Hans Journal of Data Mining 数据挖掘, 2025, 15(2), 159-175
Published Online April 2025 in Hans. https://www.hanspub.org/journal/hjdm
https://doi.org/10.12677/hjdm.2025.152014

Hans汉斯

时频融合和特征交叉融合的序列推荐算法

张金文，李征宇，孙　平

沈阳建筑大学计算机科学与工程学院，辽宁 沈阳

收稿日期：2025年3月7日；录用日期：2025年3月21日；发布日期：2025年4月3日

摘　要

为了有效融合项目ID嵌入和文本嵌入，提出一种时频融合和特征交叉融合的序列推荐算法(Time-frequency fusion and feature cross-fusion sequential recommendation algorithm, TFFCRec)。使用RoBERTa对项目文本进行编码，多样化混合专家调制方法获得的是易于区分的文本表示，将项目ID嵌入和文本嵌入通过结合快速傅里叶变换(FFT)和短时傅里叶变换(STFT)，提取用户的全局频域特征和局部时频特征。这样的方法使得算法既能考虑用户的长期兴趣偏好，又能捕捉用户的短期兴趣变化。此外，我们引入了特征交叉融合，并通过优化的Mamba-like的线性注意力(OMLLA)来捕获特征之间更深层次的非线性关系，提取更深层次的特征表示。我们设计了一个融合网络，自适应地学习不同嵌入表示的权重，将FFT、STFT和OMLLA得到的特征向量进行加权融合，通过SASRec来进行序列推荐。在Instant video、Beauty、Digital Music、Tools and Home improvement数据集上进行实验，本文方法较基准方法在Recall@10指标上分别提升了6.3%、13.2%、3.7%、6.5%。

关键词

快速傅里叶变换，短时傅里叶变换，时频融合，特征交叉融合

Time-Frequency Fusion and Feature Cross-Fusion Sequence Recommendation Algorithm

Jinwen Zhang, Zhengyu Li, Ping Sun

School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang Liaoning

Received: Mar. 7th, 2025; accepted: Mar. 21st, 2025; published: Apr. 3rd, 2025

Abstract

To effectively integrate project ID embeddings and text embeddings, we propose a sequence recom-

文章引用: 张金文, 李征宇, 孙平. 时频融合和特征交叉融合的序列推荐算法[J]. 数据挖掘, 2025, 15(2): 159-175.
DOI: 10.12677/hjdm.2025.152014

mendation algorithm called Time-frequency Fusion and Feature Cross-fusion Sequential Recommendation Algorithm (TFFCRec). RoBERTa is used to encode the project text, and a diversity mixture expert modulation method is applied to obtain distinguishable text representations. Project ID embeddings and text embeddings are combined using Fast Fourier Transform (FFT) and Short-Time Fourier Transform (STFT), extracting the user's global frequency-domain features and local time-frequency features. This approach enables the algorithm to capture both the user's long-term interest preferences and short-term interest variations. In addition, we introduce feature cross-fusion and use the optimized Mamba-like Linear Attention (OMLLA) to capture deeper non-linear relationships between features and extract more profound feature representations. We design a fusion network that adaptively learns the weights of different embedding representations and performs weighted fusion of the feature vectors obtained from FFT, STFT, and OMLLA. These fused features are then passed into SASRec for sequence recommendation. Experiments are carried out on the Instant Video, Beauty, Digital Music, and Tools and Home Improvement datasets. Compared with the benchmark methods, the proposed method in this paper has improved the Recall@10 metric by 6.3%, 13.2%, 3.7%, and 6.5% respectively.

Keywords

Fast Fourier Transform, Short-Time Fourier Transform, Time-Frequency Fusion, Feature Cross-Fusion

1. 引言

顺序推荐系统作为推荐系统的一个关键分支，主要通过分析用户的历史行为序列来预测未来的兴趣和行为，广泛应用于亚马逊、淘宝等在线平台。与传统的协同过滤和基于内容的推荐方法不同，顺序推荐更侧重于捕捉动态的用户行为模式，以适应用户兴趣的快速变化和多样化的消费习惯。不同用户在兴趣方面各不相同，同一用户也可能对多种物品感兴趣。因此，捕捉用户多样化兴趣的能力至关重要。研究者们开发了多种模型架构来表征用户的顺序模式，主要包括卷积神经网络(CNN)、循环神经网络(RNN)、Transformer。这些模型各具优势，能够从不同角度捕捉用户行为的特征。CNN 擅长识别局部特征，RNN 能够处理时间序列依赖，Transformer 则利用自注意力机制建模长距离依赖性，提升预测准确性。通过结合这些模型，推荐系统能够更全面地理解用户偏好，从而增强预测能力。现有的序列推荐模型[1]主要依赖于项目 ID 来建模用户行为，但这种方法难以充分利用丰富的上下文数据，限制了推荐效果。为了解决这一问题，研究者提出引入额外的辅助信息，特别是与项目相关的文本。这些文本提供了详细的特征信息，有助于捕捉项目之间的差异和用户的潜在兴趣，从而更全面地理解用户的多样化需求，进而在复杂场景中提升模型性能。基于文本的推荐算法通过分析物品的文本描述、用户评论和其它相关文本信息，能够深入挖掘物品特征和用户偏好。例如，利用自然语言处理技术从文本中提取主题、情感和关键特征，然后将这些信息融合到序列模型中，增强模型对物品特征和用户兴趣的理解。然而，现有的文本处理方法在推荐系统中仍面临挑战。近年来，研究者转向使用预训练模型如 RoBERTa [2]来提取项目的文本特征。尽管这类模型在自然语言处理任务中表现出色，但在处理推荐系统的特定任务时也存在局限性。由于 RoBERTa 预训练的目标是通用的语言理解，其学习到的语义表示未必能够充分捕捉与推荐相关的细粒度特征。根据上述的问题，TedRec [3]通过多专家调制的方法来增强文本的区分能力和

通过快速傅里叶变换将文本和 ID 嵌入转换到频域进行有效的融合，从而捕捉全局上下文信息。但是它也面临以下的局限性，使用多专家调制的时候仅使用线性变换对捕捉不同粒度的特征不够充分，在使用快速傅里叶变换时重点关注全局特征，对局部特征和动态变化特征的捕获不够充分，缺乏显式特征交互，对稀疏特征组合的捕捉能力不足。因此，如何有效融合项目 ID 嵌入和文本嵌入，已成为推荐系统领域的一个关键研究方向。

针对上述问题，本文通过对比学习不同推荐方法，提出了一种 TFFCRec 推荐方法，其主要贡献如下：

1) 我们通过引入一个多样化的混合专家(Diverse Mixture-of-Experts, DMoE)适配器，利用多样化的专家架构和位置调制，提高了文本嵌入的区分度。

2) 对于融合项目的 ID 嵌入和文本嵌入时，通过快速傅里叶变换(FFT)和短时傅里叶变换(STFT)，我们成功地捕获了全局特征和局部时频特征。此外，我们引入了特征交叉融合，并通过 OMLLA 来捕获特征之间更深层次的非线性关系，提取更深层次的特征表示。我们设计了一个融合网络，自适应地学习不同嵌入表示的权重，将 FFT、STFT 和 OMLLA 得到的特征向量进行加权融合。

