
生成式推荐的模块化解析：从表征到推理的统一范式
A Modular Pipeline for Generative Recommendation:
Representation → Tokenization → Backbone → Training → Inference

目录

生成式推荐简述

从“预测”到“生成”的范式转变

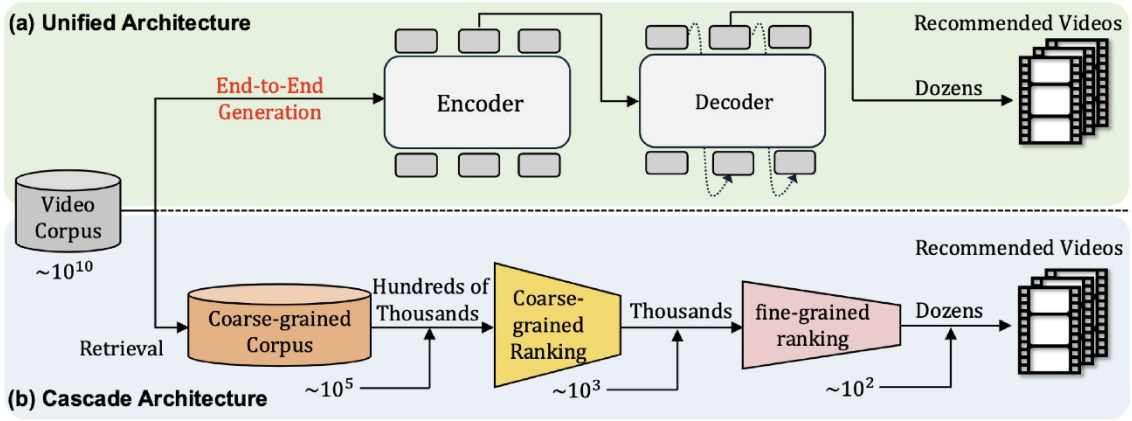
01

1.1 为什么需要生成式推荐（GenRec）？

推荐系统在电商、短视频、新闻、广告等场景中已成为核心基础设施，几乎所有用户触达都离不开推荐服务。传统方法（如协同过滤、矩阵分解、BERT4Rec、LightGCN、SASRec）长期依赖离散 ID-based 表示，在工程落地上取得了显著成功。工业界也形成了“召回 - 粗排 - 精排”的多阶段级联架构，各阶段以不同目标优化，带来了工程可控性，但同时造成整体优化割裂、难以端到端统一建模。

但随着业务规模增长和场景多样化，这一传统范式逐渐暴露出两个根本性问题：

- ID-based 建模限制语义表达与泛化能力: 传统方法将物品视为离散 ID，仅通过 latent 向量建模，缺乏显式语义，难以处理冷启动与跨域场景。
- 多阶段级联导致目标割裂与系统碎片化: 召回、粗排、精排分别基于不同模型与目标训练，整体系统难以获得统一的最优解，只能以各种 heuristic 进行补丁式优化。

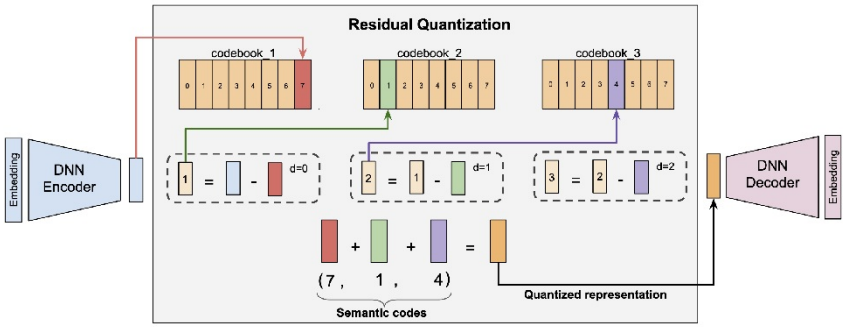


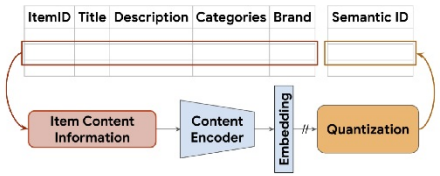
1.2 从“预测”到“生成”：TIGER 的启示

TIGER [1]这一工作可以看作SIDs based生成式推荐的“开山之作”之一，它第一次比较系统地将推荐任务从“预测评分/点击概率”重写为“生成语义 ID 序列”，从而让推荐问题与机器翻译等序列生成任务站在了同一个技术平台上。

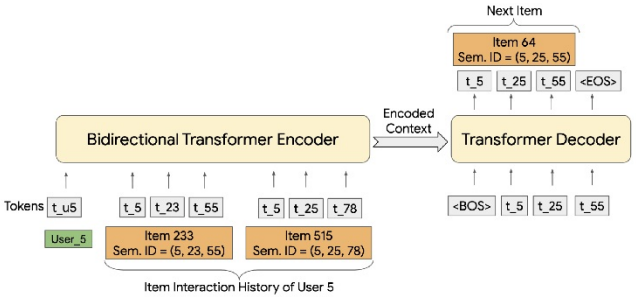
在 TIGER 中，模型先使用 RQ-VAE 这类残差量化结构，把连续的 item embedding 映射为短的 Semantic ID（即由少量离散 code 组成的 token 序列），相当于为每个物品学习了一段更紧凑的“语义拼音”。随后，TIGER 利用 Encoder - Decoder Transformer 对用户历史交互对应的语义 token 序列进行自回归建模，通过类似文本生成的方式一步步生成下一个物品的语义 ID，再通过 beam search 和前缀约束得到最终的 Top-K 推荐列表。

这种改写带来的关键观点是：推荐可以被看作在一个专门构造的语义 token 词表上进行序列生成，而不再局限于对离散 item ID 做分类或打分。





(a) Semantic ID generation for items using quantization of content embeddings.



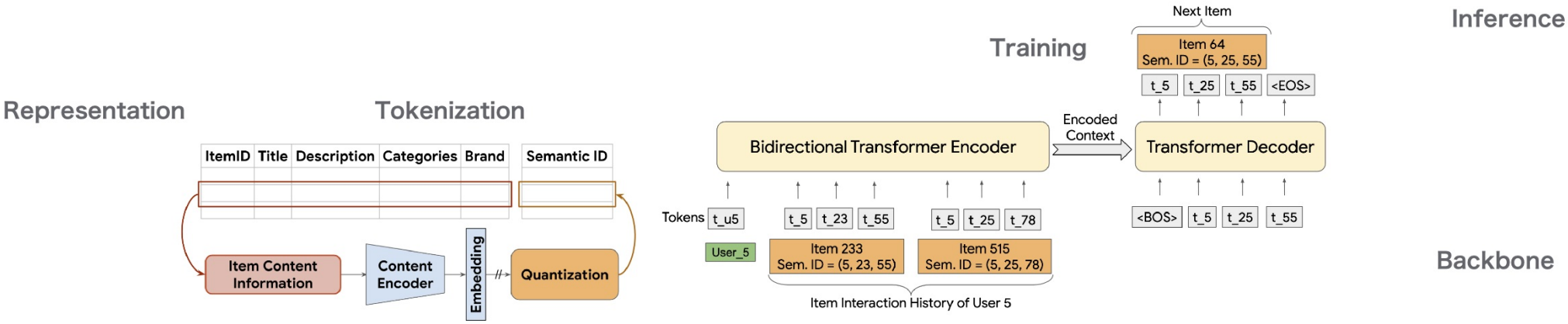
(b) Transformer based encoder-decoder setup for building the sequence-to-sequence model used for generative retrieval.

[1] Recommender Systems with Generative Retrieval

2.1 生成式推荐并不是“一个模型”，而是一条流水线

从系统角度看，任何一个实用的 SID-based GenRec 方法都可以拆解为一条从输入特征到最终推荐列表的流水线，而不仅仅是某个单一的神经网络结构。

Representation → Tokenization → Backbone → Training → Inference



(a) Semantic ID generation for items using quantization of content embeddings.

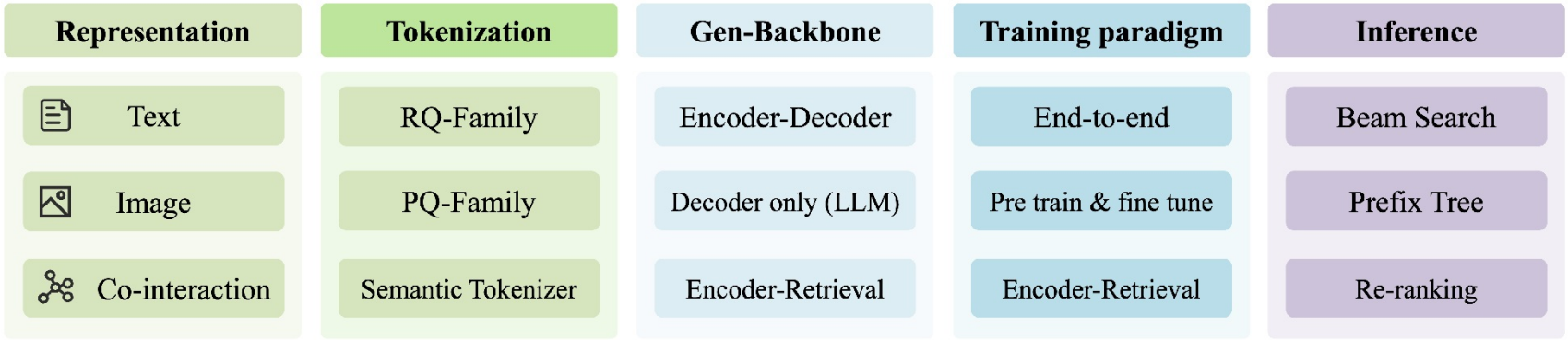
(b) Transformer based encoder-decoder setup for building the sequence-to-sequence model used for generative retrieval.

生成式推荐五阶段模块化框架

Representation → Tokenization → Generation → Training → Inference

02

2.2 生成式推荐五阶段模块化框架



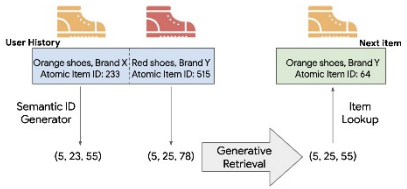
为了梳理和统一现有工作，我们提出一个由**五个阶段组成的模块化框架**，用于刻画生成式推荐从输入到输出的完整流程。

- **Representation** 负责将原始的用户 - 物品信息、文本、图像、图结构等编码为高质量的连续 embedding；
- **Tokenization** 将这些连续表示离散化为语义 token 或 Semantic ID，使其能够被生成模型直接处理；
- **Generative Backbone** 则基于 token 序列建模用户行为，生成下一个 token 或推荐列表；
- **Training Paradigm** 决定整个系统在训练阶段采用何种优化策略，如两阶段、端到端、预训练+微调、多任务联合等；
- **Inference** 则对应线上推理与解码策略，包括 beam search、前缀树结构、Softmax 检索和重排序等。

Recommender Systems with Generative Retrieval (NIPS 23)

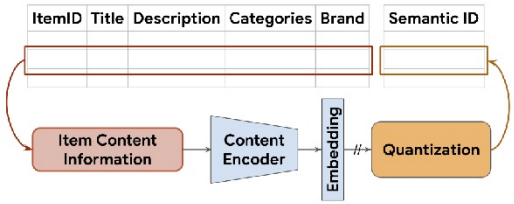
问题

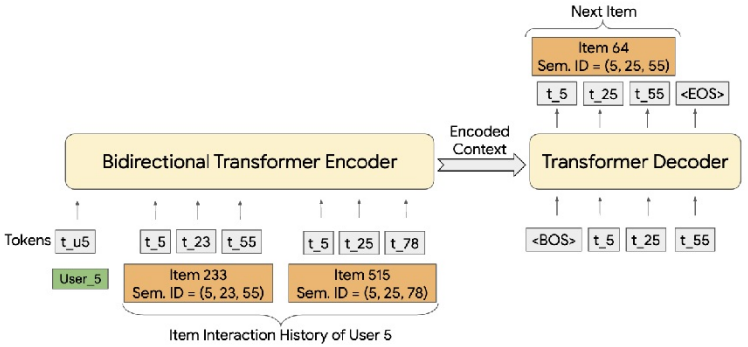
传统召回依赖向量检索，无法直接“生成 item”，缺乏语义表达能力，冷启动严重。
如何构造一种可生成、可泛化、可扩展、并能替代 ItemID 的统一语义离散空间?



