用于序列推荐的对比增强滑动滤波器混合器

Xinyu Du[1†], Huanhuan Yuan[1†], Pengpeng Zhao[1*], Junhua Fang[1],Guanfeng Liu[2],Yanchi Liu[3],Victor S. Sheng[4],和 Xiaofang Zhou[5]苏州大学计算机科学与技术学院，中国苏州[2]澳大利亚麦考瑞大学计算系[3]美国罗格斯大学 [4]美国德克萨斯理工大学[5]中国香港科技大学{xydu, hhyuan}@stu.suda.edu.cn {ppzhao, jhfang}@suda.edu.cn guanfeng.liu@mq.edu.au yanchi.liu@rutgers.edu victor.sheng@ttu.edu zxf@cse.ust.hk

arXiv:2305.04322v1 [cs.IR] 7 May 2023

摘要——**序列推荐（SR）旨在通过从用户历史物品交互数据中捕捉行为模式来建模用户偏好。大多数现有方法都在时域中建模用户偏好，却忽略了用户行为还会受到各种频率模式的影响，而这些模式难以从纠缠在一起的时间序列物品中分离出来。然而，鲜有研究尝试在频域中训练 SR，如何利用频率分量学习合适的用户表示仍不明确。为解决这一问题，我们将视角转向频域，并提出一种新颖的用于序列推荐的对比增强滑动滤波器混合器，简称 SLIME4Rec。具体而言，我们设计了一种频率斜坡结构，使可学习滤波器能够在不同层的频谱上滑动，从而捕捉不同的频率模式。此外，我们提出动态频率选择（DFS）和静态频率划分（SFS）模块，以两种方式有效提取频率信息，取代自注意力模块。DFS 用于动态选择有用的频率分量，而 SFS 与动态频率选择模块结合，以提供更细粒度的频率划分。最后，我们利用对比学习提升从频域学习到的用户嵌入质量。在五个广泛使用的基准数据集上进行的大量实验表明，我们提出的模型显著优于最先进的方法。代码可在 https://github.com/sudaada/SLIME4Rec 获取。**

索引词—**序列推荐、滤波算法、周期模式、对比学习。**

一、引言

推荐系统（归一化折损累计增益）已在各种众多在线平台依靠其准确推荐用户可能感兴趣的一系列合适物品的能力。不同于通常假设用户偏好静态的传统推荐系统 [1]，序列推荐算法会考虑时间序列信息 [2]–[4] ，以便更好地从用户交互序列中识别用户行为模式，并捕捉用户兴趣的演变。

然而，大多数序列模型主要从时域的角度处理历史交互，很少充分利用用户在频域中的特征，而频域常用于数字信号处理领域以滤除噪声 [5], [6]。例如，图1展示了 Alice 和 Bob 分别交互过的历史物品序列。从时域的角度观察，所有物品都按时间顺序排列，并沿着 t轴交织在一起，本质上其中包含噪声，甚至可能包含恶意伪造内容 [5]。通过离散傅里叶变换 [7], [8],将输入的时间特征转换到频域后，可以将每位用户的历史物品序列分解为不同频谱，从而在不同频率上发现更细粒度的序列模式。

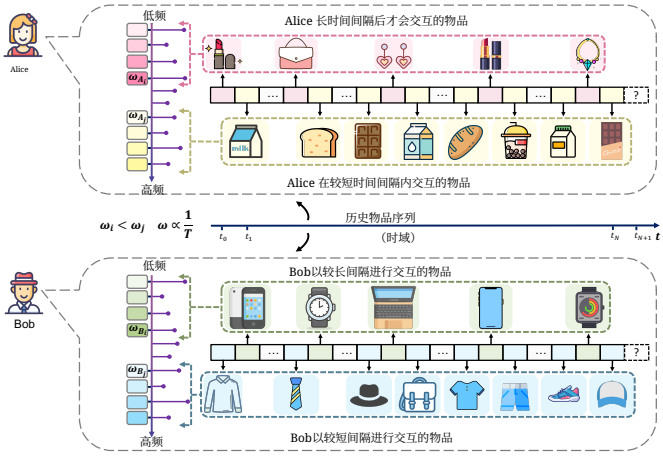


图 1. 从两个不同视角观察用户历史物品序列的示意图。从时域视角观察时，Alice 和 Bob 交互过的物品沿 t 轴按时间顺序排列。转而从频域视角观察时，每位用户的历史物品序列会沿 ω 轴分解为多个具有不同频率和周期的行为模式。Alice 和 Bob 都具有低频行为（$i.e.$, ω_{A_i}, ω_{B_i}）和高频行为（$i.e.$, ω_{A_j}, ω_{B_j}）。

在时域中，深度推荐模型往往会对噪声交互数据过拟合 [5]在时域中，这是因为现有方法为所有交互物品分配权重，以便在物品层面发现用户行为模式

†他们是共同第一作者，贡献相同。*通讯作者

。此外，一些研究表明，广泛使用的自注意力操作在汇聚全局低频信息方面效果显著更好，但会削弱高频信号 [9], [10], 而高频信号可以在频域中轻松获取。为此，FMLP-Rec [5] 尝试在频域中处理序列数据，以减弱特征中隐藏的噪声信息，并捕捉用户的频率模式。

对于基于频率的方法，如何有效利用频率分量来表示用户行为是一个关键问题。一方面，虽然通过简单地乘以一个覆盖所有频率分量的滤波器，模型能够在频域中最大程度地保留用户的全面信息，但噪声输入可能会导致表示效果较差 [11]。另一方面，同时学习不同的频率特性并不可取。频域特征还可以进一步划分为其他分量（例如高频和低频），这有助于学习不同频率下的变化模式。这一优势对于推荐任务尤其重要，因为用户行为往往呈现出一定的周期性趋势。对于推荐系统，高频主要包括在短时间间隔内经常购买的物品（例如服装），而相对低频则通常包括在较长时间间隔内购买的物品（例如手机或电脑）。在图1中，行为 ω_{B_j}（例如服装和户外用品）的频率明显高于 ω_{B_i}（例如电子产品），因为 $\omega_{B_j} > \omega_{B_i}$。但是，当所有频率的物品都交织在时间序列中时，很难识别这些变化模式。因此，分别灵活地学习高频行为（服装和户外用品）和低频行为（电子产品），或许能够更好地理解整体用户偏好。

基于上述原因，我们提出了一种简单高效的模型，称为面向序列推荐的对比增强滑动滤波器混合器（Contrastive Enhanced Slide Filter Mixer for Sequential Recommendation），简称 SLIME4Rec。我们持续探索频域中的频率特征，并发现用户历史序列由多种具有不同周期和频率的行为模式构成。SLIME4Rec 的核心组件是滤波器混合器，它在频域中以动态频率选择模块和静态频率划分模块取代自注意力模块。首先，与 FMLP-Rec [5] 利用全局可学习滤波器覆盖所有频率分量不同，我们提出动态频率选择模块，使可学习滤波器能够在频域中滑动。具体而言，为权衡用户的不同频率模式，频率斜坡结构模块通过不同的神经网络层，在频域中分层覆盖特定的频率范围。该模块的主要作用是使 SLIME4Rec 能够自适应地强调或滤除某些频率分量，并缓解噪声导致的过拟合现象。其次，为确保考虑所有频率分量，从而最大程度地保留用户有意义的周期性特征，我们将静态频率划分模块与动态频率选择模块结合起来。该模块根据层数对所有频率分量进行平均划分，以实现更细粒度的频率划分，并重新捕获动态频率选择模块遗漏的频率。最后，为提升用户嵌入的质量，我们设计了一项对比学习任务，以增强监督信号，并将其作为推荐任务的补充。本文的主要贡献总结如下：

- 我们将视角转向频域，并设计了一种频率斜坡结构，利用滑动滤波器在不同层中捕获整体频率特征的不同子集。
- 我们分别提出动态频率选择模块和静态频率划分模块，用于混合由不同尺寸的滤波器捕获的不同细粒度频率特征。
- 我们将基于频率特征的模型引入对比学习范式。通过统一推荐任务与用户行为层面的自监督任务，可以显著提升推荐性能。
- 我们在五个公开数据集上进行了大量实验，结果表明，与最先进的基线方法相比，SLIME4Rec在计算复杂度更低的情况下表现更优。

II. 预备知识

在详细介绍所提出的 SLIME4Rec 之前，我们先给出一些必要的符号来形式化序列推荐问题，然后介绍傅里叶变换和卷积定理的一些背景知识。

A. 问题陈述

序列推荐的目标是根据用户先前的交互，预测用户接下来会点击的物品。我们用以下形式化定义来描述序列推荐问题。设 $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$表示所有不重复物品的集合，$\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$表示所有用户的集合，其中 $u \in \mathcal{U}$ 表示一个用户，$v \in \mathcal{V}$ 表示一个物品。用户数和物品数分别记为 $|\mathcal{U}|$ 和 $|\mathcal{V}|$。用户行为集合可表示为 $S = \{s_1, s_2, ..., s_{|\mathcal{U}|}\}$。在 SR 中，用户行为序列通常按时间顺序排列。这意味着每个用户序列由（按时间顺序排列的）物品交互 $s_u = [v_1^{(u)}, v_2^{(u)}, ..., v_t^{(u)}, ..., v_n^{(u)}]$构成，其中 $s_u \in S$, $v_t^{(u)} \in \mathcal{V}$是用户$u$在步骤 t交互的物品，n是序列长度。$s_{u,\ t} = [v_1^{(u)}, v_2^{(u)}, ..., v_t^{(u)}]$ 表示用户 u 在 $t+1$之前交互过的物品子序列。序列通常会截断至最大长度 N。如果序列长度大于 N，则仅考虑最近的 N 个交互元素。如果序列长度小于 N，则在左侧插入零填充物品，直至序列长度达到 N。因此，如果 $t > N$，我们会截断输入

序列 $s_{u,\ t}$ 在推断时间步 $t+1$用户u 的表示时，保留最后的 N 个元素：

$$s_{u,t} = [v_{t-N+1}, v_{t-N+2}, \ldots, v_t] \tag{1}$$

具体而言，推荐模型经过训练后，会为每个候选项生成概率分数， $i.e.$， $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \ldots\ldots, \hat{y}_{|\mathcal{V}|}\}$，其中$\hat{y}_i$表示物品 v_i的预测分数。给定用户的历史交互序列，且不包含任何其他辅助上下文信息，序列推荐任务以$s_{u,\ t}$ 为输入，预测时间戳 $t+1$处最有可能的前K 个物品，可表述如下：

$$v_u^* = \arg\max_{v_i \in \mathcal{V}} P\left(v_{t+1}^{(u)} = v_i \mid s_{u,t}\right) \tag{2}$$