2. 相关工作

文本嵌入在自然语言处理任务中起着至关重要的作用，它能够将文本数据转换为低维连续空间中的向量表示，为下游任务提供丰富的语义和语法特征支持。随着深度学习的发展，诸如 Word2Vec [4]、GloV [5]、BERT [6]、RoBERTa [2]、WhiteningBERT [7]等模型相继涌现，极大地提升了文本表示的质量。然而，这些模型在捕获复杂、多样的语言特征时仍存在一定的局限性，难以全面表征文本的多维度信息，导致文本表示的区分度较低。

早期的序列推荐研究通常基于马尔可夫链假设[8]。随着深度学习的发展，各种神经网络架构被引入以更好地捕捉序列模式，包括 RNN、CNN、Transformer 等。接着，许多基于深度学习的序列推荐模型被开发出来，例如 GRU4Rec [9]基于门控循环单元(GRU)建模用户行为序列的动态演化，捕捉长期兴趣依赖。Caser [10]通过水平卷积和垂直卷积联合建模用户行为序列的局部与全局特征。后来，自注意力网络在建模序列数据方面显示出巨大的潜力，并且开发了各种相关模型。例如，SASRec [1]基于 Transformer 编码器的单向自注意力机制，建模用户行为序列中的长期依赖关系。BERT4Rec [11]借鉴 BERT 的双向 Transformer，通过掩码语言模型(MLM)学习序列上下文表征。S3Rec [12]通过自监督学习(SSL)增强序列建模，解决数据稀疏性问题。GCSAN [13]图卷积网络(GCN)与自注意力的混合架构，联合学习序列局部结构与全局依赖。LightSANs [14]通过低秩分解与核化注意力降低计算复杂度。最近，最近将傅里叶变换引入到了序列推荐的任务中 FMLP-Rec [15]首次引入了一种滤波增强的 MLP 用于顺序推荐，该方法通过乘以一个全局滤波器来去除频域中的噪声。然而，全局滤波器倾向于给低频赋予更大的权重而相对低估高频。为了进一步探索傅里叶变换在顺序推荐中的应用，TedRec [3]通过快速傅里叶变换将文本嵌入和 ID 嵌入从时间域转换到频率域，从而更好地整合全局上下文信息。虽然 FMLP-Rec、TedRec 都使用快速傅里叶变换取得了不错的效果，但是它们也面临以下的局限性。在使用快速傅里叶变换时重点关注全局特征，对局部特征和动态变化特征的捕获不够充分，缺乏显式特征交互，对稀疏特征组合的捕捉能力不足。

3. 预备知识

3.1. 傅里叶变换

对于长度为 N 的信号 $x[n]$，其离散傅里叶变换(DFT)公式为：

$$X[k]=\sum_{j=0}^{N-1}x[n]\cdot e^{-j\frac{2\pi}{N}kn},k=0,1,\cdots,N-1 \tag{1}$$

其中 $X[k]$ 是频域信号序列， $x[n]$ 是时间域信号序列。

DFT 的逆变换用于将频域信号转换回时间域，恢复原始信号。对于一个长度为 N 的序列，其逆离散傅里叶变换(IDFT)的公式为：

$$x[n]=\frac{1}{N}\sum_{j=0}^{N-1}X[k]\cdot e^{j\frac{2\pi}{N}kn},n=0,1,\cdots,N-1 \tag{2}$$

3.2. 快速傅里叶变换

快速傅里叶变换(FFT)是一种计算 DFT 的高效算法。快速傅里叶变换(FFT)的公式为：

$$X[k]=\sum_{j=0}^{N-1}x[n]\cdot e^{-j\frac{2\pi}{N}kn},k=0,1,\cdots,N-1 \tag{3}$$

逆快速傅里叶变换(IFFT)的公式为：

$$x[n]=\frac{1}{N}\sum_{j=0}^{N-1}X[k]\cdot e^{j\frac{2\pi}{N}kn},n=0,1,\cdots,N-1 \tag{4}$$

3.3. 短时傅里叶变换

短时傅里叶变换(STFT)的公式为：

$$X(m,k)=\sum_{n=0}^{N-1}x[n+mR]\omega[n]e^{-j\frac{2\pi}{N}kn},k=0,1,2,\cdots,N-1 \tag{5}$$

其中 $x[n]$ 是原始信号， $\omega[n]$ 是窗口函数，N 是每帧的傅里叶变换长度，m 是帧的索引，R 是帧移。

逆短时傅里叶变换(ISTFT)的公式为：

$$x[n]=\frac{1}{C}\sum_{m}\omega[n-mR]\left(\frac{1}{N}\sum_{k=0}^{N-1}X(m,k)e^{j\frac{2\pi}{N}k(n-mR)}\right),k=0,1,2,\cdots,N-1 \tag{6}$$

其中 $x[n]$ 是重构后的时间域信号，C 是归一化常数，$\omega[n-mR]$ 是移位的窗口函数，N 是每帧的傅里叶变换长度。

4. 方法

本文提出的模型整体结构如图 1 所示。使用 RoBERTa 对项目文本进行编码和多样化混合专家调制方法获得的是易于区分的文本表示。将项目的 ID 嵌入和文本嵌入通过快速傅里叶变换(FFT)和短时傅里叶变换(STFT)，我们成功地捕获了全局频域特征和局部时频特征。此外，我们引入了特征交叉融合，并通过 OMLLA 来捕获特征之间更深层次的非线性关系，提取更深层次的特征表示。我们设计了一个融合网络，自适应地学习不同嵌入表示的权重，将 FFT、STFT 和 OMLLA 得到的特征向量进行加权融合。通过 SASRec 处理特征最终通过预测层输出推荐结果，预测下一个项目的可能性。

4.1. 文本数据处理

4.1.1. 可区分文本表示编码

我们使用 RoBERTa 对项目文本进行编码。对于项目 v 的相关文本 $\{w_1;w_2;w_3;\cdots,w_C\}$，我们在序列的起始位置插入一个[CLS]标记，并将扩展后的序列输入到 RoBERTa 中：

$$t_v = \mathrm{RoBERTa}\left(\left\{\left[\mathrm{CLS}: w_1; w_2; w_3; \cdots, w_C\right]\right\}\right) \tag{7}$$

