{i=1}^{N} \alpha_i M_i \tag{22}$$

Memory Writing Stage: To dynamically store new sequence features from the Transformer, we design an external memory update strategy based on attention-weighted aggregation and gating mechanisms. Given the Transformer output at time step t, h_t^{trm}, the model first computes its compatibility with the write weight matrix W_w to obtain the write attention distribution:

$$\beta = \text{Softmax}(h_t^{\text{trm}} \cdot W_w) \tag{23}$$

Based on this attention distribution, the new sequence representation is aggregated into the memory slots to form the write value:

$$V = \beta^\top \cdot h_t^{\text{trm}} \tag{24}$$

The write value is then concatenated with the old memory and fed into the gating mechanism to generate the update gate and reset gate:

$$\begin{aligned} g_u &= \text{Sigmoid}(W_u[M; V]) \\ g_r &= \text{Sigmoid}(W_r[M; V]) \end{aligned} \tag{25}$$

Here, the reset gate modulates the contribution of the old memory M when generating the candidate memory. When the model determines that certain old information is outdated or irrelevant, the reset gate suppresses the corresponding dimensions, effectively "forgetting" them in the candidate memory computation. The candidate memory is calculated as:

$$\tilde{M} = \tanh(W_c[g_r \cdot M; V]) \tag{26}$$

Finally, the update gate is used to control the integration of the candidate memory with the existing memory, enabling a smooth memory update:

$$M_{\text{new}} = g_u \odot \widetilde{M} + (1 - g_u) \odot M \tag{27}$$

In the EMM, memory reading and writing form a dynamic closed loop: In the reading phase, the mechanism extracts long-term interests related to the current sequence to enhance the Transformer output, while in the writing phase, it updates the memory to store new sequence features, thereby achieving an effective fusion of short-term behaviors and long-term interests.

4.4 Multi-Task Training Strategy

To optimize the sequential recommendation model, a multi-task training strategy is implemented by jointly optimizing the contrastive learning task, the generation task, the discrimination task, and the next-item prediction (NIP) task. The overall loss function is defined as follows:

$$L = L_{\text{rec}} + \lambda_3 L_{\text{cl}} + \lambda_4 L_{\text{gen}} + \lambda_5 L_{\text{disc}} \tag{28}$$

By appropriately adjusting the weight coefficients λ_3, λ_4, and λ_5, the contribution of each task to the overall loss can be flexibly controlled. Ultimately, this multi-task learning strategy effectively facilitates the coordinated optimization of the model across multiple objectives.

5 Experiments

To evaluate the proposed EASNRec framework, extensive experiments are conducted on public datasets to address the following research questions:

- **RQ1:** Does EASNRec generate better recommendation results compared to existing baseline methods?
- **RQ2:** How does the dynamic random augmentation strategy affect the model's generalization ability compared to existing baselines?
- **RQ3:** What are the contributions of different components in EASNRec to the recommendation performance?
- **RQ4:** What is the impact of various Siamese Network architectures?
- **RQ5:** How do key hyperparameters in EASNRec affect the performance?

5.1 Experimental Settings

5.1.1 Datasets

Experiments are conducted on three widely used benchmark datasets.

- **ML-1M:** A widely used dataset from MovieLens containing 1 million rating records, each with precise timestamps and ratings on a scale of 1 to 5.
- **Yelp:** A classic business recommendation dataset covering user consumption behaviors and detailed reviews from 2018 to the present.
- **Sports:** Sourced from Amazon review data, this dataset records user purchase sequences for sports-related products along with corresponding product reviews.

For data preprocessing, we follow the methodologies in [12] to standardize all datasets by removing users and items that appear less than 5 times. Table 1 presents the statistical details of the preprocessed datasets.

Table 1 Statistics of the three processed datasets

Dataset	Sports	Yelp	ML-1M
#Users	35,598	30,499	6,041
#Items	18,357	20,068	3,471
#Avg. Actions/User	8.3	10.4	165.5
#Avg. Actions/Item	16.1	15.8	292.6
#Actions	296,337	317,182	999,611
Sparsity	99.95%	99.95%	95.15%

5.1.2 Evaluation Metrics

To comprehensively assess all models, the widely used metrics Recall@K and NDCG@K are employed to evaluate model performance in sequential recommendation. For the purpose of model evaluation, we adopt the leave-one-out splitting strategy, where the last two interactions in each user's interaction sequence are used as the validation set and test set, respectively, while the remaining interactions are used as the training set.