TIGER 在五阶段范式下的完整方法对齐

1. Representation（基于内容的连续语义表征）:TIGER 使用 SentenceT5 提取 item 文本 embedding，得到高质量的内容语义表示，这是整个量化和生成的基础。
2. Tokenization（用 RQ-VAE 构造 Semantic ID）:采用多级残差量化（RQ-VAE）将 item embedding 压缩为一个短序列（如 5-23-55）：
    1. 离散化 → 生成友好
    2. 语义相邻性 → 相似 item 共享前缀
    3. 巨大ID空间 → 可覆盖亿级商品
3. Generative Backbone（Transformer 做检索引擎）:将用户行为序列转为 Semantic ID 序列，然后用 Seq2Seq Transformer 直接预测下一条 Semantic ID。模型参数本身成为“内置索引”（Index-in-Model），抛弃传统 ANN + candidate tower。
4. Training Paradigm（标准 Seq2Seq 端到端训练）:使用下一 token 预测（Next Semantic ID Generation）进行端到端训练，不需要额外的多任务、对比学习或两阶段框架，直接让模型学会：按序生成下一个商品的离散 token 序列。
5. Inference（Beam Search 生成候选 item）:在线阶段通过 beam search 生成若干可能的 Semantic ID，然后映射回 item。可以自然控制：1多样性（通过 beam 宽度）2





Representation

从“表征物品”到“压缩信息的最小单位”

03

3 Representation：从“表征物品”到“压缩信息的最小单位”

Representation 是什么?

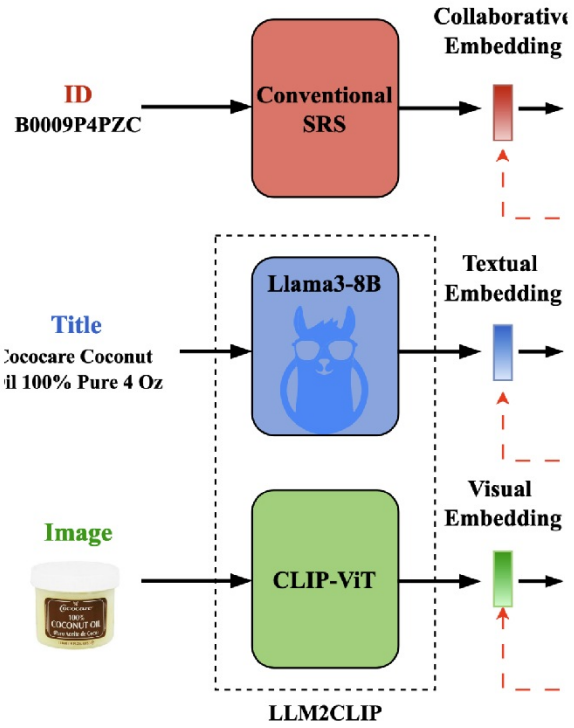
在生成式推荐中，Representation 指将每个物品从原始内容（文本、交互行为、多模态内容）映射到一个 连续的语义向量空间，作为后续 量化（RQ-VAE / PQ / OPQ 等）生成 Semantic IDs 的基础。

Representation的多种特征:

1. 文本语义（Text）:通过标题、描述等文本获取可泛化的语义向量，补足 ID 系方法的冷启动弱点。
   常用提取方式：Sentence-T5 / BERT / MiniLM / T5-encoder 等文本编码器。
   2. 协同信号（Collaborative）:利用用户行为共现关系学习隐式偏好结构，为量化提供用户行为驱动的分布基础。
   常用提取方式：MF / LightGCN / SASRec / 行为序列 Transformer 产生的序列 embedding。
   3. 多模态融合（Multimodal）:融合文本、图像、音频、结构等多源特征，构建更完整统一语义空间。
   常用提取方式：CLIP / ViT / ResNet（图像），MLP（结构特征），跨模态对齐（如 Cross-Attention）。

Representation的两种融合方式:

1.早期融合（Early Fusion）：在量化之前将文本、图像、行为 embedding 直接拼接或加权融合，形成一个统一的连续向量作为 Tokenizer 输入。
   → 优点：表达能力强，量化后 SID 更丰富；缺点：容易被行为噪声污染。
1.晚期融合（Late Fusion）:分别对文本/图像/行为独立编码，再在量化后或下游模型中进行融合。
   → 优点：噪声隔离、模块解耦；缺点：SID 间信息交互弱。



Case Study 1 早期融合

FORGE: Forming Semantic Identifiers for Generative Retrieval in Industrial Datasets

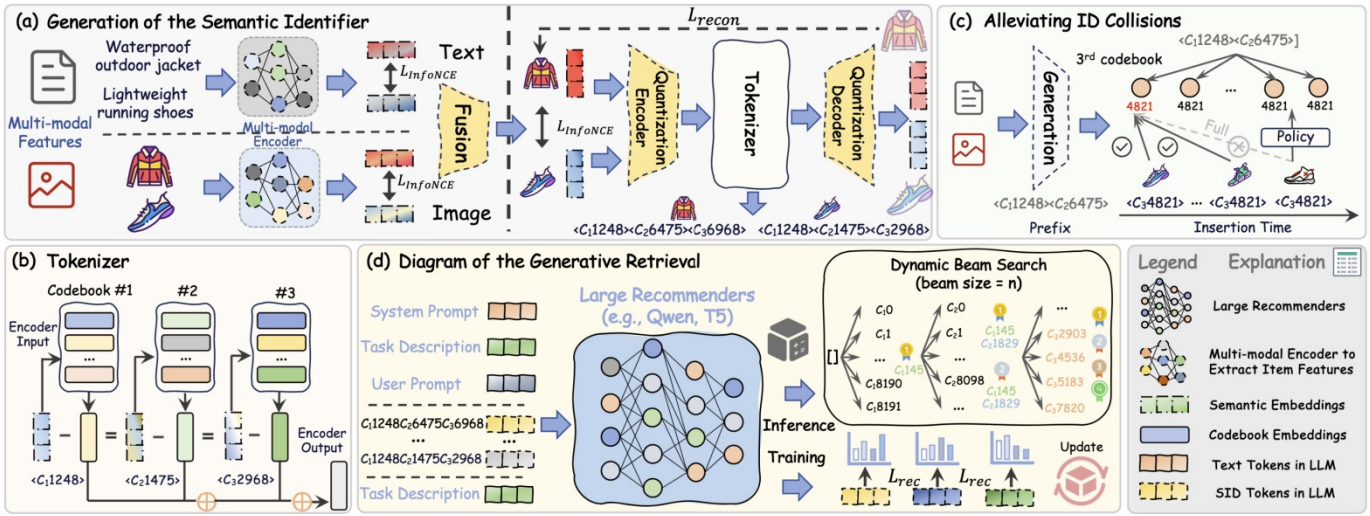
FORGE 首次从“数据、指标、优化、上线”四个层面系统性解决 SID 的工业落地问题，让 Generative Retrieval 真正具备工程可用性。

问题

现有 SID 研究缺乏工业级大规模多模态数据，SID 优化无标准、验证成本高、上线收敛慢。如何在真实超大规模场景下，高效构造、评估并部署高质量的语义 ID（SID）？

方法

1. 构建首个真正工业级 GenRec 数据集
    1. 来自淘宝真实流量
    2. 三模态：ID + 文本 embedding + 图像 embedding
    3. 支持 SID 构造、碰撞研究、GR 训练、搜索任务
2. 系统研究 SID 构造与优化（基于 RQ-VAE）
    1. 不同模态组合方案
    2. 协同信号增强
    3. RQ-VAE 构造 SID 的参数影响
3. 离线 Pretraining 加速工业部署
    1. 预训练 + 上线微调
    2. 将线上收敛时间 减少 50%（从数天 → 一半时间）



Case Study 2 晚期融合

MMQ-v2: Align, Denoise, and Amplify: Adaptive Behavior Mining for Semantic IDs Learning in Recommendation

MMQ-v2 在传统内容量化（VQ/PQ）基础上，加入“自适应对齐 + 混合量化 + 动态权重”，使 SID 更干净、更稳健、更能表达行为偏好。

问题

当前基于内容的 SID（文本/图像量化）无法覆盖真实用户行为，导致推荐对“热门—长尾”差异极度敏感。同时，简单地把行为信号加入内容量化，会让噪声直接侵入内容表征，使 SID 不稳定、不可靠。

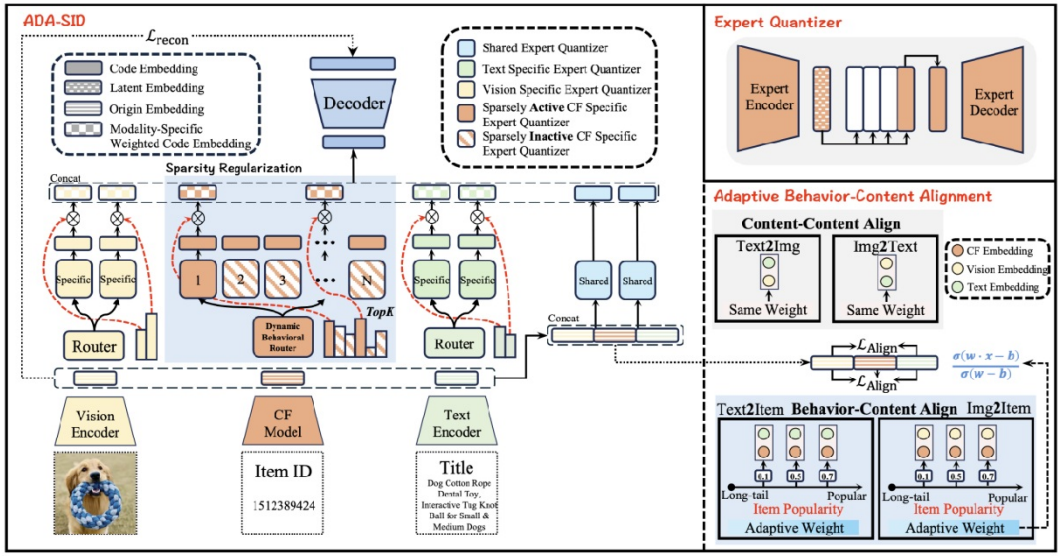
挑战

- 行为信号的质量差异巨大（热门丰富、长尾稀疏），若统一处理就会出现“热门信息被过压缩、长尾信息被放大噪声”的反效果。
- 此外，不同来源的 SID（视觉/文本/行为）在重要性上完全等权，下游模型无法知道“哪些 SID 是有价值的

方法

1. 自适应行为 - 内容对齐（Adaptive Alignment）:根据行为 embedding 的“信量”自动调整对齐强度:
    1. 热门 item 强对齐（保留真实行为结构）
    2. 长尾 item 弱对齐（避免噪声污染内容表示）
2. 混合量化（Mixture-of-Quantization）结构
    1. 共享专家学习跨模态共性（文本 + 视觉 + 行为），
    2. 特定专家学习模态独特性，最终生成 多视角、更稳健的 SID（ADA-SID）。
3. 动态行为路由（Dynamic Router）

