面向推荐系统的多基础模型用户自适应融合

Sunwoo Kim
KAIST 首尔，大韩民国
kswoo97@kaist.ac.kr

Hyunjin Hwang
KAIST
首尔，大韩民国
hyunjinhwang@kaist.ac.kr

Geon Lee
KAIST
首尔，大韩民国
geonlee0325@kaist.ac.kr

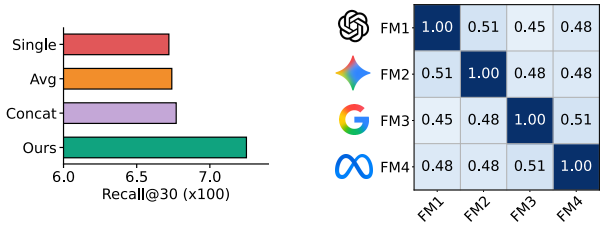
Kyungho Kim
KAIST 首尔，大韩民国
k
kyungho@kaist.ac.kr

Kijung Shin
KAIST 首尔，大韩民国
kijungs@kaist.ac.kr

arXiv:2611.31951v1 [cs.IR] 24 Nov 2026

摘要

推荐系统能否从融合模态内外的一组基础模型中获益？尽管不同的基础模型可能捕捉到物品元数据的互补方面，但我们发现，直接融合它们的表示，与使用单个基础模型相比，带来的提升微乎其微。我们观察到一个关键现象：从不同基础模型中受益的用户各不相同。受此启发，我们提出 UF^{2}O，一种用于推荐的用户自适应多基础模型融合方法。简而言之，UF^{2}O 将每个基础模型的贡献视为因用户而异：它为每个模型维护独立的推荐视图，并针对每位用户自适应地融合各视图的分数。通过在分数层级进行融合，UF^{2}O 还能够融合异构基础模型，而无需直接对齐其表示空间。在四个数据集上联合融合四个基础模型的大量实验表明，在静态和序列推荐设置中，UF^{2}O 的表现均优于现有融合方法，相比最强的融合基线，提升幅度最高可达 7%。本研究的代码、数据集和补充材料可在 https://github.com/kswoo97/UFFO 获取。



(a) 推荐性能　　**(b) 用户集合之间的 Jaccard相似度 从每个基础模型中获益**

图 1：关键动机。与使用单个基础模型（*Single*）相比，直接串接多个基础模型的嵌入，例如平均（*Avg*）或串接（*Concat*），仅带来有限的性能提升，如（a）所示。我们的分析表明，不同基础模型受益的用户各不相同。具体而言，如（b）所示，根据每个基础模型生成的 Top-K 推荐结果中，下一次购买记录出现的用户集合在不同模型之间存在差异。这一观察促使我们设计了 UF^{2}O，以用户自适应的方式融合基础模型。如（a）所示，UF^{2}O（*Ours*）显著优于简单的融合基线方法。结果来自Sports数据集的序列设置。

CCS 概念

• **信息系统** →**推荐系统**。

关键词

推荐系统，基础模型

ACM 参考文献格式：
Sunwoo Kim、Hyunjin Hwang、Geon Lee、Kyungho Kim 和 Kijung Shin。2027。《面向推荐系统的多基础模型用户自适应融合》。载于《第二十届 *ACM* 网络搜索与数据挖掘国际会议论文集（*WSDM '27*）》，*2027*年*2*月*15*日至*19*日，香港，香港。ACM，纽约州纽约市，美国，11 页。https://doi. org/10.1145/3862322.3863490

本作品采用知识共享署名 4.0 国际许可协议。*WSDM '27*，香港，香港© 2027 版权所有归所有者/作者所有。ACM ISBN 979-8-4007-3117-4/2027/02 https://doi.org/10.1145/3862322.3863490

1 引言

推荐系统是各种网络服务的核心，它们从海量物品目录中检索符合用户兴趣的物品 [14, 17, 19, 20, 23–26, 30, 49]。通过从购买等历史商品交互中推断用户偏好，推荐系统能够提供个性化推荐，从而提升用户满意度并促进业务收入。

各种方法利用基础模型从物品元数据中捕获语义信息的能力来进行推荐 [13, 18, 22, 33, 34]。其中，一个有前景的方向是利用基础模型对物品元数据（如标题和图像）进行编码，并将生成的嵌入用作 LightGCN [10] 或 SAS-Rec [15]，等推荐系统骨干网络中的物品表示；这一策略已在许多研究中得到广泛探索 [29, 32, 43, 54]。

自然语言处理和计算机视觉领域的既有研究表明，即使属于同一模态，不同的基础模型也可能捕捉相同输入数据的不同方面 [8, 16, 53]。例如，对于同一输入图像，一些视觉基础模型往往更加关注纹理，而

另一些则倾向于捕捉物体的整体形状和空间组织 [39]。这表明，从物品元数据中提取的信息可能因用于编码的基础模型而异，即使是在同一模态内也是如此，但推荐系统文献对这一特性的关注仍然有限。

受这一特性启发，我们探索使用一组基础模型对推荐所需的物品元数据进行编码。然而，简单地引入多个基础模型并不一定能显著提升推荐性能。如图 1(a) 所示，简单拼接或平均来自不同基础模型的物品嵌入，效果并不比使用单个基础模型好多少，甚至没有提升。基于这一观察，我们推测，要有效利用多个基础模型，需要一种更精细的融合策略，既能保留模型特定表示，又能自适应地确定它们对推荐的贡献。

针对这一挑战，我们提出**UF^{2}O**（**U**ser-ada**p**tive**F**usion of **FO**undation models），一种面向推荐的多基础模型用户自适应分数级融合方法。该方法的设计基于我们的实证分析；分析表明，从不同基础模型中获益的用户并不相同。若某用户的测试物品出现在采用某基础模型的物品表示的推荐系统所生成的 Top-K 条推荐中，我们便认为该基础模型使该用户受益。如图 1(b) 所示，从不同基础模型中获益的用户集合有所重叠，但并不完全相同，其 Jaccard相似度均值为 0.48。这一发现促使我们按用户分别建模各基础模型的贡献，而非对所有用户采用相同的融合策略。

基于这一发现，UF^{2}O 采用了用户自适应分数级融合方法。UF^{2}O 不会将来自不同基础模型的异构嵌入直接组合成一个供所有用户共享的单一物品表示，而是为每个模型保留独立的推荐视图，并仅在获得其推荐分数后执行融合。这种设计使具有不同架构、模态和表示空间的基础模型能够发挥作用，同时无需直接对齐其嵌入，也不会面临丢失模型特定信息的风险。

具体而言，对于每个用户–物品对，UF^{2}O 首先以每个基础模型的物品表示作为推荐系统骨干网络的输入，为其构建独立的推荐视图。随后，UF^{2}O 从每个视图中独立计算推荐分数，并通过用户条件注意力模块对这些分数进行组合；该模块会根据目标用户为每个基础模型分配不同的重要性。该模块为推荐分数分配用户特定权重，并计算其加权平均值，从而增加或减少每个基础模型对最终推荐结果的贡献。此外，我们引入了一种正则化技术，通过防止融合权重集中于单一模型，帮助 UF^{2}O 有效利用多个基础模型。值得注意的是，UF^{2}O 同时适用于静态和序列推荐设置。

通过在每个推荐设置下使用 4 个数据集和 13 种基线方法进行广泛实验，我们证明了 UF^{2}O 在推荐任务中的有效性。值得注意的是，UF^{2}O 联合融合了四个基础模型，并且优于现有的基础模型融合基线，相较于最强的融合基线，性能最高提升了 7%。此外，UF^{2}O 在 16 个设置中的 15 个设置上优于多种多模态推荐系统，突显了利用多个基础模型的有效性，包括在每种模态中使用多个模型。值得注意的是，尽管使用了多个基础模型，与使用单个基础模型相比，UF^{2}O 为每位用户带来的额外推理时间不到 10^{-4} 秒。这种极低的开销得益于预先计算物品嵌入，并在推理过程中并行处理多个基础模型视图。我们的主要贡献如下：

- ***C1.*** 新发现。我们发现，基础模型对不同用户提供准确推荐的能力各不相同。
- ***C2.*** 新方法。 我们提出 UF^{2}O，这是一种用户自适应的分数级融合方法，可融合多个基础模型，并结合一种用于稳定训练的正则化技术。
- ***C3.*** 性能优异。 通过自适应地融合四种基础模型，UF^{2}O 在序列推荐和静态推荐设定中的表现均优于现有融合方法。

我们工作的代码、数据集和补充材料可在 https://github.com/kswoo97/UFFO 获取。

2 相关工作与预备知识

2.1 相关工作

2.1.1 结合多个基础模型。 自然语言处理（NLP）和计算机视觉（计算机视觉）领域的研究表明，不同的基础模型可能捕捉同一物品元数据的不同方面 [8, 16, 39, 53]。以往研究将这一倾向归因于基础模型架构以及各模型所使用的训练数据集之间的差异 [37, 39]。受这一特性启发，许多方法结合从不同基础模型中获得的多个表示，以构建能够捕捉数据不同方面的更丰富表示 [7, 16, 42]。这些方法通常学习一个在所有用户之间共享的通用融合表示，隐式地将每个基础模型的效用视为用户无关。相比之下，UF^{2}O 将基础模型的效用视为用户相关，为每个模型保留单独的推荐视图，并在分数层级执行用户自适应融合。