其中， v_u^* 表示 u的真实物品。

B. 傅里叶变换

离散傅里叶变换（DFT） [7], [8]是一种应用最广泛的计算方法，在数据分析、信号处理和机器学习领域有着诸多应用。为了将序列行为从时域转换到频域，我们将离散傅里叶变换引入SLIME4Rec。由于序列推荐中的输入数据是序列形式，我们只考虑一维离散傅里叶变换。给定一个由 $\{x_n\}_{n=0}^{N-1}$构成的有限序列，一维离散傅里叶变换将原始序列转换为频域中的复数序列，具体如下：

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}, \quad 0 \le k \le N-1 \tag{3}$$

其中， W_N^{nk} 是旋转因子。根据欧拉公式，可以将其展开为以下形式：

$$W_N^{nk} = e^{-\frac{2\pi i}{N} nk} = \cos\left(\frac{2\pi}{N} kn\right) - i \cdot \sin\left(\frac{2\pi}{N} kn\right) \tag{4}$$

其中， i 是复数的虚数单位。 X_k 是表示频率为 $\omega_k = 2\pi k/N$ 的信号频谱的复数。DFT 通过将一系列数值分解为不同频率的分量来获得。请注意，DFT 是时域和频域之间的一一映射操作。频率表示序列 $\{X_k\}_{k=0}^{N-1}$ 可通过逆离散傅里叶变换（IDFT）转换到时间特征域，其公式如下：

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-nk} \tag{5}$$

对于实输入 x_n，已证明其离散傅里叶变换具有共轭对称性， $i.e.$， $X_k = X_{N-k}^*$，其中 X_{N-k}^* 表示 X_{N-k}的共轭。这一性质意味着，仅凭离散傅里叶变换的一半即可恢复实离散信号 x_n $\{X_k\}_{k=0}^{\lceil N/2 \rceil}$ [6]。得益于这一性质，快速傅里叶变换（FFT）算法 [12], [13] 被广泛用于计算离散傅里叶变换，可将计算复杂度从 $\mathcal{O}(N^2)$ 降至 $\mathcal{O}(N \log N)$。因此，在本文中，我们通过快速傅里叶变换将序列行为转换到频域，并将其记为 $\mathcal{F}(\cdot)$。与逆离散傅里叶变换类似，逆快速傅里叶变换（IFFT）（记为 $\mathcal{F}^{-1}(\cdot)$）也用于高效地将频率特征转换回时域。

C. 卷积定理

近年来，频域卷积受到了越来越多的关注，为深度学习领域开辟了另一种研究视角。根据卷积定理，时域中的循环卷积等价于频域中的乘法。这一性质直接有助于我们设计滤波器，以捕获不同的频率分量。在离散傅里叶变换的背景下，卷积定理可以证明如下。

给定隐式周期信号 $x[n]$，在此条件下 $x[n]$ 与 $f[n]$ 的卷积称为循环卷积，可用如下数学公式表示：

$$f[n] * x[n] = \sum_{m=0}^{N-1} f[n] x_N[n-m] \tag{6}$$

其中，“*”表示循环卷积， $x_N[\cdot]$ 是 $\{x_n\}_{n=0}^{N-1}$的周期延拓：

$$x_N[n-m] \overset{\text{def}}{=} \sum_{k=-\infty}^{\infty} x[n-kN] = \sum_{k=-\infty}^{\infty} x[n+kN] \tag{7}$$

这是两个具有相同周期的周期函数之间的循环卷积的特例。我们有以下推导：

$$\begin{aligned}
\mathcal{F}(f[n] * x[n]) &= \mathcal{F}\Big(\sum_{m=0}^{N-1} f[m] x_N[n-m]\Big) \\
&= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f[m] x_N[n-m] W_N^{nk} \\
&= \sum_{u=-m}^{N-1-m} \sum_{m=0}^{N-1} f[m] x_N[u] W_N^{(u+m)k} \\
&= \sum_{m=0}^{N-1} f[m] W_N^{mk} \sum_{u=-m}^{N-1-m} x_N[u] W_N^{uk} \\
&= \sum_{m=0}^{N-1} f[m] W_N^{mk} \sum_{u=0}^{N-1} x[u] W_N^{uk} \\
&= \mathcal{F}(f[n]) \cdot \mathcal{F}(x[n])
\end{aligned}$$

总之，给定变换后的频率特征 $X[k]$:

$$y[n] = f[n] * x[n] = \mathcal{F}^{-1}(\mathcal{F}(f[n]) \cdot X[k]) \tag{8}$$

在本文中， $\mathcal{F}(f[n])$ 是可学习滤波器， $y[n]$ 是经过滤波后转换回时域的特征。

III. 方法

本节介绍用于序列推荐的对比增强滑动滤波器混合架构（命名为**SLIME4Rec**）。如图 2 所示，SLIME4Rec 是一种无需注意力机制的 Transformer架构，其中每个模块以滤波器混合层开始，以前馈层结束。本质上，我们使用由

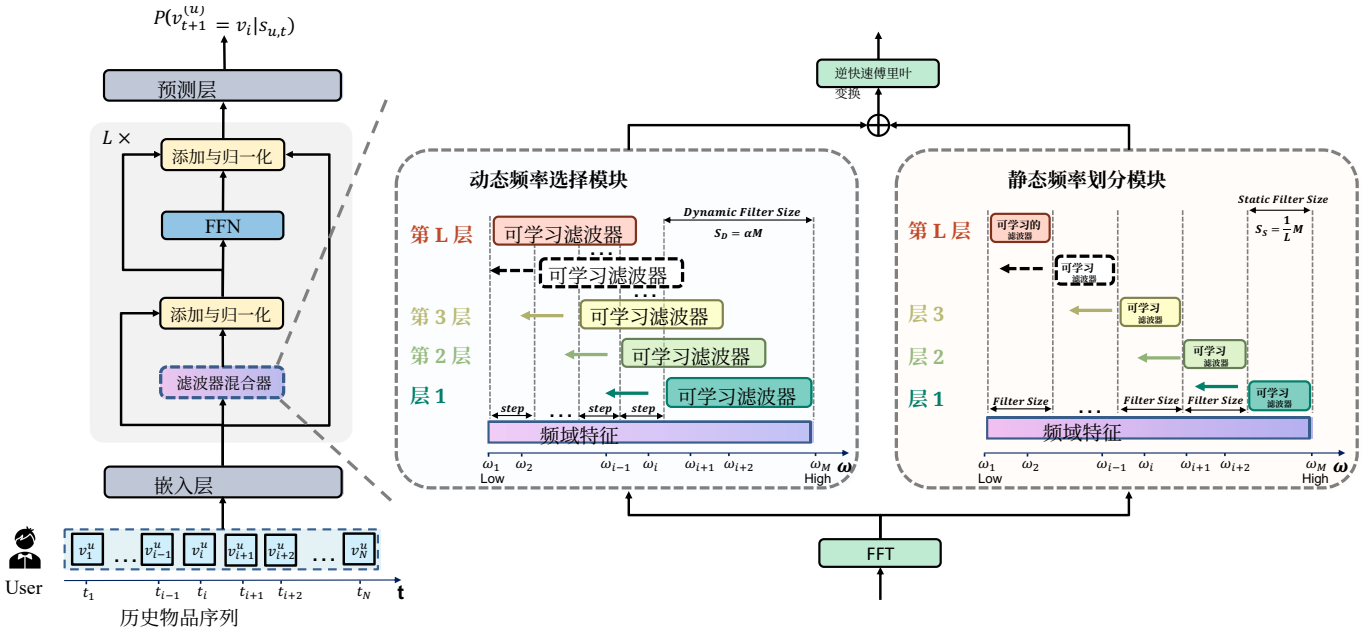


图 2。SLIME4Rec 的模型架构类似于 Transformer 编码器。它首先通过嵌入层，利用位置嵌入生成物品嵌入；随后，不同于自注意力层，它通过滤波器混合层从频域中提取用户偏好。滤波器混合器的详细结构如右侧所示，由两个模块组成： *i.e.*动态频率选择（DFS）模块和静态频率拆分（SFS）模块。为简化起见，我们将不同层中的所有可学习滤波器绘制在同一个模块中，以展示频率斜坡结构在两个模块上的作用；该结构看起来就像滤波器在不同层的频域特征上滑动。最后，预测层会计算所有候选物品的推荐分数。

动态频率选择（DFS）和静态频率分割（SFS）模块组成的滤波器混合层，替换 Transformer 编码器中的自注意力层。随后，我们将详细讨论这两个模块上的频率斜坡结构以及训练目标。

A. 嵌入层

序列推荐侧重于对隐式反馈的用户行为序列进行建模，该序列在 SR 中表现为物品 ID 列表。对于 $\mathcal{V}$ 中给出的所有物品，我们为所有物品创建一个物品嵌入矩阵 $\mathbf{M}^{\mathcal{V}} \in \mathbb{R}^{|\mathcal{V}| \times d}$，其中$d$表示嵌入维度。用户行为序列 $[v_1^{(u)}, v_2^{(u)}, ..., v_{|s_u|}^{(u)}]$ 的嵌入表示为：

$$\mathbf{E} = [\mathbf{e}_1^{(u)}, \mathbf{e}_2^{(u)}, ..., \mathbf{e}_n^{(u)}], \mathbf{e}_k^{(u)} = \text{LookUp}(v_k^{(u)}, \mathbf{M}^{\mathcal{V}}) \quad (9)$$

其中，LookUp(·, ·) 从嵌入矩阵中检索物品嵌入。

位置嵌入用于在保持物品原始嵌入维度的同时，添加更多位置信息，从而使我们的模型对物品的位置更加敏感。我们还执行层归一化和丢弃 [14] 操作，以稳定训练过程。因此，我们通过以下方式生成序列表示 $\mathbf{E} \in \mathbb{R}^{N \times d}$：

$$\mathbf{E} = \text{Dropout}(\text{LayerNorm}(\mathbf{E} + \mathbf{P})) \quad (10)$$