    模型自动为每个行为 SID 分配权重：热门 SID 放大、无效长尾 SID 降权，让下游推荐只关注“真正有意义的行为信号”。



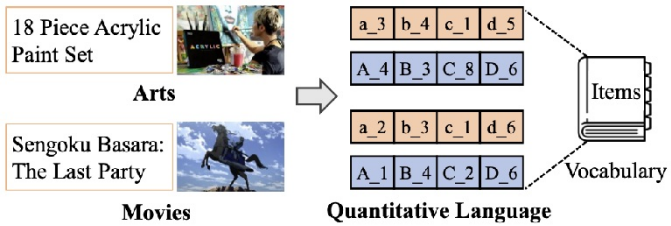
Case Study 3 晚期融合

MULTIMODAL QUANTITATIVE LANGUAGE FOR GENERATIVE RECOMMENDATION (ICLR 2025)

MQL4GRec 通过“量化语言”把文本与图像映射到统一 token 空间，让生成式推荐第一次实现真正的跨模态、跨领域知识迁移。

问题

现有基于 PLM 的生成式推荐直接用自然语言 token 做物品 ID，但 PLM 的通用语言知识与推荐任务存在巨大鸿沟。同时，多模态信息（文本、图像）之间缺乏统一表征，难以捕捉用户多维偏好，也阻碍跨领域知识迁移。

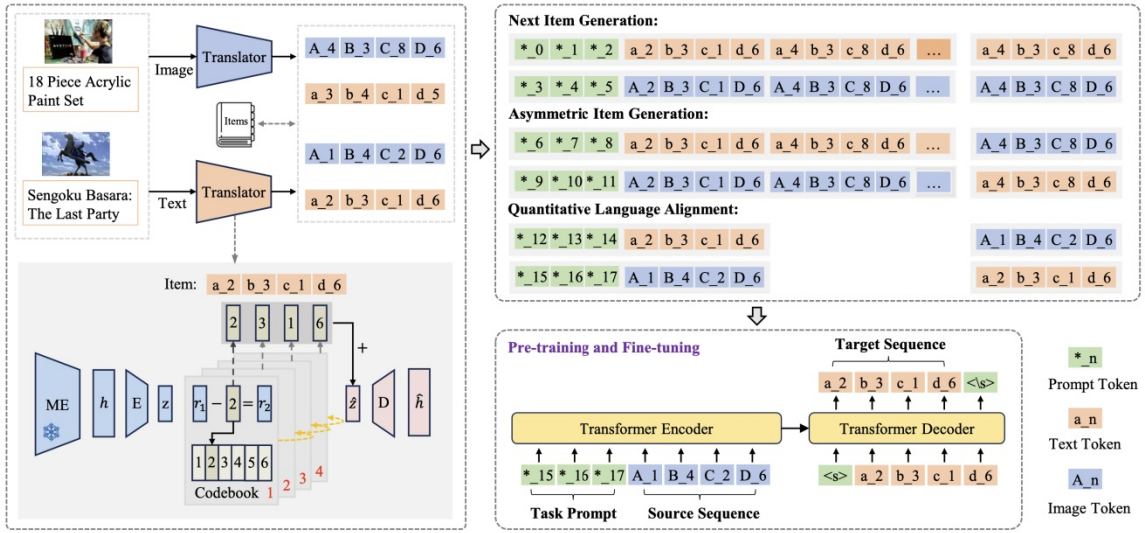


挑战

- PLM 无法自然理解商品的图像与文本结构，将原始内容直接生成为 token 会导致冗长、噪声大、难训练。
- 不同模态之间没有统一“语言空间”，推荐知识无法在文本→图像→跨域之间迁移。

方法

1. 多模态量化翻译器（Quantitative Translator）：将文本与图像分别编码→再用 RQ-VAE 量化成 统一的“量化语言 Token”（如 a_2、B_4…）所有模态共享同一词表，让多域、多模态商品可以相互“对话”。
2. 量化语言生成任务（Quantitative Language Tasks）:设计多种任务来注入推荐语义, 这些任务让量化语言逐渐具备“可理解、可推理、可迁移”的推荐知识:
    1. Next-item Generation（主任务）
    2. Asymmetric Generation（跨模态生成）
    3. Text↔Image Alignment（显式对齐）
3. 预训练 → 微调（Knowledge Transfer Pipeline）:先在源域做量化生成预训练，迁移跨域/跨模态知识；再在目标域微调，捕捉当前用户偏好。最终的多模态结果通过 re-ranking 融合。



Tokenization

设计推荐系统的“可生成语言”

04

4.1 Tokenization:量化方法

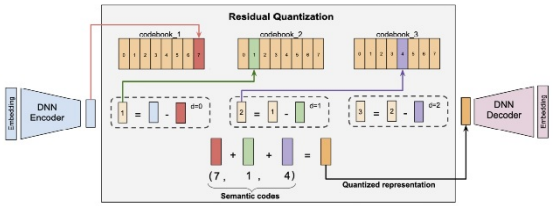
短语义 ID（RQ-VAE / RKMeans）vs 长语义 ID（PQ ）

在生成式推荐中，“量化”不仅决定离散 token 的形式，更决定 semantic ID 的结构形态。

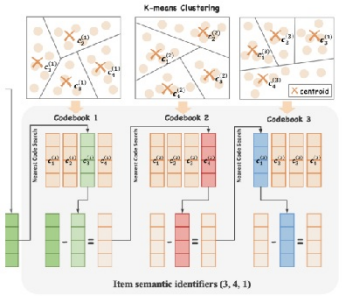
从整体来看，量化方法主要分成 两大阵营：

（1）短语义 ID：4 - 8 token、结构上天然有层次: RQ-VAE / R-KMeans / RVQ / VQVAE

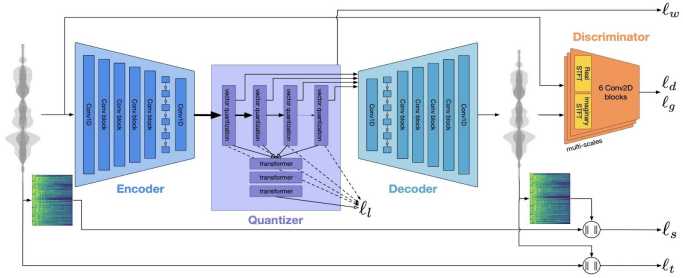
（2）长语义 ID：32 - 64 token、结构完全平行: Product Quantization



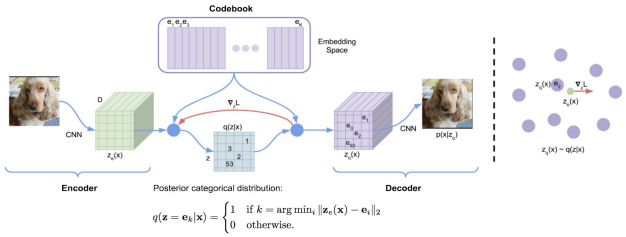
(1) RQ-VAE



(2) R-KMeans

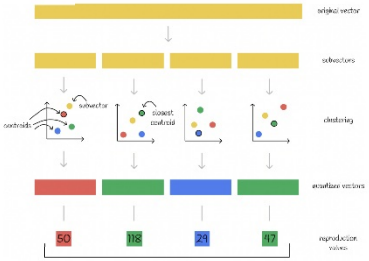


(3) RVQ



$$q(\mathbf{z}=\mathbf{e}_k|\mathbf{x}) = \begin{cases} 1 & \text{if } k = \arg\min_i \|\mathbf{z}_e(\mathbf{x}) - \mathbf{e}_i\|_2 \\ 0 & \text{otherwise.} \end{cases}$$

(4) VQ-VAE



(4) Product Quantization

Case Study 1 R-KMeans在生成式推荐的应用

OneRec: Unifying Retrieve and Rank with Generative Recommender and Preference Alignment

OneRec 以“Session 级生成 + MoE 扩容 + DPO 偏好对齐”打破了传统三级级联模式，实现了真正意义上的端到端工业级生成式推荐。

问题

传统工业级推荐依赖“召回 - 粗排 - 精排”多阶段级联，各阶段目标独立、优化割裂，使整体效果受限于最弱环节。
现有生成式推荐虽能直接生成候选，但仅能作为召回器使用，难以替代复杂的多级排序体系，精排效果不足。

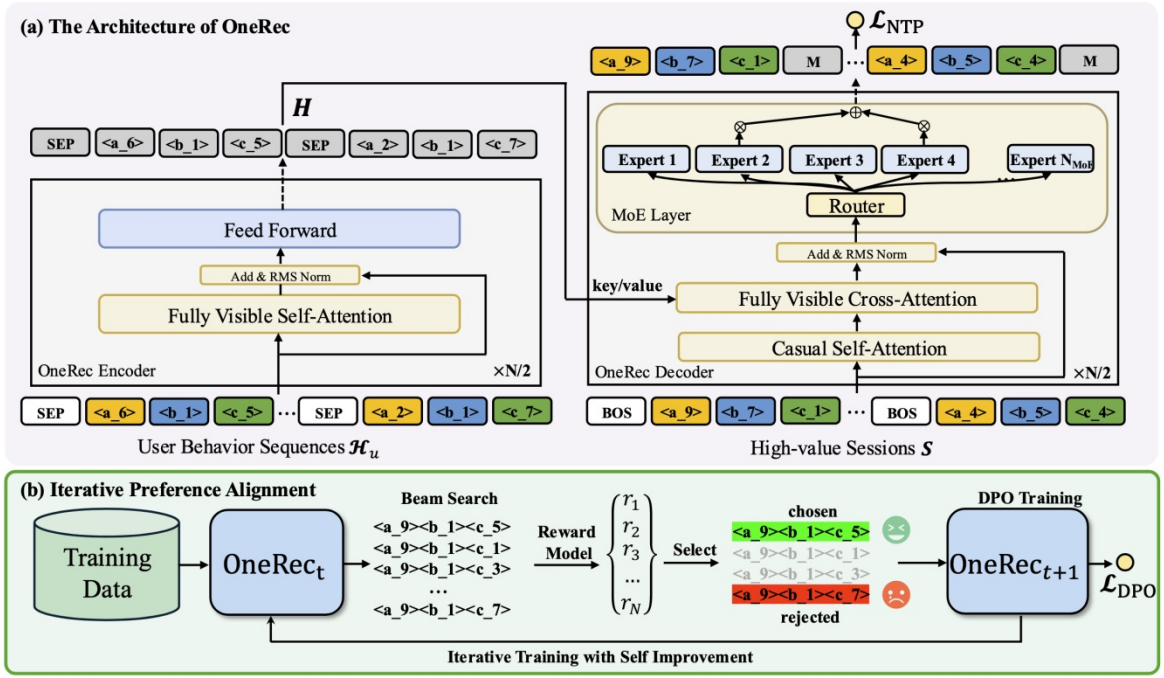
挑战

- 要让生成模型同时承担召回与排序，需要更强的模型容量、更长序列建模能力，以及跨 item 的整体 session 级理解。
- 同时，推荐系统缺乏 NLP 场景中常见的大规模人工偏好数据，使得对生成结果进行“偏好对齐（preference alignment）”极具挑战。

方法

用单一的 Encoder - Decoder + MoE 大模型直接生成整个推荐 Session，并通过 DPO 对齐用户偏好。

- 端到端 Session-wise 生成（替代多级排序）:不再逐个预测“下一个 item”，而是一次生成整段 session（5 - 10 个视频）。结合 MoE 扩展模型容量，使模型能同时完成召回与精排的语义建模。
- Iterative Preference Alignment（IPA）+ DPO: 用 Reward Model 模拟用户观看偏好，从 beam search 结果中自动构造“胜/负”偏好对。迭代式自提升（self-improvement）使模型逐步对齐真实用户行为，显著提升生成结果质量。



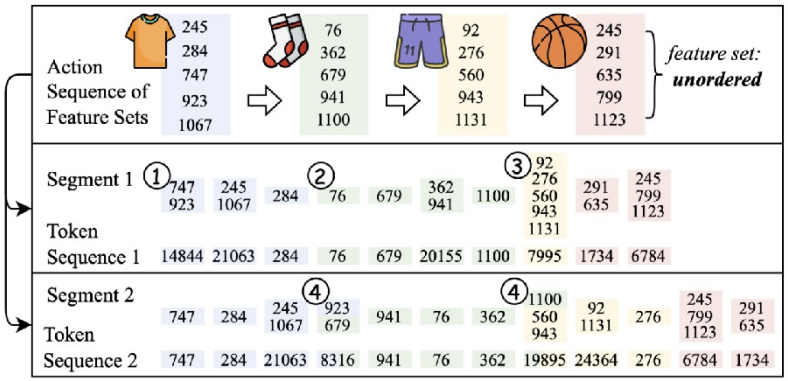
Case Study 2 PQ在生成式推荐的应用

ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation (ICML 2025)