2.1.2 使用基础模型进行物品编码。 各种推荐系统利用物品元数据（如标题、描述和图像）来捕获有关物品的丰富语义信息 [3, 28, 46, 48, 52]。在这些方法中，一个有前景的方向是使用预训练基础模型 对物品元数据进行编码，并利用所得嵌入来获取物品表示 [11, 22, 31, 51]。具体而言，这些嵌入可以（1）直接用作物品表示 [51]，或（2）与传导式物品ID嵌入 [22]相结合。与我们的研究最密切相关的研究领域之一是多模态推荐 [2, 4, 35, 40],，该领域利用物品元数据的多模态表示进行推荐。我们的研究设置与传统多模态推荐互为补充。多模态推荐系统主要研究如何结合不同模态，但通常为每种模态依赖一个单独的编码器。相比之下，我们研究如何利用多个独立预训练的基础模型，包括同一模态中的多个模型，因此还面临着结合异构的模型特定表示这一额外挑战，涵盖

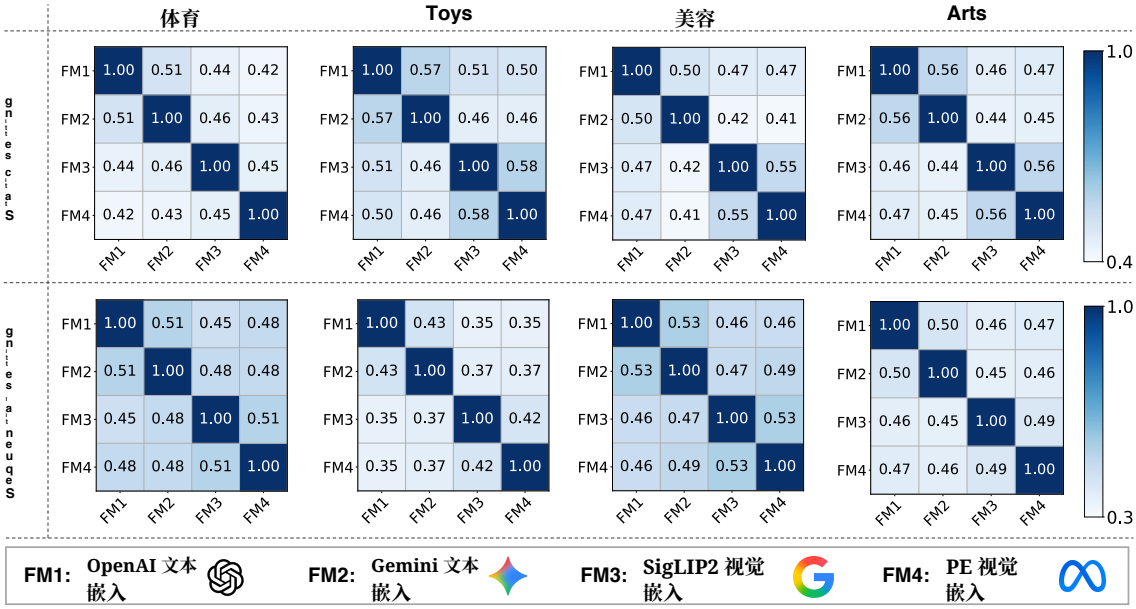


图 2：从不同基础模型中受益的用户重合情况。对于每一对基础模型，报告了从一个基础模型获得准确推荐的用户集合（即其下一次购买出现在 Top-K 推荐中的用户）与从另一个基础模型获得准确推荐的相应用户集合之间的 Jaccard相似度。在静态和序列推荐设置中，平均 Jaccard相似度均低于 0.5，分别为 0.48 和 0.46。

同一模态，并因此面临着结合异构的模型特定表示这一额外挑战。

2.2 预备知识

*2.2.1*物品信息。 设 I 表示物品集合。每个物品 $i_k \in \mathrm{I}$ 都与其对应的元数据 D_k 相关联。这些元数据包括文本信息（例如标题或描述）和/或视觉信息（例如图像）。我们将基于基础模型的编码器集合表示为 $\mathrm{F} = \{f_{(1)} f_{(2)}, \cdots \ f_{(|\mathcal{F}|)}\}$。最后，我们将基础模型 $f_{(p)}$ 生成的物品 i_k 的元数据嵌入记为 $\mathbf{x}_k^{(p)}$（即 $\mathbf{x}_k^{(p)} = f_{(p)}(\mathrm{D}_k)$）。需要注意的是，基础模型不一定要联合处理多个模态；它也可以只处理单一模态。例如，Gemini Text Embedding [27] 仅对文本信息进行编码。

2.2.2 静态设置。 令 U 表示用户集合。在静态设置下，我们假设给定用户-物品二部图 $\mathrm{G} = (\mathrm{U} \cup \mathrm{I}, \mathrm{E})$，其中 E 表示已观测的用户-物品交互集合。具体而言，用户 $u \in \mathrm{U}$ 与物品 $i_k \in \mathrm{I}$ 之间的交互由一条边 $(u, i_k) \in \mathrm{E}$ 表示。

*2.2.3*序列设置。 与静态设置一样，我们用 U 表示用户集合。在序列设置下，每个用户 $u \in \mathrm{U}$ 都关联一个交互序列 $s_u = (i_{u,1}, i_{u,2}, \ldots\ldots, i_{u,n_u})$，其中 $i_{u,t} \in \mathrm{I}$ 表示用户按时间顺序交互过的第 t 个物品，而 n_u 表示序列长度。

3 分析

本节将展示一项分析，说明不同基础模型能为不同用户带来益处，这也构成了我们提出的方法的关键动机。为此，我们首先介绍分析设置（第 3.1 节），随后呈现分析结果及其主要启示（第 3.2 节）。

3.1 分析设置

*3.1.1*数据集。 我们使用 Hou 等人发布的四个领域特定的电子商务数据集 [12]：(1) 体育与户外（体育），(2) 玩具与游戏（玩具），(3) 美容与个人护理（美容），以及 (4) 艺术、手工艺与缝纫（艺术）。我们使用 Kim 等人提供的这些数据集的预处理版本 [22]。在这里，我们使用文本和视觉元数据，其中每个商品的标题作为其文本元数据，图像作为其视觉元数据。

3.1.2 基础模型。 我们采用四种基础模型对物品元数据进行编码：两种文本基础模型和两种视觉基础模型。两种文本基础模型是 OpenAI 的 text-embedding-3-large [36] 和 Gemini Text Embedding [27],，二者均以每个物品的标题作为输入。两种视觉基础模型是 SigLIP2 [45] 和 Perception Encoder（PE） [1],，二者均以每个物品的图像作为输入。在接下来的分析中，我们分别考察每种基础模型，并比较由此得到的推荐结果。

3.1.3 静态设置下的前向传播与训练。 在静态推荐设定中，我们使用 LightGCN [10] 作为骨干推荐系统。我们为每个用户 $u \in \mathrm{U}$ 学习基于 ID 的传导式嵌入，记为 $\mathbf{t}_u \in \mathrm{R}^d$；并为每个物品 $i_k \in \mathrm{I}$ 学习基于 ID 的传导式嵌入，记为 $\mathbf{t}_k \in \mathrm{R}^d$。

考虑使用单个基础模型 $f_{(p)}$。对于每个物品 $i_k \in \mathrm{I}$，我们使用 $f_{(p)}$ 对其元数据进行编码，以获得其元数据嵌入 $\mathbf{x}_k^{(p)} = f_{(p)}(\mathrm{D}_k)$。然后，我们使用一个 MLP 投影器（记作 $\mathrm{MLP}^{(p)}$），将物品的元数据嵌入 $\mathbf{x}_k^{(p)}$ 映射为

一个 d 维的嵌入 $\tilde{\mathbf{x}}_k^{(p)} \in \mathbb{R}^d$（即 $\tilde{\mathbf{x}}_k^{(p)} = \mathrm{MLP}^{(p)}(\mathbf{x}_k^{(p)})$），使其维度与用户嵌入和物品 ID 嵌入的维度相匹配。

接下来，我们构建两个视图：ID 视图和元数据视图。这两个视图共享用户 ID 嵌入 $\mathbf{t}_u$，将其用作用户 u的输入用户表示，但采用不同的物品表示。具体而言，ID 视图使用物品 i_k 的 ID 嵌入 $\mathbf{t}_k$，而元数据视图使用投影后的元数据嵌入$\tilde{\mathbf{x}}_k^{(p)}$。基于这些输入表示，我们分别在两个视图中，对用户–物品交互图 G 执行基于LightGCN的消息传递，从而得到各视图专属的最终用户和物品表示。

最后，对于给定的用户-物品对，我们将该视图中最终用户和物品表示的内积作为视图专属的推荐分数，并将两个视图的分数相加，得到最终推荐分数。所有可训练参数，包括用户和物品 ID 嵌入以及 MLP 投影器，均按照原始 LightGCN 论文 [10]采用的训练流程，通过BPR 损失 [41], 进行优化。

3.1.4 序列设置中的前向传播与训练。在序列推荐设置中，我们使用 SASRec [15] 作为骨干推荐系统。与静态设置不同，我们仅使用物品 ID 嵌入（$\mathbf{t}_k$，对每个 $i_k \in \mathrm{I}$），并遵循序列推荐中的标准做法，省略用户 ID 嵌入 [15, 44, 55]。对于物品 i_k，我们按照静态设置中的方法，使用单个基础模型 $f_{(p)}$ 和 MLP 投影器$^{(p)}$，获得投影后的元数据嵌入 $\tilde{\mathbf{x}}_k^{(p)}$。

