

# Diffusion for Sequential Recommendation

## Diffusion for Recommendation

### 1. Generate Perspective

[SIGIR 2023] Diffusion Recommender Model

Diffusion Recommender Model

Generative models such as Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs) are widely utilized to model the generative process of user interactions. However, these...

 <https://arxiv.org/abs/2304.04971>

#### 1.1. Setting

给定user  $u$ 和他在item set  $\mathcal{I}$ 上的交互历史 $\mathbf{x}_u = \{x_u^1, x_u^2, \dots, x_u^{|\mathcal{I}|}\}$ , 其中 $x_u^i = 0$  or  $1$ 表示user是否与第*i*个item进行交互，目的是推断出user与所有item的交互概率。

#### 1.2. DiffRec

##### 1.2.1. Motivation

- **GAN-based Models**

- 基于GAN的模型利用生成器来估计用户的交互概率并利用对抗训练来优化参数。然而，对抗性训练通常是不稳定的，导致性能不理想。

- **VAE-based Models**

- 基于VAE的模型使用编码器来近似潜在因素的后验分布，并最大化观察到的交互的可能性。虽然VAE在推荐中通常优于GAN，但VAE存在易处理性和表示能力之间的权衡。  
tractable和简单的编码器可能无法很好地捕获异构用户偏好，而复杂模型的后验分布难以处理。

- **Diffusion Model**

- 类似生成模型中的解释，diffusion能够克服GAN和VAE的缺陷，是一种更优的模式。
- 与推荐结合的难点：
  - 由于生成个性化推荐的必要性，不能直接将图像域中的前向过程嫁接，即不适合加噪到pure noise
  - 生成模型需要大量的资源成本来同时预测所有项目的交互概率，限制了它们在大规模项目推荐中的应用
  - 生成模型必须捕获交互序列中的时间信息，这对于处理用户偏好shift至关重要。

- 解决方法：
  - 前向过程并不加噪到pure noise
  - 将item聚类分组，压缩至latent space进行建模
  - 对不同时间item进行加权

### 1.2.2. Diffusion Recommender Model (DiffRec)

- Training

---

## Algorithm 1 DiffRec Training

---

**Input:** all users' interactions  $\bar{X}$  and randomly initialized  $\theta$ .

- 1: **repeat**
- 2:     Sample a batch of users' interactions  $X \subset \bar{X}$ .
- 3:     **for all**  $x_0 \in X$  **do**
- 4:         Sample  $t \sim \mathcal{U}(1, T)$  or  $t \sim p_t$ ,  $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$ ;
- 5:         Compute  $x_t$  given  $x_0$ ,  $t$ , and  $\epsilon$  via  $q(x_t|x_0)$  in Eq. (3);
- 6:         Compute  $\mathcal{L}_t$  by Eq. (11) if  $t > 1$ , otherwise by Eq. (12);
- 7:         Take gradient descent step on  $\nabla_\theta \mathcal{L}_t$  to optimize  $\theta$ ;
- 8: **until** converged

**Output:** optimized  $\theta$ .

---

1. training 类似 DDPM，但 loss 是估计 input  $x_0$  而非 noise
  2. 前向加噪并不逼近标准正态分布，这里是否有理论保证？
  3. importance sampling
- Inference

---

## Algorithm 2 DiffRec Inference

---

**Input:**  $\theta$  and the interaction history  $x_0$  of user  $u$ .

- 1: Sample  $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$ .
- 2: Compute  $x_{T'}$  given  $x_0$ ,  $T'$ , and  $\epsilon$  via Eq. (3), and set  $\hat{x}_T = x_{T'}$ .
- 3: **for**  $t = T, \dots, 1$  **do**
- 4:      $\hat{x}_{t-1} = \mu_\theta(\hat{x}_t, t)$  calculated from  $\hat{x}_t$  and  $\hat{x}_\theta(\cdot)$  via Eq. (10);

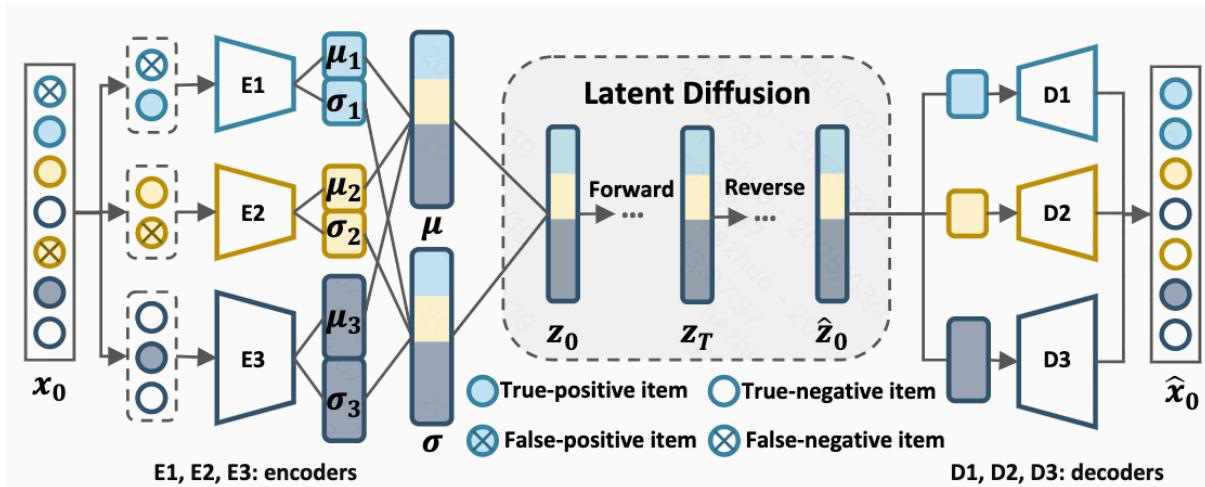
**Output:** the interaction probabilities  $\hat{x}_0$  for user  $u$ .

---

1. inference阶段并非直接denoise，而是用预测的正态均值进行替换

- L-DiffRec

为解决扩散模型资源消耗问题，将item聚类降维，随后在latent space建模扩散模型



1. 利用 LightGCN 得到item embedding

2. k-means聚类成 $c$ 个类别

3. 每类用一个 VAE 编码到latent space，在latent space进行forward和reverse

4. 通过多个decoder重构 $\hat{x}_0$

- T-DiffRec

user的perference会随时间变化，近期的交互能更准确表达用户的perference，因此为最近交互的数据分配更大的权重

Formally, for user  $u$  with  $M$  interacted items, the interaction time is available and the interaction sequence is formulated as  $\mathcal{S} = \{i_1, i_2, \dots, i_M\}$ , where  $i_m$  denotes the ID of the  $m$ -th interacted item. We define the weights of interacted items  $\mathbf{w} = [w_1, w_2, \dots, w_M]$  via a time-aware linear schedule<sup>2</sup>:  $w_m = w_{\min} + \frac{m-1}{M-1}(w_{\max} - w_{\min})$ , where the two hyper-parameters  $w_{\min} < w_{\max} \in (0, 1]$  represent the lower and upper bounds of interaction weights. Thereafter, the interaction history  $\mathbf{x}_0$  of user  $u$  is reweighted as  $\bar{\mathbf{x}}_0 = \mathbf{x}_0 \odot \bar{\mathbf{w}}$ , where

## 2. Representation Perspective

[SIGIR 2024] Denoising Diffusion Recommender Model

## Denoising Diffusion Recommender Model

Recommender systems often grapple with noisy implicit feedback. Most studies alleviate the noise issues from data cleaning perspective such as data resampling and reweighting, but they are...