基于PQ的量化方法，ActionPiece 通过对无序特征集合进行上下文重组，实现了既紧凑又语义敏感的离散化 token 表示。

问题

现有生成式推荐（GenRec）使用的 Tokenizer 都是 上下文无关的：同一个 item 在不同序列中总是被分成相同的 token pattern，无法表达上下文语义（如品牌搭配、价格区间、风格一致性等）。导致生成模型负担加重，推荐质量受限。



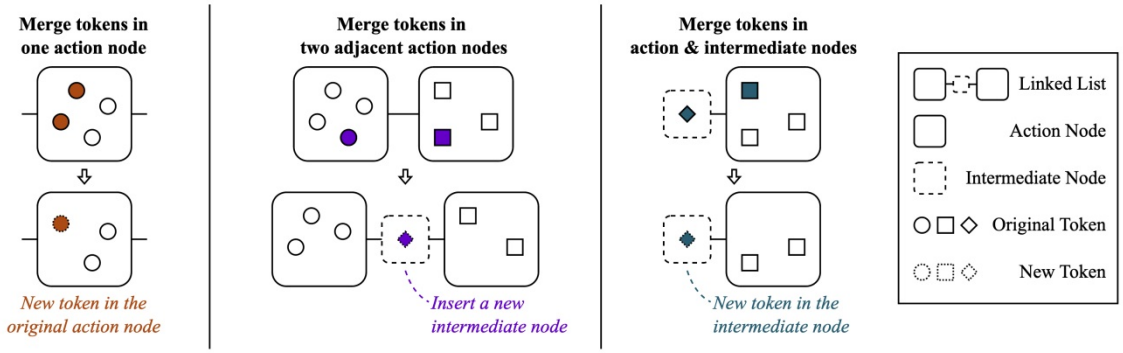
挑战

- 动作是“无序特征集合”（brand、category、price…），不能简单做 BPE 式序列 Tokenization。
- 需要同时建模“动作内部”和“相邻动作”之间的特征共现。
- 单一 segmentation 会导致 token 使用不均衡，影响训练与推理。

方法

ActionPiece 提出 上下文感知（context-aware）动作 Tokenizer：

- 把 item 表示成无序特征集合（feature set）
- 按权重统计“集合内”和“相邻集合间”特征共现 → 动态合并成新的 token
- 引入中间节点结构 处理跨 item 合并

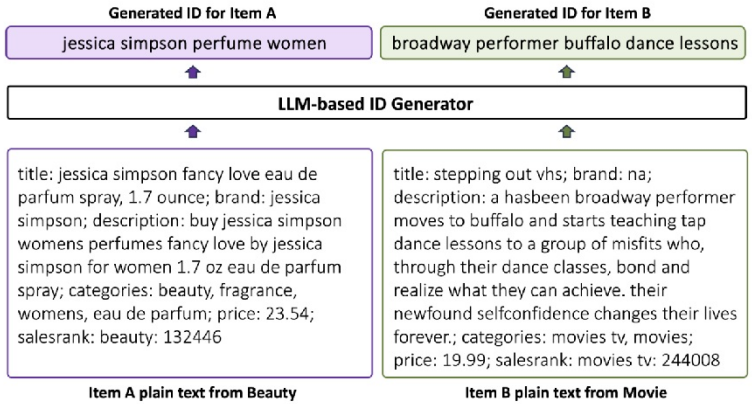


4.2 Tokenization: 语义 Tokenization 与文本 ID

量化并不是唯一的离散化方式。当推荐系统越来越“语言化”，新的方向开始出现：**让 item 直接拥有可读性强、语义友好的“文本式 Token ID”**。这让 Tokenization 从数据处理步骤，演变为一种“推荐语言的设计”。

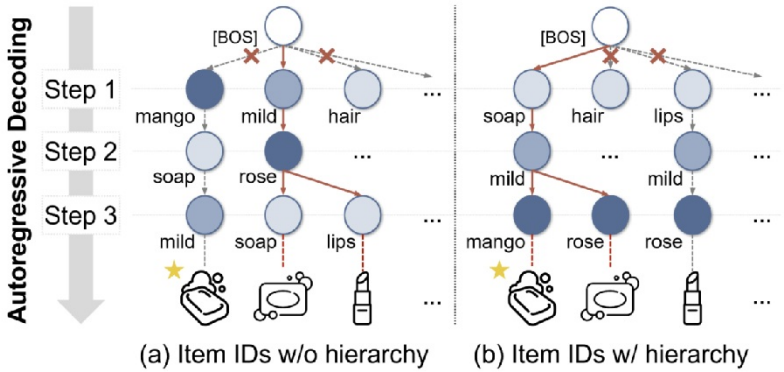
文本式 Token：LLM 直接命名 item

- IDGenRec[1]: 让 LLM 给每个 item 生成一个短文本式的 ID（如短名称、概念短句）



多粒度 Tokenization：层级化语义ID

- GRAM[2]: 通过“语义→词汇”的多粒度文本化翻译构建层次可读的 Item ID



[1]IDGenRec: LLM-RecSys Alignment with Textual ID Learning http://arxiv.org/abs/2403.19021

[2]GRAM: Generative Recommendation via Semantic-aware Multi-granular Late Fusion http://arxiv.org/abs/2506.01673

Case Study 1 Token语义化的尝试

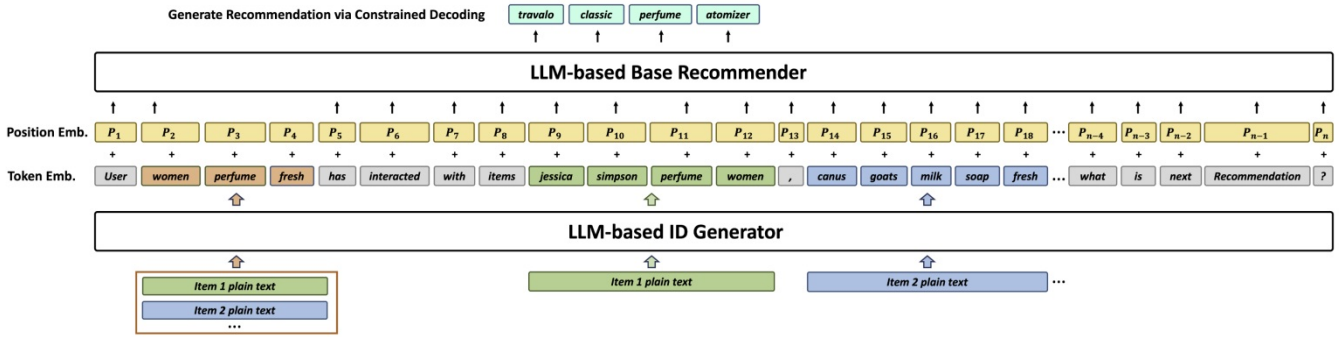
IDGenRec: LLM-RecSys Alignment with Textual ID Learning (SIGIR 2024)

IDGenRec 用语言模型把 item 的文本压缩成短、小、唯一的‘语义化 Textual ID’，相当于把 PQ 的‘向量压缩’变成 LLM 里的‘文本语义压缩’。

问题

现有生成式推荐框架仍依赖 无语义的数字型 Item ID（OOV tokens），与 LLM 的语言空间完全脱节：

- 无法让 LLM 利用其丰富的语义理解能力
- Item 表示缺乏可迁移性，不支持跨平台、跨数据集
- 训练出的模型几乎 不具备零样本泛化能力



挑战

- 如何从冗长、含噪声的 item metadata 中抽取 简短且有判别性 的文本 ID?
- 如何确保所有 item 的 ID 唯一、不冲突?
- 如何使“ID 生成器”与“LLM 推荐模型” 协同优化、目标一致?

方法

1）Textual ID Generator（基于 T5）: 利用 item 文本生成 简短、语义明确、可读的自然语言 ID，替代 OOV 数字 token。

2）Diverse ID Generation（确保 ID 唯一）:引入 多样性 beam search，并动态调整搜索多样性与 ID 长度，确保所有 item 的 ID 唯一。

3）Alternating Training（交替训练）:采用“推荐模型训练 → ID 生成器训练”的交替优化策略，实现 ID 与推荐模型的 双向对齐。

Case Study 2 Token语义的层级划分

GRAM: Generative Recommendation via Semantic-aware Multi-granular Late Fusion (ACL 2025)

GRAM通过“语义转文本 + 多粒度晚融合”让 LLM 同时理解 item 的层次关系与丰富内容，实现高语义保真、低复杂度的生成式推荐。

问题

当前的生成式推荐方法普遍存在两类核心限制：

- 无法充分建模隐式的 item 关系:现有 ID 构建方式（数值 ID / 压缩 ID）缺乏层次性与协同语义，LLM 很难理解 item 之间的真实关联。
- 无法有效利用 item 丰富但冗长的文本信息: 将完整 metadata 直接输入 LLM 会导致序列过长、成本极高，因此现有方法都只用部分属性，造成语义损失。

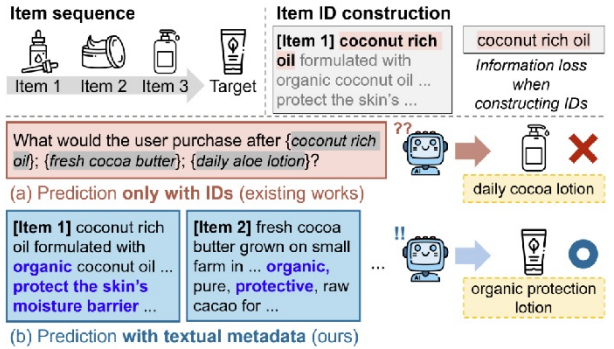
挑战

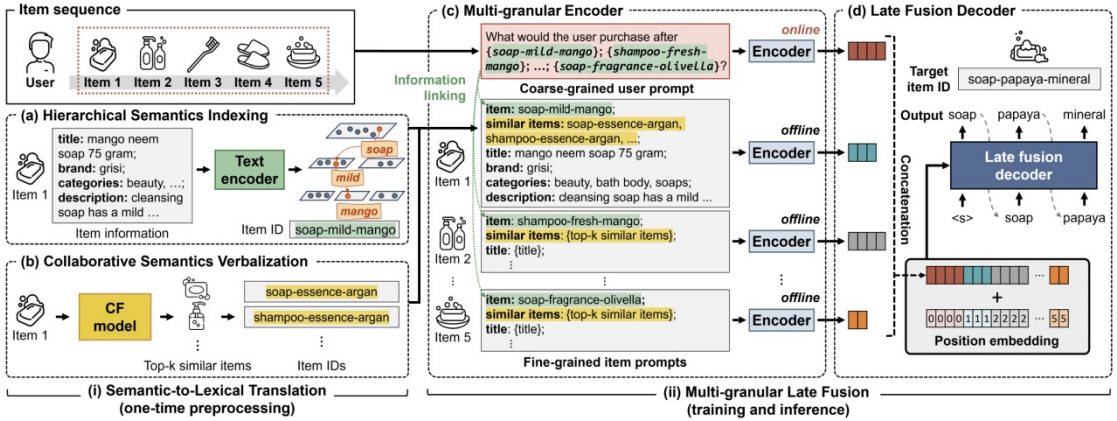
- 如何把 item 的层次语义（taxonomy、语义聚类）与协同语义（相似 item）转化成 LLM 能理解的 textual token?
- 如何在不显著增加序列长度的前提下利用丰富的 item metadata?
- 如何避免“早期融合”的平方级复杂度，让 LLM 能处理长序列?