B. 滤波器混合器

为了将输入物品表示矩阵$\mathbf{E}$ 转换到频域，我们首先沿物品维度执行快速傅里叶变换：

$$\mathbf{H}^0 = \mathbf{E} \quad (11)$$

$$\mathcal{F}(\mathbf{H}^l) \rightarrow \mathbf{X}^l \in \mathbb{C}^{M \times d} \quad (12)$$

其中，$\mathbf{H}^l \in \mathbb{R}^{N \times d}$ 是第 l层的时间特征，$\mathcal{F}(\cdot)$表示一维快速傅里叶变换。请注意，$\mathbf{X}^l$ 是一个复数张量，表示$\mathbf{H}^l$的频谱。如第 II-B 节所述，由于频域中的共轭对称性质，离散傅里叶变换的一半包含了时域中周期性特征的全部信息。 M 计算如下：

$$M = \lceil N/2 \rceil + 1 \quad (13)$$

经过傅里叶变换后，我们得到 M 个频率分量 $\{\mathbf{X}_k^l\}_{k=0}^{M-1}$，来自 $\mathbf{X}^l \in \mathbb{C}^{M \times d}$。频率 $\omega_k = 2\pi k/N$ 用于描述这些频率分量在频域中沿 ω轴的位置。为了对频域特征 $\mathbf{X}_k^l$建模，我们提出了滤波器混合块，每个块由两个模块组成： *i.e.*、DFS 模块和 SFS 模块，以两种不同的方式捕获频率信息。

1）动态频率选择模块：在 SLIME4Rec 中，我们没有保留所有频率分量，而是设计了一个模块，用于在频域中动态选择特定频率的特征。正如第 II-C 节所证明的，频域中的滤波器操作等同于循环卷积，它与 Transformer架构中的自注意力机制一样具有全局感受野，并能捕捉物品序列中的周期性特征。通过乘以相应的可学习滤波器 $\mathbf{W}_D^l \in \mathbb{C}^{M \times d}$，我们可以动态选择频率特征，如下所示：

$$\widehat{\mathbf{X}}_D^l = \mathbf{X}^l \odot \sigma_{ij}^l(\omega) \odot \mathbf{W}_D^l \quad (14)$$

其中，$\odot$ 表示逐元素乘法，l 表示第 i 层。如图 2 所示，$\mathbf{X}^l$ 包含 M 个频率分量，我们使用 ω 表示每个频率分量的索引。因此，σ_{ij}^l（ω）定义如下：

$$\sigma_{ij}^l(\omega) = \mathbb{I}(i^l \le \omega \le j^l), \quad 0 \le i < j \le M \tag{15}$$

其中，$\mathbb{I}$ 是一个指示函数，由 i 和 j决定。滤波器 $\mathbf{W}_D^l$ 是可学习的，因为它由复数参数组成，这些参数可以通过随机梯度下降（SGD）进行优化，从而在频域中自适应地捕获序列模式的特征。当将可学习滤波器在频域中的感受野设置为固定长度的窗口时，周围的其他频率分量将被屏蔽，以消除对无关频率的依赖。因此，在捕获当前特定频率范围的特征时，可学习滤波器将直接禁用所有不属于当前范围的特征。

2）频率斜坡结构：如图2所示，为了使每一层的滤波器覆盖特定范围，我们进一步设计了一种频率斜坡结构，使可学习滤波器能够在不同层的频谱上滑动。具体而言，对于动态频率选择模块，我们将 $\sigma_{ij}^l(\omega)$ 定义为 $\leftarrow$ $\sigma_D^l(\omega)$，其中 $\leftarrow$ 表示滤波器滑动的方向，$i.e.$，从高频到低频（低 $\leftarrow$ 高），而 $\rightarrow$表示从低频到高频的方向（低 $\rightarrow$高）。其定义如下：

$$\overleftarrow{\sigma_D^l}(\omega) = \begin{cases} 1 & \text{if } \omega \in [\overleftarrow{i_D^l}, \overleftarrow{j_D^l}] \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$\overleftarrow{i_D^l} = M * (1-\alpha) - (l * step) \tag{17}$$

$$\overleftarrow{j_D^l} = M - (l * step) \tag{18}$$

其中，α 是动态滤波器大小 S_D 相对于频率特征范围 M的比例：

$$\alpha = \frac{S_D}{M} \tag{19}$$

给定 L为滤波器混合块的数量：

$$step = \frac{(1-\alpha)M}{L-1} \tag{20}$$

通过这种方式，不同的层可以捕获多样的频率特性。

当 α 设置为 1 时，SLIME4Rec 的动态滤波器大小与 FMLP-Rec [5],相同，其 Eq. (20) 的值等于 0，且滤波器不会在频谱上滑动。对于一个包含 L 层的模型，我们得到一个长度为 L 的指示$\leftarrow$函数列表，可表示为 $\boldsymbol{\sigma}_D\ \omega\ =$ ( ) $\leftarrow\quad\leftarrow\quad\leftarrow$ $\{\sigma_D^0(\omega)\ \sigma_D^1(\omega)\ ...,\ \sigma_D^{L-1}(\omega)\}$。可以很容易地证明，$\rightarrow\qquad\leftarrow\boldsymbol{\sigma}_D(\omega)\ =\ \text{inverse}(\boldsymbol{\sigma}_D(\omega))$，其中 inverse(·) 表示将输入顺序反转。考虑到顶层更关注对低频全局信息的建模，而底层对于捕捉高频细节更为重要，因此我们选择从高频到低频的方向，并将 Eq.(14) 改写为：

$$\widehat{\mathbf{X}}_D^l = \mathbf{X}^l \odot \overleftarrow{\sigma_D^l}(\omega) \odot \mathbf{W}_D^l \tag{21}$$

3）静态频率划分模块：静态频率划分模块旨在弥补动态频率选择模块中可能存在的部分频域特征缺失问题。具体而言，当动态滤波器大小为 $S_D < step$时，动态滤波器将无法覆盖当前步与前一步之间的频率特征，$i.e.$:

$$\begin{aligned} \alpha M &< \frac{(1-\alpha)M}{L-1} \\ (L-1)\alpha &< (1-\alpha) \\ \alpha &< \frac{1}{L} \end{aligned}$$

因此，我们将静态滤波器大小的相对比例定义为 β:

$$\beta = \frac{1}{L} = \frac{S_S}{M} \tag{22}$$

其中，S_S 是静态滤波器的大小。因此，σ_S^l（ω）的滑动 $\leftarrow$索引 i 和 j 可表示为：

$$\overleftarrow{i_S^l} = M * (1-\beta) - (l * S_S) \tag{23}$$

$$\overleftarrow{j_S^l} = M - (l * S_S) \tag{24}$$

之后，通过乘以静态可学习滤波器 $\mathbf{W}_S^l \in \mathbb{C}^{M \times d}$，我们可以修改频谱，以捕获动态滤波器无法捕获的缺失频率特征，即使 $\alpha < \beta$。

$$\widehat{\mathbf{X}}_S^l = \mathbf{X}^l \odot \overleftarrow{\sigma_S^l}(\omega) \odot \mathbf{W}_S^l \tag{25}$$

最后，我们将动态滤波器提取的特征与静态滤波器提取的特征混合，并采用逆快速傅里叶变换将调制后的频谱 $\widehat{\mathbf{X}}_{mix}^l$ 转换回时域：

$$\widehat{\mathbf{X}}_{mix}^l = (1-\gamma)\widehat{\mathbf{X}}_D^l + \gamma\widehat{\mathbf{X}}_S^l \tag{26}$$

$$\widehat{\mathbf{H}}^l \leftarrow \mathcal{F}^{-1}(\widehat{\mathbf{X}}_{mix}^l) \in \mathbb{R}^{N \times d} \tag{27}$$

其中 $\mathcal{F}^{-1}$ 表示一维逆快速傅里叶变换，它将复数张量转换为实数张量。为了解决梯度消失和训练不稳定的问题，我们还采用了跳跃连接、层归一化和丢弃操作，如下所示：

$$\widehat{\mathbf{H}}^l = \text{LayerNorm}(\mathbf{H}^l + \text{Dropout}(\widehat{\mathbf{H}}^l)) \tag{28}$$

C. 逐点前馈网络

与自注意力机制类似，滤波器混合器中的频域卷积仍是一种线性运算，无法建模复杂的非线性关系。为了使模型具备时域中不同维度之间的非线性特征，我们还在每个滤波器混合器之后添加了一个前馈网络，该网络由采用 GELU 激活函数的多层感知机构成。该过程的

逐点前馈神经网络（前馈神经网络）定义如下：

$$\tilde{\mathbf{H}}^{l}=\mathrm{FFN}(\widehat{\mathbf{H}}^{l})=(\mathrm{GELU}(\widehat{\mathbf{H}}^{l}\mathbf{W}_{1}+\mathbf{b}_{1}))\mathbf{W}_{2}+\mathbf{b}_{2} \quad (29)$$

其中 $\mathbf{W}_1,\mathbf{W}_2\in\mathbb{R}^{d\times d}$和 $\mathbf{b}_1$,$\mathbf{b}_2\in\mathbb{R}^{1\times d}$是可学习参数。为防止过拟合，我们在每个隐藏层上方添加一个丢弃层，并在输出$\mathbf{H}^{l+1}$上采用密集残差连接结构再次执行层归一化操作，如下所示：

$$\mathbf{H}^{l+1}=\mathrm{LayerNorm}(\mathbf{H}^{l}+\widehat{\mathbf{H}}^{l}+\mathrm{Dropout}(\tilde{\mathbf{H}}^{l})) \quad (30)$$

D. 预测层

经过 L个滤波器混合块，自适应地以分层方式提取先前交互物品的行为模式信息后，我们得到代表用户偏好的物品最终组合表示。基于 L层编码器的输出 $\mathbf{H}^{L}$，我们选取$\mathbf{H}^{L}=$[中的最后一个隐藏向量 $\mathbf{h}_t^L$ $\mathbf{h}_0^L,\mathbf{h}_1^L,\cdots,\mathbf{h}_t^L]$ 作为该用户序列的用户表示，并计算候选物品 v 的推荐概率，以预测用户采纳该物品的可能性。具体而言，对应的预测概率$\hat{}\mathbf{y}$ 可由下式生成：

$$\hat{\mathbf{y}}=\mathrm{softmax}((\mathbf{M}^{\mathcal{V}})^{\top}\mathbf{h}^{L}) \quad (31)$$