与静态设置类似，我们构建两个视图：ID 视图和元数据视图。ID 视图使用传导式物品嵌入 $\mathbf{t}_k = \tilde{\mathbf{x}}_k^{(ID)}, \forall i_k \in \mathrm{I}$ 作为物品表示，而元数据视图使用 $\tilde{\mathbf{x}}_k^{(p)}, \forall i_k \in \mathrm{I}$ 作为物品表示。然后，对于具有交互序列 $s_u = (i_u,1, i_u,2, \ldots, i_u,n_u)$ 的用户 u，我们将每个$q \in \{p, \mathrm{ID}\}$对应的表示序列 $[\tilde{\mathbf{x}}_{u1}^{(q)}, \tilde{\mathbf{x}}_{u2}^{(q)}, \ldots, \tilde{\mathbf{x}}_{u}^{(q)}, {}_{n_u}]$输入 SASRec，其中 $\tilde{\mathbf{x}}_{ut}^{(q)}$, 表示视图 q中物品 i_u,t的表示。为了计算物品 i_j作为用户 u的下一个物品的推荐分数，我们计算 SASRec 在第 n_u个位置的输出与每个 $q \in \{p, \mathrm{ID}\}$对应的物品表示$\tilde{\mathbf{x}}_j^{(q)}$ 之间的内积。随后，我们将两个视图的分数相加，得到最终得分。所有可训练参数，包括物品 ID 嵌入、MLP 投影器和 SASRec 参数，均采用 SASRec 原始论文 [15]中使用的基于二元交叉熵的训练过程进行优化。

3.1.5 最终评估方案。 训练完成后，我们衡量从不同基础模型中获益的用户群体之间的重叠程度。这里，如果用户的下一次购买（即测试物品）出现在使用基于该基础模型得到的物品表示的推荐系统所生成的 Top-K 推荐结果中，我们就认为该基础模型使这位用户受益。我们在序列设置和静态设置下均进行此分析。具体而言，对于每个基础模型 $f_{(p)} \in \mathrm{F}$，我们首先分别按照第 3.1.3 节和第 3.1.4 节所述流程，在静态设置和序列设置下训练推荐系统。训练完成后，对于每个基础模型 $f_{(p)}$，我们为每位用户生成一份 Top-30 推荐列表，并将 $\mathrm{U}^{(p)} \subseteq \mathrm{U}$ 定义为其下一次购买出现在相应推荐列表中的用户集合。最后，对于每一对基础模型 $\{f_{(p)} f_{(q)}\} \in \binom{\mathcal{F}}{2}$，我们衡量 $\mathrm{U}_{(p)}$和 $\mathrm{U}^{(q)}$之间的 Jaccard 相似度（即 $(|\mathrm{U}_{(p)} \cap \mathrm{U}_{(q)}|)/(|\mathrm{U}_{(p)} \cup \mathrm{U}_{(q)}|)$.

3.2 分析结果与启示

分析结果如图 2 所示。总体而言，从不同基础模型中受益的用户群体彼此重叠，但并不完全相同；在静态推荐设置和序列推荐设置中，平均 Jaccard 相似度分别为 0.48 和 0.46。此外，使用不同输入模态的基础模型，其用户群体重叠程度低于使用相同模态的模型，平均 Jaccard 相似度分别为 0.45 和 0.51。作为比较，当我们比较使用不同随机种子训练的同一基础模型时，平均 Jaccard 相似度约为 0.8，显著高于不同基础模型之间的相似度；详情见补充材料 [21]。这些结果表明，不同基础模型会使不同用户受益，这意味着对所有用户采用相同的融合策略，可能无法充分利用他们各自独特的推荐信号。结合图 1 中观察到的简单表示级融合效果有限这一结果，这促使我们保留各个基础模型的独特信号，并针对每位用户进行自适应组合。

4 提出的方法：UF²O

我们介绍**UF²O**（**U**户自适应**F**融合**FO**基础模型），这是一种用于在推荐系统中融合多个基础模型的用户自适应方法。受第1节和第3节中观察结果的启发，UF²O遵循两项关键设计原则：（1）保留基础模型特有的推荐信号，而不是直接将异构表示合并为单一表示；（2）根据目标用户自适应地确定这些信号的贡献。为此，UF²O以每个基础模型的物品表示作为骨干推荐系统的输入，为每个基础模型构建一个独立的推荐视图，每个视图分别生成一个推荐分数。随后，UF²O通过用户条件注意力模块，在这些视图之间执行用户自适应分数级融合。图3展示了静态设置下UF²O的概览，其序列版本见补充材料 [21]。下面，我们首先介绍如何使用基础模型对物品元数据进行编码（第4.1节）。然后介绍如何计算基础模型特有的推荐分数（第4.2节），以及如何以用户自适应的方式对这些分数进行融合（第4.3节）。之后，我们将介绍采用专门正则化损失的训练策略，以及最终推荐分数的推理时计算（第4.4节）。

4.1 项目元数据编码

UF²O 的物品元数据编码过程在静态和序列设置中完全相同。编码时，我们使用多个基础模型为每个物品获取多个元数据嵌入。考虑一个物品 i_k 以及一组基础模型 $\mathrm{F} = \{f_{(1)} f_{(2)}, \cdots f_{(|\mathcal{F}|)}\}$。对于每个基础模型 $f_{(p)} \in \mathrm{F}$,

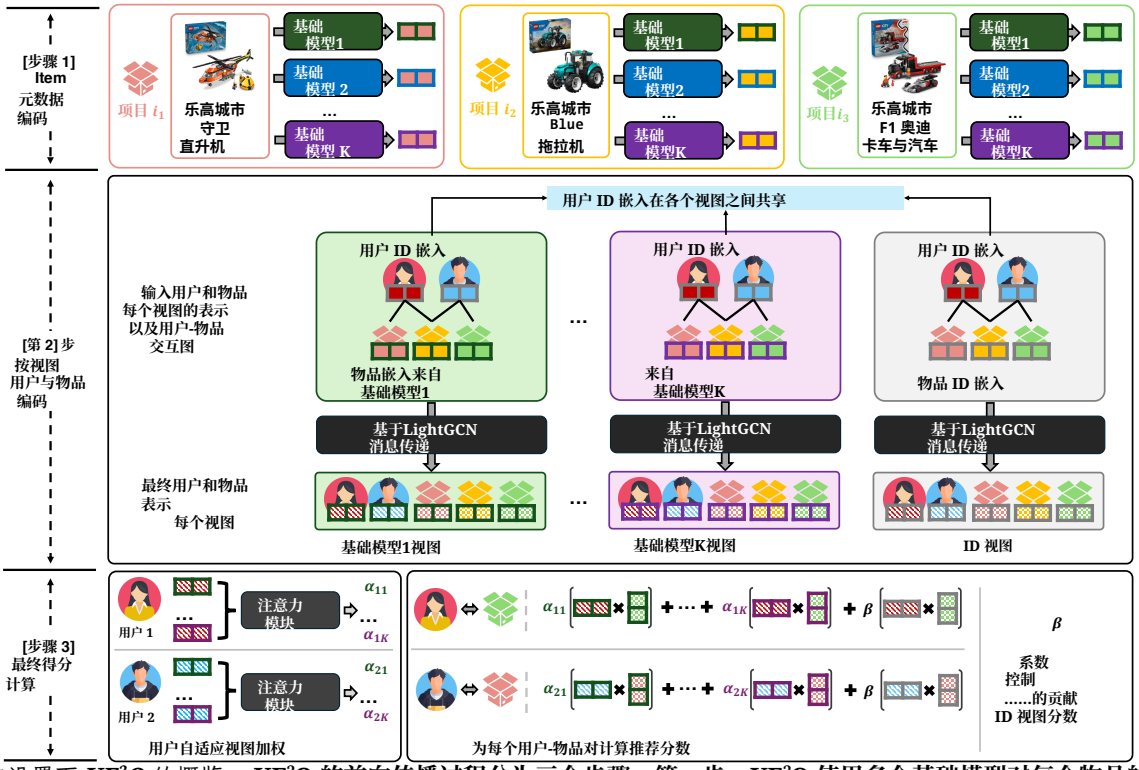


图3：静态设置下 *UF²O* 的概览。 UF²O 的前向传播过程分为三个步骤。第一步，UF²O 使用多个基础模型对每个物品的元数据进行编码（见第一行）。第二步，UF²O 为每个视图分别获取用户和物品表示；在静态设置下使用 LightGCN [10] 作为骨干推荐器，在序列设置下使用 SASRec [15]（见第二行）。最后，第三步，UF²O 使用用户自适应加权策略，对不同基础模型的分数进行加权平均（见第三行）。

我们对物品 i_k 的元数据 D_k 进行编码，以获取其特定于基础模型的表示 $\mathbf{x}_k^{(p)}$（即 $\mathbf{x}_k^{(p)} = f_{(p)}(\mathrm{D}_k)$）。请注意，所有基础模型均保持冻结状态，不会通过推荐损失进行端到端更新，因为我们的目标是保留每个模型捕捉物品特定于基础模型方面的固有能力。此外，（1）许多基础模型的权重并未公开；（2）微调多个基础模型会产生高昂的计算成本。对元数据进行编码后，我们将每个生成的嵌入投影到一个d维特征空间，以确保其与用户和/或物品 ID 嵌入在维度上兼容。具体而言，对于每个特定于基础模型的表示 $\mathbf{x}_k^{(p)}$，其中 $p \in [|\mathbb{F}|]$，我们使用特定于基础模型的 MLP 投影器 $\mathrm{MLP}^{(p)}$，得到投影后的表示 $\tilde{\mathbf{x}}_k^{(p)} \in \mathrm{R}^d$（即 $\tilde{\mathbf{x}}_k^{(p)} = \mathrm{MLP}^{(p)}(\mathbf{x}_k^{(p)})$）。请注意，MLP 投影器 $\mathrm{MLP}^{(p)}$，$\forall p \in [|\mathbb{F}|]$的所有参数均可学习，并通过推荐损失进行端到端训练。

4.2 基础模型特定评分

获得每个基础模型生成的物品元数据表示后，我们为每个模型分别计算推荐分数，而不是先将它们的表示合并为单个嵌入。这样便将不同基础模型生成的异构表示转换为用户-物品推荐分数，为组合这些分数提供了统一的基础。具体而言，