其中 t_v 代表输入标记“CLS”的最后一个隐藏状态向量。“[;]”表示拼接操作。通过这种方式，每一个项目 v 都被编码为一个唯一的文本嵌入 t_v。

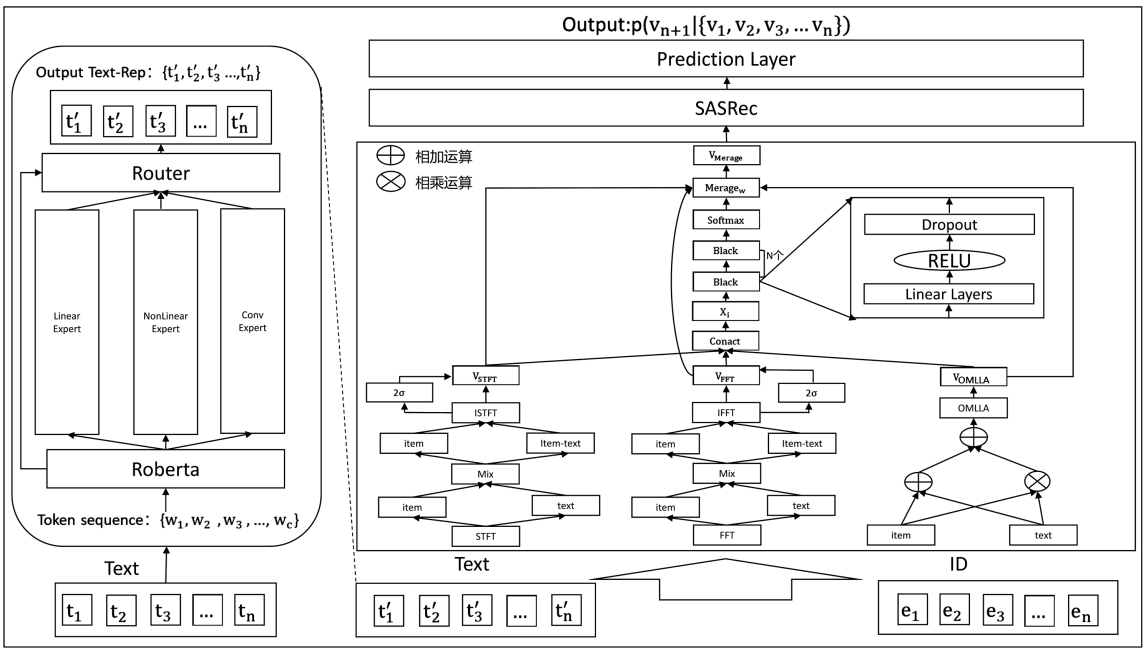

Figure 1. TFFCRec structure diagram
图 1. TFFCRec 结构图

4.1.2. 多样化专家调制

为了更好地捕获用户交互行为中的复杂模式和多样化的序列语义，我们在多样化的混合专家(DMoE)架构中引入了多样化专家的设计。不同于传统 MoE 架构[16]同质专家的方式，我们的模型集成了具有不同网络结构的专家模块，包括线性、非线性和卷积专家。这样的设计使模型能够从多角度、多层次地对输入文本序列进行建模，提升了模型对复杂序列的适应性和表达能力。具体来说，我们定义了一个包含 G 个专家的专家库，每个专家 E_k 的结构可以是以下三种类型之一：

1) 线性专家(Linear Expert)：适用于捕获输入与输出之间的线性关系。并且结构是一个无偏置的线性层：

$$E_{\text{linear}}(x) = W_{\text{linear}} \cdot x \tag{8}$$

2) 非线性专家(Nonlinear Expert)：用于捕获输入之间的非线性关系。由两层无偏置线性层和 ReLU 激活函数组成：

$$E_{\text{nonlinear}}(x) = W_{\text{linear2}}\mathrm{ReLU}\left(W_{\text{linear1}} \cdot x\right) \tag{9}$$

3) 卷积专家(Convolutional Expert)：在捕获输入序列中的局部模式和邻域特征。由两个无偏置一维卷积层和 ReLU 激活函数构成：

$$E_{\text{conv}}(x) = \mathrm{Conv1D}_2\left(\mathrm{ReLU}\left(\mathrm{Conv1D}_1(x)\right)\right) \tag{10}$$

每种专家类型的基础数量，将专家数量平均分为三类。剩余的专家数量，如果专家数量不能被 3 整除。如果有剩余的专家，就从三种类型中随机选择 n 个添加到专家列表。首先，专家数量平均分为三类，

计算出每种类型的基础数量。如果专家数量不能被 3 整除，会有一个余数。对于剩余的专家数量，从三种专家类型中随机选择余数个类型，并添加到专家列表中。在生成余数并添加到专家列表之前，使用固定种子(seed = 42)保证实验的可复现性。最后，随机打乱专家列表的顺序，以确保每次实验得到一致的结果。

通过为每个序列位置引入独立的偏置，模型可以学习到与位置相关的特征调整，增强不同位置表示的区分度。然后每个专家都会对其进行处理，得到相应的输出：

$$t_j = \left(t_j + q_j\right) \tag{11}$$

$$\text{ExpertOutput}_k = E_k\left(t_j\right) \tag{12}$$

q_j 相当于第 j 个位置的调制嵌入，所使用的调制嵌入与 Transformer 中使用的绝对位置嵌入相似。为了自适应地融合这些专家的输出，我们引入了一个带噪声的门控路由器来计算各个专家的权重。首先，计算每个专家的未归一化权重(logits)：

$$\text{logits} = t_j \cdot W^G \tag{13}$$

其中，W^G 是可学习的参数矩阵。为了增加模型的鲁棒性和探索能力，我们在训练过程中向 logits 添加可控的噪声：

$$\text{noisy\_logits} = \text{logits} + \epsilon \tag{14}$$

其中，ϵ 是基于输入计算的随机噪声，其标准差由另一个可学习的参数 W^{noise} 确定：

$$\epsilon = \sigma \cdot N\left(0,1\right) \tag{15}$$

$$\sigma = \text{Softplus}\left(t_j \cdot W^{\text{noise}}\right) + \delta \tag{16}$$

其中，σ 使噪声标准差，Softplus 是激活函数，δ 是一个小的常数，用于避免数值不稳定。然后，通过 Softmax 函数将 noisy_logits 转换为概率分布，得到各个专家的门控权重：

$$g = \text{Softmax}\left(\text{noisy\_logits}\right) \tag{17}$$

最终，我们对所有专家的输出按照门控权重进行加权求和，得到增强后的文本表示：

$$t'_j = \sum_{k=1}^{G} g_k \cdot \text{ExpertOutput}_k \tag{18}$$

通过这种多样化专家和带噪声门控的设计，模型能够捕获多种特征形式、自适应地融合专家信息和增加模型的鲁棒性。

4.2. 项目 ID 嵌入和文本嵌入的特征融合

4.2.1. 快速傅里叶变换特征融合

ID 嵌入 $E = \left\{e_1, e_2, \cdots, e_n\right\} \in R^{n*d}$ 和文本嵌入 $T = \left\{t'_1, t'_2, \cdots, t'_n\right\} \in R^{n*d}$，我们首先使用快速傅里叶变换(FFT)将 T 和 E 从时域转换到频域，在频域中，信号被表示为一系列不同频率的正弦和余弦成分的组合，有助于捕捉全局的特征信息：

$$\tilde{T} = \text{FFT}\left(T\right) \in C^{n*d} \tag{19}$$

$$\tilde{E} = \text{FFT}\left(E\right) \in C^{n*d} \tag{20}$$

其中 $\tilde{T}$ 和 $\tilde{E}$ 分别表示文本数据和 ID 嵌入的频谱。为了衰减 ID 嵌入中的噪声，我们引入一个可学习的滤波器矩阵 $W \in C^{n*d}$，并应用它到 ID 嵌入的频谱上：

$$\tilde{E}' = W \odot \tilde{E} \in C^{n*d} \tag{21}$$

其中“$\odot$”表示逐元素乘积。接下来，我们在频域中执行逐元素乘积，相当于在频域中将文本和 ID 嵌入进行特征融合：

$$\tilde{F} = \tilde{T} \odot \tilde{E} \in C^{n*d} \tag{22}$$

频域中的逐元素乘积，实现两个嵌入的特征融合。通过在频域中融合不同类型得嵌入，可以更好地整合它们得特征，发挥各自的优势。这种操作可以增强或抑制特定的频率成分，从而捕捉全局范围内的特征互动。我们采用 IFFT 将表示 $\tilde{E}'$ 和 $\tilde{F}$ 转换回时域，经过频域处理的信号在转换回时域后，包含了全局特征的信息，这些信息是在原始时域中难以直接获取的：

$$E' \leftarrow \mathrm{IFFT}\left(\tilde{E}'\right) \in R^{n*d} \tag{23}$$

$$F \leftarrow \mathrm{IFFT}\left(\tilde{F}\right) \in R^{n*d} \tag{24}$$

经过上述处理，最终的输出将包含两部分特征表示：一部分是通过文本和 ID 嵌入的融合得到的特征，另一部分是去噪后的 ID 嵌入。在获得 E' 和 F 之后，我们应用门控函数，以自适应地控制每个模态的信息流：

$$g_E = \sigma G_E E' \in R^{n*d} \tag{25}$$

$$g_F = \sigma G_F F \in R^{n*d} \tag{26}$$

其中 G_E 和 G_F 是可学习的门控函数，σ 是 Sigmoid 激活函数，g_E 和 g_F 是对应的 E' 和 F 的门控权重。最后，我们在时域中设计了一个双门控机制，用于将这两部分特征进行组合，这使得模型能够有选择地关注更重要的全局特征，提高特征表达的有效性。

$$V_{\mathrm{FFT}} = 2 \cdot \left(g_E \odot E' + g_F \odot F\right) \in R^{n*d} \tag{27}$$