方法

核心思想：“把 item 的结构化语义转成简洁文本，再用多粒度方式输入，让 LLM 在解码时整合信息。”

- 语义转文本（Semantic→Lexical）：把 item 的层次关系和相似关系翻译成简短文本 ID 或文本属性。
- 多粒度晚融合（Late Fusion）：粗粒度用短文本表示用户历史，细粒度用长文本表示 item 内容，在解码阶段再融合，避免长序列开销。





Generative Backbone

从 Encoder–Decoder 到 LLM：生成架构的演进

05

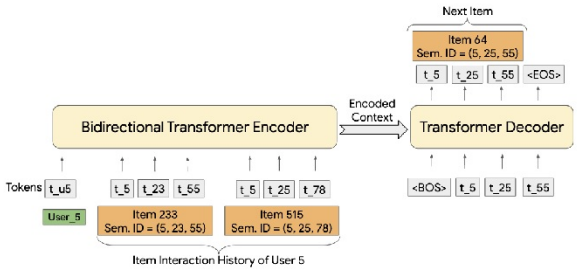
5 Generative Backbone-生成骨干的不同架构

随着推荐从“预测”走向“生成”，生成骨干的结构也随之出现分化, 不同的架构直接决定了模型的 表达能力、训练效率与推理方式。

Encoder - Decoder

Encoder - Decoder 结构延续了机器翻译的成功范式:

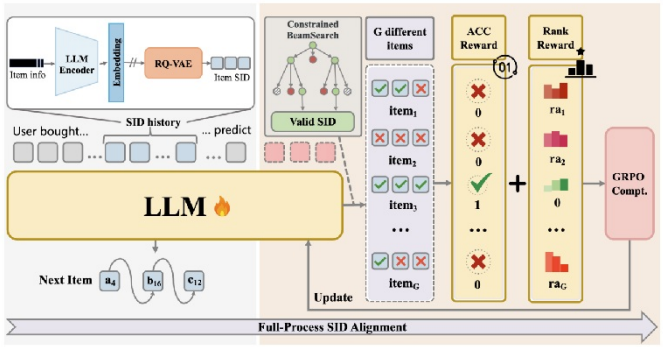
- Encoder 负责吸收并整合用户历史、上下文、时序模式;
- Decoder 再逐步生成下一物品的语义 token。
- TIGER[1]



Decoder-only

Decoder-only 结构本质上是 GPT 类自回归模型容易复用已有 LLM 权重（如 Qwen、LLaMA），适合 指令微调式 GenRec。

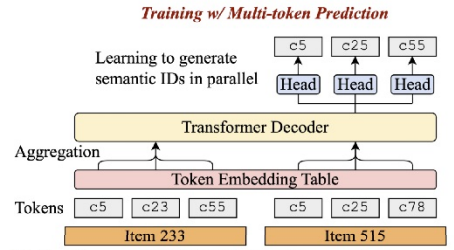
- MiniOneRec: 基于LLM的指令微调推荐框架



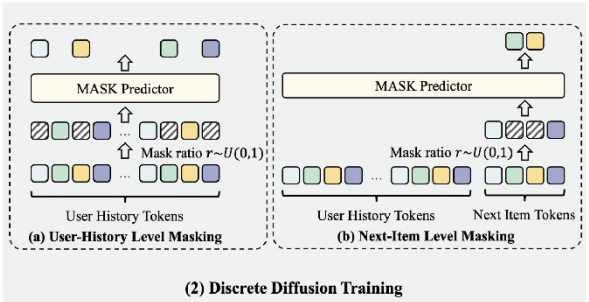
Hybrid：Parallel prediction

将 Tokenizer、Backbone、推荐生成统一为一个可训练链路，从输入 embedding 到输出语义 ID 全部可端到端优化。

- RPG: 长SIDs + 并行预测 + 相似度匹配



- LLaDA-Rec: 离散扩散模型mask-based预测每一位



[1]Recommender Systems with Generative Retrieval https://arxiv.org/pdf/2305.05065
[2]MiniOneRec: An Open-Source Framework for Scaling Generative Recommendation http://arxiv.org/abs/2510.24431
[3]Generating Long Semantic IDs in Parallel for Recommendation http://arxiv.org/abs/2506.05781
[4]LLaDA-Rec: Discrete Diffusion for Parallel Semantic ID Generation in Generative Recommendation http://arxiv.org/abs/251

Case Study 1 Encoder + 检索的应用

Generating Long Semantic IDs in Parallel for Recommendation

RPG 以“无序长 ID + 并行生成 + 图约束解码”实现高表达语义 ID 与高效推理.

问题

现有生成式推荐依赖自回归 + Beam Search 推理，每生成一个 token 都需要一次前向，导致延迟高、无法支持更长的语义 ID。因此大多数模型只能使用 4-token 的短语义 ID，表达能力有限，难以充分刻画复杂的 item 语义。

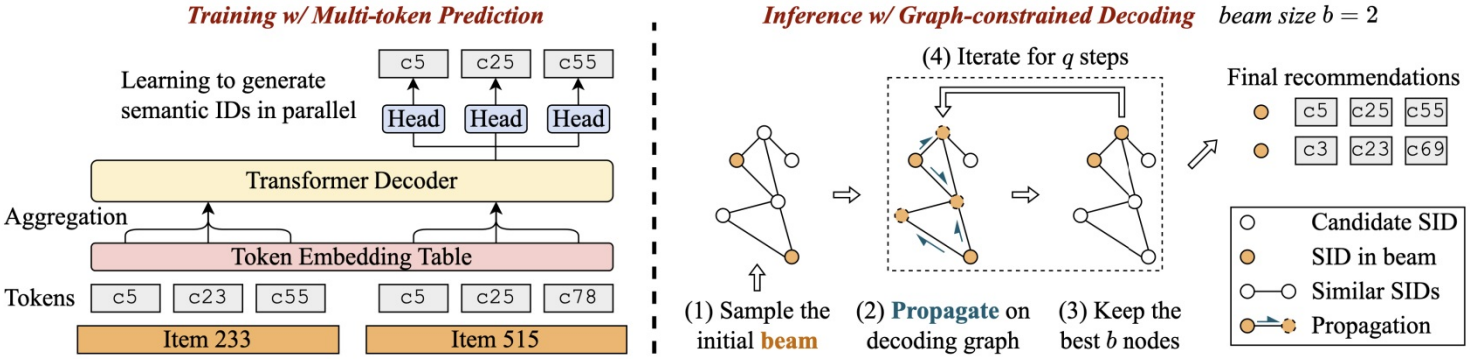
挑战

并行生成会导致语义 ID 的 解码空间极其稀疏（如 256^32 级别），模型很难从独立预测的 tokens 组合出真实存在的 item。
同时，长语义 ID 是“无序”的，传统基于 token 顺序的 beam search 完全无法使用，需要一种新的高效解码机制。

方法

把语义 ID 设计为“无序长向量”并用多 token 并行预测，再通过图约束解码找到真实 item。

- 无序长语义 ID（基于 OPQ）：使用 OPQ 将 item 表达拆为最多 64 个独立 token，使每一位都可独立预测，避免自回归瓶颈。
- 并行生成 + 图约束解码（Graph-Constrained Decoding）：用 Multi-Token Prediction（MTP）一次性预测所有 digits;
- 推理阶段构建“相似语义 ID 图”，通过迭代图传播找到最可能的合法 item。



Case Study 2 并行预测的应用

LLaDA-Rec: Discrete Diffusion for Parallel Semantic ID Generation in Generative Recommendation

LLaDA-Rec 以“并行量化 + 双向扩散 + 动态纠错”的方式突破自回归瓶颈，是生成式推荐从串行走向并行的新范式。

问题

当前生成式推荐依赖 左到右的自回归生成，存在单向信息受限、误差累积、语义建模不足等结构性瓶颈。

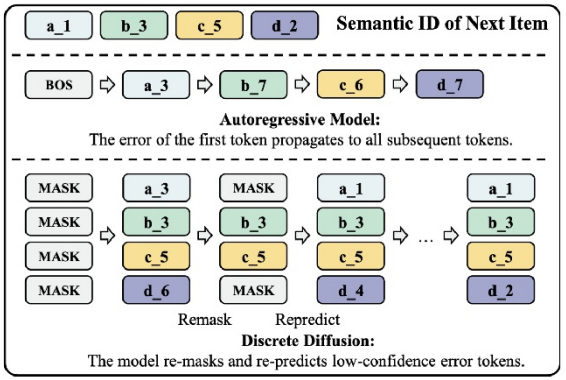
挑战

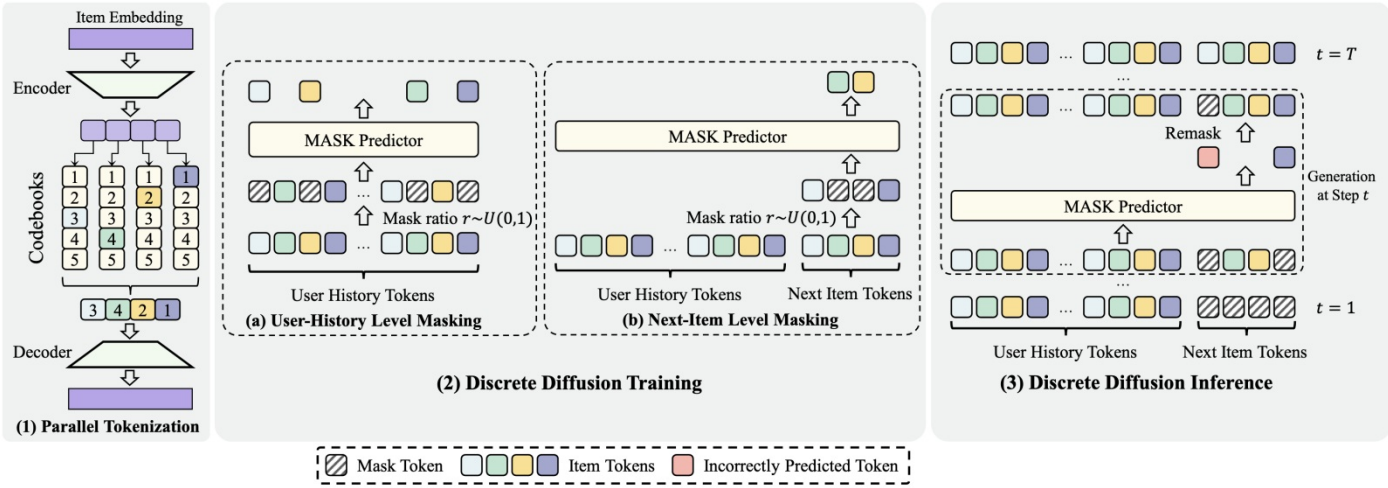
如何实现 并行、双向、可自校正 的语义 ID 生成，同时兼容推荐中固定长度的离散 token 表达?