我们将每个基础模型视为一个独立视图，利用相应模型生成的物品表示。由于静态和序列设置下的分数计算过程有所不同，我们将在下文分别介绍。

4.2.1 静态设置下的分数计算。 除了 $|\mathbb{F}|$ 基础模型视图外，我们还纳入了一个 ID 视图，该视图使用传导式物品 ID 嵌入。对于静态设置下的骨干推荐系统，UF²O 使用第 3.1.3 节所采用的 LightGCN [10],。具体而言，对于与基础模型 $f_{(p)}$相关联的视图，我们使用物品 $i_k \in \mathrm{I}$ 的投影表示 $\tilde{\mathbf{x}}_k^{(p)} \in \mathrm{R}^d$ 作为输入物品表示。对于 ID 视图，我们使用传导式物品 ID 嵌入作为输入物品表示。在所有视图中，我们共享相同的输入用户表示，即所有 $u \in \mathrm{U}$ 对应的传导式用户 ID 嵌入$\mathbf{t}_u$。

然后，我们针对每个视图分别执行基于LightGCN的消息传递。因此，对于每个视图 $p \in [|\mathbb{F}|] \cup \{ID\}$，我们得到每个 $u \in \mathrm{U}$对应的最终用户表示 $\mathbf{e}_u^{(p)} \in \mathrm{R}^d$，以及每个物品 $i_k \in \mathrm{I}$对应的物品表示 $\mathbf{e}_k^{(p)} \in \mathrm{R}^d$。请注意，尽管不同视图共享相同的输入用户嵌入，但它们的最终用户表示并不相同，因为消息传递基于各视图特有的物品表示。最后，为了计算用户 u对物品 i_k的推荐分数，我们为每个视图 p计算一个视图特有的分数：$s_{uk}^{(p)} = (\mathbf{e}_u^{(p)})^T \mathbf{e}_k^{(p)} \in \mathrm{R}$。换言之，对于每个用户-物品对，我们得到 $|\mathbb{F}|+1$ 个分数。

4.2.2 序列设置下的分数计算。在序列设置下，我们还加入一个基于传导式物品 ID 嵌入的 ID 视图，与 $|\mathcal{F}|$ 个基础模型视图并列。在这种设置下，对于骨干推荐系统，UF²O 使用 SASRec [15]，，其参数在各视图之间共享。具体而言，对于与基础模型 $f^{(p)}$关联的视图，我们使用所有物品 $i_k \in \mathcal{I}$的投影后的元数据表示 $\tilde{\mathbf{x}}_k^{(p)} \in \mathbb{R}^d$ 作为物品表示。对于 ID 视图，我们使用物品 $i_k \in \mathcal{I}$的传导式物品嵌入 $\mathbf{t}_k \in \mathbb{R}^d$ 作为物品表示。

接下来，对于具有交互序列 $s_u = (i_{u,1}, i_{u,2}, \ldots, i_{u,n_u})$ 的用户 u，我们将表示序列 $[\tilde{\mathbf{x}}_{u1}^{(p)}, \tilde{\mathbf{x}}_{u2}^{(p)}, \ldots, \tilde{\mathbf{x}}_{u,n_u}^{(p)}]$输入 SASRec，其中，对于每个视图 $p \in [|\mathcal{F}|] \cup \{ID\}$，$\tilde{\mathbf{x}}_{ut}^{(p)}$ 表示物品 $i_{u,t}$ 的物品表示。为了计算用户 u 可能交互的下一个物品的推荐分数，我们使用与用户最后交互物品 i_{u,n_u} 对应的 SASRec输出嵌入 $\mathbf{e}_u^{(p)} \in \mathbb{R}^d$ 作为最终用户表示。具体而言，为了计算物品 i_k 作为用户 u 的下一次交互的推荐分数，我们针对每个视图 $p \in [|\mathcal{F}|] \cup \{ID\}$ 获取视图特定分数，即 $s_{uk}^{(p)} = (\mathbf{e}_u^{(p)})^T \tilde{\mathbf{x}}_k^{(p)} \in \mathbb{R}$。

4.3 用户自适应分数融合

回顾第 3 节的一个关键观察：从不同基础模型中受益的用户各不相同。因此，在 UF²O 中，我们会在计算最终推荐分数时，根据目标用户调整每个基础模型的贡献。为此，我们采用用户自适应注意力机制，为每个基础模型视图产生的推荐分数分配针对用户的权重，并计算其加权平均值，以得到最终推荐分数。考虑为用户 u计算物品 i_k的推荐分数。首先，我们使用可学习投影向量 $\mathbf{a} \in \mathbb{R}^d$，将每个最终用户表示$\mathbf{e}_u^{(p)}$投影为一个标量注意力logit。随后，我们对各视图对应的这些logit应用 Softmax，得到相应的注意力权重：

$$[\alpha_{u1}, \alpha_{u2}, \cdots, \alpha_{u|\mathcal{F}|}] = \texttt{Softmax}\left(\frac{\mathbf{a}^T \mathbf{e}_u^{(1)}}{\tau}, \frac{\mathbf{a}^T \mathbf{e}_u^{(2)}}{\tau}, \cdots, \frac{\mathbf{a}^T \mathbf{e}_u^{(|\mathcal{F}|)}}{\tau}\right), \tag{1}$$

其中，$\alpha_{up} \in [0, 1]$ 表示分配给用户的基础模型视图 p的注意力权重，τ表示温度超参数。与全局共享的融合权重不同，α_{up} 会为每个基础模型分配针对用户的贡献权重，从而反映第 3 节中观察到的用户层面的异质性。然后，我们基于这些注意力权重计算基础模型分数的加权平均值。在这两种设置下，用户 u的物品 i_k推荐分数（记为 s_{uk}）按如下方式计算：$s_{uk} = \sum_{p=1}^{|\mathcal{F}|} \alpha_{up} s^{(p)uk} + \beta s_{(ID)uk}$，其中 β是一个超参数，用于控制ID视图分数相对于基础模型视图分数的贡献。请注意，我们通过实证观察到，各基础模型视图的分数大小相近（见 [21]），这表明最终得分不会仅因分数尺度而由某个视图主导。[1] 因此，注意力权重控制各视图的相对贡献，权重越高，该视图的预测结果影响越大。

[1]我们可以通过标准化显式对齐不同视图中分数的尺度，但在我们的实验中，这样做会降低推荐性能（见 [21]）。

4.4 训练与推理

4.4.1 训练。UF²O 的训练损失由两部分组成：（1）推荐监督损失；（2）注意力坍缩正则化损失，用于防止注意力权重坍缩到单个基础模型视图上。

对于推荐监督损失 L_{rec}，我们采用原始骨干推荐器所使用的训练目标：在静态设置下，LightGCN [10] 使用 BPR 损失 [41]，在序列设置下，SASRec [15] 使用二元交叉熵损失。

当 UF²O 仅使用推荐监督损失进行训练时，我们观察到，基础模型融合过程中注意力权重通常会集中到单个基础模型上，使该模型在所有用户上的权重都显著高于其他模型。为缓解这一问题，我们引入注意力坍缩正则化损失 L^{reg}，以防止注意力权重过度集中于单个基础模型。具体而言，该损失的计算如下：

$$\mathcal{L}_{reg} = \sum_{p=1}^{|\mathcal{F}|} \overline{\alpha}_p \log(\overline{\alpha}_p), \text{ where } \overline{\alpha}_p = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \alpha_{up}.$$

请注意，L^{reg} 会惩罚单个基础模型在用户之间获得主导性注意力权重的情况。重要的是，该正则化作用于跨用户聚合的融合权重，而不是分别作用于每个用户的权重。因此，它会抑制单个基础模型在全局范围内占据主导地位，同时仍允许不同用户对不同模型赋予较高权重。[2] 总体训练目标定义为 $\mathrm{L} = \mathrm{L}_{rec} + \lambda \mathrm{L}^{reg}$，其中 λ 控制正则化损失的贡献，并对所有可学习参数进行联合优化，以最小化 L。

4.4.2 推理。在推理时，UF²O 使用与训练期间相同的前向传播过程计算推荐分数。具体而言，在使用 $|\mathcal{F}|$ 基础模型对物品元数据进行编码之后（第 4.1 节），UF²O 为每个基础模型计算一个推荐分数（第 4.2 节），然后对这些分数执行用户自适应融合（第 4.3 节），从而获得每个目标用户-物品对的最终推荐分数。

5 实验

在本节中，我们通过回答以下五个问题来评估 UF²O 在推荐任务中的有效性：

- 问题***1***。UF²O 在静态设置下的效果如何?
- 问题***2***。UF²O 在序列设置下的效果如何?
- 问题***3***。 与基线方法相比，UF²O 的推理速度有多快?
- ***Q4.*** 学习到的用户注意力权重分布是否符合我们的直觉?
- ***Q5.*** UF²O 的每个关键组件如何影响其整体性能?