5.1.3 Baseline Models

To validate the effectiveness of our proposed method, we compare it against the following representative baseline models:

- **BPR** [6]: An implicit feedback recommendation model based on matrix factorization, trained by optimizing the ranking differences between preferred and non-preferred items for users.
- **GRU4Rec** [8]: A session-based recommendation model utilizing GRU and employing novel loss functions and efficient sampling strategies.
- **Caser** [7]: A sequential recommendation model based on CNN that captures high-order Markov Chain patterns to model user behavior.
- **BERT4Rec** [12]: A sequential recommendation model based on a bidirectional Transformer architecture, which leverages self-attention mechanisms for dynamic sequence modeling.
- **SASRec** [11]: A self-attention model based on a unidirectional Transformer, focusing on capturing the evolution of users' temporal preferences.
- **ELECRec** [14]: A sequential recommendation model based on an adversarial training framework where the sequential recommender is trained as a discriminator.
- **S^3-Rec** [29]: A self-supervised sequential recommendation model that leverages contrastive learning and multi-task data augmentation to model implicit dependencies in user behavior sequences.
- **CL4SRec** [30]: A sequential recommendation model based on contrastive learning that employs sequence data augmentation strategies.
- **CoSeRec** [31]: A contrastive self-supervised recommendation model based on item co-occurrence analysis for constructing high-quality positive sample pairs based on semantic relevance.
- **ICLRec** [32]: A sequential recommendation method based on latent intent modeling, which can effectively capture users' multi-dimensional interest representations through unsupervised learning.
- **ECL-SR** [20]:A sequential recommendation method based on equivariant contrastive learning that enhances model robustness by enforcing equivariance in augmentation strategies.
- **MSSR** [33]: A multi-behavior sequential attention model based on auxiliary information, which integrates heterogeneous interaction information to achieve precise preference modeling.
- **xLSTM-LSR** [34]: A lifelong sequential recommendation model that captures long- and short-term user preferences with a dual-stream xLSTM architecture.

5.1.4 Implementation Details

To ensure a fair comparison, all baseline models and the EASNRec model are implemented and evaluated using the open-source recommendation framework RecBole [35], with training conducted under the same Adam optimizer. The number of layers and attention heads in EASNRec and attention-based baseline models are set to 2, and the latent dimension for all models is fixed at 256. The learning rate is set to 0.0001, and the batch size is set to 128. The maximum sequence length is set to 200 for ML-1M and 50 for both Yelp and Sports. The similarity loss weights for the generator and the discriminator are fine-tuned within the range of $\{0.02, 0.03, 0.04, 0.05, 0.06, 0.07\}$. The standard deviation σ of the Gaussian noise introduced in the UAMM is set to 0.01. The temperature τ in the contrastive loss is set to 0.05, and the masking ratio γ in the generator is set to 0.2 for Sports and ML-1M, and 0.6 for Yelp. The model is implemented in Python 3.7 and trained on an NVIDIA Tesla P100 GPU.

5.2 Overall Performance Comparison (RQ1)

Table 2 presents the overall performance of EASNRec compared to other baselines across all datasets. From the table, it can be observed that:

(1) Superiority of the EASNRec Model: The experimental results show that the EASNRec model exhibits significant performance advantages across multiple benchmark datasets. Specifically, compared with the second-best model, EASNRec improves the Recall@10 metric by 10.2% and 3.4% on the Yelp and ML-1M datasets, respectively; it also improves the Recall@20 metric by 5.1% on the Sports dataset. These performance improvements are attributable to the core innovations described below. First, by introducing a Siamese network architecture and similarity loss functions into the GAN framework, the model innovatively addresses the problem of "hard-boundary" decision bias caused by the binary classification of traditional discriminators, enabling collaborative optimization between the generator and the discriminator. Additionally, the UAMM is applied to the encoder, generator, and discriminator, thus effectively suppressing high-uncertainty features and noise interference while capturing long-term dependency patterns.

(2) Fundamental Recommendation Models: BPR, as a fundamental recommendation approach, exhibits significantly lower performance compared to other sequential recommendation models. This suggests that relying solely on simple scoring or ranking mechanisms is insufficient to fully exploit user behavior sequences, further highlighting the importance of sequence modeling.

(3) Classical Sequential Recommendation Models: SASRec consistently outperforms GRU4Rec and Caser, demonstrating the superiority of the self-attention mechanism in modeling complex behavioral patterns. While ELECRec matches SASRec 's baseline performance, fine-tuned SASRec remains dominant, indicating that direct discriminator architectures may not optimally address NIP tasks. BERT4Rec slightly underperforms SASRec, which might be attributed to its "cloze-style" pre-training strategy resulting in insufficient utilization of sequential information, thereby limiting its effectiveness in NIP tasks.