🔗 <https://arxiv.org/abs/2401.06982>



## 2.1. Setting

给定用户集  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ , 项目集  $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$ , 每个用户的交互历史  $\mathcal{S}_u = \{v_1^u, \dots, v_{n_u}^u\}$ , 需要利用历史信息为user进行推荐。

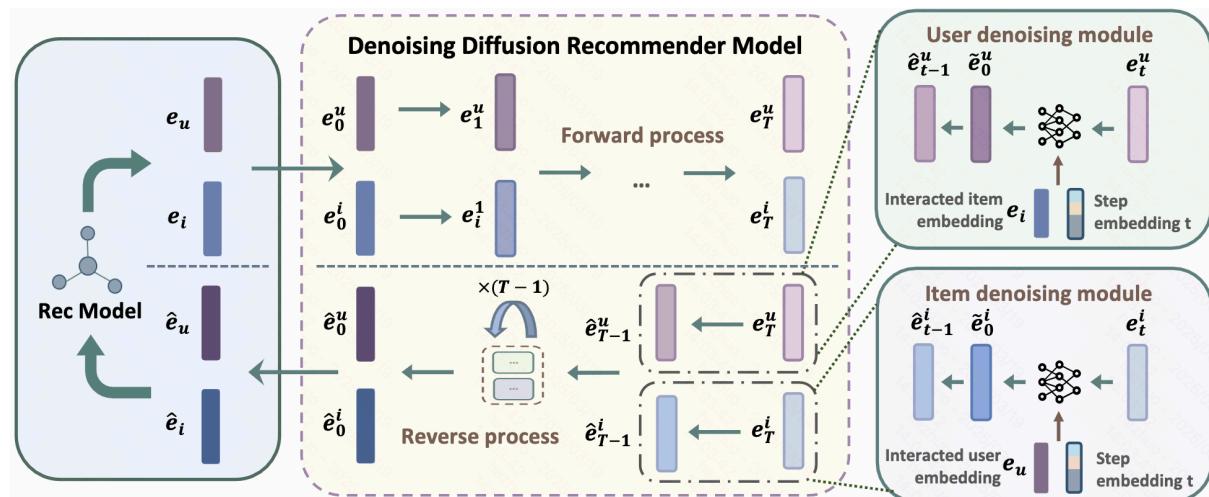
## 2.2. DDRM

### 2.2.1. Motivation

- 关键问题：推荐系统数据经常存在noise的feedback，需要学习更加鲁棒的representation
- 现存方法：一类方法从data-cleaning的角度缓解noise，但这类方法依赖于确定性的启发式假设，因此方法的适应性受到限制。另一类方法从model角度处理noise，它们通过主动注入噪声来增强模型去噪能力，学习更加robust representation，这对模型捕捉noise pattern提出了高的要求，diffusion model具备解决这些问题的能力。
- 关键挑战：
  - 如何设计适当的guidance去引导reverse denoise过程，以得到更加准确的clean embedding？
  - 如何设计适当的start point去整合user个性化信息？
- 解决方法：
  - 利用推荐系统中的协同信息作为guidance
  - 以要预测的user历史交互信息作为start point，融入个性化的信息

### 2.2.2. Denoising Diffusion Recommender Model (DDRM)

- Framework



- Training

---

## Algorithm 1 DDRM Training

---

**Input:** interaction data  $\bar{D}$ , pre-trained user embedding  $e_0^u$ , pre-trained item embedding  $e_0^i$ , diffusion step  $T$ , user reconstruction MLP  $f_\theta$ , item reconstruction MLP  $f_\psi$

```

1: repeat
2:   Sample a batch of interactions  $D \subset \bar{D}$ .
3:   for all  $(u, i, j) \in D$  do
4:     Sample  $t \sim \mathcal{U}(1, T)$ 
5:     Compute  $e_t^u$  given  $e_0^u$  and  $t$  via  $q(e_t^u | e_0^u)$  in Eq. (6);
6:     Compute  $e_t^i$  given  $e_0^i$  and  $t$  via  $q(e_t^i | e_0^i)$  in Eq. (7);
7:     Reconstruct  $\tilde{e}_0^u$  and  $\tilde{e}_0^i$  through  $f_\theta$  and  $f_\psi$ ;
8:     Calculate  $\mathcal{L}_{\text{final}}$  by Eq. (22);
9:     Take gradient descent step on  $\nabla_\theta(\mathcal{L}_{\text{final}})$  to optimize  $\theta$ ;
10:    Take gradient descent step on  $\nabla_\psi(\mathcal{L}_{\text{final}})$  to optimize  $\psi$ ;
11: until converged

```

**Output:** optimized  $\theta, \psi$ .

---

1. 涉及diffusion部分类似 DDPM，只需注意：
    - a. DDRM对pre-trained user, item embedding进行forward和reverse;
    - b. reverse过程额外利用了协同信息作为guidance，即 $\tilde{e}_0^u = f_\theta(\hat{e}_t^u, c_u, t)$ ,  $\tilde{e}_0^i = f_\theta(\hat{e}_t^i, c_i, t)$ ，其中 $c_u, c_i$ 分别为user和item的协同信息，在实现中分别取item和user的embedding  $e_i, e_u$
  2. 在通过DDRM获得去噪后的user和positive embedding后，这些embedding去计算BPR损失  $\mathcal{L}_{bpr}$ ，单个user的loss为diffusion部分loss  $\mathcal{L}_{re}$ 和  $\mathcal{L}_{bpr}$ 之间的trade off。作为扩展，作者还对每个user的loss计算了score  $s(u, i)$ 并利用softmax进行重加权
- inference

---

## Algorithm 2 DDRM Inference

---

**Input:** all users  $\bar{U}$ , diffusion step  $T$ , item reconstruction MLP  $f_\psi$

- 1: Sample a batch of users  $U \subset \bar{U}$ .
- 2: **for all**  $u \in U$  **do**
- 3:     Compute average embeddings of users' historically liked items  $\bar{e}_i$  via Eq. (23);
- 4:     Compute  $\bar{e}_T^i$  given  $\bar{e}_0^i$  and  $T$  via  $q(\bar{e}_T^i | \bar{e}_0^i)$  in Eq. (7);
- 5:     **for**  $t = T, \dots, 1$  **do**
- 6:         Reconstruct  $\hat{e}_0^i$  through  $f_\psi$ ;
- 7:         Compute  $\hat{e}_{t-1}^i$  from  $\hat{e}_t^i$  and  $\hat{e}_0^i$  via Eq. (13);
- 8:     Rounding via Eq. (24) to get the ideal item embedding  $e$ ;

**Output:** the ideal item embedding  $e$ .

---

1. inference阶段，利用user历史的平均item embedding作为start point，reverse阶段依然用预测均值替换
2. 在生成 ideal item embedding后，利用rounding function  $s(e, e_i)$ 选出top-k候选item进行推荐

item embedding的shape?

---

# Diffusion for Sequential Recommendation

## 1. Setting

先前提到的推荐系统建模可能会遗漏user的序列行为信息，因此sequential recommendation旨在显式建模用户的序列行为，提升推荐系统的效果。

给定user set  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ , item set  $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$ , 每个user的交互历史  $\mathcal{S}_u = \{v_1^u, \dots, v_{n_u}^u\}$ 由交互时间进行组织，需要利用历史交互序列信息为user进行推荐。

## 2. Sequence as Diffusion Target

### 2.1. DiffuASR

[CIKM 2023] Diffusion Augmentation for Sequential Recommendation

## Diffusion Augmentation for Sequential Recommendation

Sequential recommendation (SRS) has become the technical foundation in many applications recently, which aims to recommend the next item based on the user's historical interactions. However,...