乘以 2 是为了保持权重的动态范围，使得平均权重为 1。这个步骤整合了来自 ID 嵌入和文本嵌入，整合这些特征后的表示 $V_{\mathrm{FFT}} = \{v_1, v_2, v_3, \cdots, v_n\} \in R^{n*d}$。

4.2.2. 短时傅里叶变换特征融合

(1) 短时傅里叶变换(STFT)模块

短时傅里叶变换(STFT)模块专为序列数据的时频特征提取与重建设计，STFT 模块图如图 2 所示。

压缩网络(CompressNet)：对输入数据进行降维，得到压缩后的表示，压缩网络图如图 3 所示。

$$z^{(n)} = W^{(n)} a^{(n-1)} + b^{(n)} \tag{28}$$

$$a^{(n)} = \tanh\left(z^{(n)}\right) \tag{29}$$

$$a^{(n)} = \mathrm{Dropout}\left(a^{(n)}, p\right) \tag{30}$$

输入层输入数据 $a^{(0)} = x$ 依次通过各层，最终得到 $a^{(n)}$ 到输出层，相当于经过了 n 个 Block，W 相当于权重矩阵，b 相当于偏置向量。

短时傅里叶变换(STFT)：对于压缩后的每个维度的特征，我们应用 STFT 将时域信号转换到时频域，以捕获局部的频域信息。公式表示为：

$$S_i = \mathrm{STFT}\left(X_{\mathrm{compress},i}\right) \tag{31}$$

逆短时傅里叶变换(ISTFT)：在重建过程中，对每个维度的 STFT 结果应用 ISTFT，恢复到时域信号：

$$X_{\mathrm{istft},i} = \mathrm{ISTFT}\left(S_i\right) \tag{32}$$

扩展网络(EnlargeNet)：对 ISTFT 后的数据进行重构，恢复到原始维度，扩展网络图如图 4 所示。

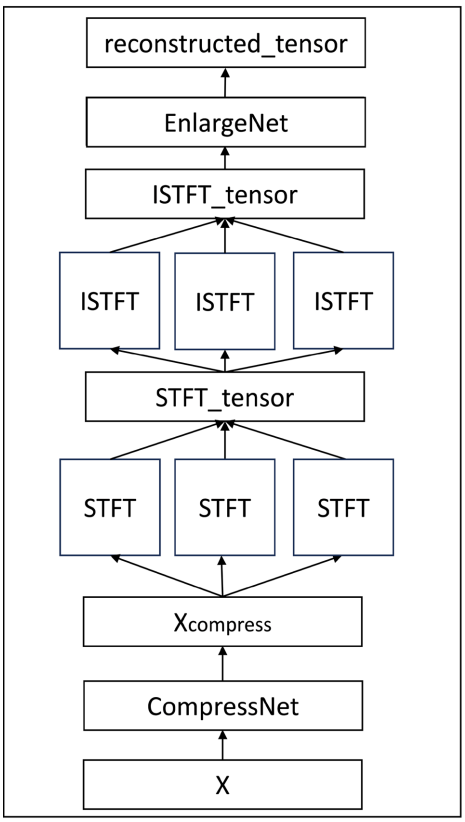

Figure 2. STFT module diagram
图 2. STFT 模块图

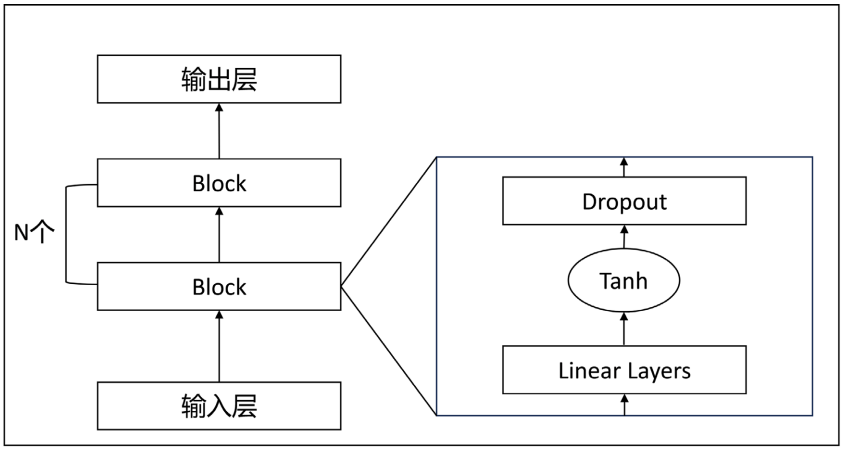

Figure 3. Compressed network diagram
图 3. 压缩网络图

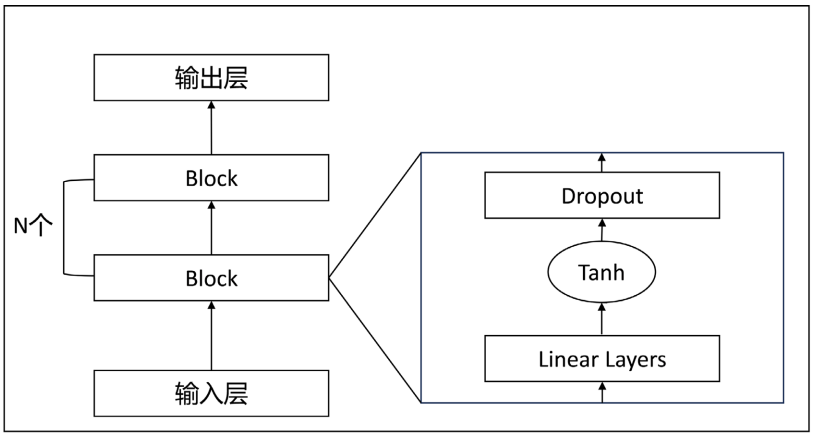

Figure 4. Expanded network diagram
图 4. 扩展网络图

$$y^{(n)} = W^{(n)} c^{(n-1)} + b^{(n)} \tag{33}$$

$$c^{(n)} = \tanh\left(y^{(n)}\right) \tag{34}$$

$$c^{(n)} = \text{Dropout}\left(c^{(n)}, p\right) \tag{35}$$

输入层输入数据 $c^{(0)} = x$ 依次通过各层，最终得到 $c^{(n)}$ 到输出层，相当于经过了 n 个 Block，W 相当于权重矩阵，b 相当于偏置向量。