(4) Contrastive Learning-Based Models: S^3-Rec leverages self-supervised learning to provide auxiliary training signals. CoSeRec introduces collaborative self-supervised signals by jointly modeling sequence-level and item-level contrastive objectives, enabling the model to capture both global sequential patterns and local item transitions more effectively. CL4SRec constructs contrastive views through data augmentation strategies. ICLRec adopts learnable intent prototypes as positive samples for sequences, achieving outstanding performance in invariant contrastive learning. ECL-SR incorporates equivariant contrastive learning and effectively utilizes both mild and invasive augmentations to enhance user behavior representation. EASNRec further proposes a dynamic random augmentation strategy, demonstrating more adaptive performance across diverse scenarios.

(5) Auxiliary Information-Based Models: As shown in Table 2, EASNRec outperforms the state-of-the-art MSSR model. Although MSSR integrates multi-sequence attention and side information to enhance recommendation performance, its ability to resist interference from uncertain and noisy data is insufficient. In contrast, EASNRec comprises a UAA-Attention mechanism, which incorporates penalty correction terms and adversarial Gaussian noise into attention computation. This enables the model to dynamically suppress the interference of high-uncertainty and noisy features, thereby significantly improving recommendation accuracy.

(6)Lifelong Long-Sequence Models: xLSTM-LSR is a representative model for lifelong sequential recommendation. However, EASNRec consistently outperforms xLSTM-LSR across all datasets and evaluation metrics. This advantage is mainly attributed to the UAA-Attention and EMM to explicitly handle uncertainty and noise in long behavior sequences, whereas xLSTM-LSR lacks dedicated mechanisms to suppress high-uncertainty or noisy interactions.

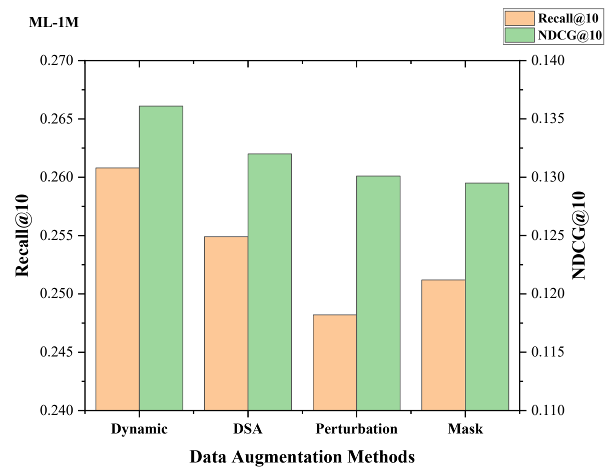

Fig. 6 Performance Comparison of Different Data Augmentation Methods.

5.3 Impact of Dynamic Random Augmentation (RQ2)

This study adopts a dynamic random augmentation strategy where one augmentation method—{Dropout, Perturbation, Mask}—is randomly selected with equal probability for each input sequence in every training batch and independently injected into its latent feature representation. Compared to baseline methods such as CL4SRec and CoSeRec, EASNRec applies augmentation to the continuous hidden representations output by the sequence encoder, achieving data augmentation without altering the original sequences, whereas baselines perform discrete operations—such as cropping, reordering, and masking—directly on the item sequences.

To further verify the effectiveness of our proposed augmentation strategy, we designed four contrastive variants by replacing only the augmentation module while keeping the core architecture of EASNRec unchanged, with the experimental results shown in Fig. 6. In this setup, Dynamic represents the proposed method, while **D**iscrete **S**equence **A**ugmentation (DSA) refers to the CL4SRec approach, where combinatorial transformations are applied in parallel to discrete item sequences to generate two augmented views per round for dual-view InfoNCE contrastive loss. The Mask and Perturbation variants consistently apply their respective augmentation to the hidden representations, with all other settings identical to ours. Experimental results demonstrate that our method significantly outperforms all variants in terms of Recall@10 and NDCG@10. This performance gap indicates that sequence augmentation on discrete items alters the input structure, which tends to destroy the temporal causal dependencies of user behavior and introduce noise. The lowest performance observed in single-representation augmentation further confirms the insufficient expressive power of a single augmentation perspective.

5.4 Ablation Study (RQ3)

To verify the effectiveness of each component in EASNRec, ablation experiments are conducted on three datasets. The variants of EASNRec are as follows: (1) w/o Siamese: The Siamese network is removed while retaining the fundamental adversarial training framework. (2) w/o UAA-Attention: The UAA-Attention mechanism is replaced with standard scaled dot-product attention.(3) w/o EMM: The EMM module is removed.(4) w/o UAMM: The UAA-Attention mechanism and the EMM module are removed while retaining only the standard Transformer.