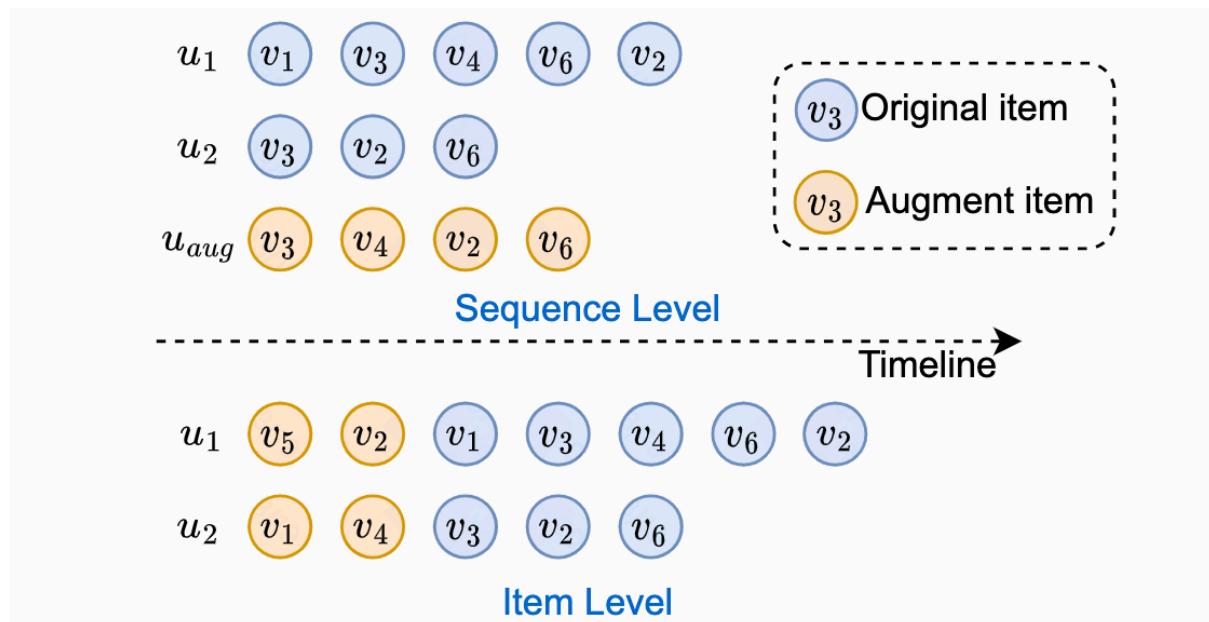
✖ <https://arxiv.org/abs/2309.12858>



通过扩散模型对用户历史交互序列进行数据增强

### 2.1.1. Motivation

- **关键问题：**序列推荐通常面临**数据稀疏**和**长尾问题**两大挑战，合适的数据增强策略是较好的应对方法
  - 数据稀疏：大量用户和项目之间只会发生一些交互，它很容易导致序列推荐的次优问题
  - 长尾问题：交互较少的用户通常会得到更糟糕的推荐
- **现存方法：**
  - sequence-level augmentation：由于直接生成sequence往往容易包含较多的noise，难度较大，因此需要设计复杂的训练流程；此外这种方法不能很好解决长尾问题；
  - item-level augmentation：此类方法能处理数据稀疏和长尾问题，它为每个用户交互序列生成pseudo pre-order items。但现存方法关注如何有效地使用增强数据集，而忽略了增强数据本身的质量与生成数据与原始数据的shift。
  - diffusion model 具备生成高质量数据的能力，且作者认为借助diffusion一次生成能减缓生成的sequence与原始sequence之间的shift，因而提议借助扩散模型解决问题。

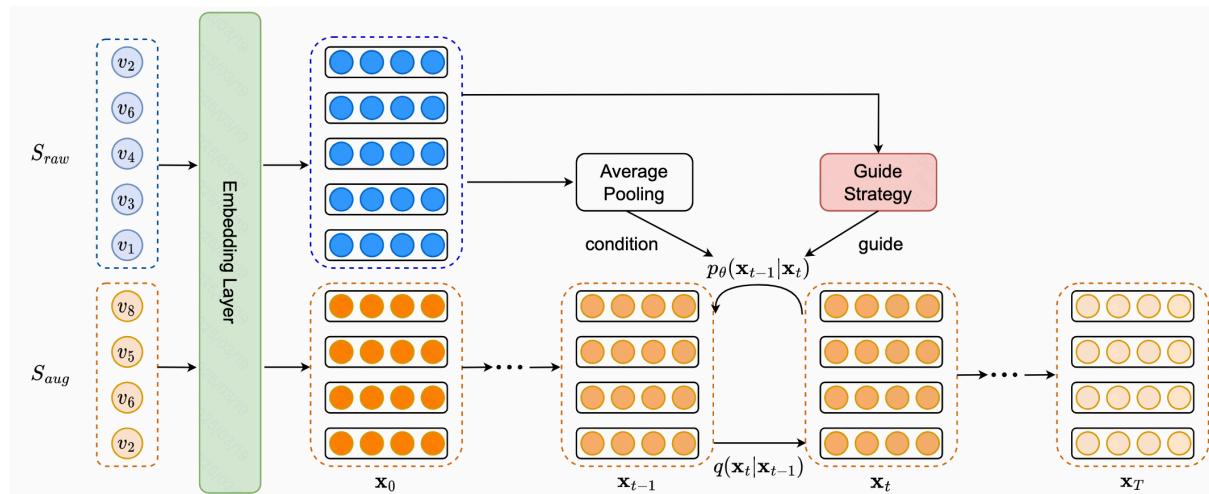


- **关键挑战：**

- diffusion model 最初为生成图像而设计，在二维连续空间执行forward和reverse，但是user的交互历史处于一维离散空间，其中存在gap

- 为保证生成质量，需要确保生成的 pseudo item 应当与 user 的原始 preference 相关
- 解决方法：
  - 设计了一个适配于推荐领域 denoise 的网络 Sequential U-Net
  - 将 user 原始交互序列作为 guidance (因为其中包含了 preference 信息)，并利用两种 guidance 策略

### 2.1.2. Diffusion Augmentation for Sequential Recommendation (DiffuASR)



#### • Forward Process

- 利用一个 learnable embedding table  $\mathbf{E} \in \mathbb{R}^{|\mathcal{V} \times d|}$  将 item identity  $\mathcal{S}_{aug} = \{v_{-M}, v_{-M+1}, \dots, v_{-1}\}$  映射为 dense representation 并在第 0 维上 stack，得到  $\mathbf{x}_0 = \{e_{-M}, e_{-M+1}, \dots, e_{-1}\} \in \mathbb{R}^{M \times d}$
- 对  $\mathbf{x}_0$  加噪至  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}^{M \times d})$
- 在实际实现中，取 user 交互 sequence 的前  $M$  个 item 作为  $\mathcal{S}_{aug}$

#### • Reverse Process

- 类似 diffusion 做图像生成任务用 U-Net 建模  $\epsilon_\theta$  预测 noise，这里也采用类似的结构预测 noise，但是直接利用并不 match 并且会丢失 sequence 信息，因此设计 Sequential U-Net (SU-Net)
  - SU-Net
    - 将每个交互过的 item embedding reshape 为  $\sqrt{d} \times \sqrt{d}$
    - 将 reshape 后的每个 embedding 独立视作一个 channel，因此得到了形状为  $\mathbb{R}^{M \times \sqrt{d} \times \sqrt{d}}$  的三维矩阵（因此为了统一 channel nums，不够  $M$  个 item 的交互 sequence 会被略过）
    - 这样一方面适配 U-Net 的输入，另一方面卷积操作能对多通道信息进行融合，能一定程度上保留原来的 sequence 信息
- Guide Strategy

高质量的增强应该对应于原始用户交互序列中包含的偏好，因为不相关的项目可以被认为是 noise 的交互，从而损害序列推荐模型

- a. 实际实现中，取 user 交互 sequence 剩下的部分为  $S_{raw}$ ，其中包含 user 的 preference 信息，因此考虑将其作为 guidance 注入  $\epsilon_\theta$ ，实现中是取历史交互 embedding 的 average
  - b. 由于建模方式是利用  $S_{aug}$  预测  $v_1$ ，因此将  $v_1$  作为 condition，并类似图像生成设计 classifier-guided 和 classifier-free guidance 策略
3. 获取 augmented sequence
    - a. 对生成对 sequence 的每个位置，与 item set 中的 item embedding 计算余弦相似度选取最终 item

- **Inference**

对于数据集中的一个 sequence，从标准高斯分布采样并选取适当的 guide strategy 进行 reverse，做 round 后得到增强的 item，随后以 pre-order 形式加入原 sequence

### 3. Sequence as Diffusion Guidance

#### 3.1. DiffRec2

##### [arXiv 2023] Sequential Recommendation with Diffusion Models

###### Sequential Recommendation with Diffusion Models

Generative models, such as Variational Auto-Encoder (VAE) and Generative Adversarial Network (GAN), have been successfully applied in sequential recommendation. These methods require sampling from...