其中 $\hat{}\mathbf{y}\in\mathbb{R}^{|\mathcal{V}|}$。因此，我们期望用户 u 采纳的真实物品 v能够获得更高的分数 $\hat{}y_i$。为优化模型参数，我们采用交叉熵损失。序列推荐的目标函数可表示为：

$$\mathcal{L}_{Rec}=-\sum_{i=1}^{|\mathcal{V}|}y_i\log(\hat{y}_i)+(1-y_i)\log(1-\hat{y}_i) \quad (32)$$

E. 对比学习

为增强动态频率选择和静态频率划分模块中滤波器的训练，使其能够捕获用户序列的核心频率分量，我们采用多任务训练策略，联合优化主要推荐损失和辅助对比损失。这有助于缩小同一用户不同增强视图之间的差异，并扩大不同用户所生成的增强序列之间的差异。

尽管以往的增强方法 [16]（包括物品裁剪、掩码和重新排序）有助于提升序列推荐模型的性能，但数据级增强无法保证较高的语义相似性 [17]。我们不采用常见的数据增强，而是让同一用户的序列经过网络两次，并对频率分量进行建模：在逆傅里叶变换之前，将从 DFS 和 SFS 提取的频率特征混合，从而构造更难的正样本。由于网络模块中存在不同的丢弃层，我们会得到两个数值特征不同但语义相似的输出视图。此外，为了增加对比学习的监督信号，我们遵循 DuoRec [17]，使用目标相同的序列作为监督对比学习的正样本。训练批次中的所有其他增强样本都被视为负样本，以便为一对增强样本（表示为 neg）高效构造负样本。

对于批次 $\mathcal{B}$，对比正则化定义如下：

$$\mathcal{L}_{\mathrm{CLReg}}=\mathcal{L}_{\mathrm{CLReg}}(\mathbf{h}',\mathbf{h}'_s)+\mathcal{L}_{\mathrm{CLReg}}(\mathbf{h}'_s,\mathbf{h}') \quad (33)$$

$$\mathcal{L}_{\mathrm{CLReg}}(\mathbf{h}',\mathbf{h}'_s)=-\log\frac{\exp(\mathrm{sim}(\mathbf{h}',\mathbf{h}'_s))}{\sum_{neg}\exp(\mathrm{sim}(\mathbf{h}',\mathbf{h}_{neg}))} \quad (34)$$

其中$\mathbf{h}'$ 和$\mathbf{h}'_s$ 分别表示无监督和有监督的增强视图，定义如下：

$$\mathbf{h}'=\mathbf{H}^{L'}[-1],\quad \mathbf{h}'_s=\mathbf{H}_s^{L'}[-1] \quad (35)$$

因此，SLIME4Rec 的总体目标为：

$$\ell=\ell_{\mathrm{Rec}}+\lambda\ell_{\mathrm{CLReg}} \quad (36)$$

其中 λ 是用于控制对比正则化强度的超参数。

F. 复杂度分析

与依赖自注意力机制的基于 Transformer 的模型不同，SLIME4Rec 是一种结合动态滤波器和静态滤波器的无注意力架构。传统自注意力的计算复杂度为 $\mathcal{O}($ n^2d+nd^2)，其中n 表示用户的序列长度，d 表示隐藏维度。相比之下，借助 FFT [18]，滤波器混合模块的时间复杂度可降低至 $\mathcal{O}(n\log(nd))$。我们提出的动态频率选择模块和静态频率划分模块中逐元素乘积的时间复杂度为 $\mathcal{O}(nd)$，而前馈网络的时间开销为 $\mathcal{O}(\frac{nd^2}{2})$。因此，考虑到对比目标和层数，SLIME4Rec 的总时间复杂度为 $\mathcal{O}(3L(n\log nd+nd+\frac{nd^2}{2}))$，与输入序列长度 n的对数线性复杂度成正比。

IV. 实验

在本节中，我们首先简要介绍实验中使用的数据集、八种基线方法、评估指标以及实验设置中的实现细节。随后，我们将所提出的模型 SLIME4Rec 与最先进的基线方法进行比较，展示各模型的实验结果并分析其原因。此外，我们还探究模型组件和参数对模型 SLIEM4Rec 性能的影响。最后，我们讨论 SLIME4Rec 对合成噪声的鲁棒性，并展示动态滤波器和静态滤波器的可视化结果。具体而言，为了研究 SLIME4Rec 的有效性，我们开展实验，尝试回答以下问题：

- **RQ1** SLIME4Rec 的表现是否优于最先进的基线方法?
- **RQ2** 动态与静态滤波器模块对 SLIME4Rec 有何影响?
- **RQ3** 滤波器滑动方向对模型性能有何影响?

- **RQ4** 动态滤波器尺寸比 α、最大物品长度 N、隐藏维度 d 以及滤波器混合模块数量 L 等参数如何影响 SLIME4Rec 的有效性?
- **RQ5** 所提出的模型对合成噪声有多鲁棒?

A. 数据集

我们在从真实平台收集的五个公开数据集上开展实验，以全面评估 SLIME4Rec。这些数据集在应用场景、规模和稀疏度方面各不相同，常用于测试序列推荐方法。表 I 报告了五个数据集预处理后的主要统计信息。下面将详细介绍各个数据集。

- **美妆、服装和运动** [19] 是从亚马逊这一全球最大的电子商务平台之一收集的三个数据集。它们按照亚马逊上最高层级的产品类别划分。高稀疏度以及丰富多样的评分评论类别是亚马逊数据集的特点。我们遵循 DuoRec 中的设置，采用“美妆”“服装、鞋履和珠宝”以及“运动与户外”这三个类别。
- **MovieLens-1M** [20] 基于从非商业电影推荐网站 MovieLens 收集的电影评论。*ML-1M* 中的交互数量约为 100 万。
- **Yelp** [17] 是一个知名的商业推荐数据集。由于交易规模的限制，我们仅使用2019年1月1日之后的交易记录。

与 [21], [22], 一致，我们也采用 5核设置，滤除交互次数少于 5 次的用户。

表 I 预处理后数据集的统计信息。

规格。	美妆	服装	运动	ML-1M	Yelp
# 用户数	22,363	39,387	35,598	6,041	30,499
# 物品数	12,101	23,033	18,357	3,417	20,068
# 平均长度	8.9	7.1	8.3	165.5	10.4
# 行为数	198,502	278,677	296,337	999,611	317,182
稀疏度	99.93%	99.97%	99.95%	95.16%	99.95%

B. 评估指标

在我们的评估中，我们采用留一法策略，即将用户交互的最后一个物品留作测试，将倒数第二个物品留作验证，其余物品则用于训练。按照 [23], 的建议，我们在整个物品集合上对预测分数进行排序，而不使用负采样，从而确保评估过程不受偏差影响。我们采用多种评估指标来衡量性能，包括所有数据集上的 K 命中率（HR@K）和 K 归一化折损累计增益（NDCG@K）。需要注意的是，这两个指标的值越高，表示性能越好。HR@K 是一种基于召回率的指标，用于衡量真实物品出现在前 K 个推荐列表中的平均比例。NDCG@K 是一种考虑位置的指标，用于逐个位置评估前 K 个推荐列表的排序质量。本文报告了 $K = 5$ 和 10 对应的 HR 和 NDCG。

C. 基线模型

为展示所提出模型的有效性，我们将 SLIME4Rec 与应用最广泛、最先进的方法进行比较，包括朴素矩阵分解方法（BPR-MF）、采用不同神经网络架构的序列模型（GRU4Rec、Caser、SASRec、BERT4Rec、FMLP-Rec），以及使用自监督学习对比目标函数的竞争性序列模型（CL4SRec、ContrastVAE、CoSeRec 和 DuoRec）。所有基线方法如下所述:

BPR-MF [1] 是一种经典的非序列方法，通过隐式反馈学习个性化排序，并利用成对贝叶斯个性化排序（BPR）损失函数优化矩阵分解。

GRU4Rec [2] 是首个将门控循环单元（GRU）应用于用户行为序列建模的序列推荐模型。GRU 的最终隐藏特征被视为输入序列的潜在表示。

Caser [3] 是一种基于卷积神经网络的方法，通过在时间维度上使用水平和垂直卷积滤波器来捕捉用户活动的局部动态模式。

SASRec [4] 是采用多头自注意力机制的基于 Transformer 的模型之一。凭借强大的长期依赖建模能力，SASRec 已成为序列推荐中的一个强大基线模型。