The experimental results in Table 3 show that: (1) The full version of EASNRec outperforms its variants on all three datasets, demonstrating the importance of each component. (2) Removing the Siamese network (w/o Siamese) leads to a significant decline in model performance, particularly on the ML-1M dataset, highlighting its critical role in modeling similarity within dense behavioral sequences. (3) When the UAA-Attention mechanism (w/o UAA-Attention) is removed, the Recall metric decreases across all datasets, indicating that the UAA-Attention mechanism can improve recommendation accuracy by dynamically adjusting attention weights to suppress the interference of uncertain and noisy data. (4) The removal of the EMM module (w/o EMM) results in performance degradation, with the effect being more distinct in long-sequence datasets such as ML-1M, indicating its effectiveness in capturing long-term dependencies. (5) Notably, when the UAMM is removed (w/o UAMM), the model performs worse than when only the UAA-Attention or EMM is removed, further confirming the synergistic effect of the UAA-Attention and EMM modules. In summary, the ablation experiments above clearly demonstrate the effectiveness of the collaborative mechanism among multiple components in the model architecture.

5.5 Effect Analysis of Different Siamese Network Architectures (RQ4)

To comprehensively evaluate the impact of feature mapping architectures on model performance, we investigate several Siamese projection structures, including a two-layer fully connected network (FC), CNN, Transformer, and long short-term memory network (LSTM). Systematic comparisons are conducted on three benchmark datasets, and the experimental results are reported in Table 4. Overall, the FC-based projection consistently achieves the best performance across all datasets, with this advantage being particularly pronounced on the large-scale ML-1M dataset. In the sparser Yelp and Sports scenarios, although the absolute performance gaps become smaller, the FC structure still maintains a stable lead. By contrast, LSTM generally outperforms Transformer and CNN, yet remains inferior to the FC-based design.

Further analysis reveals that more complex architectures do not yield additional benefits in this task. The fundamental reason is that the SAS encoder has already assumed the responsibility of capturing temporal dependencies, and the resulting representations inherently contain high-order sequential semantics. The primary role of the Siamese network is therefore not to re-model sequential dependencies, but to project the extracted sequence semantics into a low-dimensional space that is more suitable for feature similarity measurement. Although CNNs, Transformers, and LSTMs are powerful due to their local receptive fields, self-attention mechanisms, and recurrent structures, respectively, they introduce strong inductive biases. Incorporating such biases at the projection stage often leads to representation redundancy and may even cause performance degradation. In contrast, FC layers perform feature mapping through direct global transformations, which aligns more naturally with the design objective of the projection module.

Table 2 Overall performance. The best results are highlighted in bold, while the second-best scores are underlined.

Datasets	Methods	NDCG		Recall	
		@10	**@20**	**@10**	**@20**
ML-1M	BPR(2012)	0.0115	0.0188	0.0260	0.0553
	GRU4Rec(2016)	0.0886	0.1123	0.1781	0.2725
	Caser(2018)	0.0624	0.0992	0.1579	0.2515
	SASRec(2018)	0.1074	0.1346	0.2145	0.3228
	BERT4Rec(2019)	0.0584	0.0852	0.1275	0.2370
	S^3-Rec(2020)	0.0961	0.1237	0.1992	0.3086
	CoSeRec(2021)	0.1032	0.1285	0.2214	0.3210
	ELECRec(2022)	0.0902	0.1154	0.1917	0.2917
	CL4SRec(2022)	0.1120	0.1408	0.2200	0.3341
	ICLRec(2022)	0.1140	0.1431	0.2291	0.3445
	ECL-SR(2023)	0.1270	0.1547	0.2462	0.3563
	MSSR(2024)	0.1333	0.1586	0.2465	0.3470
	xLSTM-LSR(2025)	0.1340	0.1589	0.2523	0.3483
	EASNRec(ours)	**0.1361**	**0.1646**	**0.2608**	**0.3748**
Yelp	BPR(2012)	0.0324	0.0384	0.0589	0.0830
	GRU4Rec(2016)	0.0206	0.0271	0.0418	0.0679
	Caser(2018)	0.0197	0.0255	0.0380	0.0608
	SASRec(2018)	0.0405	0.0481	0.0667	0.0969
	BERT4Rec(2019)	0.0327	0.0385	0.0524	0.0756
	S^3-Rec(2020)	0.0392	0.0465	0.0637	0.0929
	CoSeRec(2021)	0.0379	0.0454	0.0648	0.0947
	ELECRec(2022)	0.0406	0.0482	0.0677	0.0980
	CL4SRec(2022)	0.0387	0.0462	0.0640	0.0937
	ICLRec(2022)	0.0422	0.0499	0.0701	0.1005
	ECL-SR(2023)	0.0395	0.0475	0.0709	0.1029
	MSSR(2024)	0.0427	0.0513	0.0715	0.1063
	xLSTM-LSR(2025)	0.0439	0.0501	0.0722	0.0996
	EASNRec(ours)	**0.0459**	**0.0542**	**0.0796**	**0.1130**
Sports	BPR(2012)	0.0144	0.0188	0.0302	0.0480
	GRU4Rec(2016)	0.0195	0.0251	0.0386	0.0609
	Caser(2018)	0.0118	0.0153	0.0227	0.0364
	SASRec(2018)	0.0226	0.0294	0.0511	0.0781
	BERT4Rec(2019)	0.0135	0.0173	0.0295	0.0465
	S^3-Rec(2020)	0.0236	0.0306	0.0509	0.0787
	CoSeRec(2021)	0.0223	0.0299	0.0514	0.0778
	ELECRec(2022)	0.0234	0.0301	0.0480	0.0758
	CL4SRec(2022)	0.0248	0.0320	0.0497	0.0782
	ICLRec(2022)	0.0236	0.0298	0.0527	0.0773
	ECL-SR(2023)	0.0230	0.0293	0.0457	0.0706
	MSSR(2024)	0.0262	0.0328	0.0559	0.0820
	xLSTM-LSR(2025)	0.0239	0.0306	0.0460	0.0725
	EASNRec(ours)	**0.0262**	**0.0334**	**0.0574**	**0.0862**