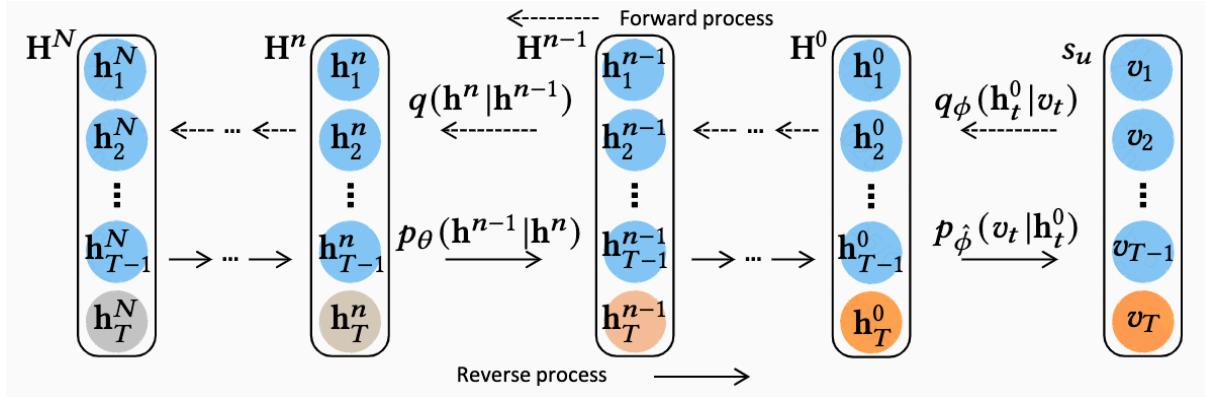
🔗 <https://arxiv.org/abs/2304.04541>



##### 3.1.1. Motivation

- 以往确定式序列推荐系统的缺陷：
  - 将 item representation 建模为固定的向量，无法捕捉 user 兴趣和行为的不确定性
  - exposure bias，确定式方法预设与 user 交互最多的 item 是最相关的，这样将交互较少但可能是 user 感兴趣的 item 置于不利地位
- 生成式模型 GAN 和 VAE 能解决上述问题，但类似生成模型领域广泛阐述的那样，训练不稳定等问题限制了它们的性能，有更强能力的 diffusion model 更加合适
- 关键挑战：
  - diffusion model 作用于连续空间，而推荐系统领域的数据分布在离散空间，之间存在 gap
  - diffusion model 的训练和推理需要消耗很大成本
- 解决方法：
  - 用一个 learnable 的 embedding table 映射到连续空间
  - 取多个随机种子，然后在 reverse 阶段一步 denoise，对不同种子下的结果做 average

### 3.1.2. Sequential Recommendation with Diffusion Models (DiffRec2)



- **Forward Process**

- 为了在连续空间建模，利用一个learnable的embedding table  $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$  将sequence  $\{v_1, v_2, \dots, v_{T-1}, v_T\}$  中的每一个  $v_t$  映射到  $\mathbb{R}^d$  空间，得到  $\mathbf{H}^0 = \{h_1^0, h_2^0, \dots, h_{T-1}^0, h_T^0\}$ ；由于序列推荐系统的目标是预测next item，因此只对最后一个item embedding  $h_T^0$  加噪

- **Reverse Process**

- 在reverse过程的loss采用预测最开始的embedding的形式，对  $f_\theta(h_T^n, n)$  对结构进行如下设计：
  - 借助 Transformer Encoder 将前  $T - 1$  个时刻的embedding  $h_{1:T-1}^0$  的信息融入最后一个 embedding  $h_T^n$ ，这可以理解成一种隐式的condition
  - 设计一个learnable position embedding matrix  $\mathbf{Y} \in \mathbb{R}^{T \times d}$ ，连同diffusion step embedding  $z_n$ ，连同第  $n$  步denoising的representation进行concat:  $\hat{\mathbf{H}}^n = [h_1^0 + y_1 + z_n, h_2^0 + y_2 + z_n, \dots, h_T^0 + y_T + z_n]$
  - $f_\theta(h_T^n, n) = \text{TransformerEncoder}(\hat{\mathbf{H}}^n)[-1]$
  - 利用接softmax的linear将连续空间中的  $h^0$  映射回离散空间，即  $p_{\phi}(v_t|h_t^0) = \text{Softmax}(W h_t^0 + b)$ ， $W \in \mathbb{R}^{|\mathcal{V}| \times d}$

- **Objective**

$$\mathcal{L} = \underbrace{\mathbb{E}_{q(\mathbf{h}_T^{1:N} | v_T)} \left[ \left( \sum_{n=2}^N \|\mathbf{h}_T^0 - f_\theta(\mathbf{h}_T^n, n)\|^2 \right) + \|\mathbf{E}(v_T) - f_\theta(\mathbf{h}_T^1, 1)\|^2 \right]}_{\text{MSE Loss}} + \underbrace{\mathbb{E}_{q(\mathbf{h}_T^0 | v_T)} - \log p_{\hat{\phi}}(v_T | \mathbf{h}_T^0)}_{\text{Rec Loss}}$$

- Inference

---

**Algorithm 2** Inference of  $v_{T+1}$  for user sequence  $s_u$ 


---

**Input:** inference sequence  $s_u = [v_2, v_3, \dots, v_T, \text{unk}]$ , sequence length  $T$ , total diffusion steps  $N$

- 1: **for**  $n = N, N - 1, \dots, 2$  **do**
- 2:     Sample random Gaussian noise  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- 3:     Calculate  $\mathbf{h}_t^0 (1 \leq t \leq T)$  via  $q_\phi(\mathbf{h}_t^0 | v_t)$  (Eq. 6)
- 4:     Calculate  $\mathbf{h}_t^N (1 \leq t \leq T)$  via  $q(\mathbf{h}_t^N | \mathbf{h}_t^0)$  (Eq. 7)
- 5:      $\mu_\theta(\mathbf{h}_T^n) = \frac{\sqrt{\bar{\alpha}_{n-1}}\beta_n}{1-\bar{\alpha}_n} f_\theta(\mathbf{h}_T^n, n) + \frac{\sqrt{\alpha_n}(1-\bar{\alpha}_{n-1})}{1-\bar{\alpha}_n} \mathbf{h}_T^n$
- 6:     Calculate  $\mathbf{h}_T^{n-1}$  via  $p_\theta(\mathbf{h}_T^{n-1} | \mathbf{h}_T^n)$  (Eq. 4)
- 7: **end for**
- 8: Predict  $\mathbf{h}_T^0$  via  $f_\theta(\mathbf{h}_T^1, 1)$  (Eq. 16)
- 9: Obtain probability distribution  $p \leftarrow p_{\hat{\phi}}(v_T | \mathbf{h}_T^0)$  (Eq. 8)
- 10: **return**  $p$

---

1. 做inference时，待生成的item取一个额外的占位符 [unk]
2. 论文后续设计的efficient inference其实是取多个随机种子，然后在reverse阶段一步 denoise，对不同种子下的结果做average，一步denoise虽然确实快了，但是效果是比较存疑的

### 3.2. DiffuRec

[ACM Trans. Inf. Sys. 2024] DiffuRec: A Diffusion Model for Sequential Recommendation

DiffuRec: A Diffusion Model for Sequential Recommendation

Mainstream solutions to Sequential Recommendation (SR) represent items with fixed vectors. These vectors have limited capability in capturing items' latent aspects and users' diverse preferences....