方法

用离散扩散模型替代自回归，让 SID 生成从“定序串行”变为“并行预测 + 动态纠错”。

- 并行 Tokenization（Multi-Head VQ-VAE）：将 item 表达拆成多头子向量，各自量化，得到适配双向建模的“平行 SID”。
- 离散扩散生成（Dual-Mask + Adaptive Beam）：训练阶段用两类掩码（用户历史 & 下一物品）学习 全局语义 + 局部细粒度语义；
- 推理阶段 所有 token 并行预测，低置信度 token 自动重置并重新生成，并配合扩散式 beam search 得到 Top-K。





Training Paradigm

两阶段 → 联合 → 预训练：训练流程的自然演进

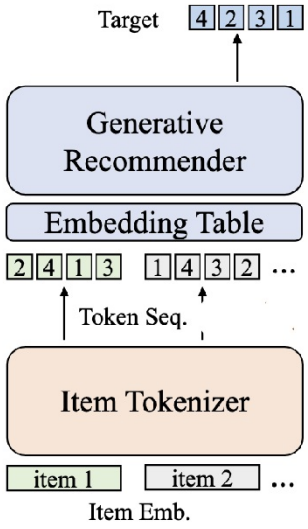
06

6 Training Paradigms-训练范式

生成式推荐的训练范式正在从传统的“两阶段流程（先量化、后推荐）”演化为一个更加统一、可对齐、可扩展的训练体系。

Two-Stage Training

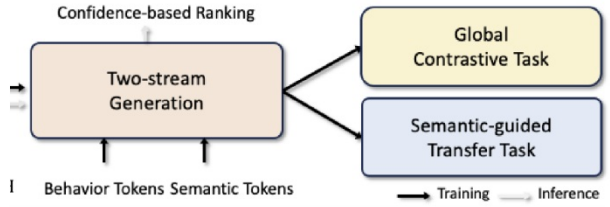
先训练 Tokenizer（VQ-VAE、RQ-VAE、PQ/OPQ），再固定 Tokenizer 训练生成式推荐模型。最普遍的模式



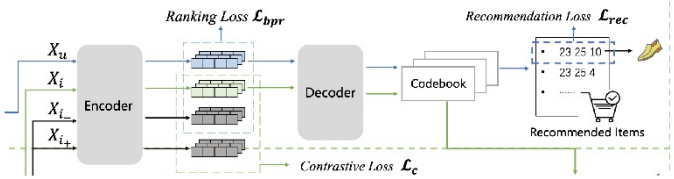
Joint / Multi-Task Learning

同时优化多个目标，在多重优化中对齐语义，协同等信息。

- EAGER[1]: 行为流 + 语义流双路生成架构，通过协作学习融合两类信号。



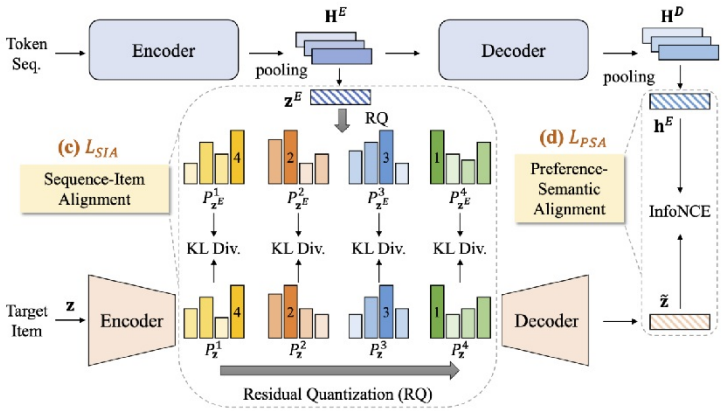
- ColaRec[2]: 联合学习推荐 + 索引的多任务生成框架



End-to-End Training

将 Tokenizer、Backbone、推荐生成统一为一个可训练链路，从输入 embedding 到输出语义 ID 全部可端到端优化。

- ETEGRec[3]: 同时优化量化和推荐backbone

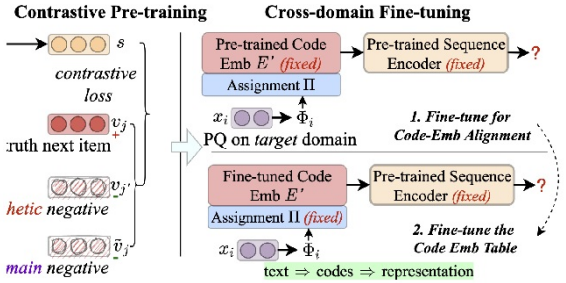


[1]EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration http://arxiv.org/abs/2406.14017
[2]Content-Based Collaborative Generation for Recommender Systems http://arxiv.org/abs/2403.18480
[3]Generative Recommender with End-to-End Learnable Item Tokenization http://arxiv.org/abs/2409.05546

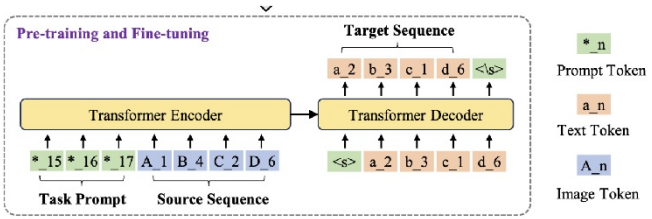
Pretrain → Finetune

在大规模交互/文本/多模态数据上自监督预训练，再在目标推荐任务上微调。

- VQ-Rec[1]: 通过预训练+微调对齐不同域的特征



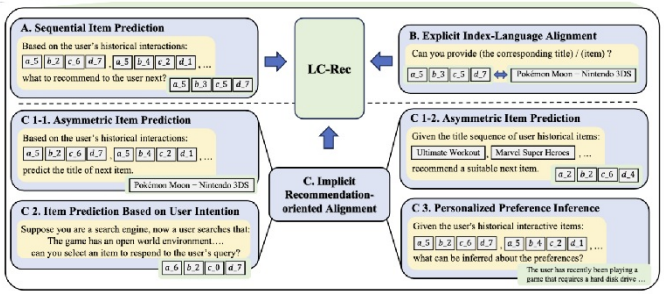
- MQL4GRec[2]: 通过预训练+微调对齐不同模态的码本



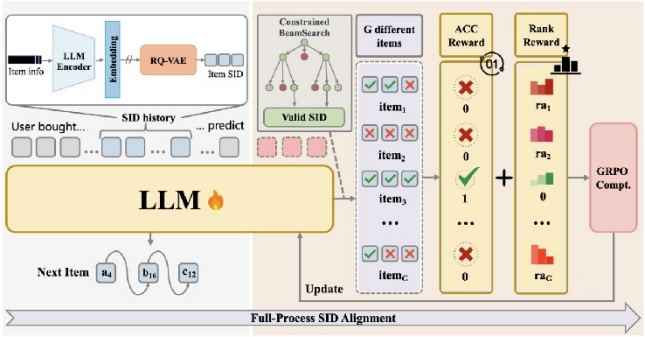
Instruction Tuning

使用自然语言指令，让生成式推荐模型学会“按用户需求生成推荐”。

- LC-Rec[3]: 通过指令微调为SIDs注入语义，协同等信息



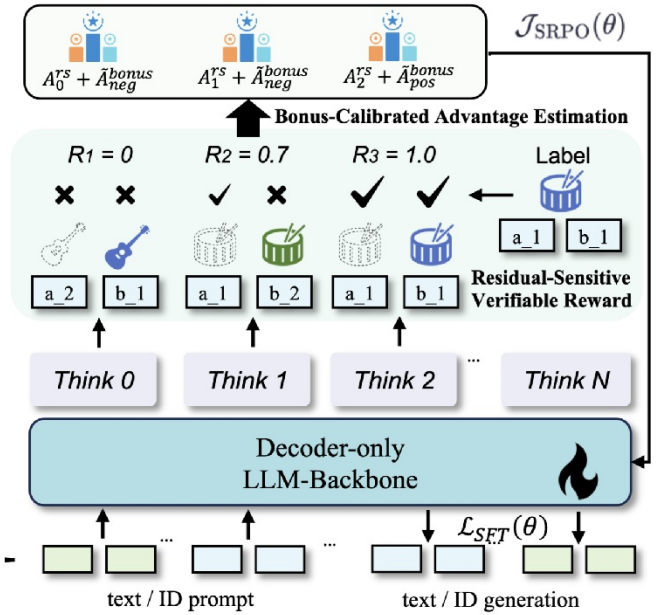
- MiniOneRec[4]: SIDs对齐LLM, 强化式偏好优化



Reinforcement Learning

让LLM与SIDs以及协同信息等通过强化学习和推理的方式对齐。

- GREAM[5]: 基于SIDs层级结构利用稀疏可验证奖励进行 RL 优化



[1]Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders http://arxiv.org/abs/2210.12316
[2]MULTIMODAL QUANTITATIVE LANGUAGE FOR GENERATIVE RECOMMENDATION https://arxiv.org/pdf/2504.05314
[3]Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation http://arxiv.org/abs/2311.09049
[4]MiniOneRec: An Open-Source Framework for Scaling Generative Recommendation http://arxiv.org/abs/2510.24431
[5]Generative Reasoning Recommendation via LLMs http://arxiv.org/abs/2510.20815

Case Study 1 端到端的生成式推荐

Generative Recommender with End-to-End Learnable Item Tokenization

ETEGRec = RQ-VAE Tokenizer + T5-style 生成模型 + 双重对齐机制 + 交替训练，实现真正端到端的生成式推荐。

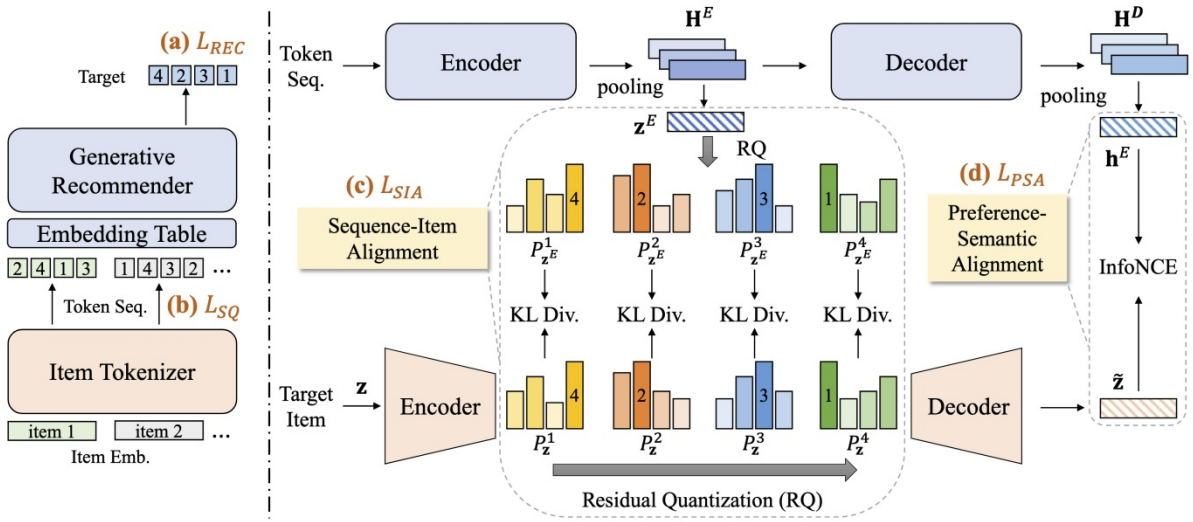
问题

现有生成式推荐 Tokenization 与生成模型分离，互不优化，导致语义不对齐、推荐效果不稳定。

挑战

如何让 item token（量化） 和 生成器（Transformer） 在同一个系统中 共同学习、互相增强?