(2) 特征融合

ID 嵌入 $E = \{e_1, e_2, e_3, \cdots, e_n\} \in R^{n*d}$ 和文本嵌入 $T = \{t_1', t_2', t_3', \cdots, t_n'\} \in R^{n*d}$，我们首先应用短时傅里叶变换(STFT)将 T 和 E 从时域转换到频域，STFT 将原始时域信号分割成短时间窗口，得到每个时间段的频谱信息，从而保留了信号的局部时频特征：

$$\tilde{T} = \text{STFT}(T) \in C^{n*d} \tag{36}$$

$$\tilde{E} = \text{STFT}(E) \in C^{n*d} \tag{37}$$

其中 $\tilde{T}$ 和 $\tilde{E}$ 分别表示文本和 ID 嵌入的短时频谱。为了减弱 ID 嵌入中的噪声，我们引入一个可学习的滤波器矩阵 $W \in C^{n*d}$，并应用它到 ID 嵌入的频谱上：

$$\tilde{E}' = W \odot \tilde{E} \in C^{n*d} \tag{38}$$

其中“$\odot$”表示逐元素乘积。接下来，我们在频域中执行逐元素乘积，在短时傅里叶变换中相当于在频域中将文本和 ID 嵌入进行特征融合，可以突出文本和 ID 嵌入在不同频率的共同特征：

$$\tilde{F} = \tilde{T} \odot \tilde{E} \in C^{n*d} \tag{39}$$

通过频域中的逐元素乘积有效地融合了文本和 ID 的频率特征，突出它们的共同频率成分，捕获了它们之间的频率相关性，可以抑制在其中一个特征中不存在的频率成分，从而减少噪声和不相关信息的影响，提升特征表示的质量。接着，我们使用 ISTFT 将表示 $\tilde{E}'$ 和 $\tilde{F}$ 转换时域，ISTFT 后的信号包含了在频域中被强调或抑制的局部特征：

$$F \leftarrow \text{ISTFT}\left(\tilde{F}\right) \in R^{n*d} \tag{40}$$

$$E' \leftarrow \text{ISTFT}\left(\tilde{E}'\right) \in R^{n*d} \tag{41}$$

经过上述处理，最终的输出将包含两部分特征表示：一部分是通过文本和 ID 嵌入融合得到的特征，另一部分是去噪后的 ID 嵌入。在获得 E' 和 F 之后，我们应用门控函数以自适应地控制每个模态的信息流：

$$g_E = \sigma G_E E' \in R^{n*d} \tag{42}$$

$$g_F = \sigma G_F F \in R^{n*d} \tag{43}$$

其中 G_E 和 G_F 是可学习的门控函数，σ 是 Sigmoid 激活函数，g_E 和 g_F 是对应的 E' 和 F 的门控权重。最后，我们在时域中设计了一个双门控机制，用于将这两部分特征进行组合，得到融合了局部频率特征。

$$V_{\text{STFT}} = 2 \cdot \left(g_E \odot E' + g_F \odot F \right) \in R^{n*d} \tag{44}$$

乘以 2 是为了保持权重的动态范围，使得平均权重为 1。这个步骤整合了来自 ID 嵌入和文本嵌入，整合这些特征后的表示 $V_{\text{STFT}} = \{v_1, v_2, v_3, \cdots, v_n\} \in R^{n*d}$ 。

4.2.3. 特征交叉融合

ID 嵌入 $E = \{e_1, e_2, \cdots, e_n\} \in R^{n*d}$ 和文本嵌入 $T = \{t'_1, t'_2, \cdots, t'_n\} \in R^{n*d}$，为了有效融合 ID 嵌入和文本嵌入。首先通过对 ID 嵌入与文本嵌入进行逐元素相加，生成初步的融合表示。这一操作是为了整合项目及其相关特征的基本信息。捕捉两者之间的线性关联。接着，利用 ID 嵌入与文本嵌入进行逐元素乘积，进一步捕捉二者之间的非线性交互关系。这一操作有助于模型学习更复杂的特征交互，提高表示能力。最终，将上述两部分结果相加得到融合后的特征，该特征同时包含了物品与特征的线性和非线性交互信息。为了进一步提取和优化特征表达能力。我们将融合后的特征输入到 OMLLA 中，最终得到的特征表示更加全面和深入。整合这些特征后的表示 $V_{\text{OMLLA}} = \{v_1, v_2, v_3, \cdots, v_n\} \in R^{n*d}$ 。OMLLA 图如图 5 所示，公式如下：

$$V_{\text{Res}} = e_i + t'_i + e_i \odot t'_i \tag{45}$$

$$V_{\text{OMLLA}} = \text{OMLLA}\left(V_{\text{Res}}\right) \tag{46}$$

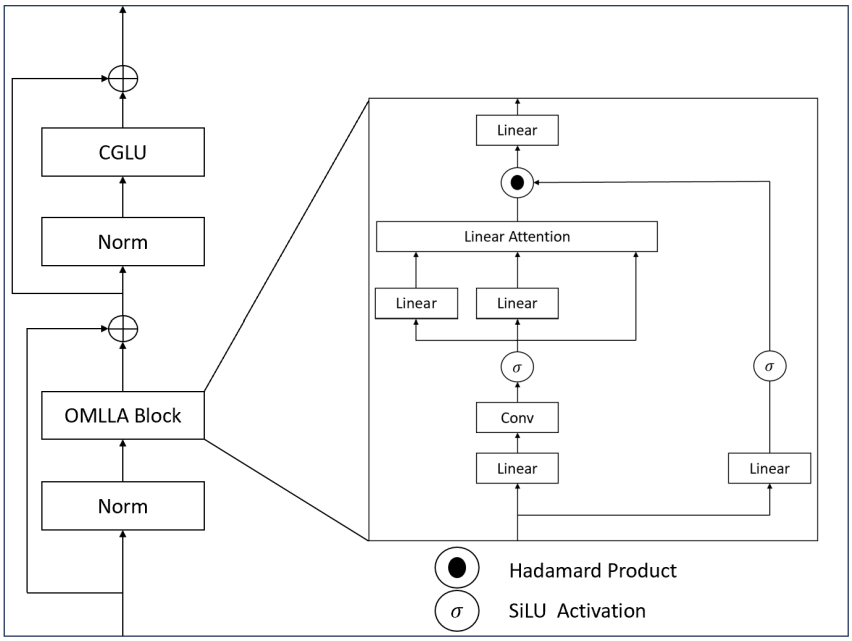

Figure 5. OMLLA diagram
图 5. OMLLA 图

MLLA [17]在处理图像任务的时候是出色的，在 MLLA 启发下，我们想要把它应用在处理文本信息序列的任务中，因此我们对 MLLA 进行了优化并且命名为 OMLLA，在线性注意力中通过高斯特征映射函数替换了 ELU 激活函数和 RoPE 位置编码。线性注意力的计算公式如下：

$$\text{Attention}(Q,K,V)=\frac{\varnothing(Q)\left(\varnothing(K)^{\mathrm{T}}V\right)}{\varnothing(Q)\left(\varnothing(K)^{\mathrm{T}}P\right)} \tag{47}$$

其中 $\varnothing(Q)$，$\varnothing(K)$ 是经过高斯特征映射的 Q 和 K，P 是元素全为 1 的向量，用于归一化。特征映射公式：

$$\varnothing(Q)=\exp\left(-\frac{1}{2}Q^2\right) \tag{48}$$

$$\varnothing(K)=\exp\left(-\frac{1}{2}K^2\right) \tag{49}$$

用 CGLU 模块[18]替换了 MLP 模块。CGLU 中的深度卷积(Dwconv)层捕捉输入特征的局部空间依赖性。CGLU 使用门控机制，将深度卷积的输出与一个门控向量 V 逐元素相乘，选择性地强调或抑制特征，增强特征表示。通过组合线性层、深度卷积和门控机制，CGLU 增加了模型的非线性和复杂性，从而更好地捕捉数据中的复杂模式和关系，CGLU 图如图 6 所示。