🔗 <https://arxiv.org/abs/2304.00686>

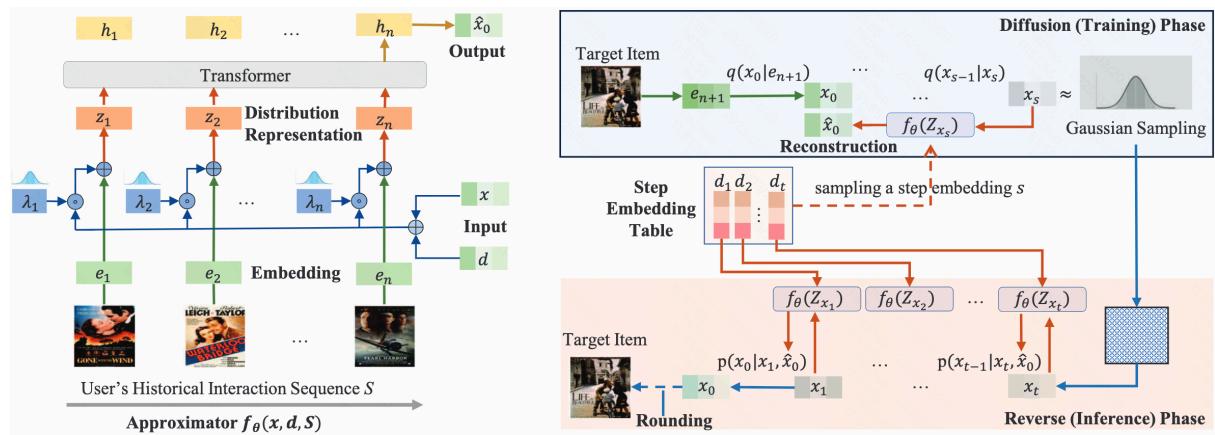


#### 3.2.1. Motivation

- 以往序列推荐系统通常将item representation学成一个vector，这会有以下几个不足：
  - item往往包含多个潜在方面，例如一部电影经常被列在多个分区，在单个向量中编码复杂的潜在方面仍然具有挑战性
  - user的兴趣和偏好是多样且多变的
  - 由于用户兴趣的多样性和演变，用户当前的偏好和意图在一定程度上变得不确定

- 目标item的guidance，目标item可以促进用户当前的意图理解，它可以作为辅助信号来指导后续过程
- 当前multi-interest modeling方法部分解决了上述问题，但受限于需要预先定义interest的数量来启发式地预定义以进行模型训练。VAE通过建模分布能解决多个潜在方面的问题并注入不确定性，但它们自身的问题导致性能受限，作者提出使用扩散模型来解决这些问题
- 和VAE一样，本工作使用diffusion建模item embedding的分布能注入不确定性且建模item潜在方面，利用target item做guidance也是diffusion做recommendation常见操作，**这份工作相较于DiffRec2，本质的区别在于将user交互历史序列建模为分布，从而表示多样且多变的兴趣，作为condition引导denoise过程**

### 3.2.2. DiffuRec



#### • Forward Process

和平常的DDPM forward一样，这里对target item进行前向加噪，noise变化遵循线性schedule  
将item映射到embedding空间的encoder是与训练好的吗？noise embedding会影响forward？

#### • Reverse Process

reverse阶段的关键在于设计恰当的方法预测原始embedding，DiffuRec设计了一个  
Approximator  $f_\theta$

1.  $f_\theta(x, d, S)$ ， $d$ 为step embedding， $x$ 为对应step下的target item的noise embedding， $S$ 为历史交互序列的信息
2.  $f_\theta$ 的设计如下：
  - 将user的历史交互序列映射为embeddings:  $e_1, e_2, \dots, e_n$ ，随后利用 $x$ 和 $d$ 建模它们的分布，即  $z_i = e_i + \lambda_i \odot (x + d)$ ， $\lambda_i \sim \mathcal{N}(\delta, \delta)$  控制着noise的scale
  - 将  $[z_1, z_2, \dots, z_n]$  送入一个transformer，取transformer output最后一位作为原始 embedding 的估计，这也可以理解为一种隐式的condition
  - 没有position embedding？
3. 文中提到

On the other hand, this setting introduces uncertainty to model the user's interest evolution for reverse phase. That is, the importance of each latent aspect of a historical item can be adjusted iteratively along the reverse steps in a user-aware manner.

这表明在reverse阶段，通过不断预测原始target item embedding，transformer会逐渐从历史交互的item中学习到不同item对user的重要性

- **Loss Function and Rounding**

1. 不同于一般的DDPM中用MSE进行约束，DiffuRec利用cross-entropy进行约束：

$$\hat{y} = \frac{\exp(\hat{\mathbf{x}}_0 \cdot \mathbf{e}_{n+1})}{\sum_{i \in \mathcal{I}} \exp(\hat{\mathbf{x}}_0 \cdot \mathbf{e}_i)} \quad (18)$$

$$\mathcal{L}_{CE} = \frac{1}{|\mathcal{U}|} \sum_{i \in \mathcal{U}} -\log \hat{y}_i \quad (19)$$

这基于两点理由：1) item embedding在latent space还是离散的；2) 在序列推荐领域，计算两个向量的内积取表示相关性更加普遍

- **Inference**

---

### Algorithm 1: Reverse or Inference

---

```

1: Input:
2:   Historical Sequence:  $(i_1, \dots, i_n)$ ;
3:   Target item Representation  $\mathbf{x}_t$ :  $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;
4:   Total Reverse Steps:  $t$ ;
5:   Schedule  $\beta$ : Linear;
6:   Approximator:  $f_\theta(\mathbf{x})$ ;
7: Output:
8:   Predicted Target Item:  $i_{n+1}$ ;
9: while  $t > 0$  do
10:    $(\mathbf{z}_1, \dots, \mathbf{z}_n) \leftarrow (\mathbf{e}_1 + \lambda_1(\mathbf{x}_t + \mathbf{d}_t), \dots, \mathbf{e}_n + \lambda_n(\mathbf{x}_t + \mathbf{d}_t))$ , ;
    // Distribution Representation
11:    $\hat{\mathbf{x}}_0 \leftarrow f_\theta(\mathbf{z}_1, \dots, \mathbf{z}_n)$ ; // Reconstruction
12:    $\mathbf{x}_{t-1} \leftarrow \tilde{\mu}(\hat{\mathbf{x}}_0, \mathbf{x}_t) + \tilde{\beta}_t * \epsilon'$ , ( $\epsilon' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ); // Reversion
13:    $t = t - 1$ ; // Iteration
14: end while
15:  $i_{n+1} \leftarrow \text{Rounding}(\mathbf{x}_0)$ ;

```

---

### 3.3. DreamRec

[NeurIPS 2023] Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion

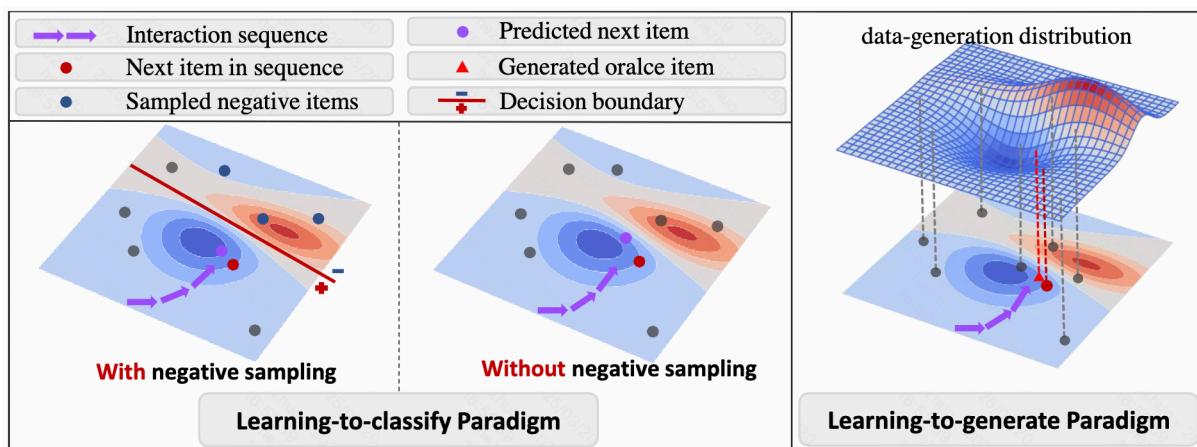
Generate What You Prefer: Reshaping Sequential Recommendation via...

Sequential recommendation aims to recommend the next item that matches a user's interest, based on the sequence of items he/she interacted with before. Scrutinizing previous studies, we can...