5.1 实验设置

5.1.1 数据集。 与我们的分析（第 3 节）一样，我们使用了 Hou 等人 [12] 发布的四个特定领域电子商务数据集：(1) 体育

[2]一个潜在的担忧是，$\mathcal{L}_{\text{reg}}$ 可能会为每位用户在各个基础模型之间生成均匀的权重。即使不同用户为同一个基础模型分配不同的权重，只要所有基础模型在所有用户中的平均权重相等，损失也同样可以达到最小值。我们的实验表明，为基础模型分配的注意力权重会因用户而异（第 5.5 节）。

表1： 静态设置下的推荐性能。**报告了五次实验试验中召回率@K（R@K）和NDCG@K（N@K）的均值和标准差，其中** $K \in \{20, 30\}$。**为提高可读性，所有数值均乘以100。表现最佳的结果以绿色突出显示。*表示在相应设置下，基于单侧Wilcoxon符号秩检验（[2]），UF**$p < 0.05$**O在统计上显著优于表现最佳的基线。**

	方法	体育与户外				玩具与游戏				美容与个人护理				艺术、工艺与缝纫			
		R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30
Single	OpenAI [36]	6.08 (.08)	7.78 (.11)	2.64 (.05)	3.01 (.05)	6.58 (.08)	8.19 (.12)	2.95 (.05)	3.29 (.06)	5.15 (.05)	6.61 (.03)	2.15 (.02)	2.46 (.02)	7.65 (.09)	9.77 (.07)	3.26 (.04)	3.71 (.03)
	Gemini [27]	5.87 (.09)	7.50 (.12)	2.49 (.04)	2.84 (.05)	6.15 (.06)	7.65 (.09)	2.74 (.02)	3.06 (.03)	4.61 (.05)	6.02 (.06)	1.90 (.02)	2.21 (.02)	7.33 (.05)	9.36 (.06)	3.09 (.02)	3.52 (.01)
	SigLIP2 [45]	5.81 (.02)	7.42 (.07)	2.55 (.01)	2.89 (.02)	6.57 (.11)	8.18 (.10)	2.95 (.05)	3.30 (.05)	5.23 (.04)	6.61 (.05)	2.18 (.03)	2.51 (.03)	7.33 (.08)	9.28 (.09)	3.13 (.03)	3.55 (.02)
	位置编码 [1]	5.82 (.06)	7.40 (.09)	2.51 (.03)	2.85 (.03)	6.41 (.06)	7.94 (.08)	2.85 (.03)	3.18 (.04)	5.19 (.03)	6.55 (.04)	2.18 (.02)	2.47 (.02)	7.31 (.07)	9.26 (.13)	3.14 (.04)	3.55 (.05)
Align	OpenCLIP [38]	6.04 (.12)	7.78 (.10)	2.60 (.06)	2.97 (.05)	6.47 (.04)	8.00 (.05)	2.92 (.02)	3.24 (.02)	5.05 (.04)	6.50 (.05)	2.13 (.03)	2.44 (.03)	7.63 (.10)	9.79 (.11)	3.26 (.07)	3.72 (.08)
	MetaCLIP2 [6]	6.15 (.07)	7.84 (.05)	2.65 (.03)	3.01 (.03)	6.70 (.12)	8.27 (.03)	3.00 (.06)	3.33 (.04)	5.18 (.04)	6.69 (.04)	2.18 (.02)	2.51 (.02)	7.72 (.06)	9.83 (.04)	3.30 (.03)	3.75 (.02)
Multi	FREEDOM [56]	6.11 (.10)	7.88 (.08)	2.61 (.05)	2.99 (.05)	6.83 (.06)	8.48 (.08)	3.14 (.03)	3.47 (.03)	5.13 (.04)	6.56 (.02)	2.17 (.02)	2.48 (.01)	7.65 (.09)	9.74 (.09)	3.31 (.04)	3.73 (.03)
	DA-MRS [50]	6.19 (.05)	7.99 (.07)	2.77 (.02)	3.15 (.01)	6.87 (.07)	8.51 (.06)	3.14 (.06)	3.49 (.05)	5.13 (.03)	6.44 (.03)	2.20 (.01)	2.48 (.01)	7.85 (.07)	10.01 (.07)	3.35 (.03)	3.80 (.03)
Fusion	Mean	6.06 (.03)	7.81 (.05)	2.66 (.01)	3.03 (.01)	6.59 (.07)	8.27 (.07)	2.93 (.03)	3.29 (.03)	5.25 (.03)	6.68 (.04)	2.19 (.01)	2.50 (.01)	7.73 (.05)	9.75 (.02)	3.27 (.02)	3.70 (.01)
	串接	6.17 (.08)	7.85 (.05)	2.69 (.05)	3.05 (.04)	6.68 (.08)	8.30 (.09)	3.00 (.03)	3.35 (.03)	5.26 (.03)	6.68 (.03)	2.22 (.02)	2.53 (.02)	7.86 (.09)	9.91 (.12)	3.36 (.01)	3.79 (.02)
	DME [16]	6.10 (.10)	7.79 (.10)	2.68 (.03)	3.03 (.03)	6.66 (.06)	8.36 (.08)	2.99 (.04)	3.35 (.05)	5.13 (.03)	6.59 (.07)	2.15 (.02)	2.46 (.02)	7.56 (.12)	9.72 (.11)	3.25 (.04)	3.70 (.04)
	SimMMDG [7]	6.22 (.13)	7.90 (.11)	2.72 (.05)	3.07 (.05)	6.66 (.10)	8.25 (.12)	2.98 (.06)	3.31 (.07)	5.21 (.07)	6.64 (.09)	2.19 (.05)	2.49 (.05)	7.70 (.06)	9.87 (.09)	3.31 (.02)	3.76 (.02)
	MoME [42]	6.31 (.03)	7.99 (.03)	2.73 (.01)	3.08 (.01)	6.76 (.07)	8.49 (.06)	3.05 (.01)	3.42 (.02)	5.09 (.01)	6.62 (.01)	2.12 (.01)	2.45 (.01)	7.85 (.01)	10.02 (.04)	3.33 (.02)	3.79 (.02)
	UF²O（本文方法）	6.56 (.06)*	8.40 (.04)*	2.86 (.03)*	3.25 (.03)*	6.93 (.06)*		3.11 (.05)			6.86 (.09)*	2.24 (.02)*	2.56 (.02)*	8.02 (.05)*	10.32 (.10)*	3.42 (.03)*	3.91 (.04)*

以及（1）户外（运动）、（2）玩具与游戏（玩具）、（3）美容与个人护理（美容）和（4）艺术、工艺与缝纫（艺术）。我们使用 Kim 等人提供的预处理数据集 [22], , 其中每个物品的标题和图像分别作为其文本和视觉元数据。有关数据集的更多细节见 [21]。

5.1.2 基线方法和 ***UF²O***。 作为比较，我们考虑了四类基线方法，共包含 13 种方法。下面将分别介绍各类别的详细信息。

• **基线 1（Single）：基于单个基础模型的方法**。 该基线类别用于评估相较于使用单个基础模型，使用多个基础模型所带来的收益。具体而言，我们使用分析中所考虑的四个基础模型（第 3 节）：OpenAI 的大型文本嵌入模型（Ope-nAI） [36], Gemini 文本嵌入模型（Gemini） [27],SigLIP2 [45] 以及 Perception Encoder（PE） [1]。这些方法使用与第 3.1 节相同的骨干模型。

• **基线 2（Align）：基于对齐的多模态基础模型的方法**。该基线类别用于与几种使用对齐的多模态嵌入的代表性方法进行比较；由于不同模态的表示共享同一嵌入空间，因此可以直接进行融合。具体而言，我们使用两种多模态基础模型：Open-CLIP [38] 和 MetaCLIP2 [6]。在这些方法中，我们将文本表示和视觉表示视为独立的基础模型视图，按照第 4.1 节和第 4.2 节分别从每个视图获得推荐分数，并对所得分数取平均值。

• **基线 3（多模态）：多模态推荐系统**。该基线类别用于与利用多模态物品元数据表示的推荐系统进行比较，其中每种模态使用一个基础模型。具体而言，我们使用 FREEDOM [56] 和 DA-MRS [50] 作为静态多模态推荐系统，使用 M³Rec [9]和 DeCoRec [5]作为序列式多模态推荐系统。由于这些方法使用一个文本基础模型和一个视觉基础模型，我们分别使用 OpenAI 的大型文本嵌入模型 [36] 和 SigLIP2 [45], 。需要注意的是，这些基础模型在各自的文本和视觉类别中取得了最佳的单基础模型性能。

• **基线方法 4（融合）：表示级融合方法**。这些基线方法使我们能够评估：相比于直接学习融合表示，保留特定于基础模型的推荐视图并在分数层级执行融合是否能够带来收益。为此，我们考察了两种启发式融合方法和三种可学习融合方法。对于前者，即平均池化和串接，我们首先将表示投影到公共维度空间中，然后将其组合为单一表示。对于后者，我们纳入了计算机视觉和自然语言处理领域中用于融合不同基础模型表示的代表性方法，包括 DME [16], SimMMDG [7], 和 MoME [42]。每种方法都利用其融合机制，组合上述四个基础模型（即 OpenAI、Gemini、SigLIP2 和 PE）的表示。随后，将得到的融合表示视为单一表示，并将其作为第 3.1 节所述相同骨干推荐系统的输入。

• **我们的方法：UF²O**。对于 UF²O，我们使用上述四个基础模型，包括两个文本模型（OpenAI 和 Gemini）以及两个视觉模型（SigLIP2 和 PE），这与融合基线所使用的模型一致。需要注意的是，尽管我们使用这四个代表性模型对 UF²O 进行评估，但其形式化方法可以适应数量不同的基础模型。

5.1.3 训练、评估与超参数。 对于静态设置和序列设置中的训练集/验证集/测试集划分，我们采用留一法协议 [15, 44],：使用每位用户倒数第二次交互作为验证集，最后一次交互作为测试集，其余之前的交互作为训练集。对于每个模型–数据集配置，我们使用不同的模型初始化进行五次实验运行。在评估中，我们使用 Recall@K 和 NDCG@K，其中 $K \in \{20,30\}$。为评估 UF²O 性能提升的统计显著性，我们针对每种设置下表现最佳的基线，采用精确的单侧 Wilcoxon 符号秩检验，并使用按随机种子 [47] 配对的五次运行。所有模型均训练 50 个训练周期，并在每个训练周期结束时使用验证集评估各个检查点，

表 2： 序列设置下的推荐性能。**五次实验试验中召回率@K（R@K）和 NDCG@K（N@K）的均值和标准差报告如下，其中 $K \in \{20,30\}$。为便于阅读，所有数值均乘以 100。表现最佳的结果以绿色标出。* 表示在相应设置下，UF²O 根据单侧 Wilcoxon符号秩检验，在统计学上显著优于表现最佳的基线（$p < 0.05$）。**