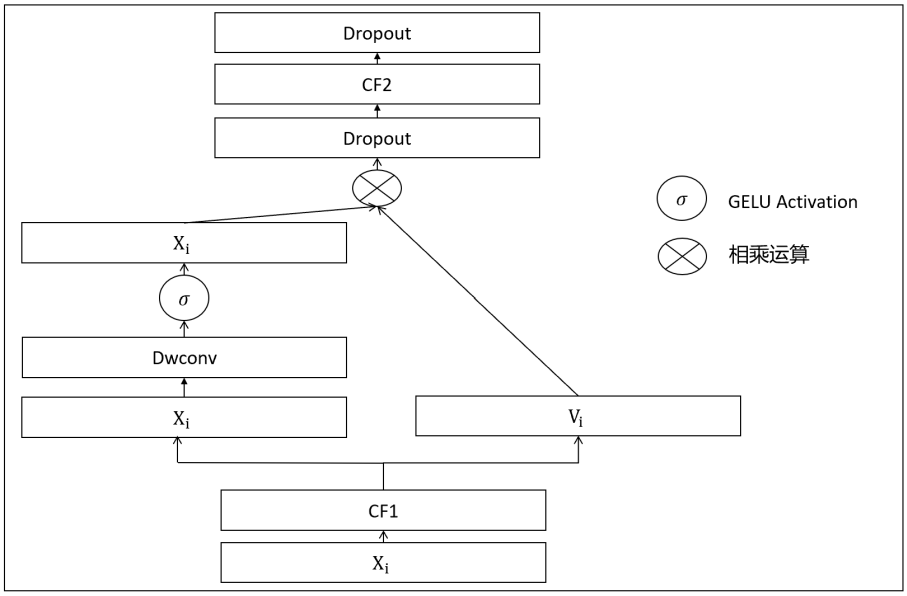

Figure 6. CGLU diagram
图 6. CGLU 图

最后我们可以通过 OMLLA 的条件位置编码来增强序列中的位置信息，线性注意力来提高长序列处理的效率，通过深度可分离卷积来有效捕获局部特征，通过 CGLU 来增强非线性表达能力，以及残差连接和层归一化来改善梯度流动和训练稳定性。因此通过 OMLLA 提取了更高阶的特征关系，提升了特征表示的表达能力，使得最终得到的特征表示表示更加全面和深入。

4.2.4. 加权特征融合

定义了一个名为 MergeNet 的神经网络模型，用于将 V_{FFT}、V_{STFT}、V_{OMLLA} 三个特征向量在最后一个维

度上进行拼接成一个高维向量，融合了不同来源得信息。使用多层感知机(MLP)对拼接后的特征进行深度特征提取，最后输出经过 Softmax 激活函数的权重向量。将得到得权重向量按最后一个维度进行分割，每个分割得到的权重向量尺寸为 1。这样得到V_{FFT}、V_{STFT}、V_{OMLLA}对应得权重，将各个嵌入向量与其对应的权重相乘，然后将结果相加，得到融合后的嵌入向量V_{merge}。

$$V_{\text{merge}} = V_{\text{FFT}} \cdot \text{fft}_{\text{weight}} + V_{\text{STFT}} \cdot \text{stft}_{\text{weight}} + V_{\text{OMLLA}} \cdot \text{omlla}_{\text{weight}} \tag{50}$$

4.3. 预测与优化

通过融合后的项目表示$V_{\text{merge}} = [v_1, v_2, v_3, \cdots, v_n] \in R^{n*d}$，我们进一步使用 SASRec 来获得序列表示。序列表示可以表述为：

$$x_j^0 = v_j + p_j \tag{51}$$

$$X^{i+1} = \text{FFN}\left(\text{MHA}\left(X^i\right)\right) \tag{52}$$

其中$X^i = \left[x_1^i, x_2^i, x_3^i, \cdots, x_n^i\right]$是第 i 层的输出表示，p_j是第 j 个位置的绝对位置嵌入，FFN 是逐点前馈网络，MHA 是多头注意力机制。$\left[\hat{x}_1, \hat{x}_2, \hat{x}_3, \cdots, \hat{x}_n\right]$为最后一层，我们选择第 n 个位置$\hat{x}_n$作为序列的表示。

我们采交叉熵(CE)损失函数、温度参数τ、标签平滑和焦点损失形成的综合损失函数：

$$\mathcal{L}_{\text{CE}} = -\log\left(\frac{\exp\left(\hat{x}_n^{\text{T}} e_j / \tau\right)}{\sum_{j'=1}^{K} \exp\left(\hat{x}_n^{\text{T}} e_{j'} / \tau\right)}\right) \tag{53}$$

$$\tilde{y}_{n,j} = (1-\epsilon) y_{n,j} + \frac{\epsilon}{K} \tag{54}$$

$$p_{n,j} = \frac{\exp\left(\hat{x}_n^{\text{T}} e_j / \tau\right)}{\sum_{j'=1}^{K} \exp\left(\hat{x}_n^{\text{T}} e_{j'} / \tau\right)} \tag{55}$$

$$\alpha_{n,j} = \left(1 - p_{n,j}\right)^{\gamma} \tag{56}$$

$$\mathcal{L} = -\sum_{j=1}^{K} \tilde{y}_{n,j} \cdot \alpha_{n,j} \cdot \log p_{n,j} \tag{57}$$

其中，$y_{n,j}$是原始标签，$\tilde{y}_{n,j}$是平滑后的标签，$p_{n,j}$是第 n 个样本属于第 j 类的概率，ϵ是平滑标签参数，K是项目总数。

计算v_{n+1}的概率为：

$$p\left(v_{n+1} \mid (v_1, v_2, v_3, \cdots, v_n)\right) = \text{Softmax}\left(\hat{x}_n \cdot e^{\text{T}}\right) \tag{58}$$

5. 实验

5.1. 数据集与评价指标

本文的实验在从 Amazon dataset 中选择 Instant video、Beauty、Digital Music、Tools and Home improvement 四个基准数据集上进行。所有数据集均含有评分(评分范围为 1 至 5)及相应评论，且在数据集的大小与稀疏程度上存在差异。表 1 详细列出了这些数据集的具体信息。针对全部数据集，我们过滤掉不受欢迎的项目和互动记录少于五次的不活跃用户。我们按照 80%、10%、10%的比例，随机划分成训练集、验证集与测试集。

Table 1. Basic information of the dataset
表 1. 数据集基本信息

数据集	用户数	商品数	评论样本数	稀疏度
Instant video	5130	1685	37,126	99.57%
Beauty	22,363	12,101	198,502	99.92%
Digital Music	5541	3568	64,706	99.67%
Tools and Home improvement	35,598	18,357	134,476	99.97%

本文在验证集上对所有方法的超参数进行调优，并通过计算两个标准评估指标：归一化折损累计增益 NDCG@K ($K \in 10,20$)和召回率 Recall@K ($K \in 10,20$)来评估在测试集上的最终性能。