🔗 <https://arxiv.org/abs/2310.20453>



#### 3.3.1. Motivation



- 过去大多序列推荐的工作都可以归类为“**learning-to-classify**”范式，即给定一个positive item，推荐模型执行负采样以添加negative item，并根据user的历史交互序列学习去判断user是否更喜欢它们。这种范式存在以下两点问题：
  - 它可能与人类行为不同，因为人们通常想象一个oracle item，并选择与oracle item最匹配的潜在item
  - 基于候选集的分类是表征用户偏好的一种迂回方式，用于解决仅有正样本可用的问题。1) 选择的负样本被限制在一个小的候选集之中，无法精细建模正负样本的决策边界；2) 对比学习领域也研究过，简单的负样本无法贡献有效的信息，而过于困难的负样本可能是假阴性的，这会为模型学习引入噪声
- 为此，本工作提出了“**learning-to-generate**”范式，核心思想是基于历史交互序列描述底层数据生成分布，直接生成表示用户偏好的oracle item，并推断出与oracle最匹配的真实item (p.s. 也就是说生成范式能够充分探索item space，并以历史交互作guidance，从而在item space中找到用户最偏好的oracle item)
- 对于所提出的“**learning-to-generate**”范式，本文作者对已有的diffusion for recommendation方法做了简要探讨：

To our knowledge, recent studies have explored integrating diffusion models into sequential recommendation. However, these

approaches still adhere to the learning-to-classify paradigm, inevitably requiring negative sampling during training. For instance, Li et al. and Du et al. apply softmax cross-entropy loss on the predicted logits of candidates, treating all non-target items as negative samples.

In contrast, our proposed DreamRec reshapes sequential recommendation as a learning-to-generate task. Specifically, DreamRec directly generates the oracle item tailored to user behavior sequence, transcending limitations of the concrete items in the candidate set and encouraging exploration of the underlying data distribution without the need of negative sampling.

与其他diffusion方法相比最本质的区别像是通过非参数分类取消了候选集的限制，无需负采样

### 3.3.2. DreamRec

- Forward

将target item映射为embedding  $e_n^0$ , 随后类似DDPM进行加噪, 直到 $e_n^T$

- Reverse

1. 将历史交互序列 $e_{1:n-1} = [e_1, e_2, \dots, e_{n-1}]$ 送入transformer encoder获得编码后的交互序列, 即 $c_{n-1} = \text{TransformerEncoder}(e_{1:n-1})$
2. 将 $c_{1:n-1}$ 作为guidance, 利用classifier-free guidance方式训练, 损失函数为

$$L_{t-1} = \mathbb{E}_{\mathbf{e}_n^0, \epsilon} \left[ \frac{\bar{\alpha}_{t-1}}{2\tilde{\beta}_t} \|\mathbf{e}_n^0 - f_\theta(\sqrt{\bar{\alpha}_t} \mathbf{e}_n^0 + \sqrt{1-\bar{\alpha}_t} \epsilon, \mathbf{c}_{n-1}, t)\|^2 \right] + C.$$

其中会以一定的概率将 $c_{1:n-1}$ 置为dummy token $\Phi$ 来训练无条件diffusion model

- Inference

---

**Algorithm 2** Generation phase of DreamRec

---

1: $\mathbf{e}_{1:n-1} \sim \mathcal{D}_t$	▷ Sample and embed a data from testing set.
2: $\mathbf{e}_n^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$	▷ Sample Gaussian noise.
3: $\mathbf{c}_{n-1} = \text{T-enc}(\mathbf{e}_{1:n-1})$	▷ Encode interaction sequence.
4: <b>for</b> $t = T, \dots, 1$ <b>do</b>	▷ Denoise for T steps.
5: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ , else $\mathbf{z} = \mathbf{0}$	▷ Sample denoising variance.
6: $\tilde{f}_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t) = (1+w) f_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t) - w f_\theta(\mathbf{e}_n^t, \Phi, t)$	▷ Control the strength of guidance.
7: $\mathbf{e}_n^{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \tilde{f}_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{e}_n^t + \sqrt{\tilde{\beta}_t} \mathbf{z}$	▷ Denoise for one step.
8: <b>end for</b>	
9: <b>return</b> $\mathbf{e}_n^0$	

---

- **Retrieval of Recommendation List**

在生成 oracle item后，后续步骤为获取针对特定用户量身定制的recommendation list。本工作从候选集中选取 $K$ 个与oracle item内积最大的item来实现。从概念上讲，DreamRec 超越了有限候选集的范围，在追求 oracle item时渗透到整个item space中。

### 3.4. DimeRec

**[WSDM 2025] A Unified Framework for Enhanced Sequential Recommendation via Generative Diffusion Models**

DimeRec: A Unified Framework for Enhanced Sequential...

Sequential Recommendation (SR) plays a pivotal role in recommender systems by tailoring recommendations to user preferences based on their non-stationary historical interactions. Achieving...

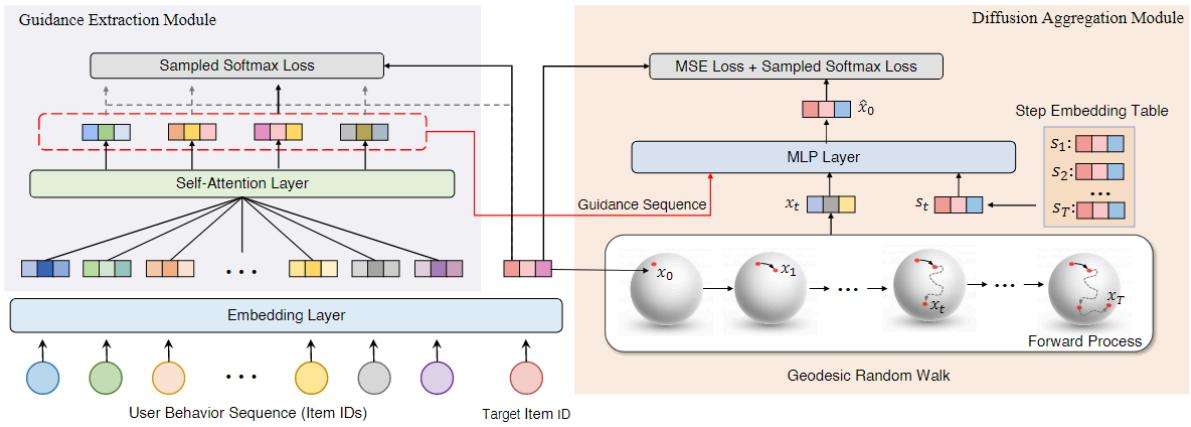
🔗 <https://arxiv.org/abs/2408.12153>



#### 3.4.1. Motivation

- 以往的确定式推荐系统在充分表示多样性和不确定性上的能力有所欠缺，无法准确掌握用户兴趣的底层分布，并且会受到exposure bias的影响。
- 借助VAE，GAN等生成式推荐系统能克服上述问题，但存在各自的缺陷限制了它们的性能。相较之下，diffusion model有更强的能力，在推荐领域的应用有着很大的潜力。
- 目前有不少工作开始探索将diffusion model应用于序列推荐系统中，一个主流的思想是将item映射到连续空间，并生成next item的embedding，使用用户的整个历史行为序列作为去噪的条件，并拟合整个item space的分布。然而，它们通常存在以下几点问题：
  - 由于现实世界user与item交互的稀疏性，历史行为序列中包含的信息可能并不充分
  - 推荐系统的分类损失与diffusion model的重建损失优化方向存在出入，导致了训练的模型性能次优
  - 这些方法涉及以end-to-end的方式训练模型，其中item的初始representation是无序的，这意味着它们的分布也可能毫无意义。在没有任何帮助的情况下直接在这样的空间中训练diffusion model是困难的
- 为了解决这些关键挑战，本工作提出了以下三点对策：
  - 针对稀疏性问题，本工作提出不直接在item space建模去噪condition，而是在更高级抽象的**interest space**建模condition，与item space相比，interest space更加稳定，信息更充分
  - 针对优化方向问题，本工作引入了一个新的噪声空间，在其中两类损失可以同时优化
  - 针对item rep. 问题，本工作提出一个**独立的guidance loss**，这个额外的损失函数增强了学习item rep. 的稳定性和有效性