方法



1. End-to-End 双模块（RQ-VAE + Transformer）
    1. Tokenizer：用 RQ-VAE 把 item embedding 转成多级 token
    2. Recommender：T5/Transformer 生成下一个 item token
    3. Tokenization 与生成器统一训练，不再分离
2. Recommendation-Oriented Alignment（核心创新）
    1. SIA（序列 - 物品对齐）：序列状态的 token 分布要接近目标 item→ token 更能反映行为信号
    2. PSA（偏好 - 语义对齐）：decoder 偏好向量要匹配 item 语义→ 生成器更懂 codebook 语义
3. Alternating Optimization（稳定训练）
    1. 先训 Tokenizer → 冻结, 再训 Recommender → 解冻
    2. 两者逐步对齐，训练稳定且效果更好

Case Study 2 LLM微调对齐推荐

Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation

ETEGRec = RQ-VAE Tokenizer + T5-style 生成模型 + 双重对齐机制 + 交替训练，实现真正端到端的生成式推荐。

问题

传统 LLM 只理解 语言语义，而推荐系统依赖 协同语义（共现、序列行为、嵌入空间）

挑战

tem ID 不在 LLM 词表中：LLM 无法“生成”推荐物品。语言语义 ≠ 协同语义：文本相似≠行为相似。

简单 fine-tuning 无法桥接语义鸿沟：模型只会表面拟合，无法真正理解 collaborative signal。

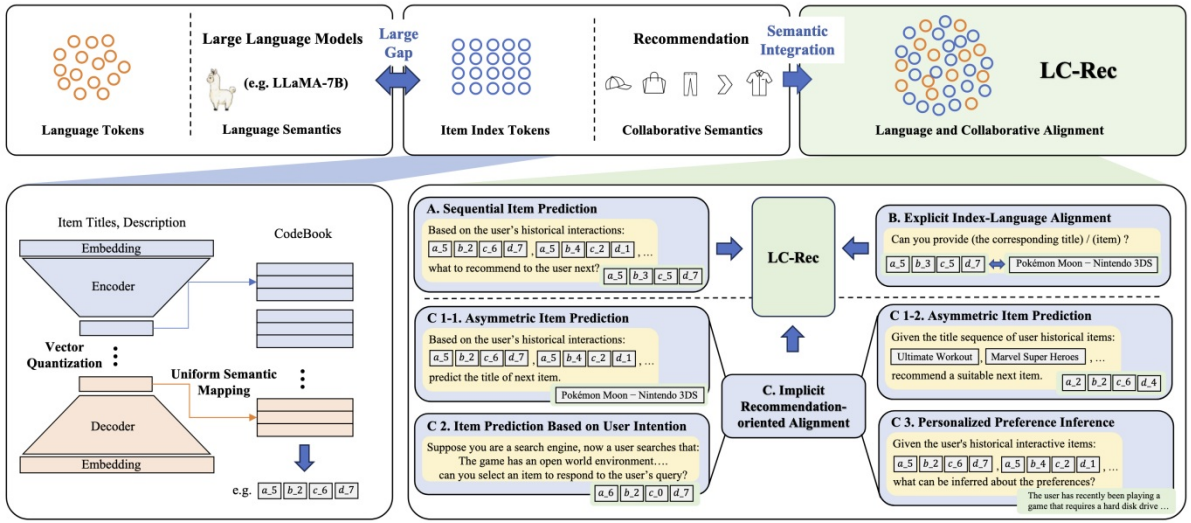
方法

Tree-Structured Vector Quantization (VQ) Item Indexing

- 用 LLaMA 提取 item 文本 embedding → RQ-VAE 建树状多级 codebook。
- 每个 item 对应多个 token（如 a_5 b_2 c_6 d_7 ），可被 LLM 直接生成。
- 通过 uniform semantic mapping 保证不同 item 不冲突、分布更均匀。

  Explicit Index - Language Alignment（显式语义对齐）
- 让 LLM 从标题生成 item index。再让 LLM 从 index 反推标题与描述。→ 把 item index 拉进 LLM 的语言语义空间。

  Implicit Recommendation-Oriented Alignment（隐式行为语义对齐）
- 设计多种推荐相关任务，让 LLM 真正吸收 collaborative signal:
- 序列预测（index→index）



Case Study 3 多任务联合训练

Content-Based Collaborative Generation for Recommender Systems

ColaRec 用“协同构造的 GID + 内容对齐任务”实现内容与协同信号的统一生成式推荐框架，大幅提升 GenRec 的稳定性与可用性。

问题

现有 GenRec 要么只利用内容（content-based GID），要么只利用协同信号（CF-based GID），两者缺乏统一生成框架，导致推荐质量受限。

挑战

内容与协同割裂： 文本表示与交互信号来自不同语义空间，GID 很难同时包含两类信息。

缺乏对齐机制： 内容空间 ↔ 协同空间 没有显式映射，导致量化 token 不稳定、生成效果差。

方法

1.Collaborative-GID Construction（协同驱动 GID 构建）

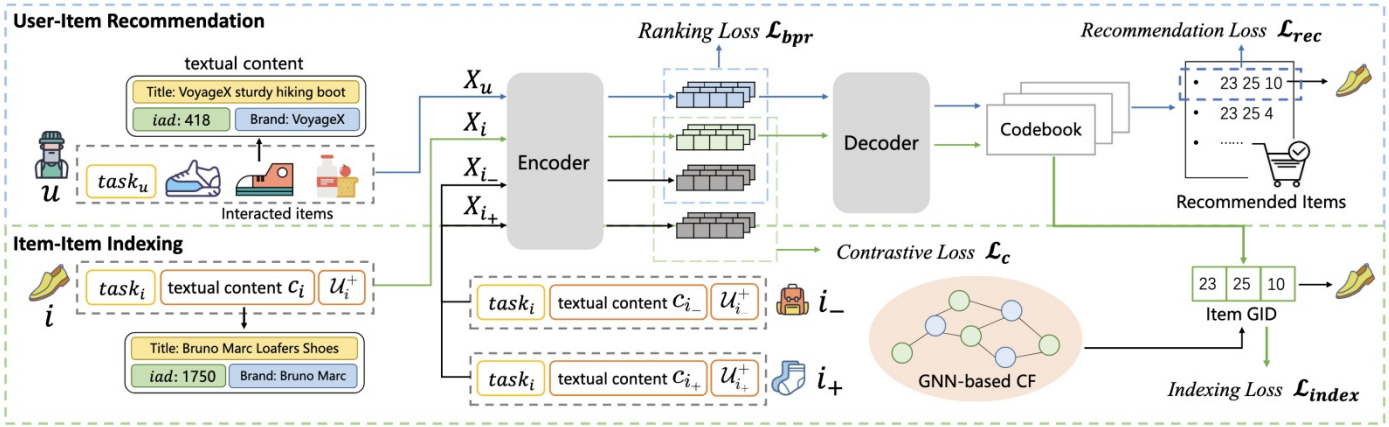
1.先用 LightGCN 学到协同 embedding

2.再用 层次化 K-Means 构造 GID（树路径作为 token 序列）→ 让 GID 天然编码协同结构，比随机/内容 GID 更稳、更有效

2.Content - Collaborative Alignment（内容 ↔ 协同显式对齐）

1.Item Indexing Task： 用同一个 encoder - decoder 将 item 文本映射回其协同 GID

2.Contrastive Loss： GID 相似 → 文本 embedding 也要相似→ 内容信息和协同信号在同一生成框架统一建模



Case Study 4 强化学习微调

Generative Reasoning Recommendation via LLMs

GREAM 通过“协同对齐 + CoT 推理 + 稳定 RL”实现可解释、可生成、可优化的 LLM 推荐

问题

LLM 虽具强推理能力，但与推荐系统的 协同语义、稀疏反馈、离散 item space 存在巨大鸿沟，无法直接作为“可生成、可推理”的推荐模型。

挑战

GREAM 通过“协同对齐 + CoT 推理 + 稳定 RL”实现可解释、可生成、可优化的 LLM 推荐，是当前最完整的“推理式 GenRec”框架。

语义错位： LLM 的语言空间与推荐的协同空间不对齐，item index 不稳定、难以生成。

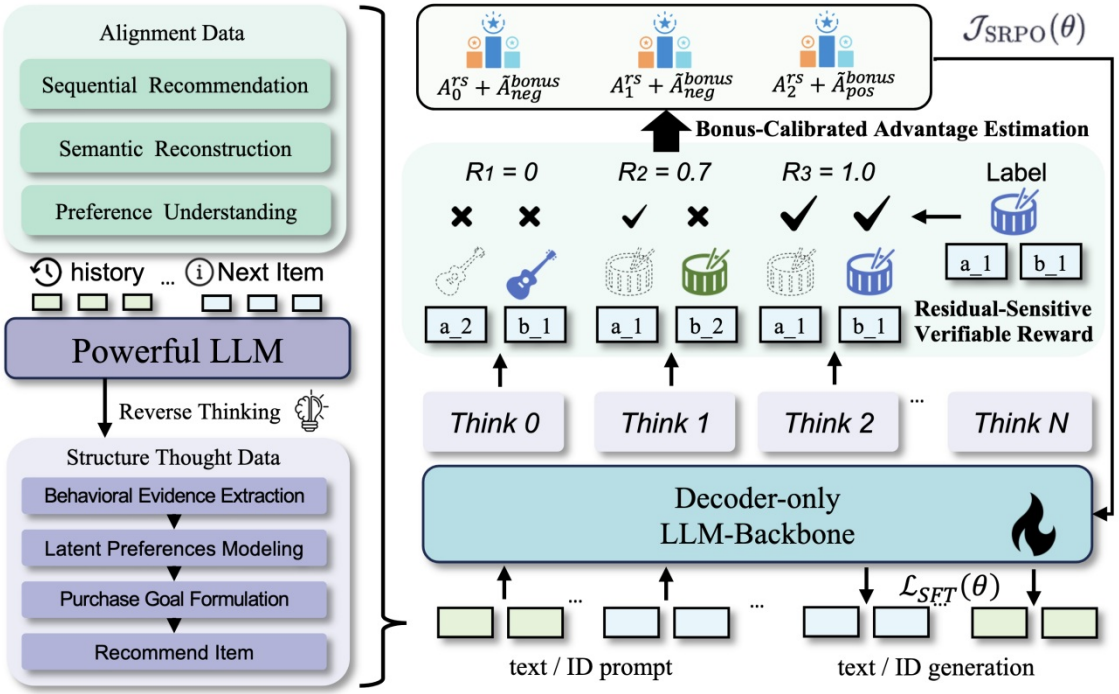
推理缺失： 传统 GenRec 只做模式匹配，没有“因果推理链”，难以解释。

反馈稀疏： RLVR 在推荐中极不稳定，真实点击稀疏且难以提供可验证奖励。

方法

1. Collaborative - Semantic Alignment（语义构建 + 对齐）
    1. 通过 多源文本融合（标题+描述+评论）→ 高保真 item embedding
    2. 使用 RQ-KMeans 构建稳定离散 item index
    3. 设计少量对齐任务（如 index↔text、序列预测）→ 让 LLM 同时掌握语言语义 + 协同语义
2. Reasoning + RL Optimization（推理激活 + 稳定强化学习）
    1. 构造 结构化 Chain-of-Thought（5 阶段）：行为证据 → 潜在偏好 → 需求推断 → 推荐推理 → 序列去噪
    2. 引入 SRPO（Sparse-Regularized PPO）：
        1. residual-sensitive reward（前缀级奖励）
        2. bonus-calibrated advantage（稀疏成功的强化信号）

    → 让 LLM 能做可解释的因果推理，并在稀疏奖励下稳定学习



Inference

从生成到检索：生成式推荐的高效推理策略

07

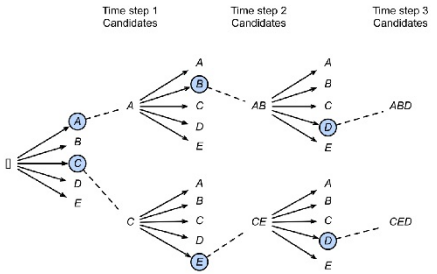
6 Inference - 推理与解码

生成式推荐的推理机制从早期的简单生成式解码，逐渐演变为结构约束、多路径探索与检索增强的混合推理方式。

Beam Search

TIGER[1]

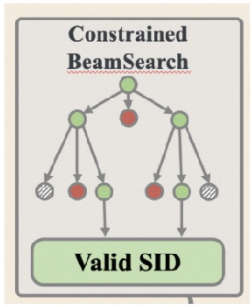
- 自回归方式逐 token 生成 Semantic ID
- 保留 top-K beams，扩展到完整的 Semantic ID 序列
- 可自动控制多样性
- 易受 token 逐位误差累积影响
- 推理成本与 Semantic ID 长度呈线性增长



Constraint Beam Search: Trie

MiniOneRec[2]

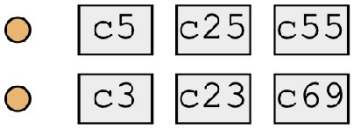
- 使用 Trie（前缀树）约束生成路径
- 只允许生成 合法的 semantic ID 前缀
- 防止产生不存在的 item（illegal IDs）
- 显著减少搜索空间，提高效率



Matching-based Inference

RPG[3]