Recall@K：Recall@K 衡量了在推荐系统给出的前 K 个推荐项目中，用户真实感兴趣的项目占用户所有感兴趣项目的比例。其公式为：

$$\text{Recall@K} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{59}$$

其中，TP 指的是推荐列表的前 K 个项目中，成功包含了用户真实感兴趣的项目的数量。FN 指的是用户真实感兴趣，但未出现在推荐列表的前 K 个项目中的项目数量。

NDCG@K：NDCG@K 综合评估推荐前 K 项排序质量，依据相关性及排名(高排名项权重更高)，并以理想排序 DCG@K 为基准归一化。其公式为：

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \tag{60}$$

其中，DCG@K 通过累计推荐列表前 K 项的相关性得分评估推荐性能，以反映用户对前列的关注。IDCG@K 是相关项按最优排序时的 DCG@K 最大值，作为归一化基准。

5.2. 基线模型

为了验证本文算法效果，本文介绍训练了 6 个算法与本章算法进行对比实验。

SASRec：利用自注意力机制捕捉用户行为序列中的依赖关系，进行个性化推荐。

GRU4Rec：使用门控循环单元(GRU)捕捉用户行为序列的动态特性，进行序列推荐。

GCSAN：结合图卷积网络和序列注意力机制，捕捉用户行为序列和项目间的关系。

LightSANs：轻量级自注意力网络，专注于高效且准确地处理用户行为序列。

FEARec [19]：特征增强的注意力机制与循环神经网络结合，捕捉用户行为序列中的重要特征。

TedRec：在文本和 ID 特征的融合方面表现出色，使得序列推荐系统能够更准确地预测用户的兴趣。

5.3. 实验细节

采用 RecBole [20]和 Pytorch [21]框架实现各模型。确保实验的一致性和可比性，所有模型均在 RTX4090 机器上进行，训练批次大小设置为 512，验证批次大小设置为 512，最大序列长度设置为 50、学习率为 0.01，并采用早停策略，在验证集上的 NDCG@10 连续 20 个 epoch 没有改善时结束训练，使用 Adam 优化器来优化模型。我们采用每个基线模型的最优参数配置，保持所有模型的核心参数一致，并在可行范围内进行调整以提升性能。

5.4. 性能评估

Instant video 实验结果如表 2 所示。

Table 2. Instant video Experimental Results
表 2. Instant video 实验结果

Model	Recall@10	Recall@20	NDCG@10	NDCG@20
SASRec	0.2156	0.2903	0.1033	0.1221
GRU4Rec	0.1754	0.2411	0.0927	0.1092
GCSAN	0.2111	0.2852	0.1091	0.1277
LightSANs	0.2211	0.2893	0.1094	0.1267
FEARec	0.2074	0.2860	0.1031	0.1229
TedRec	0.2339	0.3021	0.1536	0.1707
TFFCRec	0.2487	0.3179	0.1603	0.1778

Beauty 实验结果如表 3 所示。

Table 3. Beauty experimental results
表 3. Beauty 实验结果

Model	Recall@10	Recall@20	NDCG@10	NDCG@20
SASRec	0.0710	0.1069	0.0317	0.0407
GRU4Rec	0.0577	0.0927	0.0297	0.0385
GCSAN	0.0707	0.1066	0.0324	0.0415
LightSANs	0.0757	0.1158	0.0349	0.0450
FEARec	0.0715	0.1089	0.0319	0.0413
TedRec	0.0814	0.1183	0.0457	0.0550
TFFCRec	0.0922	0.1298	0.0508	0.0602

Digital Music 实验结果如表 4 所示。

Table 4. Digital Music experimental results
表 4. Digital Music 实验结果

Model	Recall@10	Recall@20	NDCG@10	NDCG@20
SASRec	0.1742	0.2482	0.0775	0.0962
GRU4Rec	0.1301	0.1996	0.0636	0.0812
GCSAN	0.1700	0.2480	0.0781	0.0978
LightSANs	0.1691	0.2447	0.0773	0.0965

续表

FEARec	0.1444	0.2144	0.0654	0.0830
TedRec	0.1853	0.2568	0.1053	0.1233
TFFCRec	0.1922	0.2660	0.1103	0.1289

Tools and Home improvement 实验结果如表 5 所示。

Table 5. Tools and Home improvement experimental results
表 5. Tools and Home improvement 实验结果

Model	Recall@10	Recall@20	NDCG@10	NDCG@20
SASRec	0.0451	0.0686	0.0205	0.0264
GRU4Rec	0.0321	0.0521	0.0166	0.0216
GCSAN	0.0485	0.0713	0.0223	0.0280
LightSANs	0.0456	0.0686	0.0205	0.0264
FEARec	0.0448	0.0664	0.0204	0.0258
TedRec	0.0516	0.0767	0.0282	0.0345
TFFCRec	0.0550	0.0789	0.0304	0.0364

5.5. 结果分析

SASRec 基于自注意力机制的模型能够捕获任意距离的依赖关系，适用于长序列。对于稀疏数据，可能需要大量训练数据才能表现良好。GRU4Rec 能够捕获用户点击序列中的短期和长期依赖关系。难以建模复杂的长序列依赖，相较于基于自注意力的模型，可能在性能上有所欠缺。GCSAN 在自注意力机制中引入全局上下文信息，能够捕获序列中更广泛的依赖关系。如果全局上下文信息质量不高，可能引入噪声，反而影响模型性能。LightSANs 相比于传统的自注意力模型，LightSANs 通过削减不必要的计算，显著降低了时间和空间复杂度，适合于处理长序列和大规模数据，一些需要捕获全局复杂依赖的情况下，轻量级的自注意力机制可能无法完全胜任。FEARec 能够有效地融合用户和物品的多种特征信息，需要对用户和物品的特征进行充分的提取和处理，输入特征的数据质量要求高，噪声或缺失的数据可能对模型性能产生较大影响。TedRec 通过将文本和 ID 特征在频域内进行融合，利用傅里叶变换整合全局上下文信息，TedRec 的性能高度依赖于高质量的文本嵌入。如果文本嵌入质量不高，可能会影响推荐效果。总体而言，我们方法在四个数据集上均优于所有基线模型，验证了 TFFCRec 的有效性。

5.6. 消融实验

在本部分中，我们评估了每个提出的组件对最终性能的积极影响，消融实验图如图 7 所示。为了进行消融研究，我们分析了以下四种方法变体进行比较：(1) 去掉(w/o)多样化专家调制；(2) 去掉(w/o)快速傅里叶变换；(3) 去掉(w/o)短时傅里叶变换；(4) 去掉(w/o)特征交叉融合。