BERT4Rec [22] 采用与自然语言处理（NLP）中的掩码语言模型类似的掩码物品训练方案。其骨干网络是双向自注意力机制。

FMLP-Rec [5] 是一种全多层感知机模型，采用可学习滤波器增强模块来消除嵌入矩阵中的噪声。与传统的序列建模技术相比，这种结构有显著差异。

CL4SRec [16] 采用序列裁剪、掩码和重新排序这三种数据增强技术，为辅助对比学习任务生成同一用户交互序列的不同对比表示。

ContrastVAE [24] 是一种基于 VAE 的方法，结合了数据增强、模型增强和变分增强这三种增强策略进行对比学习。

CoSeRec [25] 利用物品之间的相关性，提高对比学习框架中数据增强的鲁棒性。

DuoRec [17]采用无监督的模型级增强和有监督的语义正样本进行对比学习。它是最新且性能强劲的序列推荐基线模型。

D. 实现细节

我们使用 PyTorch 实现了 SLIME4Rec 模型。所有实验均在一台配备有.....的 Linux 服务器上进行。

表 II 总体性能。粗体分数表示所有方法中的最佳结果。下划线分数表示所有方法中的次优结果。SLIME4Rec 在所有评估指标上均以明显优势超越所有基线模型。

数据集	指标	BPR-MF	GRU4Rec	Caser	SASRec	BERT4Rec	FMLP-Rec	CL4SRec	ContrastVAE	CoSeRec	DuoRec	SLIME4Rec	提升
美妆	HR@5	0.0120	0.0164	0.0259	0.0365	0.0193	0.0398	0.0401	0.0422	0.0537	0.0546	**0.0621**	13.74%
	HR@10	0.0299	0.0365	0.0418	0.0627	0.0401	0.0632	0.0683	0.0681	0.0752	0.0845	**0.0910**	7.69%
	NDCG@5	0.0040	0.0086	0.0127	0.0236	0.0187	0.0258	0.0223	0.0268	0.0361	0.0352	**0.0396**	12.50%
	NDCG@10	0.0053	0.0142	0.0253	0.0281	0.0254	0.0333	0.0317	0.0350	0.0430	0.0443	**0.0489**	10.38%
服装	HR@5	0.0067	0.0095	0.0108	0.0168	0.0125	0.0126	0.0168	0.0161	0.0175	0.0193	**0.0225**	16.58%
	HR@10	0.0094	0.0165	0.0174	0.0272	0.0208	0.0206	0.0266	0.0247	0.0279	0.0302	**0.0343**	13.58%
	NDCG@5	0.0052	0.0061	0.0067	0.0091	0.0075	0.0082	0.0090	0.0105	0.0095	0.0113	**0.0126**	11.50%
	NDCG@10	0.0069	0.0083	0.0098	0.0124	0.0102	0.0107	0.0121	0.0133	0.0131	0.0148	**0.0164**	10.81%
运动	HR@5	0.0092	0.0137	0.0139	0.0218	0.0176	0.0218	0.0227	0.0225	0.0287	0.0326	**0.0373**	14.42%
	HR@10	0.0188	0.0274	0.0231	0.0336	0.0326	0.0344	0.0374	0.0366	0.0437	0.0498	**0.0565**	13.45%
	NDCG@5	0.0040	0.0096	0.0085	0.0127	0.0105	0.0144	0.0129	0.0151	0.0196	0.0208	**0.0243**	16.83%
	NDCG@10	0.0051	0.0137	0.0126	0.0169	0.0153	0.0185	0.0197	0.0184	0.0242	0.0262	**0.0305**	16.41%
ML-1M	HR@5	0.0078	0.0763	0.0816	0.1087	0.0733	0.1356	0.1147	0.1406	0.1262	0.2038	**0.2237**	9.76%
	HR@10	0.0162	0.1658	0.1593	0.1904	0.1323	0.2118	0.1975	0.2220	0.2212	0.2946	**0.3156**	7.13%
	NDCG@5	0.0052	0.0385	0.0372	0.0638	0.0432	0.0870	0.0662	0.0895	0.0761	0.1390	**0.1567**	12.73%
	NDCG@10	0.0079	0.0671	0.0624	0.0910	0.0619	0.1113	0.0928	0.1157	0.1021	0.1680	**0.1864**	10.95%
Yelp	HR@5	0.0127	0.0152	0.0156	0.0161	0.0186	0.0179	0.0216	0.0177	0.0241	0.0441	**0.0516**	17.01%
	HR@10	0.0245	0.0263	0.0252	0.0265	0.0291	0.0304	0.0352	0.0294	0.0395	0.0631	**0.0766**	21.39%
	NDCG@5	0.0760	0.0104	0.0096	0.0102	0.0118	0.0113	0.0130	0.0113	0.0151	0.0325	**0.0359**	10.46%
	NDCG@10	0.0119	0.0137	0.0129	0.0134	0.0171	0.0153	0.0185	0.0147	0.0205	0.0386	**0.0439**	13.73%

32GB NVIDIA Tesla V100 图形处理器。对于基线模型，我们采用原论文中报告的最佳超参数设置；由于这些工作所使用的数据集和评估指标与我们的完全一致，因此如果有可用的重新实现结果，我们将直接报告这些结果。

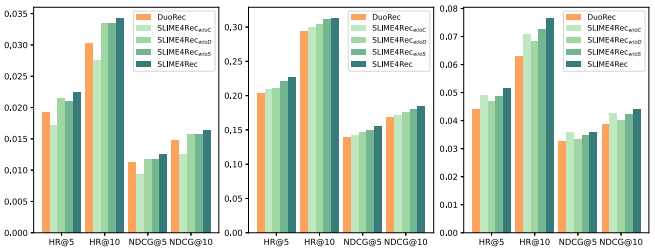
模型采用 Adam 优化器进行优化，学习率设为 0.001。前馈网络的维度和物品嵌入维度均设为 64。嵌入矩阵和滤波器混合块的丢弃率从 {0.1、0.2、0.3、0.4、0.5} 中选择。得益于频率斜坡结构，我们堆叠了更多滤波器层，并在 {2, 4, 8} 中搜索滤波器混合层的总数量。由于 $\mathcal{O}(N \log N)$ 的复杂度，最大序列长度 N 可从 {25, 50, 75, 100} 中选择，而不会增加过多的计算成本。对于动态频率选择模块，我们将α 设为超参数，并以 0.1 为步长从 [0, 1] 中进行选择。所有这些参数均在验证集上进行调优。我们报告每个模型在其最优超参数设置下的结果。我们的模型实现可在 https://github.com/sudaada/SLIME4Rec 找到。

E. 推荐性能比较

为验证我们模型 SLIME4Rec 的序列推荐性能，我们将其与其他最先进的方法进行比较（**RQ1**）。表 II 展示了各模型的详细评估结果，其中，我们的 SLIME4Rec 和性能最强的基线方法分别以粗体和下划线标出。根据这些结果，我们可以得出以下观察和结论。

首先，毋庸置疑，非序列推荐方法BPR-MF在所有数据集上的表现都最低，因为它忽略了序列信息，这表明挖掘用户行为序列中隐藏的序列模式至关重要。其次，所有神经网络方法（GRU4Rec、Caser、SASRec、BERT4Rec和FMLP-Rec）的表现都显著优于传统方法，证明了基于神经网络的模型能够捕捉复杂的序列模式以进行推荐。例如，GRU4Rec利用循环结构捕捉用户的总体偏好，因此表现优于BPR-MF。由此可得，利用序列信息能够提升性能。此外，Caser在大多数数据集上的表现都优于GRU4Rec，这表明卷积核能够有效捕捉更复杂的行为模式。与此前基于RNN和基于CNN的方法相比，先进的基于Transformer的方法（例如SASRec）大幅提升了性能。这表明，自注意力机制在SR中对交互序列的建模能力更强。例如，BERT4Rec采用掩码项目预测目标。尽管这类任务能够为模型提供有意义的信号，但其表现并不稳定，因为掩码项目预测与推荐任务的契合度不高。近来，滤波器增强型多层感知机结构通过衰减频域中的噪声，在大多数数据集上取得了与SASRec几乎相同、甚至更好的表现。

第三，与原始方法相比，加入辅助自监督学习任务的模型取得了不错的提升。例如，CL4SRec 通过三种数据增强提升了 SASRec 的性能。ContrastVAE 使用变分增强创建更强的增强视图，以进行对比学习，其结果与 CL4SRec 相当，甚至更好。CoSeRec 沿用 CL4SRec 的流程，利用物品相关性增强对比序列。这表明，对比学习



(a) 服装 (b) ML-1M (c) Yelp

图 3. 不同滤波器模块和对比学习的消融研究。

可以成为序列推荐任务中的一种有效方法。DuoRec 通过模型增强和语义增强，大幅超越所有基线方法，这验证了有监督对比学习与无监督对比学习相结合的有效性。

最后，我们的模型在稀疏和稠密数据集上的所有指标均显著优于其他竞争方法，证明了模型的优越性。具体而言，与最强基线方法相比，SLIME4Rec 在亚马逊运动和 ML-1M 上的 NDCG@5 分别提升了 16.58% 和 12.73%，在 Yelp 上的 HR@10 提升了 21.39%。这一观察结果证明了 SLIME4Rec 的有效性，并表明将用户序列从时域转换到频域，再在频率分量上乘以尺寸适当的可学习滤波器，有望提取有助于实现准确推荐的有效信息。

F. 消融研究

如第III节所述，SLIME4Rec 的骨干架构由两个主要的新颖组件构成：*i.e.*动态与静态滤波器模块以及频率斜坡结构。为了更好地理解我们方法的设计原理，我们针对所提出的模型开展了消融研究，以分别探究这些组件的有效性（**RQ2**）。受篇幅限制，我们仅展示 HR@5 和 NDCG@5 方面的分析结果。在其他指标方面，我们也获得了类似的实验结果。

表III 不同滤波器模块设计的性能（HR@5 和 NDCG@5）。

层 DFS	SFS	美妆	服装	运动	ML-1M	Yelp
		HR@5 归一化折损累计增益	HR@5 归一化折损累计增益	HR@5 归一化折损累计增益	HR@5 归一化折损累计增益	HR@5 归一化折损累计增益
L=2	α=0.3 ✗	0.0588 0.0360	0.0209 0.0116	0.0357 0.0227	0.1876 0.1287	0.0449 0.0317
	α=O_o 3 β=O_o 5	**0.0604 0.0370**	**0.0210 0.0118**	**0.0358 0.0228**	**0.1907 0.1312**	**0.0454 0.0320**
L=4	α=0.2 ✗	0.0594 0.0373	0.0213 0.0121	0.0367 0.0234	0.1874 0.1273	0.0467 0.0327
	α=O_o 2 β=0.25	**0.0599 0.0376**	**0.0217 0.0124**	**0.0369 0.0235**	**0.1879 0.1274**	**0.0481 0.0337**
L=8	α=0.1 ✗	0.0570 0.0371	0.0203 0.0120	0.0365 0.0232	0.1945 0.1357	0.0452 0.0312
	α=O_o 1 β=O_o 125	**0.0591 0.0379**	**0.0211 0.0128**	**0.0369 0.0239**	**0.2020 0.1384**	**0.0460 0.0327**

1）滤波模块设计研究：我们首先探究不同滤波器模块设计的影响，以及对比学习如何增强这些模块。为研究各组件对最终推荐性能的贡献，我们设计了 SLIME4Rec 的三个变体。从 SLIME4Rec 中移除对比学习、动态滤波器或静态滤波器后，模型分别退化为 SLIME4Rec$^{w/oC}$、SLIME4Rec $^{w/oD}$ 和 SLIME4Rec$^{w/oS}$。SLIME4Rec 同时包含动态与静态滤波器模块，是我们提出模型的完整版本。

具体而言，我们在三个数据集上进行了消融研究，结果如图3所示。从结果中可以观察到，移除任何组件都会导致性能下降，且 SLIME4Rec 的所有变体均优于 DuoRec，这证明了所提出模型的所有组件都有效且必要。此外，我们可以看到，SLIME4Rec 的表现优于所有仅包含单个滤波器模块的其他变体。这是因为动态和静态特征混合可以联合从不同感受野中学习序列模式，有望更好地捕捉用户偏好。