#### 3.4.2. Diffusion with Multi-interest Enhanced Recommender (DimeRec)



- **Guidance Extraction Module (GEM)**

- Rule-based GEM

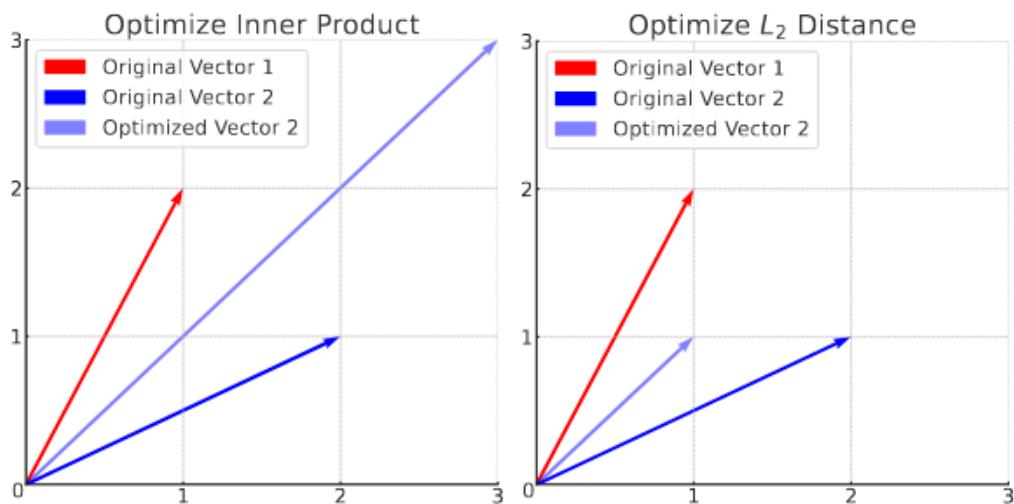
这种方式是parameter-free的，通常通过设计合理的规则来提取guidance。在本工作中，一个基本的方式是将历史序列 $\mathcal{S}^u \in \mathbb{R}^N$ 裁剪至更小的size  $K$ ，然后encode到连续空间作为guidance，即 $g^u = \mathcal{F}(\mathcal{S}^u[-K :])$

- Model-based GEM

设计了一个learnable的attentive matrix  $A \in \mathcal{R}^{N \times K}$ ，通过激活函数为tanh的两层 mlp 来实现，提取的guidance为  $g^u = A^T(\mathcal{F}(\mathcal{S}^u) + \mathcal{P})$ ， $\mathcal{P}$ 为position embedding（区别于 DreamRec，这里压缩到更低维的 interest space 中）

- **Diffusion Aggregation Module (DAM)**

- 本工作使用当前target item embedding, guidance和step embedding预测原始 embedding
  - 在此处探讨了推荐系统分类loss与diffusion重建loss优化方向不一的问题，即diffusion重建的mse loss会减小向量的norm，而增大内积则可能使得向量norm增大。



- 为此提出将向量限制在一个超球面中以保持norm不变，但由此带来的是在黎曼流形中的 diffusion modeling与欧氏空间中diffusion modeling不一致的问题。不过geodesic random walk理论告诉我们，对于球面流形，如果测地线随机游走的步骤较小且各向同性分布（即均匀分布在所有方向上），则此类游走端点的分布可以在游走均值点处逼近切线空间中的高斯噪声。因此可以直接利用DDPM的结论进行反向过程，而不失有效性，唯一需要做的事情是将生成的向量映射到球面空间

- **Loss Function**

- 为促进item embedding的学习，设计了对比学习损失来增强item embedding的语义信息， $e_a$ 为target item embedding：

$$g_u = \mathbf{g}^u[:, \arg \max(\mathbf{g}^u \cdot \mathbf{e}_a)]$$

The GEM module can be trained with the following objective function

$$\mathcal{L}_{gem}(\phi) = \frac{1}{|\mathcal{U}|} \sum_{a \in \mathcal{I}_u} -\log \left( \frac{\exp(g_u \cdot \mathbf{e}_a)}{\exp(g_u \cdot \mathbf{e}_a) + \sum_{i^- \in \mathcal{I}_{\text{sample}}} \exp(g_u \cdot \mathbf{e}_{i^-})} \right)$$

- 重建损失与分类损失：

$$\mathcal{L}_{recon}(\theta) = \mathbb{E}_{t, x_0, \epsilon} [\|x_0 - f_\theta(x_t, t, \mathbf{g}^u)\|_2^2]$$

$$\mathcal{L}_{ssm}(\theta) = \frac{1}{|\mathcal{U}|} \sum_{a \in \mathcal{I}_u} -\log \left( \frac{\exp(\hat{x}_0 \cdot \mathbf{e}_a)}{\exp(\hat{x}_0 \cdot \mathbf{e}_a) + \sum_{i^- \in \mathcal{I}_{\text{sample}}} \exp(\hat{x}_0 \cdot \mathbf{e}_{i^-})} \right)$$

- total loss:  $\mathcal{L} = \mathcal{L}_{gem} + \lambda \mathcal{L}_{recon} + \mu \mathcal{L}_{ssm}$

- **Algorithm**

---

**Algorithm 1:** DimeRec Training

---

**Input:**

Historical sequence  $\mathcal{S}^u$ ; Target item embedding  $\mathbf{e}_a$ ; Max diffusion step  $T$ ; Pre-designed diffusion noise schedule  $\beta$ ; Random initialized GEM module  $G_\phi$ ; Random initialized DAM module:  $D_\theta$ ;

**Output:**

Well-trained **GEM** $_\phi$  and **DAM** $_\theta$ .

```

1 while not converged do
2   # GEM
3    $\mathcal{H}^u \leftarrow \mathcal{F}(\mathcal{S}^u)$  ;
4   extract guidance through  $\mathbf{g}^u \leftarrow G_\phi(\mathcal{H}^u)$  ;
5   # DAM
6   sample timestep  $t \sim \text{Uniform}[0, T]$  ;
7   Map the embedding onto a sphere:  $\mathbf{e}'_a \leftarrow L_2\text{-}Norm(\mathbf{e}_a)$  ;
8   Sample a Gaussian in  $\mathbf{e}'_a$ 's tangent space:  $\epsilon \sim \mathcal{N}(0, \mathbb{I})$ ;
9   #  $\exp_x[v] = \cos(\|v\|)x + \sin(\|v\|)\frac{v}{\|v\|}$ 
10  Move along the geodesic:  $x_t \leftarrow \exp_{\mathbf{e}'_a}[\sqrt{1 - \alpha_t}\epsilon]$ ;
11  reconstruction  $\hat{x}_0 = f_\theta(x_t, t, \mathbf{g}^u)$  ;
12  optimize  $\theta, \phi$  with loss  $\mathcal{L}$  via Eq. (23) with Adam ;

```

---

---

**Algorithm 2:** DimeRec Reversing

---

**Input:**

Historical sequence  $\mathcal{S}^u$ ; Max diffusion step  $T$ ; Pre-designed diffusion noise schedule  $\beta$ ; Well-trained GEM module  $G_\phi$ ; Well-trained DAM module:  $D_\theta$ ;

**Output:**

User's next interest embedding  $e_u$ .

```
1  $t \leftarrow T$ ;  
2 while  $t > 0$  do  
3    $\mathcal{H}^u \leftarrow \mathcal{F}(\mathcal{S}^u)$ ;  
4   extract guidance through  $\mathbf{g}^u \leftarrow G_\phi(\mathcal{H}^u)$ ;  
5   reconstruction  $\hat{x}_0 = f_\theta(x_t, t, \mathbf{g}^u)$ ;  
6   move one step  $x_{t-1} \leftarrow L_2\text{-}Norm(\tilde{\mu}_t(x_t, x_0) + \tilde{\beta}_t \cdot \epsilon'), \epsilon' \sim \mathcal{N}(0, \mathbb{I})$ ;  
7   iteration  $t \leftarrow t - 1$ ;  
8  $e_u \leftarrow x_0$ ;
```

---

### 3.5. DiQDiff

#### [WWW 2025] Distinguished Quantized Guidance for Diffusion-based Sequence Recommendation

Distinguished Quantized Guidance for Diffusion-based Sequence...