	方法	体育与户外				玩具与游戏				美容与个人护理				艺术、工艺与缝纫			
		R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30
Single	OpenAI [36]	5.29 (.10)	6.72 (.11)	2.18 (.05)	2.48 (.05)	5.44 (.08)	6.96 (.09)	2.35 (.04)	2.67 (.05)	4.34 (.06)	5.72 (.04)	1.81 (.03)	2.10 (.03)	6.69 (.12)	8.57 (.12)	2.73 (.05)	3.13 (.06)
	Gemini [27]	5.10 (.06)	6.47 (.07)	2.10 (.03)	2.39 (.07)	4.95 (.10)	6.14 (.12)	2.13 (.07)	2.38 (.08)	4.10 (.05)	5.46 (.09)	1.69 (.02)	1.99 (.02)	6.00 (.11)	7.84 (.16)	2.42 (.05)	2.81 (.05)
	SigLIP2 [45]	4.92 (.07)	6.38 (.06)	2.06 (.04)	2.37 (.05)	5.28 (.08)	6.77 (.08)	2.24 (.04)	2.56 (.04)	4.17 (.05)	5.45 (.04)	1.67 (.03)	1.94 (.03)	5.99 (.09)	7.86 (.11)	2.47 (.05)	2.86 (.06)
	位置编码 [1]	4.90 (.06)	6.41 (.04)	2.10 (.02)	2.42 (.02)	5.20 (.10)	6.60 (.13)	2.19 (.04)	2.49 (.05)	4.15 (.06)	5.41 (.06)	1.69 (.02)	1.93 (.02)	6.11 (.14)	7.92 (.15)	2.55 (.04)	2.93 (.04)
Align	OpenCLIP [38]	4.98 (.10)	6.54 (.12)	2.08 (.03)	2.41 (.03)	5.68 (.07)	7.20 (.07)	2.43 (.05)	2.75 (.05)	4.32 (.07)	5.62 (.09)	1.77 (.03)	2.05 (.03)	6.26 (.17)	8.12 (.17)	2.59 (.08)	2.99 (.08)
	MetaCLIP2 [6]	5.26 (.07)	6.84 (.05)	2.17 (.05)	2.50 (.05)	5.88 (.10)	7.61 (.09)	2.53 (.06)	2.89 (.05)	4.58 (.07)	5.88 (.06)	1.91 (.02)	2.19 (.02)	6.82 (.04)	8.75 (.04)	2.80 (.03)	3.21 (.04)
Multi	M³Rec [9]	5.45 (.09)	7.04 (.05)	2.25 (.03)	2.59 (.03)	5.79 (.11)	7.44 (.13)	2.43 (.07)	2.78 (.07)	4.78 (.03)	6.13 (.05)	2.00 (.02)	2.29 (.02)	7.10 (.06)	9.02 (.08)	2.95 (.06)	3.38 (.06)
	DeCoRec [5]	5.24 (.07)	6.77 (.10)	2.22 (.07)	2.55 (.06)	6.30 (.11)	7.85 (.12)	2.73 (.04)	3.09 (.04)	4.88 (.03)	6.18 (.03)	2.04 (.02)	2.30 (.02)	7.23 (.06)	9.13 (.06)	3.01 (.03)	3.41 (.03)
Fusion	Mean	5.30 (.11)	6.74 (.08)	2.30 (.03)	2.62 (.03)	6.28 (.13)	7.86 (.13)	2.72 (.06)	3.06 (.06)	4.50 (.07)	5.86 (.11)	1.88 (.04)	2.17 (.04)	7.16 (.10)	9.00 (.11)	2.96 (.04)	3.35 (.04)
	串接	5.24 (.05)	6.77 (.07)	2.17 (.04)	2.49 (.04)	5.84 (.08)	7.63 (.08)	2.49 (.04)	2.87 (.03)	4.43 (.04)	5.83 (.08)	1.84 (.03)	2.14 (.03)	6.98 (.10)	8.89 (.12)	2.88 (.05)	3.28 (.06)
	DME [16]	5.42 (.06)	6.98 (.16)	2.24 (.06)	2.59 (.06)	5.50 (.07)	7.16 (.12)	1.73 (.02)	2.02 (.03)	4.42 (.02)	5.90 (.03)	1.65 (.02)	1.98 (.02)	6.68 (.14)	8.67 (.11)	2.83 (.07)	3.25 (.06)
	SimMMDG [7]	4.96 (.16)	6.50 (.14)	2.04 (.07)	2.36 (.07)	5.01 (.13)	6.74 (.15)	2.14 (.07)	2.45 (.06)	4.54 (.06)	5.88 (.06)	1.85 (.03)	2.15 (.03)	6.57 (.11)	8.71 (.12)	2.69 (.07)	3.14 (.07)
	MoME [42]	5.19 (.12)	6.78 (.12)	2.18 (.05)	2.51 (.04)	5.33 (.11)	7.02 (.10)	2.22 (.05)	2.58 (.05)	4.65 (.05)	6.10 (.06)	1.93 (.03)	2.24 (.03)	6.92 (.08)	8.90 (.11)	2.90 (.03)	3.35 (.03)
	UF²O（我们的方法）	5.60 (.04)*	7.24 (.05)*	2.34 (.04)*	2.69 (.04)*	6.54 (.06)*	8.23 (.08)*	2.83 (.06)*	3.19 (.06)*	4.94 (.04)*	6.40 (.03)*	2.06 (.02)*		7.22 (.06)		3.04 (.04)*	

我们选择验证集 Recall@30 最高的检查点，并将其用于测试集上的最终评估。我们将 ID-视图-分数平衡超参数 β 和正则化损失平衡超参数 λ 均固定为 1。更多实验细节，包括超参数搜索空间、超参数敏感性分析，以及取得最佳验证集性能的超参数配置，详见补充材料 [21]。

5.2 问题1。静态场景下的性能

如表1所示，UF²O在16种设置中的15种设置下优于所有基线方法，并且在16种设置中的13种设置下，其相较于每种设置中最强基线的提升具有统计显著性。在各基线方法中，作为一种多模态推荐系统的DA-MRS [50], 取得了最佳平均排名。UF²O和DA-MRS相较于改编自NLP和计算机视觉的基础模型融合方法所展现出的出色性能，凸显了针对推荐专门设计融合方法的重要性。然而，在大多数设置下，UF²O的性能优于DA-MRS，这表明，在同一模态内采用多个基础模型，而不是为每种模态依赖单一基础模型，可以进一步提升推荐性能。

5.3 问题2。序列场景下的性能

如表 2 所示，UF²O 在 16 种设置中的 15 种中优于所有基线方法，且在 16 种设置中的 12 种里，其表现提升相较于各设置中最强的基线具有统计显著性。总体而言，各方法的相对表现与静态设置（第 5.2 节）中观察到的情况相似。具体而言，DeCoRec [5],作为一种多模态序列推荐系统，在基线方法中的平均排名最佳。不过，UF²O 在大多数设置中的表现优于 DeCoRec，这表明在同一模态中利用多个基础模型的优势在序列设置中同样成立。

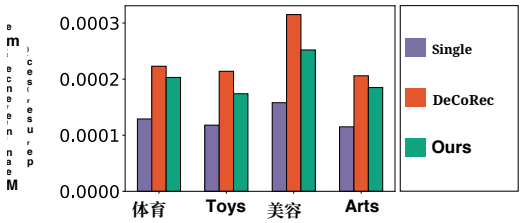


图 4： 推理时间分析。**UF²O 的用户平均推理时间低于最强基线 DeCoRec [5]。此外，与使用单个基础模型（Single）相比，UF²O 为每位用户增加的推理时间不足 10^{-4} 秒，表明其实际开销极小。**

5.4 Q3. 推理时间分析

设置。本节评估 UF²O 的推理速度，这对于在实际场景中部署推荐系统而言是一项重要考量。为此，我们将 UF²O 的推理速度与以下方案进行比较：(i) 使用 OpenAI 嵌入 [36] 的单一基础模型变体；以及 (ii) DeCoRec [5], , 后者是在序列推荐设置中表现最强的基线。

结果。结果见图 4。首先，UF²O 的推理速度快于最强竞争方法 DeCoRec。其次，尽管采用了四个基础模型视图，UF²O 所需的推理时间并非单个基础模型设置的四倍；相反，在各数据集上的平均额外推理时间不足 60%。这是因为：（1）基础模型保持冻结，因此可以预先计算物品元数据嵌入；（2）不同视图的推荐分数可以并行计算。此外，额外推理时间在实践中并不显著，每位用户平均增加的时间不足 10^{-4} 秒。这些结果表明，与使用单个基础模型相比，UF²O 带来的实际开销较小。

5.5 Q4. 用户注意力权重分析

设置。 我们的融合策略的一个关键设计目标，是使 UF²O 能够自适应地确定每个基础

表 3： 消融研究。**Recall@K (R@K) 和 NDCG@K (N@K) 在五次实验中的均值和标准差如下所示，其中** $K \in \{20,30\}$。**为便于阅读，所有数值均乘以 100。表现最佳的结果以绿色突出显示。总体而言，UF²O 在 16 个案例中的 15 个案例中取得了最佳表现。**