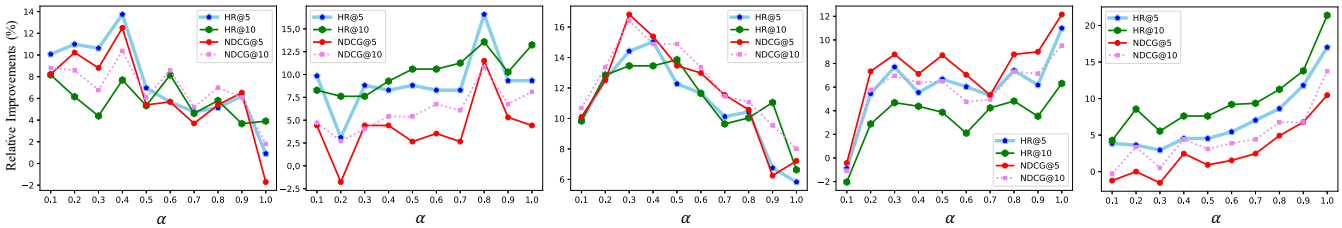
为了证明使用静态滤波器模块的必要性，我们在动态滤波器大小过小、无法捕获当前步的频率范围时进行了详细的消融研究。我们在表III中分析了局部感知的重要性。表III显示，尽管动态滤波器模块在捕获频率特征方面发挥着更关键的作用，但当 $\alpha < \beta$ 时，静态滤波器也发挥着重要的辅助作用。得益于同时使用动态与静态滤波器模块，SLIME4Rec 能够在每一层中完整保留频谱特征信息（当 $\alpha < \beta$ ），并收集不同层中的所有信息信号，同时不丢失任何频率范围的信息。

表 IV 频率斜坡结构不同滤波滑动模式的性能（HR@5 和 NDCG@5）。

滑动	DFS SFS	美妆	服装	运动	ML-1M	Yelp
		HR@5 NDCG@5	HR@5 NDCG@5	HR@5 NDCG@5	HR@5 NDCG@5	HR@5 NDCG@5
模式 1	← →	0.0577 0.0371	0.0216 0.0120	0.0360 0.0239	0.2086 0.1432	0.0486 0.0343
模式 2	→ ←	0.0563 0.0360	0.0214 0.0121	0.0361 0.0224	0.2104 0.1461	0.0489 0.0346
模式 3	→ →	0.0589 0.0371	0.0220 0.0123	0.0367 0.0233	0.2108 0.1455	0.0493 0.0343
模式 4	← ←	**0.0621 0.0396**	**0.0225 0.0126**	**0.0373 0.0243**	**0.2237 0.1567**	**0.0516 0.0359**

2）频率斜坡结构研究：SLIME4Rec 主要专注于提取频域特征，其最具创新性的结构之一是滑动滤波器。由于低频特征和高频特征在频域中分别分布于左侧和右侧，因此存在两个滑动方向。在动态模块和静态模块中，我们都采用滑动技术，将用户行为模式中不同频率的信息与频域特征进行融合。为进一步说明不同滑动模式对 SLIME4Rec 的影响，我们在五个数据集上开展了消融研究（**RQ3**）。如表 IV 所示，共有四种滑动模式。符号 $\leftarrow$ 表示滤波器从高频特征向低频特征滑动。符号 $\rightarrow$ 表示滤波器从低频特征向高频特征滑动。

我们注意到，滑动模式 4 的性能优于其他滑动模式。这表明，高频特征中的细节适合由较低层滤波器捕捉局部



(a) 美妆 (b) 服装 (c) 运动 (d) ML-1M (e) Yelp

图 4. 与最强基线模型 DuoRec 相比，不同动态滤波器大小下的相对提升 α .

频繁变化的用户兴趣。随着网络层逐渐加深，各种局部信息被逐步收集，以理解完整的全局交互序列。此外，模式 3 的性能排名第二。滑动模式 3 与模式 4 非常相似，但方向相反。模式 3 中的动态滤波器和静态滤波器分别在底层捕捉低频特征、在顶层捕捉高频信息，这与 [15] 中的结论不一致：较低层通常需要更多局部信息，而较高层则需要更多全局信息。

事实上，无论位于顶层还是底层，模式 1 和模式 2 中的动态滤波器与静态滤波器的滑动方向都相反且相互冲突。换言之，动态滤波器提取的低频信息会与静态滤波器提取的高频信息直接混合，反之亦然。因此，将这两种冲突的频率特征结合起来会导致次优结果。在滤波器模块中应用频率斜坡结构，使 SLIME4Rec 能够更好地权衡高频和低频信息，因为神经网络中不同位置的层在捕获特征细节方面发挥着不同的作用。SLIME4Rec 进一步在频率斜坡结构中从高频到低频对序列模式进行建模，这适用于深度神经网络的特征提取 [15]。

G. 超参数的影响

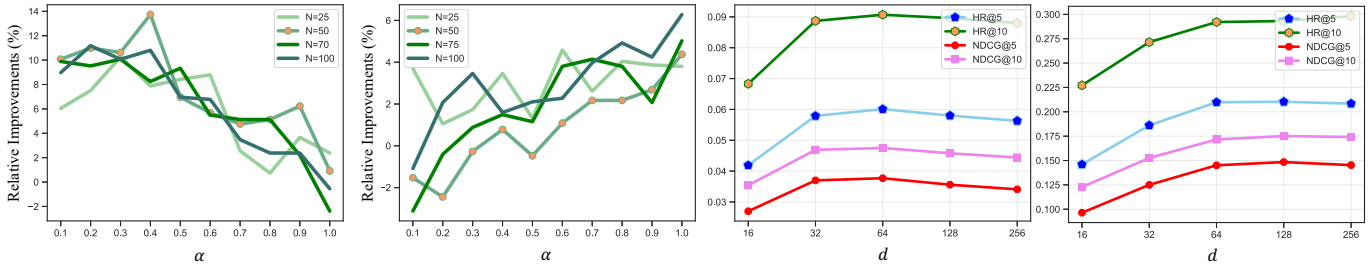
为了探究使用滑动滤波器在频域中捕获序列模式的有效性，我们研究了四个超参数——滤波器大小比 α、模型深度L、隐藏维度 d 以及用户序列的最大长度M——对 SLIME4Rec 性能的影响（**研究问题 4** ）。我们每次分析一个超参数，并将其余超参数固定为各自的最优设置。

1) 滤波器大小比的影响 α：所提出的 SLIME4Rec 模型的一项创新之处在于，我们让一个大小适当的滤波器在不同层的频域特征上滑动。因此，在我们的工作中，我们引入超参数 α 来控制滤波器大小，这会显著影响模型性能。我们在五个数据集上针对不同的 α 进行实验，并在图4中展示了该超参数的影响。

如图4所示，随着 α 增加，SLIME4Rec的性能起初开始提升，并逐渐达到峰值，此时 α 为在Amazon-Beauty上为0.4，在Amazon-Clothing上为0.8，在Amazon-Sports上为0.3。之后，性能开始下降。有趣的是，Sports数据集的最佳 α 仅为0.3，而Beauty数据集的最佳值为0.4。这表明，对于稀疏数据集，滑动滤波器大小无需设置得太大，而应重点关注重要频率特征所在的特定频带。例如，在Amazon数据集中，用户的重要频率分量更为集中，主要分布在低频区域 [5]。而在由各种电影构成的ML-1M等稠密数据集中，用户序列的频谱更加复杂，重要频率分量分散在不同频带中，因此用户在ML-1M数据集中的兴趣更加多样，也更难理解。为了捕捉用户更复杂的序列模式，动态滤波器结构需要在频域中使用更复杂的参数，并在时域中采用更大的感受野。值得注意的是，除 α 为0.1和1外，无论其取值如何，我们的方法SLIME4Rec始终优于DuoRec。然而，当 α 设置得过小（α = 0.1）时，推荐效果会不理想。

2) 输入序列长度的影响 N：如第III节所述， α 是动态滤波器大小 S_D 与频域特征范围 M 的比值，其中 M 取决于 N。为了研究沿序列长度维度使用大小适当的滑动滤波器所带来的益处，我们在不同的 N 下运行 SLIME4Rec。受篇幅限制，我们只报告 HR@5 的结果，其他指标也呈现出类似趋势。图 5(a) 和 (b) 展示了在使用不同输入序列长度时，HR@5 的相对提升如何随滤波器大小比 α 变化。对于 Beauty 数据集，将 SLIME4Rec 的 N 从 25 增加到 50，可持续提升模型性能；但当 N 进一步增大时，模型性能会下降。这些结果表明，任意增加 N 并不总能提升性能。然而，对于用户平均行为更多的稠密数据集 ML-1M，当动态滤波器大小比为 1 时，增加序列长度 N 有望提升性能，因为这样会纳入更多已有的交互物品。最后可以看出，尽管序列长度有所增加，动态滤波器大小比对序列长度的变化并不敏感。

3) 隐藏维度的影响 d：除了关于序列长度的实验外，我们还针对不同隐藏维度 d 开展了实验，以



(a) Beauty 上的 HR@5 (b) ML-1M 上的 HR@5 (c) Beauty (d) ML-1M

图 5. 在设置不同最大物品列表长度时，不同动态滤波器尺寸比的性能 α 以及不同隐藏维度 d下的性能。

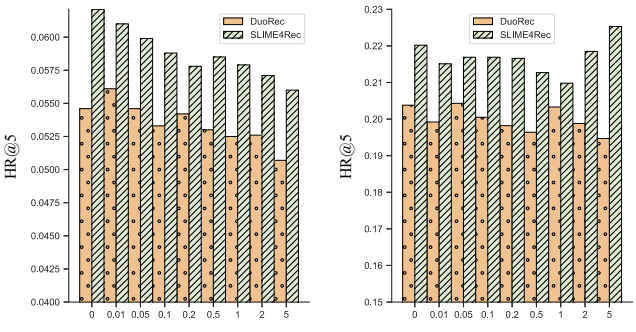
表 V 不同层数下 SLIME4Rec 与 DuoRec 的性能比较。HR 是命中率的缩写，NG 是归一化折损累计增益的缩写。每行中的最佳性能以粗体显示。