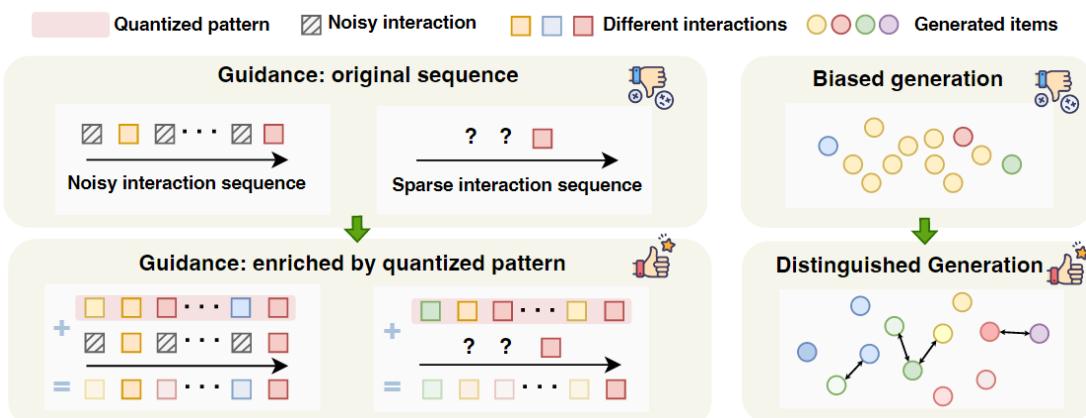
Diffusion models (DMs) have emerged as promising approaches for sequential recommendation due to their strong ability to model data distributions and generate high-quality items. Existing work...

🔗 <https://arxiv.org/abs/2501.17670>



##### 3.5.1. Motivation

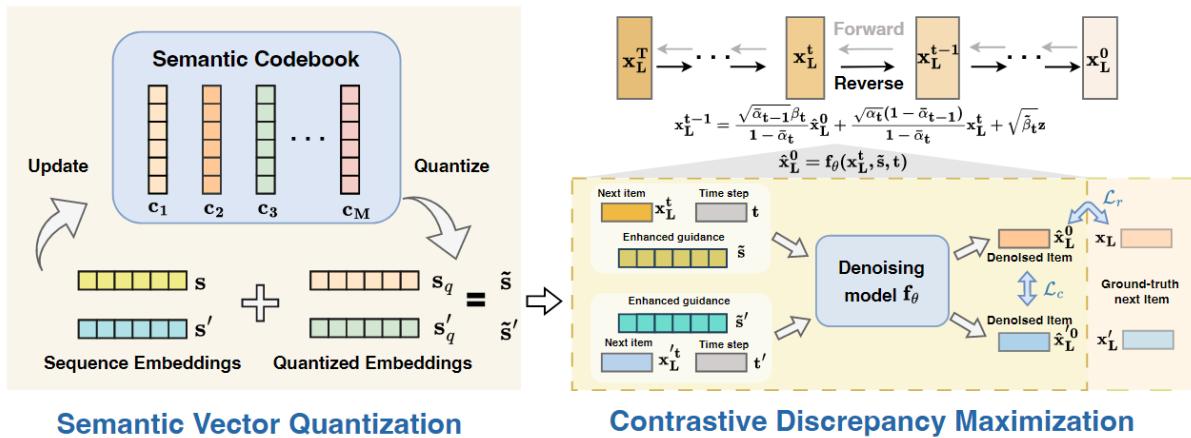
- 当前已经有不少工作开始探索diffusion在序列推荐系统中的应用，但是有两点关键的问题依然没能得到有效解决：
  - 历史交互序列是heterogeneous & noise的：**交互序列在长度与内容上非常异质，低活用户的交互序列非常稀疏，将其作为guidance不能提供足够的信息；高活用户的长交互序列中可能存在很多的噪声（因为交互行为是随机的）
  - 生成过程存在偏差：**尽管diffusion model能为user推荐个性化的item，如果训练数据分布是unbalance的，被popular item占据的概率密度很高，那么训练完全的diffusion model采样时就倾向于生成popular item，这可能损害了推荐的个性化



- 解决方案：

- 针对问题一，其本质上是要提取鲁棒的guidance，本工作提出学习semantic vector quantization (SVQ) 并融入交互序列来丰富guidance以更好理解用户兴趣
- 针对问题二，为了生成差异化的item，本工作通过**对比差异最大化 (CDM)** 以最大化不同解噪轨迹之间的距离，防止为不同user的bias generation

### 3.5.2. Distinguished Quantized Guidance for Diffusion-based Sequence Recommendation (DiQDiff)



- 借助SVQ的guidance提取

对于交互序列  $s = [x_1, x_2, \dots, x_{L-1}]$ ，首先得到它的embedding  $s = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{L-1}$ ，设定一个语义codebook  $\mathbf{C} = \{\mathbf{c}_m\}_{m=1}^M$ ，其中  $\mathbf{c}_m \in \mathbb{R}^{(L-1) \times D}$ ，可以理解成每个item embedding都有  $M$  个VQ。对于每一个item，直接去codebook里找最相似的VQ并替换的方式会使得loss对特征不可导，因此采用的是GumbelSoftmax技巧，将item embedding利用mlp  $f_\phi$  从  $\mathbb{R}^D$  映射到  $\mathbb{R}^M$ ，随后

$$\mathbf{o} = f_\phi(s), \mathbf{o} \in \mathbb{R}^M,$$

$$g_m = \frac{\exp((o_m + n_m)/\tau)}{\sum_{m'=1}^M \exp((o_{m'} + n_{m'})/\tau)},$$

$g_m$  中值最大的index即为对应VQ的index；在得到  $s$  的 quantized code  $s_q$  后，将其融入到原始embedding，即  $\tilde{s} = s + \lambda_q s_q$



Remark. 直观上看，对于交互不足的稀疏序列，最接近的code将提供额外的信息，这些信息与用户的兴趣最一致；对于noise的序列，提取的code将有助于放大可识别的模式并减少不相关噪声的影响，从而提高guidance的表达能力。

codebook  $\mathbf{C}$  的更新由下面给出：

$$\mathbf{c}'_m = \frac{1}{|S_m|} \sum_{s \in S_m} s, \quad (10)$$

where  $S_m$  denotes the set of sequences in the batch samples that select  $m$ -th code. This means that the code vectors maintain the most representative information about the semantic cluster (e.g., collaborative signals and category interests).

- 借助CDM的差异化生成

为了使得diffusion model能够进行差异化生成，对不同sequence denoise生成的 item representation进行余弦相似度最大化：

$$\mathcal{L}_c = \mathbb{E}_{\hat{\mathbf{x}}_L^0} \left[ \log \sum_{\hat{\mathbf{x}}_L'^0 \in B_x} \left[ \exp \left( \text{sim}(\hat{\mathbf{x}}_L^0, \hat{\mathbf{x}}_L'^0) \right) \right] \right], \quad (14)$$

diffusion model损失函数为：

$$\mathcal{L}_r = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[ \left\| \mathbf{x}_L - f_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_L + \sqrt{1 - \bar{\alpha}_t} \epsilon, \tilde{s}, t) \right\|^2 \right], \quad (13)$$

- 损失函数

$$\mathcal{L} = \mathcal{L}_r + \lambda_c \mathcal{L}_c$$

- Inference

---

**Algorithm 2:** Inference process of DiQDiff

---

**Input:** Sequence  $s$ , hyperparameters  $\lambda_q$ , codebook  $C$ , optimal denoising model  $f_\theta(\cdot)$ , and code selection model  $f_\phi(\cdot)$ .

**Output:** Generated item  $\mathbf{x}_L^0$ .

- 1:  $\mathbf{x}_L^T \sim \mathcal{N}(0, I)$  ▷ Sample Gaussian noise.
  - 2:  $\mathbf{s}_q \leftarrow \text{quantize } s \text{ with SVQ.}$
  - 3:  $C \leftarrow \text{Equation 10}$  ▷ Update the codebook.
  - 4:  $\tilde{s} = s + \lambda_q \mathbf{s}_q$  ▷ Enhance the guidance with  $\mathbf{s}_q$ .
  - 5: **for**  $t = T, \dots, 1$  **do**
  - 6:    $\hat{\mathbf{x}}_L^0 = f_\theta(\mathbf{x}_L^t, \tilde{s}, t)$ .
  - 7:    $\mathbf{x}_L^{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \alpha_t} \hat{\mathbf{x}}_L^0 + \frac{\sqrt{\alpha_t} (1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_L^t + \sqrt{\tilde{\beta}_t} \mathbf{z}$ .
  - 8: **end for**
  - 9: **return**  $\mathbf{x}_L^0$
-

inference阶段codebook也更新？

可以发现在 sequence as diffusion guidance 范式中，最开始探索的是如何将历史交互序列信息作为guidance进行denoise，发展出了直接 condition和借助transformer encoder融合两种方法。近期的工作开始关注历史交互序列本身的问题，并探索如何获得更好的guidance