展示出我们方法与四种变体的性能对比。可以看出 TFFCRec 中的所有提出组件都影响了整体的推荐性能，并且我们的最终方法表现最佳。

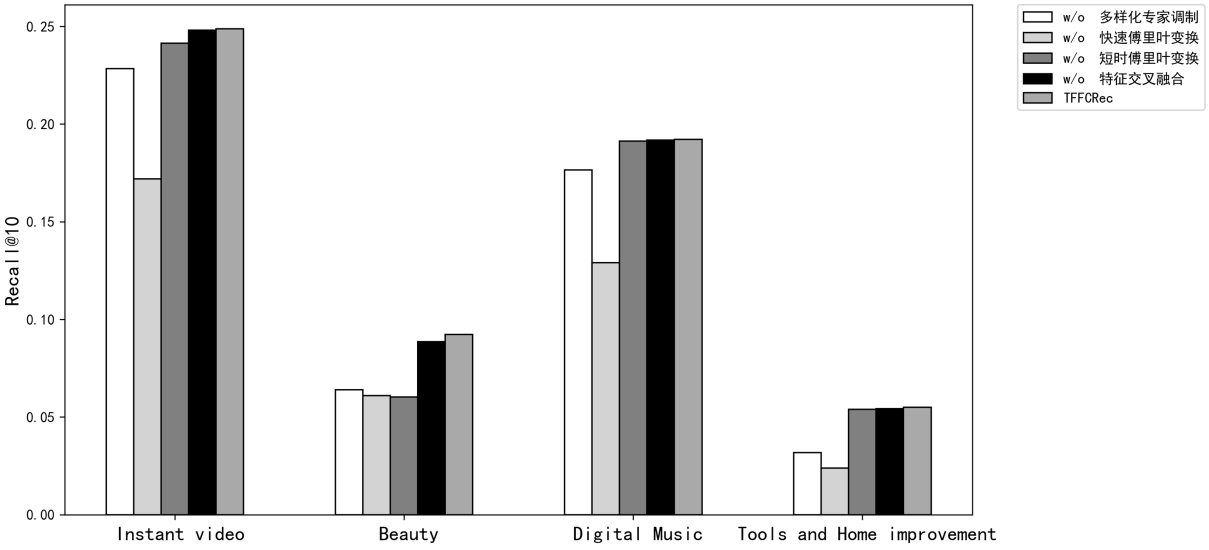

Figure 7. Ablation experiment diagram
图 7. 消融实验图

6. 结论

通过多样化专家的调制方法并动态地融入位置信息，增强了文本编码器生成的表示向量之间的差异性和独特性。通过对项目的 ID 嵌入和文本嵌入进行快速傅里叶变换(FFT)和短时傅里叶变换(STFT)，我们成功地捕获了全局频域特征和局部时频特征。此外，我们引入了特征交叉融合，并通过 OMLLA 来捕获特征之间更深层次的非线性关系，提取更深层次的特征表示。我们设计了一个融合网络，自适应地学习不同嵌入表示的权重，将 FFT、STFT 和 OMLLA 得到的特征向量进行加权融合，进而通过 SASRec 进行序列的推荐。在四个公开数据集上进行的大量实验验证了我们方法的有效性和效率。在未来研究中，我们应该考虑有效地融合多模态信息，进一步丰富用户兴趣特征的表示以及如何优化算法，提高计算效率。

参考文献

[1] Kang, W. and McAuley, J. (2018) Self-Attentive Sequential Recommendation. 2018 *IEEE International Conference on Data Mining* (*ICDM*), Singapore, 17-20 November 2018, 197-206. https://doi.org/10.1109/icdm.2018.00035

[2] Liu, Y. (2019) Roberta: A Robustly Optimized Bert Pretraining Approach. arXiv: 1907.11692.

[3] Xu, L., Tian, Z., Li, B., Zhang, J., Wang, D., Wang, H., *et al*. (2024) Sequence-Level Semantic Representation Fusion for Recommender Systems. *Proceedings of the* 33*rd ACM International Conference on Information and Knowledge Management*, Boise, 21-25 October 2024, 5015-5022. https://doi.org/10.1145/3627673.3680037

[4] Church, K.W. (2016) Word2Vec. *Natural Language Engineering*, **23**, 155-162. https://doi.org/10.1017/s1351324916000334

[5] Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, 25-29 October 2014, 1532-1543. https://doi.org/10.3115/v1/d14-1162

[6] Alaparthi, S. and Mishra, M. (2020) Bidirectional Encoder Representations from Transformers (BERT): A Sentiment analysis Odyssey. arXiv: 2007.01127.

[7] Huang, J., Tang, D., Zhong, W., Lu, S., Shou, L., Gong, M., *et al*. (2021) WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. *Findings of the Association for Computational Linguistics*: *EMNLP* 2021, Punta Cana, 16-20 November 2021, 238-244. https://doi.org/10.18653/v1/2021.findings-emnlp.23

[8] Rendle, S., Freudenthaler, C. and Schmidt-Thieme, L. (2010) Factorizing Personalized Markov Chains for Next-Basket

Recommendation. *Proceedings of the* 19*th International Conference on World Wide Web*, Raleigh, 26-30 April 2010, 811-820. https://doi.org/10.1145/1772690.1772773

[9] Jannach, D. and Ludewig, M. (2017) When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. *Proceedings of the Eleventh ACM Conference on Recommender Systems*, Como, 27-31 August 2017, 306-310. https://doi.org/10.1145/3109859.3109872

[10] Tang, J. and Wang, K. (2018) Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, Marina Del Rey, 5-9 February 2018, 565-573. https://doi.org/10.1145/3159652.3159656

[11] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., *et al*. (2019) BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Trans-Former. *Proceedings of the* 28*th ACM International Conference on Information and Knowledge Management*, Beijing, 3-7 November 2019, 1441-1450. https://doi.org/10.1145/3357384.3357895

[12] Zhou, K., Wang, H., Zhao, W.X., Zhu, Y., Wang, S., Zhang, F., *et al*. (2020) S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. *Proceedings of the* 29*th ACM International Conference on Information & Knowledge Management*, 19-23 October 2020, 1893-1902. https://doi.org/10.1145/3340531.3411954

[13] Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., *et al*. (2019) Graph Contextualized Self-Attention Network for Session-Based Recommendation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao SAR, 10-16 August 2019, 3940-3946. https://doi.org/10.24963/ijcai.2019/547

[14] Fan, X., Liu, Z., Lian, J., Zhao, W.X., Xie, X. and Wen, J. (2021) Lighter and Better: Low-Rank Decomposed Self-Attention Networks for Next-Item Recommendation. *Proceedings of the* 44*th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 11-15 July 2021, 1733-1737. https://doi.org/10.1145/3404835.3462978

[15] Zhou, K., Yu, H., Zhao, W.X. and Wen, J. (2022) Filter-Enhanced MLP Is All You Need for Sequential Recommendation. *Proceedings of the ACM Web Conference* 2022, 25-29 April 2022, 2388-2399. https://doi.org/10.1145/3485447.3512111

[16] Hou, Y., Mu, S., Zhao, W.X., Li, Y., Ding, B. and Wen, J. (2022) Towards Universal Sequence Representation Learning for Recommender Systems. *Proceedings of the* 28*th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington DC, 14-18 August 2022, 585-593. https://doi.org/10.1145/3534678.3539381

[17] Han, D., Wang, Z., Xia, Z., *et al*. (2024) Demystify Mamba in Vision: A Linear Attention Perspective. arXiv: 2405.16605.

[18] Shi, D. (2024) TransNeXt: Robust Foveal Visual Perception for Vision Transformers. 2024 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 16-22 June 2024, 17773-17783. https://doi.org/10.1109/cvpr52733.2024.01683

[19] Du, X., Yuan, H., Zhao, P., Qu, J., Zhuang, F., Liu, G., *et al*. (2023) Frequency Enhanced Hybrid Attention Network for Sequential Recommendation. *Proceedings of the* 46*th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Taipei City, 23-27 July 2023, 78-88. https://doi.org/10.1145/3539618.3591689

[20] Zhao, W.X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., *et al*. (2021) RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. *Proceedings of the* 30*th ACM International Conference on Information & Knowledge Management*, 1-5 November 2021, 4653-4664. https://doi.org/10.1145/3459637.3482016

[21] Paszke, A., Gross, S., Massa, F., *et al*. (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv: 1912.01703.