方法	体育与户外				玩具与游戏				美容与个人护理				艺术、工艺与缝纫			
	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30	R@20	R@30	N@20	N@30
无文本	5.00 (.10)	6.58 (.11)	2.10 (.05)	2.43 (.04)	5.82 (.11)	7.38 (.12)	2.49 (.04)	2.82 (.04)	4.16 (.10)	5.50 (.09)	1.72 (.04)	2.00 (.03)	6.33 (.07)	8.18 (.06)	2.63 (.02)	3.02 (.03)
无视觉	5.30 (.12)	6.93 (.12)	2.19 (.06)	2.53 (.06)	5.31 (.08)	6.82 (.10)	2.28 (.06)	2.60 (.06)	4.48 (.07)	5.85 (.07)	1.84 (.04)	2.13 (.04)	6.72 (.14)	8.68 (.15)	2.81 (.08)	3.22 (.08)
无正则化	5.52 (.09)	7.15 (.09)	2.25 (.03)	2.61 (.02)	6.09 (.13)	7.78 (.15)	2.66 (.06)	3.02 (.05)	4.64 (.06)	6.09 (.06)	2.00 (.03)	2.31 (.04)	7.06 (.11)	9.04 (.10)	2.96 (.04)	3.38 (.04)
每模态单帧	5.28 (.07)	6.88 (.08)	2.21 (.04)	2.55 (.04)	5.81 (.08)	7.50 (.10)	2.47 (.03)	2.83 (.04)	4.44 (.05)	5.86 (.05)	1.83 (.03)	2.13 (.03)	6.82 (.08)	8.81 (.08)	2.88 (.04)	3.30 (.05)
均值池化	5.28 (.08)	7.12 (.09)	2.20 (.05)	2.52 (.05)	6.41 (.09)	8.00 (.11)	2.65 (.04)	2.99 (.04)	4.62 (.06)	6.01 (.05)	1.90 (.02)	2.21 (.02)	7.05 (.07)	9.11 (.07)	3.00 (.03)	3.41 (.04)
全局池化	5.24 (.05)	7.12 (.05)	2.18 (.03)	2.57 (.04)	6.42 (.08)	8.03 (.09)	2.65 (.05)	3.00 (.05)	4.62 (.04)	5.98 (.04)	1.91 (.03)	2.22 (.02)	7.11 (.08)	9	00 (.03)	3.42 (.03)
物品池化	5.55 (.06)	7.15 (.08)	2.30 (.03)	2.64 (.03)	6.03 (.11)	7.71 (.10)	2.60 (.07)	2.96 (.07)	4.48 (.10)	5.92 (.08)	1.87 (.04)	2.17 (.03)	6.88 (.09)	8.88 (.09)	2.79 (.04)	3.22 (.04)
UF²O（我们的）	5.60 (.04)												21 (.03)	3.		

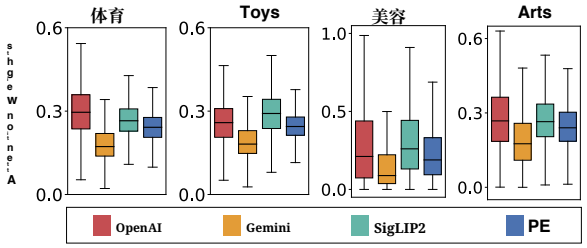


图 5：学习到的用户注意力权重的分布。不同用户和基础模型之间学习到的注意力权重存在显著差异，这表明视图加权行为并非简单模式。

模型对每位用户的贡献。因此，我们预期，学习得到的注意力权重会因用户和基础模型视图而异，而不是几乎保持均匀分布，或在全局范围内几乎完全集中于单个基础模型。为考察这种行为是否会在实践中出现，我们分析了各基础模型在不同用户上的学习所得注意力权重。具体而言，对于每个基础模型$f_{(p)} \in \mathrm{F}$，我们收集其针对所有 $u \in \mathrm{U}$ 的用户专属注意力权重α_{up}（见公式 (1)），并使用箱线图可视化其分布。我们在静态设置下评估这一行为。

结果。如图 5 所示，不同用户学习到的用户注意力权重各不相同，部分权重接近 0.9。这些分布既没有集中在均匀权重（即在我们的设置中为 0.25）附近，也没有普遍由某个单一基础模型主导，这表明 UF²O 能够针对不同用户和基础模型视图学习异质的注意力模式。此外，我们在补充材料 [21] 中提供了另外两项分析，表明学习到的注意力权重具有实际意义。首先，在训练后将学习到的注意力权重在用户之间打乱，会导致 UF²O 的性能下降，这说明这些权重与对应用户存在有意义的关联。其次，与无法使用户受益的基础模型相比，UF²O 往往会为能够使用户受益的基础模型分配更高的注意力权重。

5.6 Q5.消融研究

设置。我们分析 UF²O 中的关键组件对于其强劲性能是否必不可少。为此，我们考虑 UF²O 的七种变体：（1）无文本，排除文本基础模型；（2）无视觉，排除视觉基础模型；（3）无正则化，移除第 4.4 节所述的正则化损失；（4）每模态单帧，每种模态使用一个基础模型；文本使用 OpenAI，视觉使用 SigLIP2；（5）均值池化，以基础模型之间的均匀平均取代用户自适应融合；（6）全局池化，以在所有用户之间共享的可学习全局融合权重取代用户自适应融合；（7）物品池化，以物品自适应融合取代用户自适应融合。我们在序列推荐设置下评估这些变体，遵循第 5.3 节所述的实验设置。

结果。如表 3 所示，UF²O 在 16 种设置中的 15 种下均优于其所有变体。这一结果表明，UF²O 的关键组成部分共同促成了其出色的推荐性能：同时利用文本和图像基础模型、用户自适应融合，以及防止融合权重集中到单一基础模型上的正则化损失。

6 结论与讨论

结论。在本工作中，我们探讨了如何有效融合多个基础模型以用于推荐系统。为此，我们首先开展分析，结果表明，从基础模型中受益的用户因模型而异。受这一观察启发，我们提出了 UF²O，一种针对多个基础模型的用户自适应分数级融合方法，并配套采用一种专门的正则化损失来稳定训练的训练策略。通过广泛的实验，我们证明，在静态和序列推荐设置下，UF²O 在大多数情况下均优于现有的基础模型融合基线和多模态推荐系统基线。

未来工作。我们的分析表明，不同基础模型的优势因用户而异。这些差异可能源于模型架构或训练数据，但部分模型仅披露了有关这些因素的有限信息。分析架构和训练数据集记录更完善的模型，或有助于解释为何不同用户能从不同模型中获益。

致谢。本工作部分得到了韩国政府（MSIT）资助的韩国国家研究基金会（NRF）项目支持（编号：RS-2024-00406985，30%）。本工作部分得到了韩国政府（MSIT）资助的信息与通信技术规划评估研究院（IITP）项目支持（编号：RS-2022-II220871，AI自主性与知识增强以促进AI智能体协作，30%）（编号：RS-2024-00457882，国家人工智能研究实验室项目，30%）（编号：RS-2019-II190075，人工智能研究生院项目（KAIST），10%）。

伦理考量

我们认为，本研究不会引发重大的伦理问题，因为本研究不涉及直接的人体受试者实验（例如众包），且所有实验均使用经过匿名化处理、公开可用并允许学术使用的数据集。此外，本研究中使用的所有基础模型均允许学术使用。而且，所有基线方法的实现均由其原始提供方公开发布，并允许用于学术用途。