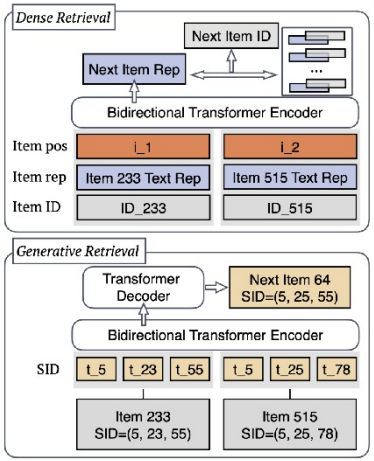
- 不生成 semantic ID，而是计算 item 得分
- Dense Retrieval: dot(user_emb, item_emb)
- 本质为 token 空间或 embedding 空间的相似度匹配
- 推理速度极快，可直接 ANN 检索



Two-stage Re-ranking

LIGER[4]

- **Stage 1：Candidate Generation**
  Beam Search / Matching（semantic ID 或 embedding）
- **Stage 2：Dense Re-ranker**
  根据 item 文本 / embedding / 图信息重新评分
- 可显著提升准确率与冷启动表现
- LIGER 是典型的 CapRec 方式



[1]Recommender Systems with Generative Retrieval https://arxiv.org/pdf/2305.05065
[2]MiniOneRec: An Open-Source Framework for Scaling Generative Recommendation http://arxiv.org/abs/2510.24431
[3]Generating Long Semantic IDs in Parallel for Recommendation http://arxiv.org/abs/2506.05781
[4]Unifying Generative and Dense Retrieval for Sequential Recommendation http://arxiv.org/abs/2411.18814

Case Study 1

MiniOneRec: An Open-Source Framework for Scaling Generative Recommendation

MiniOneRec 是第一个在公开数据集上验证生成式推荐 Scaling Law 的开源系统，通过“SID 全流程对齐 + 强化偏好优化”，显著超越现有生成式和 LLM 推荐模型。

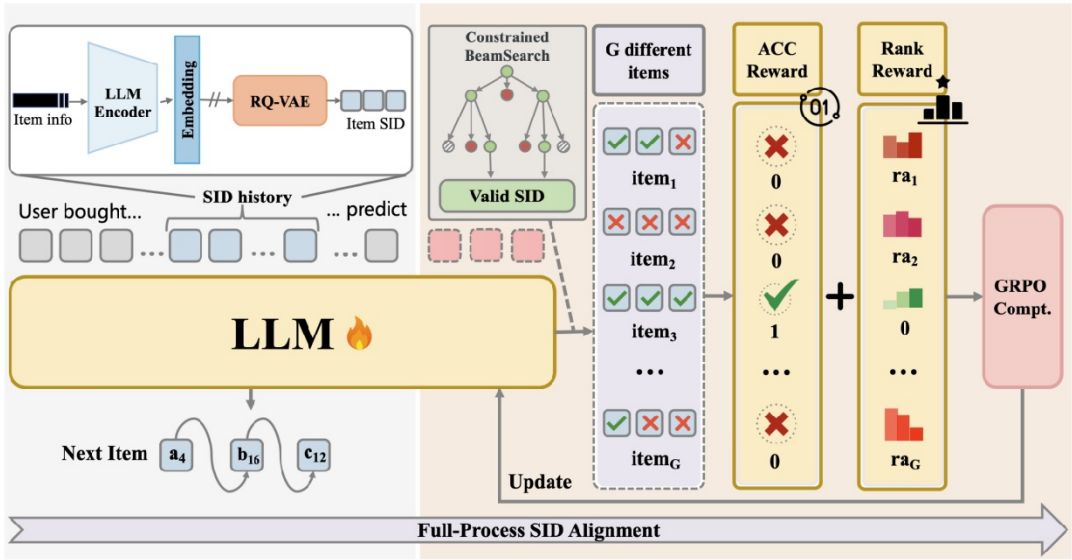
问题

生成式推荐依赖 LLM，但 如何让 LLM 真正“理解” SIDs、避免无效生成、并在公开数据集上实现可复现的 Scaling Law，一直是社区未解决的难题。

方法

全流程端到端开源框架（SID → SFT → RL: MiniOneRec 首次提供完全开源三阶段 pipeline:

- RQ-VAE 构建 SID codebook（同 OneRec / TIGER）
- SFT 对齐用户行为序列
- GRPO 强化偏好优化（RL）
  LLM × SID 的 Full-Process Alignment（两方向对齐）通过双向任务让 LLM 同时理解 “自然语言” 与 “离散 SIDs”：
- 让 LLM 用指令预测下一个 SID（Rec Task）
- 让 LLM 将文本 ↔ SID 互相对齐（Alignment Task）
- RL 阶段使用 Constrained Beam Search + Rule-based + Rank-aware Reward
- 保证模型 每一步都生成合法 SID，显著提升排序能力与稳定性。



Case Study 2

Unifying Generative and Dense Retrieval for Sequential Recommendation

LIGER 通过“生成式召回 + Dense 精排”的混合架构，在保持生成式检索效率优势的同时，大幅提升整体效果并恢复冷启动能力，实现两种范式的优势互补。

问题

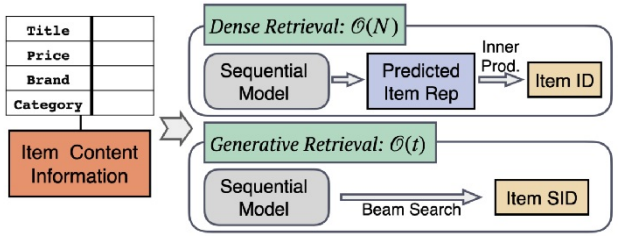
在相同输入条件下，生成式检索（如 TIGER）在小规模学术数据集上显著弱于 Dense 检索，并几乎无法生成冷启动物品。

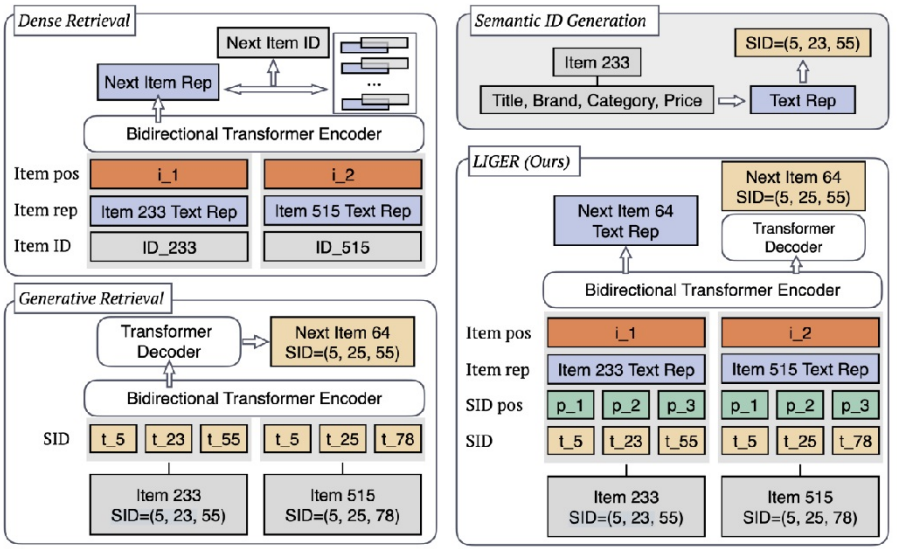
挑战

生成式检索依赖 beam search 的 token-by-token 解码，导致模型过度偏向训练集内 item，对未见过的 item（cold-start）几乎无概率生成，形成能力与效率的核心矛盾。

方法

- Dense + Generative 融合（Hybrid Retrieval）：使用生成式检索（TIGER 类）生成少量候选（减少存储成本），再由 Dense 模型对这些候选进行精确排序，从而继承 Dense 的 embedding 质量与冷启动能力。
- 引入 Text Representation 与 SID 共同建模: 在训练中同时利用 item 文本表示与语义 ID，使生成模块能捕获更丰富语义，同时通过 Dense 模块的相似度排序补齐生成器的 cold-start 弱点。





Challenges

语义容量、效率、可扩展性的三难困境

07

7 生成式推荐的挑战

挑战一:Tokenization 设计困境（语义vs可生成性）

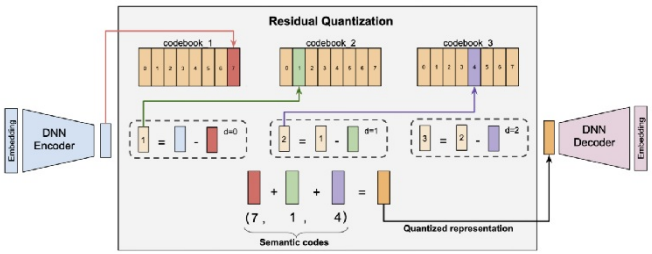
短 ID (RQVAE-based)虽生成迅速、推理友好，但语义容量有限；长 ID (PQ-based)虽表达力强，却使自回归解码成本陡增。多数 Tokenizer 仍是静态预处理器，与下游生成优化割裂。如何同时保证语义保真、生成友好，并支持端到端可学习，是当前 Tokenization 最大瓶颈。

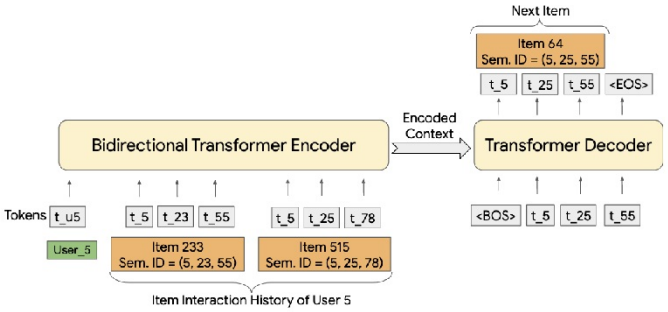
挑战二: 生成骨干可扩展性（能力 vs 延迟）

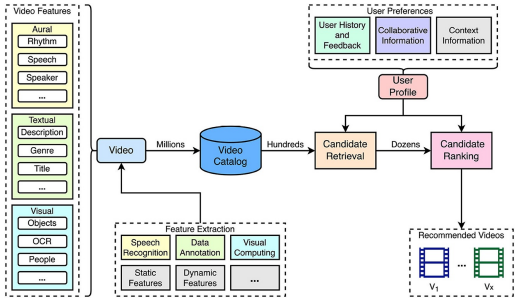
自回归 Transformer 推理延迟随 token 数线性增长，难满足在线场景；大型 Decoder-only LLM 则因参数规模与能耗过高而难以部署。未来需要更高效的并行/半并行解码和结构化生成骨干，在建模能力与推理效率之间取得可落地的平衡。

挑战三: 系统落地难点（统一 vs 工程成本）

多步生成带来的高延迟、业务中大量异构特征的统一建模，以及大规模参数的扩展成本，使 GenRec 难以直接替代传统多阶段 pipeline。如何在可接受的工程代价下构建统一、低延迟、可扩展的生成式推荐系统，是从论文到工业落地的关键挑战。







总结与展望

生成式推荐的现状: 生成式推荐通过“表征 → 离散化 → 序列生成”重构了传统推荐范式，为统一建模用户行为、语义信息与推荐推理提供了新的路径。现有工作表明，语义 Tokenization 的质量、生成骨干的建模能力以及训练与推理的一致性，仍是决定 GenRec 性能的核心因素。

从概念到大规模落地: 未来的 GenRec 需要在语义保真、生成友好、高效推理和系统可扩展性之间取得更好的平衡，包括可学习的 Tokenizer、更高效的生成结构、与行业特征体系的深度融合，以及面向业务指标的偏好对齐。随着这些关键环节逐步成熟，GenRec 有望从概念验证走向真正的大规模工业部署。

GenRec 的未来在于：更好的语义、更强的生成、更快的推理、更大的规模。

联系人

胡沛宇

联系方式

peiyuhu30@gmail.com

谢谢观看