Table 3 Ablation study of the EASNRec

Dataset	**ML-1M**		**Yelp**		**Sports**	
Metric	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
EASNRec	**0.3748**	**0.1646**	**0.1130**	**0.0542**	**0.0862**	**0.0334**
w/o Siamese	0.3629	0.1572	0.1084	0.0533	0.0844	0.0326
w/o UAA-Attention	0.3662	0.1597	0.1075	0.0531	0.0852	0.0325
w/o EMM	0.3662	0.1571	0.1084	0.0527	0.0853	0.0330
w/o UAMM	0.3588	0.1535	0.1053	0.0522	0.0846	0.0322

Moreover, the substantial number of parameters introduced by complex architectures significantly increases gradient variance in high-dimensional search spaces during adversarial and contrastive learning, resulting in less stable optimization

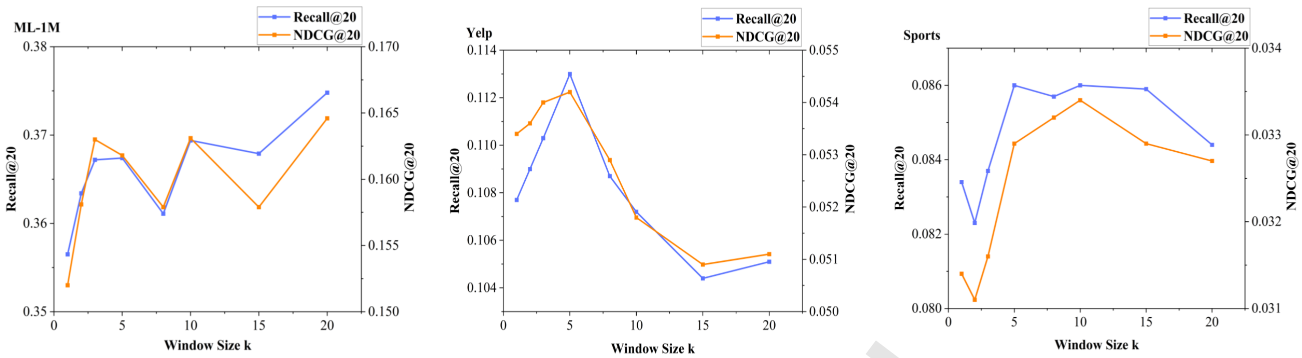

Fig. 7 Performance comparison of different window sizes on three datasets (NDCG@20 and Recall@20)

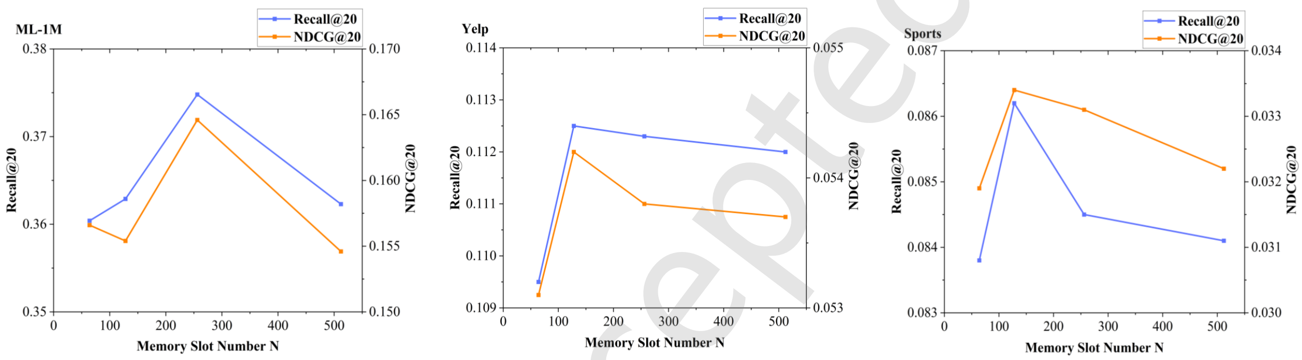

Fig. 8 Performance comparison with different numbers of memory slots on three datasets (NDCG@20 and Recall@20)