参考文献

[1] Daniel Bolya、Po-Yao Huang、Peize Sun、Jang Hyun Cho、Andrea Madotto、Chen Wei、Tengyu Ma、Jiale Zhi、Jathushan Rajasegaran、Hanoona Bangalath 等。2025年。Perception Encoder：最佳视觉嵌入并非位于网络的输出端。载于 NeurIPS。[2] Lu Chen、Peng Yi、Xiaoyue Hou、Cheng Yang 和 Xiongcai Cai。2026年。模态缺失时：面向多模态推荐的早期融合。载于 WSDM。[3] Xiwen Chen、Yen-Chieh Lien、Susan Liu、María Castaños、Abolfazl Razi、Xiaoting Zhao 和 Congzhe Su。2026年。面向多模态和多视图电子商务表示学习的因式分解传输对齐。载于 WSDM。[4] Zheyu Chen、Jinfeng Xu、Hewei Wang、Shuo Yang、Zitong Wan 和 Haibo Hu。2025年。超复数提示感知多模态推荐。载于 CIKM。[5] Zhaoqi Chen、Wanni Xu、Yunfeng Zhang、Yawei Hou、Zhenyu Wen 和 Cong Wang。2025年。DeCoRec：用于多模态序列推荐的解耦协同精炼。载于 MM。[6] Yung-Sung Chuang、Yang Li、Dong Wang、Ching-Feng Yeh、Kehan Lyu、Ramya Raghavendra、Jim Glass、Lifei Huang、Jason Weston、Luke Zettlemoyer 等。2025年。Meta CLIP 2：全球规模扩展方案。载于 NeurIPS。[7] Hao Dong、Ismail Nejjar、Han Sun、Eleni Chatzi 和 Olga Fink。2023年。SimMMDG：一种简单有效的多模态领域泛化框架。载于 NeurIPS。[8] Iker García-Ferrero、Rodrigo Agerri 和 German Rigau。2021年。元嵌入基准测试：哪些方法有效，哪些无效。载于 EMNLP。[9] Xu Guo、Tong Zhang、Yufei Xue、Chenxu Wang、Fuyun Wang 和 Zhen Cui。2025年。M 3 rec：用于多模态序列推荐的混合模态专家选择性状态空间模型。载于 ICASSP。[10] Xiangnan He、Kuan Deng、Xiang Wang、Yan Li、Yongdong Zhang 和 Meng Wang。2020年。LightGCN：简化并增强图卷积网络以用于推荐。载于 SIGIR。[11] Yupeng Hou、Zhankui He、Julian McAuley 和 Wayne Xin Zhao。2023年。学习可迁移序列推荐系统的向量量化物品表示。载于 WWW。[12] Yupeng Hou、Jiacheng Li、Zhankui He、An Yan、Xiusi Chen 和 Julian McAuley。2024年。连接语言与物品以实现检索和推荐。arXiv 预印本 arXiv:2403.03952（2024年）。[13] Guoqing Hu、An Zhang、Shuo Liu、Zhibo Cai、Xun Yang 和 Xiang Wang。2025年。AlphaFuse：在语言嵌入的零空间中学习用于序列推荐的 ID 嵌入。载于 SIGIR。[14] Feiran Huang、Yuanchen Bei、Zhenghang Yang、Junyi Jiang、Hao Chen、Qijie Shen、Senzhang Wang、Fakhri Karray 和 Philip S Yu。2025年。用于冷启动推荐的大语言模型模拟器。载于 WSDM。261–270。[15] Wang-Cheng Kang 和 Julian McAuley。2018年。自注意力序列推荐。载于 ICDM。[16] Douwe Kiela、Changhan Wang 和 Kyunghyun Cho。2018年。用于改进句子表示的动态元嵌入。载于 EMNLP。[17] Kyungho Kim、Sunwoo Kim、Geon Lee、Jinhong Jung 和 Kijung Shin。2025年。多行为推荐系统：综述。载于 PAKDD。[18] Kyungho Kim、Sunwoo Kim、Geon Lee、Shinhwan Kang、Sojeong Kim、Liam Collins、Bhuvesh Kumar、Donald Loveland 和 Kijung Shin。2026年。无需训练的基于大语言模型的推荐：利用协同信号进行大语言模型后物品精炼。载于 CIKM。[19] Kyungho Kim、Sunwoo Kim、Geon Lee 和 Kijung Shin。2024年。提升多视图在捆绑推荐中的利用效果。载于 CIKM。[20] Kyungho Kim、Sunwoo Kim、Geon Lee 和 Kijung Shin。2025年。用于多行为推荐的自监督混合专家框架。载于 CIKM。[21] Sunwoo Kim、Hyunjin Hwang、Geon Lee、Kyungho Kim 和 Kijung Shin。2027年。面向推荐系统的多基础模型用户自适应融合（补充材料）。https://github.com/kswoo97/UFFO。[22] Sunwoo Kim、Hyunjin Hwang 和 Kijung Shin。2026年。面向多模态推荐的基础模型个性化参数高效微调。载于 WWW。[23] Sunwoo Kim、Geon Lee、Kyungho Kim、Liam Collins、Neil Shah 和 Kijung Shin。2026年。面向基于大语言模型的推荐系统的检索增强生成：全面综述。（2026年）。[24] Sunwoo Kim、Geon Lee、Kyungho Kim、Jaemin Yoo 和 Kijung Shin。2026年。Item-rag：用于基于大语言模型的推荐的物品级检索增强生成。在 SIGIR。[25] Sunwoo Kim、Sunkyung Lee、Clark Mingxuan Ju、Donald Loveland、Bhuvesh Kumar、Kijung Shin、Neil Shah 和 Liam Collins。2026年。生成式推荐中大语言模型的记忆行为：观察、影响与训练策略。arXiv 预印本 arXiv:2606.17276（2026年）。[26] Geon Lee、Sunwoo Kim、Kyungho Kim 和 Kijung Shin。2026年。重新思考图协同过滤中的对比学习：局限性与一种简单的补救方法。在 ICML。[27] Jinhyuk Lee、Feiyang Chen、Sahil Dua、Daniel Cer、Madhuri Shanbhogue、Iftekhar Naim、Gustavo Hernández Ábrego、Zhe Li、Kaifeng Chen、Henrique Schechter Vera 等。2025年。Gemini嵌入：来自 Gemini 的通用嵌入。arXiv 预印本 arXiv:2503.07891（2025年）。[28] Hongji Li、Hanwen Du、Youhua Li、Junchen Fu、Chunxiao Li、Ziyi Zhuang、Jiakang Li 和 Yongxin Ni。2025年。教我如何去噪：通过引导校准实现多模态推荐系统去噪的通用框架。在 WSDM。[29] Jiacheng Li、Ming Wang、Jin Li、Jinmiao Fu、Xin Shen、Jingbo Shang 和 Julian McAuley。2023年。文本就是你所需要的一切：学习用于序列推荐的语言表示。在 KDD。[30] Canyi Liu、Tianyi Li、Wei Li、Youchen Zhang、Xiaodong Li 和 Hui Li。2026年。用于增强基于大语言模型的推荐的混合双重语义建模。在 WSDM。[31] Qijiong Liu、Nuo Chen、Tetsuya Sakai 和 Xiao-Ming Wu。2024年。Once：利用开源和闭源大语言模型增强基于内容的推荐。在 WSDM。[32] Qidong Liu、Xian Wu、Wanyu Wang、Yejing Wang、Yuanshao Zhu、Xiangyu Zhao、Feng Tian 和 Yefeng Zheng。2025年。Llmemb：大语言模型可以成为序列推荐的优秀嵌入生成器。在 AAAI。[33] Qidong Liu、Xian Wu、Yejing Wang、Zijian Zhang、Feng Tian、Yefeng Zheng 和 Xiangyu Zhao。2024年。Llm-esr：大语言模型增强长尾序列推荐。在 NeurIPS。[34] Jie Luo、Wenyu Zhang、Xinming Zhang 和 Yuan Fang。2026年。时间感知自适应侧信息融合用于序列推荐。在 WSDM。[35] Rongqing Kenneth Ong 和 Andy WH Khong。2025年。基于频谱的模态表示融合图卷积网络用于多模态推荐。在 WSDM。[36] OpenAI。2026年。text-embedding-3-large模型。https://developers.openai.com/api/docs/models/text-embedding-3-large [37] Matthew E Peters、Mark Neumann、Luke Zettlemoyer 和 Wen-tau Yih。2018年。剖析上下文词嵌入：架构与表示。在 EMNLP。[38] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark 等。2021年。从自然语言监督中学习可迁移视觉模型。在 ICML。[39] Benjamin Ramtoula、Pierre-Yves Lajoie、Paul Newman 和 Daniele De Martini。2025年。奇妙特征及其寻觅之处：一种结合多个基础模型中特征的探测方法。在 NeurIPS。[40] Kelin Ren、Chan-Yang Ju 和 Dong-Ho Lee。2025年。利用多尺度双边注意力进行模态对齐以实现多模态推荐。在 CIKM。[41] Steffen Rendle、Christoph Freudenthaler、Zeno Gantner 和 Lars Schmidt-Thieme。2009年。BPR：基于隐式反馈的贝叶斯个性化排序。在 UAI。[42] Leyang Shen、Gongwei Chen、Rui Shao、Weili Guan 和 Liqiang Nie。2024年。Mome：面向通用多模态大语言模型的多模态专家混合。在 NeurIPS。[43] Leheng Sheng、An Zhang、Yi Zhang、Yuxin Chen、Xiang Wang 和 Tat-Seng Chua。2025年。语言表示可以满足推荐系统的需求：发现与潜力。在 ICLR。[44] Fei Sun、Jun Liu、Jian Wu、Changhua Pei、Xiao Lin、Wenwu Ou 和 Peng Jiang。2019年。BERT4Rec：基于 Transformer 双向编码器表示的序列推荐。在 CIKM。[45] Michael Tschannen、Alexey Gritsenko、Xiao Wang、Muhammad Ferjad Naeem、Ibrahim Alabdulmohsin、Nikhil Parthasarathy、Talfan Evans、Lucas Beyer、Ye Xia、Basil Mustafa 等。2025年。SigLIP 2：具有改进语义理解、定位和密集特征的多语言视觉语言编码器。arXiv 预印本 arXiv:2502.14786（2025年）。[46] Hoang V. Dong、Yuan Fang 和 Hady W Lauw。2025年。用于推荐的用户、物品和评论对齐对比框架。在 WSDM。[47] Frank Wilcoxon。1945年。通过排序方法进行个体比较。《生物统计学通报》1，6（1945年），80–83。[48] Jianye Xie、Lianyong Qi、Weiming Liu、Anqi Wang、Xiaolong Xu、Haolong Xiang、Xuyun Zhang、Wenwen Gong、Yang Zhang、Amin Beheshti 等。2026年。面向多模态新物品推荐的联合相似用户探索与信息性行为引导。在 WWW。[49] Yi Xu、Moyu Zhang、Chenxuan Li、Zhihao Liao、Haibo Xing、Hao Deng、Jinxin Hu、Yu Zhang、Xiaoyi Zeng 和 Jing Zhang。2026年。Mmq：用于语义ID生成和用户行为适应的多模态量化混合分词。在 WSDM。

[50] Guipeng Xv、Xinyu Li、Ruobing Xie、Chen Lin、Chong Liu、Feng Xia、Zhanhui Kang 和 Leyu Lin。2024年。通过对多模态内容和用户反馈进行去噪与对齐，改进多模态推荐系统。在 *KDD*。[51] Zheng Yuan、Fajie Yuan、Yu Song、Youhua Li、Junchen Fu、Fei Yang、Yunzhu Pan 和 Yongxin Ni。2023年。推荐系统的下一步发展方向是什么？重新审视ID与基于模态的推荐模型。在 *SIGIR*。[52] Zhenrui Yue、Huimin Zeng、Yueqi Wang、Julian McAuley 和 Dong Wang。2025年。面向高效多模态推荐的偏好优化检索与排序。在 *KDD*。[53] Leixin Zhang、David Burian、Vojtěch John 和 Ondřej Bojar。2024年。揭示句子嵌入中的语义信息。发表于*LREC-COLING* 意义表示设计研讨会。[54] Lingzi Zhang、Xin Zhou、Zhiwei Zeng 和 Zhiqi Shen。2024年。用于序列推荐的预训练文本嵌入双视图白化。发表于*AAAI*。[55] Kun Zhou、Hui Wang、Wayne Xin Zhao、Yutao Zhu、Sirui Wang、Fuzheng Zhang、Zhongyuan Wang 和 Ji-Rong Wen。2020年。S3-rec：通过互信息最大化实现序列推荐的自监督学习。发表于*CIKM*。[56] Xin Zhou 和 Zhiqi Shen。2023年。两个图的故事：为多模态推荐冻结并去噪图结构。发表于*MM*。