方法		美妆				服装				运动			
		HR@5	NG@5	HR@10	NG@10	HR@5	NG@5	HR@10	NG@10	HR@5	NG@5	HR@10	NG@10
$L=2$	DuoRec	0.0546	0.0352	0.0845	0.0443	0.0193	0.0113	0.0302	0.0148	0.0326	0.0208	0.0498	0.0262
	Ours	**0.0604**	**0.0370**	**0.0903**	**0.0467**	**0.0225**	**0.0126**	**0.0343**	**0.0164**	**0.0364**	**0.0230**	**0.0561**	**0.0294**
		↑10.62%	↑5.11%	↑6.86%	↑5.42%	↑16.58%	↑11.50%	↑13.58%	↑10.81%	↑13.75%	↑10.58%	↑13.79%	↑12.21%
$L=4$	DuoRec	0.0551	0.0344	0.0855	0.0441	0.0197	0.0113	0.0299	0.0146	0.0315	0.0204	0.0480	0.0257
	Ours	**0.0607**	**0.0379**	**0.0920**	**0.0480**	**0.0221**	**0.0126**	**0.0341**	**0.0165**	**0.0373**	**0.0243**	**0.0565**	**0.0305**
		↑10.16%	↑10.17%	↑7.60%	↑8.84%	↑12.18%	↑11.50%	↑14.05%	↑13.01%	↑17.46%	↑19.12%	↑19.17%	↑18.68%
$L=8$	DuoRec	0.0565	0.0353	0.0867	0.0451	0.0197	0.0116	0.0316	0.0154	0.0299	0.0197	0.0460	0.0248
	Ours	**0.0621**	**0.0396**	**0.0910**	**0.0489**	**0.0221**	**0.0128**	**0.0342**	**0.0167**	**0.0365**	**0.0239**	**0.0563**	**0.0302**
		↑9.91%	↑12.18%	↑4.96%	↑8.43%	↑12.18%	↑10.34%	↑8.23%	↑8.44%	↑22.07%	↑21.32%	↑19.17%	↑21.77%

方法		ML-1M				Yelp			
		HR@5	NG@5	HR@10	NG@10	HR@5	NG@5	HR@10	NG@10
$L=2$	DuoRec	0.2038	0.1390	0.2946	0.1680	0.0441	0.0325	0.0631	0.0386
	Ours	**0.2139**	**0.1457**	**0.3026**	**0.1743**	**0.0516**	**0.0359**	**0.0766**	**0.0439**
		↑4.96%	↑4.82%	↑2.72%	↑3.75%	↑17.01%	↑10.46%	↑21.39%	↑13.7%
$L=4$	DuoRec	0.2065	0.1423	0.2917	0.1699	0.0454	0.0333	0.0643	0.0394
	Ours	**0.2202**	**0.1515**	**0.3127**	**0.1812**	**0.0502**	**0.0348**	**0.0765**	**0.0432**
		↑6.63%	↑6.47%	↑7.20%	↑6.65%	↑10.57%	↑4.5%	↑18.97%	↑9.64%
$L=8$	DuoRec	0.2164	0.1501	0.3063	0.1701	0.0438	0.0318	0.0629	0.0380
	Ours	**0.2262**	**0.1559**	**0.3132**	**0.1840**	**0.0493**	**0.0336**	**0.0745**	**0.0417**
		↑4.53%	↑3.86%	↑2.25%	↑8.17%	↑9.80%	↑5.66%	↑14.97%	↑9.74%

全面考察滤波器大小在各个维度上的影响。我们将潜在表示的隐藏维度 d 从16调整到256。实验结果如图4所示。我们观察到，对于用户序列行为建模而言，有限的维度大小无法保留足够的物品潜在信息。随着隐藏单元数量在Beauty和ML-1M上达到约64，模型性能趋于饱和，因为更大的嵌入维度能够增强每个频率分量的表示能力，从而更多维度。然而，当隐藏维度超过最优点时，结果不再改善，甚至开始恶化。这表明，当隐藏维度过大时，可能会导致过拟合。

4）模型深度的影响 L：模型的深度决定了骨干编码器的容量和参数量，从而影响序列表示学习的性能。现有的序列推荐模型通常并不深，但在我们的实验中发现，SLIME4Rec 能够从更多的层中获益。为了研究滤波器混合层数的影响，我们固定其他参数，并改变 S
LIME4Rec 和 DuoRec 的网络层数



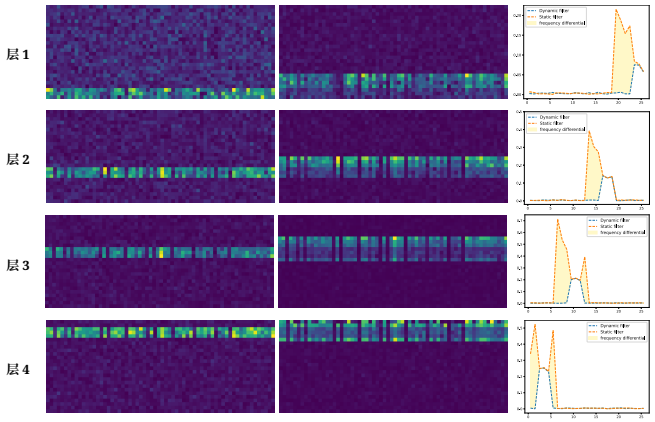
(a) 美妆 (b) ML-1M

图 6. 对不同合成噪声比例的鲁棒性。

使其保持一致，以进行公平比较。表 V 展示了实验结果。我们可以观察到，在 Beauty 和 ML-1M 数据集上，SLIME4Rec 在设置 $L=8$ 下取得了最佳性能；而在 Clothing、Sports 和 Yelp 数据集上，在设置 $L=2$ 或 $L=4$ 下取得了最佳结果。总体而言，在所有不同的网络深度设置下，SLIME4Rec 都持续优于基线模型，并且可以堆叠更多层而不会损失太多推荐性能，这主要得益于使用较小的滤波器，并允许每一层专注于捕获特定频率范围内的用户行为模式。这表明，Transformer 的性能提升无法仅通过简单增加层数来实现，而所提出的滑动滤波器混合结构能够合理利用动态和静态组件，在避免过拟合的同时捕获特征更高阶的交互。

H. 对合成噪声的鲁棒性

为验证 SLIME4Rec 对合成噪声的鲁棒性（**RQ5**），我们在 Beauty 和 ML-1M 数据集上进行了实验。具体而言，我们在每一层的原始表示中加入均匀随机噪声。从图 6 可以看出，ϵ 的增加会导致 SLIME4Rec 和 DuoRec 的性能下降。不过，SLIME4Rec 的性能始终优于 DuoRec。原因可能是，在时域中加入的噪声很容易被我们的滤波器在频域中区分出来。有趣的是，在稠密数据集 ML-1M 上，当添加噪声的比例过



(a) 动态选择模块 (b) 静态拆分模块 (c) 滤波器的振幅

图 7。SLIME4Rec 中学习到的滑动滤波器的可视化。(a) 展示动态滤波器，(b) 展示在频率斜坡结构的滑动模式 4（←）下设置的相应静态滤波器，以及 $\alpha = 0.1$ 和 $\beta = 0.25$。我们沿序列维度计算动态和静态滤波器的幅度，并在 (c) 中展示两者之间的频率差异。使用静态滤波器模块时，$\alpha < \frac{1}{L}$ 中遗漏的一些频率模式可以被重新捕获。

大时，SLIME4Rec 的推荐性能非但没有下降，反而大幅提升。原因可能是，过量噪声可以从稠密数据集的频谱中轻松去除，从而生成更适合对比学习的无噪声表示。换言之，SLIME4Rec 对随机噪声表现出很强的鲁棒性。

I. 学习到的滤波器的可视化

为了评估滑动滤波器如何在每一层捕捉序列行为，我们在 Beauty 数据集上可视化滑动滤波器中学习到的复数权重。动态和静态滤波器的可视化有助于理解频率斜坡结构的工作原理。结果如图 7 所示。根据对图 7(a) 和 (b) 的观察，我们可以得出以下结论：不同层的所有动态和静态滤波器都具有清晰的频率模式，这意味着不同层专注于捕捉用户在不同频率范围内的模式。在捕捉当前特定频率范围的特征时，可学习滤波器会直接抑制周围的频率分量。纵观全局，可以明显看出，滤波器以模式 4 滑动，动态和静态滤波器从高频特征滑向低频特征。

如第III节所述，当动态滤波器大小$S_D < step$时，相邻频带之间会缺失一些频率分量，这使得我们的模型无法对用户的多期偏好进行建模。图7(c)展示了频率差异。静态滤波器重新捕获了动态滤波器遗漏的频率模式，因为静态模块会按模型深度 L平均划分频率分量。它还提供了更细粒度的频率划分，有助于对用户的复杂偏好进行建模，尤其是在这些特征与动态模块提取的特征混合时。尽管该滤波器倾向于为更多

权重分配给低频分量，但从整个频率范围来看，这些相对较高的频率分量得到了强调，从而缓解了高频信息常被忽略的问题。

V. 相关工作

A. 序列推荐

序列推荐通过建模物品转移相关性，预测用户序列中的后续物品。早期的序列推荐研究通常基于马尔可夫链假设。此后，研究者提出了众多由深度学习驱动的序列推荐模型。GRU4Rec [2] 是首次尝试在序列推荐中使用 GRU 网络的方法。为发现序列模式，Caser [3]采用了水平和垂直卷积滤波器。随后，自注意力网络在序列数据建模方面展现出巨大潜力，并由此发展出多种相关模型，例如 SASRec[4]、BERT4Rec [22]和 S^3Rec [26]。这些方法[27]–[33] 大多采用下一物品监督训练方式。另一种训练方式通常会加入额外的辅助训练任务。CL4SRec [16] 将对比策略应用于通过数据增强生成的多个视图。CoSeRec [25] 引入了更多数据增强操作，以训练鲁棒的序列表征。DuoRec [17] 将推荐损失与无监督学习和监督式对比学习相结合，以优化序列推荐模型。尽管这些模型在序列推荐中取得了成功，却忽略了频域中隐藏的重要信息。

B. 频域学习

几十年来，傅里叶变换一直是数字信号处理中的重要工具 [34], [35]。许多研究将傅里叶变换应用于计算机视觉 [36]–[39] 和自然语言处理[40], [41]。全局滤波器网络（GFNet） [6] 学习傅里叶滤波器，以执行逐通道全局卷积。近期的研究尝试利用傅里叶变换增强的模型进行长期序列预测 [11], [42]和偏微分方程求解 [43]–[45]。然而，在序列推荐领域，与傅里叶变换相关的研究还很少。最近，FMLP-Rec [5] 采用一种不含自注意力机制的全多层感知机结构进行序列推荐。更近期地，FMLP-Rec[5]首次引入了用于序列推荐的滤波器增强型多层感知机，通过乘以一个全局滤波器来去除频域中的噪声。然而，全局滤波器往往会赋予低频更高的权重，而低估相对较高的频率。