Table 4 Performance comparison of different Siamese Networks

Dataset	Metric	FC	CNN	Transformer	LSTM
ML-1M	Recall@20	**0.3748**	0.3608	0.3632	0.3675
	NDCG@20	**0.1646**	0.1554	0.1578	0.1595
Yelp	Recall@20	**0.1130**	0.1111	0.1104	0.1113
	NDCG@20	**0.0542**	0.0535	0.0536	0.0539
Sports	Recall@20	**0.0862**	0.0843	0.0852	0.0855
	NDCG@20	**0.0334**	0.0325	0.0328	0.0331

and unnecessary computational overhead.

5.6 Sensitivity Analysis (RQ5)

This section analyzes the impact of two key hyperparameters on EASNRec: window size k and memory slot number N. Recall@20 and NDCG@20 are measured across the three datasets while changing only one hyperparameter at a time and keeping the others unchanged.

(1)Effect of window size k(Fig. 7). The window size k determines the aggregation scope of user behavior sequences. We conducted experiments with $k \in [1, 20]$ to evaluate its impact. The results show that on the ML-1M dataset, as k increases from 1 to 20, the model performance continuously improves, indicating that user sequences in this dataset are relatively long and expanding the window allows the model to capture more cross-stage interest patterns. For the sparse datasets Yelp and Sports, the performance peaks within the range $k \approx$ 5–10. Beyond this range, recommendation effectiveness starts to decline or plateau, suggesting that in sparse scenarios, an excessively large aggregation scope may introduce noise and interfere with the model's focus on meaningful behaviors. Moreover, larger windows incur higher computational costs, making it challenging to model long-term dependencies in sparse scenarios while simultaneously increasing noise and computation overhead, thereby weakening overall performance.

(2)Effect of the number of memory slots N(Fig. 8). The number of memory slots N determines the capacity of the external memory module to store historical behavior patterns. We conducted experiments with $N \in \{64, 128, 256, 512\}$. The results show that when N is small (e.g., 64), the external memory struggles to cover cross-stage interests and periodic preferences in long sequences. As N increases to 128 and 256, the model can better retrieve long-term dependency information, resulting in a significant performance improvement. However, when N further increases to 512, performance declines. This is mainly because an excessive number of memory slots leads to a more sparse attention distribution, reducing retrieval efficiency, while the larger memory matrix also introduces additional computational overhead, slowing down the overall model.

6 Conclusion

In this paper, we propose the EASNRec to address the limitations of the "hard-boundary" classification strategy in traditional GANs and the shortcomings of Transformer-based models. Specifically, by introducing the Siamese network, we refine the process of similarity mapping between real and generated sequences in the feature space and optimize the learning process of the generator and discriminator based on similarity loss, thereby generating recommendation sequences that better align with user preferences. Furthermore, the design of the UAMM effectively suppresses high-uncertainty sequence data, reduces noise interference, and facilitates long-term dependency modeling. Extensive experiments are conducted on three public datasets to verify the effectiveness of EASNRec against 13 baseline models. With its lightweight model architecture and robust feature modeling capability, EASNRec can operate efficiently on edge nodes with limited computational resources and dynamically changing environments, enabling high-precision sequential recommendation.

In the future, we will focus on incorporating multimodal auxiliary information, such as knowledge graphs or item attributes, into EASNRec. This is expected to help mitigate common issues in sequential recommendation, such as data sparsity and cold-start problems, and further enhance the model's robustness in complex interaction scenarios.

Acknowledgment

This work is supported by the Beijing Natural Science Foundation (Grant No. 9242003), the Natural Science Foundation of Chongqing, China (Grant No. CSTB2023NSCQ-MSX0391), the National Natural Science Foundation of China (Grant No. 62471493), the Natural Science Foundation of Shandong Province (Grant Nos. ZR2023LZH017 and ZR2024MF066), the Key Laboratory of Public Opinion Governance and Computational Communication (Grant No. YQKFYB202501), the Research Project on the Development of Social Sciences in Hebei Province in 2024 (Grant No. 202403150), and the National Statistical Science Research Project (Grant No. 2025LZ020).

Declaration of competing interest

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

Reference

[1] C. Luo, J. Zhang, J. Guo, Y. Hong, Z. Chen, and S. Gu, Energy efficiency maximization in riss-assisted uavs-based edge computing network using deep reinforcement learning, *Big Data Mining and Analytics*, vol. 7, no. 4, pp. 1065–1083, 2024.

[2] W. Lin, M. Zhu, X. Zhou, R. Zhang, X. Zhao, S. Shen, and L. Sun, A deep neural collaborative filtering based service recommendation method with multi-source data for smart cloud-edge collaboration applications, *Tsinghua Science and Technology*, vol. 29, no. 3, pp. 897–910, 2024.

[3] M. Zhang, X. Zhang, W. Pedrycz, S. Wang, and G. Wu, Learning fine-grained user preference for personalized recommendation, *Tsinghua Science and Technology*, vol. 30, no. 6, pp. 2544–2556, 2025.

[4] H. Kou and J. Xu, Kan-gnn: Kolmogorov-arnold network inspired graph neural networks for personalized api recommendation, *Tsinghua Science and Technology*, 2025.

[5] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010, pp. 811–820.

[6] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 452–461.

[7] J. Tang and K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 565–573.

[8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, Session-based recommendations with recurrent neural networks, in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 21–30.

[9] N. Hua, H. Dong, R. Hu, and et al, Multi-time-scale with clockwork recurrent neural network modeling for sequential recommendation, *The Journal of Supercomputing*, vol. 81, no. 2, pp. 1–31, 2025.

[10] Y. Dang, E. Yang, G. Guo, and et al, Uniform sequence better: Time interval aware data augmentation for sequential recommendation, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4225–4232.

[11] W.-C. Kang and J. McAuley, Self-attentive sequential recommendation, in *Proceedings of the 18th IEEE International Conference on Data Mining*. IEEE, 2018, pp. 197–206.

[12] F. Sun, J. Liu, J. Wu, and et al, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2019, pp. 1441–1450.

[13] S. Dai, A. Gao, Z. Li, and Y. Du, Rotary position embedding-

based transformer hawkes process for event-type big data, *Big Data Mining and Analytics*, vol. 9, no. 1, pp. 23–38, 2026.

[14] Y. Chen, J. Li, and C. Xiong, Elecrec: Training sequential recommenders as discriminators, in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2022, pp. 2550–2554.

[15] J. Huang, K. Li, J. Jia, and X. Wang, Single image super-resolution through image pixel information clustering and generative adversarial network, *Big Data Mining and Analytics*, vol. 8, no. 5, pp. 1044–1059, 2025.

[16] J. Zhao, C. Rong, X. Dang, and H. Sun, Qar data imputation using generative adversarial network with self-attention mechanism, *Big Data Mining and Analytics*, vol. 7, no. 1, pp. 12–28, 2023.

[17] D. Qiu, Research on recurrent neural network recommendation algorithm based on time series, *Applied and Computational Engineering*, vol. 87, no. 1, pp. 72–79, 2024.

[18] Y. Liu, T. Song, Y. Yang, D. Zhang, J. Xie, L. Qi, and S. Shen, Optimizing sequence-based poi recommendations: From sequence adjustment to transformer-xl integration, *Tsinghua Science and Technology*, 2025.

[19] J. Chen, G. Zou, P. Zhou, Y. Wu, Z. Chen, H. Su, H. Wang, and Z. Gong, Sparse enhanced network: An adversarial generation method for robust augmentation in sequential recommendation, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8. AAAI Press, 2024, pp. 8283–8291.

[20] P. Zhou, J. Gao, Y. Xie, Q. Ye, Y. Hua, J. Kim, S. Wang, and S. Kim, Equivariant contrastive learning for sequential recommendation, in *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, 2023, pp. 129–140.

[21] C.-Y. Chen, H.-C. Huang, J.-C. Jheng, and R.-C. Hwang, A siamese-network-based facial recognition system, *Sensors and Materials*, vol. 36, no. 6, pp. 2425–2438, 2024.

[22] Z. Liu, H. Huang, H. Dong, and et al, Iou-guided siamese network with high-confidence template fusion for visual tracking, *Neurocomputing*, vol. 614, p. 128774, 2025.

[23] X. Hu, J. He, X. Guo, and et al, Siamese networks for few-shot coffee bean defect detection, *LWT*, vol. 216, p. 118631, 2025.

[24] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, Signature verification using a "siamese" time delay neural network, in *Advances in Neural Information Processing Systems*, vol. 6. Morgan-Kaufmann, 1993, pp. 737–744.