VI. 结论

在本文中，我们提出了一种用于序列推荐的新型模型 SLIME4Rec，该模型超越了现有的基于时间特征的序列推荐方法。我们构建了两个具有频率斜坡结构的频率选择模块，即动态频率选择模块和静态频率划分模块，分别用于在不同频段中发现更细粒度的序列模式，从而缓解噪声引起的问题。最后，据我们所知，

我们是首个将对比学习范式应用于基于频率的模型的研究，并使用对比损失补充主要的推荐损失。在五个基准数据集上的实验结果表明，我们的 SLIME4Rec 模型优于所有最先进的模型。

致谢

本研究部分得到国家自然科学基金委员会（61876117、62176175）、江苏省高校自然科学研究重大项目（21KJA520004）、苏州市科技发展计划（SYC2022139）、江苏高等学校优势学科建设工程以及软件开发环境国家重点实验室探索性自主选题项目的资助。

参考文献

[1] S. Rendle、C. Freudenthaler、Z. Gantner 和 L. Schmidt-Thieme，“Bpr：基于隐式反馈的贝叶斯个性化排序”，*arXiv* 预印本 *arXiv:1205.2618*，2012。[2] D. Jannach 和 M. Ludewig，“当循环神经网络遇上基于会话的推荐邻域模型”，载于 *RecSys*，2017，第 306–310 页。[3] J. Tang 和 K. Wang，“通过卷积序列嵌入实现个性化 Top-N 序列推荐”，载于 *WSDM*，2018，第 565–573 页。[4] W.-C. Kang 和 J. McAuley，“自注意力序列推荐”，载于 *ICDM*。IEEE，2018，第 197–206 页。[5] K. Zhou、H. Yu、W. X. Zhao 和 J.-R. Wen，“滤波器增强型多层感知机足以实现序列推荐”，载于 *WWW*，2022，第 2388–2399 页。[6] Y. Rao、W. Zhao、Z. Zhu、J. Lu 和 J. Zhou，“用于图像分类的全局滤波器网络”，载于 *NeurIPS 2021*，2021，第 980–993 页。[7] L. R. Rabiner 和 B. Gold，“数字信号处理理论与应用”，*Englewood Cliffs*：*Prentice-Hall*，1975。[8] N. Park 和 S. Kim，“视觉Transformer如何工作？”*arXiv* 预印本 *arXiv:2202.06709*，2022。[10] M. Raghu、T. Unterthiner、S. Kornblith、C. Zhang 和 A. Dosovitskiy，“视觉Transformer的视觉方式与卷积神经网络相同吗？”*NeurIPS*，第 34 卷，第 12 116–12 128 页，2021。[11] T. Zhou、Z. Ma、Q. Wen、X. Wang、L. Sun 和 R. Jin，“Fedformer：用于长期序列预测的频率增强分解Transformer”，载于 *ICML*，机器学习研究论文集，第 162 卷，2022，第 27 268–27 286 页。[12] J. W. Cooley 和 J. W. Tukey，“一种用于机器计算复傅里叶级数的算法”，计算数学，第 19 卷，第 90 期，第 297–301 页，1965。[13] M. Frigo 和 S. G. Johnson，“FFTW3 的设计与实现”，*IEEE* 会议录，第 93 卷，第 2 期，第 216–231 页，2005。[14] N. Srivastava 、 G. Hinton、A. Krizhevsky、I. Sutskever 和 R. Salakhutdinov，“丢弃：一种防止神经网络过拟合的简单方法”，机器学习研究期刊，第 15 卷，第 1 期，第 1929–1958 页，2014。[15] C. Si、W. Yu、P. Zhou、Y. Zhou、X. Wang 和 S. Yan，“Inception Transformer”，*arXiv* 预印本 *arXiv:2205.12956*，2022。[16] X. Xie、F. Sun、Z. Liu、S. Wu、J. Gao、J. Zhang、B. Ding 和 B. Cui，“面向序列推荐的对比学习”，载于 *ICDE*。IEEE，2022，第 1259–1273 页。[17] R. Qiu、Z. Huang、H. Yin 和 Z. Wang，“通过对比学习解决序列推荐中的表示退化问题”，载于 *WSDM*，2022，第 813–823 页。[18] J. W. Cooley 和 J. W. Tukey，“一种用于机器计算复傅里叶级数的算法”，计算数学，第 19 卷，第 90 期，第 297–301 页，1965。[19] J. McAuley、C. Targett、Q. Shi 和 A. Van Den Hengel，“基于风格和替代品的图像推荐”，载于 *SIGIR*，2015，第 43–52 页。

[20] F. M. Harper 和 J. A. Konstan，“MovieLens 数据集：历史与背景”，ACM 交互智能系统汇刊（TIIS），第 5 卷，第 4 期，第 1–19 页，2015。[21] J. Li、Y. Wang 和 J. McAuley，“面向序列推荐的时间间隔感知自注意力”，载于 WSDM，2020，第 322–330 页。[22] F. Sun、J. Liu、J. Wu、C. Pei、X. Lin、W. Ou 和 P. Jiang，“BERT4Rec：基于Transformer双向编码器表示的序列推荐”，载于 CIKM，2019，第 1441–1450 页。[23] W. Krichene 和 S. Rendle，“关于物品推荐的采样指标”，载于 KDD，2020。[24] Y. Wang、H. Zhang、Z. Liu、L. Yang 和 P. S. Yu，“ContrastVAE：用于序列推荐的对比变分自编码器”，载于 CIKM，2022，第 2056–2066 页。[25] Z. Liu 、 Y. Chen 、 J. Li 、 P. S. Yu 、 J. McAuley 和 C. Xiong，“ 结合鲁棒增强的对比自监督序列推荐”，arXiv 预印本 arXiv:2108.06479，2021。[26] K. Zhou 、 H. Wang 、 W. X. Zhao 、 Y. Zhu 、 S. Wang 、 F. Zhang、Z. Wang 和 J.-R. Wen，“S3-Rec：通过互信息最大化进行自监督序列推荐”，载于 CIKM，2020，第 1893–1902 页。[27] L. Wu、S. Li、C.-J. Hsieh 和 J. Sharpnack，“SSE-PT：通过个性化Transformer实现序列推荐”，载于 RecSys，2020，第 328–337 页。[28] J. Lin、W. Pan 和 Z. Ming，“ FISSA：将物品相似度模型与自注意力网络融合，用于序列推荐”，载于 RecSys，2020，第 130–139 页。[29] Y. He、Y. Zhang、W. Liu 和 J. Caverlee，“面向用户生成物品列表续写的一致性感知推荐”，载于 WSDM，2020，第 250–258 页。[30] X. Fan、Z. Liu、J. Lian、W. X. Zhao、X. Xie 和 J.-R. Wen，“更轻量、更优：用于下一物品推荐的低秩分解自注意力网络”，载于 SIGIR，2021，第 1733–1737 页。[31] Z. Liu、Z. Fan、Y. Wang 和 P. S. Yu，“通过反向预训练Transformer，利用伪先验物品增强序列推荐”，载于 SIGIR，2021，第 1608–1612 页。[32] Z. Cui、Y. Cai、S. Wu、X. Ma 和 L. Wang，“基序感知序列推荐”，载于 SIGIR，2021，第 1738–1742 页。[33] J. Zhao、P. Zhao、L. Zhao、Y. Liu、V. S. Sheng 和 X. Zhou，“用于序列推荐的变分自注意力网络”，载于 ICDE。IEEE，2021，第 1559–1570 页。[34] I. Pitas，《数字图像处理算法与应用》。John Wiley & Sons，2000。[35] G. A. Baxes，《数字图像处理：原理与应用》。John Wiley & Sons, Inc.，1994。[36] K. Xu、M. Qin、F. Sun、Y. Wang、Y.-K. Chen 和 F. Ren，“频域学习”，载于 CVPR，2020，第 1740–1749 页。[37] L. Chi、B. Jiang 和 Y. Mu，“快速傅里叶卷积”，NeurIPS，第 33 卷，第 4479–4488 页，2020。[38] R. Suvorov、E. Logacheva、A. Mashikhin、A. Remizova、A. Ashukha、A. Silvestrov、N. Kong、H. Goka、K. Park 和 V. Lempitsky，“利用傅里叶卷积进行分辨率鲁棒的大掩码图像修复”，载于 WACV，2022，第 2149–2159 页。[39] I. Shchekotov、P. Andreev、O. Ivanov、A. Alanov 和 D. Vetrov，“FFC-SE：用于语音增强的快速傅里叶卷积”，arXiv 预印本 arXiv:2204.03042，2022。[40] A. Tamkin、D. Jurafsky 和 N. Goodman，“棱镜视角下的语言：多尺度语言表示的谱方法”，NeurIPS，第 33 卷，第 5492–5504 页，2020。[41] J. Lee-Thorp、J. Ainslie、I. Eckstein 和 S. Ontanon，“FNet：用傅里叶变换混合令牌”，arXiv 预印本 arXiv:2105.03824，2021。[42] H. Wu、J. Xu、J. Wang 和 M. Long，“Autoformer：利用自相关进行长期序列预测的分解变换器”，NeurIPS，第 34 卷，第 22 419–22 430 页，2021。[43] Z. Li、N. Kovachki、K. Azizzadenesheli、B. Liu、K. Bhattacharya、A. Stuart 和 A. Anandkumar，“用于参数化偏微分方程的傅里叶神经算子”，arXiv 预印本 arXiv:2010.08895，2020。[44] J. Guibas、M. Mardani、Z. Li、A. Tao、A. Anandkumar 和 B. Catanzaro，“自适应傅里叶神经算子：高效的Transformer令牌混合器”，arXiv 预印本 arXiv:2111.13587，2021。[45] G. Wen、Z. Li、K. Azizzadenesheli、A. Anandkumar 和 S. M. Benson，“U-FNO——一种用于多相流的增强型基于傅里叶神经算子的深度学习模型”，《水资源进展》，第 163 卷，104180，2022 年，第 .