[25] Z. Liu, H. Huang, H. Dong, and J. Chen, Dsiam-cnk: A cbam- and kcf-enabled deep siamese region proposal network for human tracking in dynamic and occluded scenes, *IEEE Access*, vol. 12, pp. 72 430–72 445, 2024.

[26] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, Few-shot and meta-learning methods for image understanding: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5128–5146, 2022.

[27] C. Han, Y. Wang, Y. Fu, X. Li, M. Qiu, M. Gao, and A. Zhou, Meta-learning siamese network for few-shot text classification, in *International Conference on Database Systems for Advanced Applications*. Springer, 2023, pp. 737–752.

[28] N. Serrano and A. Bellogín, Siamese neural networks in recommendation, *Neural Computing and Applications*, vol. 35, no. 19, pp. 13 941–13 953, 2023.

[29] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization, in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2020, pp. 339–348.

[30] X. Xie, F. Zhang, Z. Sun, and et al, Contrastive learning for sequential recommendation, in *Proceedings of the 38th IEEE International Conference on Data Mining (ICDE)*. IEEE, 2022, pp. 1259–1273.

[31] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. Crossane, C. Xiong, and et al, Contrastive self-supervised sequential recommendation with robust augmentation, *arXiv preprint arXiv:2108.06479*, 2021.

[32] Y. Chen, Z. Liu, J. Li, and et al, Intent contrastive learning for sequential recommendation, in *Proceedings of the Web Conference 2022 (WWW)*. ACM, 2022, pp. 2172–2182.

[33] X. Lin, J. Luo, J. Pan, W. Pan, Z. Ming, X. Liu, S. Huang, and J. Jiang, Multi-sequence attentive user representation learning for side-information integrated sequential recommendation, in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 414–423.

[34] W. Sun, R. Xie, J. Zhang, X. Zhao, Z. Kang, and J. Wen, Lifelong long sequence recommendation method based on matrix and hybrid mechanism dual-stream long short-term memory network, (in chinese), *Chinese Journal of Computers*, vol. 48, no. 12, pp. 2789–2808, 2025.

[35] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Liu, H. Jiang, Y. Chen, H. Pang, C. Yuan, and J.-R. Wen, Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms, in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM, 2021, pp. 4653–4664.

Author biography

Junsan Zhang received the Ph.D. in Computer Science and Technology from Beijing Jiaotong University, China, in 2013. He is a currently Professor and Chair of the Software Engineering Department at China University of Petroleum (East China). His research interests include image understanding, recommender systems, intelligent software engineering, and industrial software. In recent years, he has undertaken more than 30 projects funded by the National Natural Science Foundation of China, Shandong Provincial Natural Science Foundation, as well as enterprises and public institutions. He has published more than 40 SCI- and EI-indexed papers in important domestic and international academic journals and conferences, and obtained 4 authorized national invention patents.

Fengmei Ding is a Master's student in the College of Computer Science and Technology, China University of Petroleum (East China), and a CCF Student Member (A16627G). Her research interests focus on recommender systems, and she has participated in research and academic activities related to personalized recommendation and intelligent systems.

Jia Luo received a bachelor's degree from Shanghai University, China in 2011, a master's degree from Waseda University, Japan in 2015, and a Ph. D. degree from Université de Toulouse, France in 2019. Currently, she is an Associate Professor at the Interdisciplinary Faculty of Science and Engineering, Shimane University, Japan. Her research interests include deep learning, social media analysis, and parallel learning.

Yunpeng Si is a Master's student in the College of Computer Science and Technology, China University of Petroleum (East China). His research interests focus on the fairness of recommender systems. He also actively participates in academic exchanges related to data mining and intelligent information processing, focusing on the latest research progress in the field of fair recommendation.

Lei Shi was born in 1986. He received the Ph.D. degree in School of Computer Science from Beijing University of Posts and Telecommunications, Beijing, China, in 2019. He is currently an associate professor in the State Key Laboratory of Media Convergence and Communication, Communication University of China. His research interests include intelligent information processing, data mining and cross-media search.

Peiying Zhang is currently a Professor with the College of Computer Science and Technology, China University of Petroleum (East China). He received his Ph.D. in the School of Information and Communication Engineering at University of Beijing University of Posts and Telecommunications in 2019. He has published multiple high quality papers since 2016, such as IEEE T-IFS, IEEE TSC, IEEE T-ITS, IEEE TMC , etc. He served as the Technical Program Committee of IJCAI'25, AAAI' 25, AAAI'24, AAAI'23. He is the editorial board of Scientific Reports, Drones, CMC-Computers, Materials & Continua, Mobile Information Systems, International Journal of Computational Intelligence Systems and Artificial Intelligence and Applications (AIA). His research interests include future internet architecture, network virtualization, and artificial intelligence for networking.