Guoqing Hu

Master student

Lab of Data Science
University of Science and Technology of China

443 Huangshan Road, Hefei, China 230027

Advisor: Prof. Xiang Wang
Prof. An Zhang
Email: HugoChinn AT mail.ustc.edu.cn
WeChat: PB15671953077
QQ: 2306106034
• GitHub • Google Scholar

I am currently in the 2nd year of my Matser program at USTC Lab for Data Science, supervised by Prof. Xiang Wang and Prof. An Zhang. My research interests includes diffusion models and generative recommendations, particularly in discrete diffusion models.

Fading to Grow:

Growing Preference Ratios via Preference Fading Discrete Diffusion for Recommendation

Guoqing Hu[1], An Zhang[1], Shuchang Liu[2], Wenyu Mao[1], Jiancan Wu[1], Xun Yang[1], Xiang Li[2], Lantao Hu[2], Han Li[2], Kun Gai[2], Xiang Wang[1]

[1] University of Science and Technology of China

[2] KUAISHOU

Author

Code

Paper

Q0: Recommendation?

$$p(item|user)$$

- **User Condition:**

- **Item Representation:** discrete ID, continuous embedding, Semantic ID, etc

- **Data sparsity challenge:** user–item interaction data is extremely sparse.

Table 3: Statistics of datasets after preprocessing.

Dataset	# users	# items	# Interactions	sparsity
Movies	6040	3883	1001456	04.27%
Steam	39795	9265	2949605	00.80%
Beauty	22,363	12,101	198,502	00.07%
Toys	19,412	11,924	138,444	00.06%
Sports	35,598	18,357	256,598	00.04%

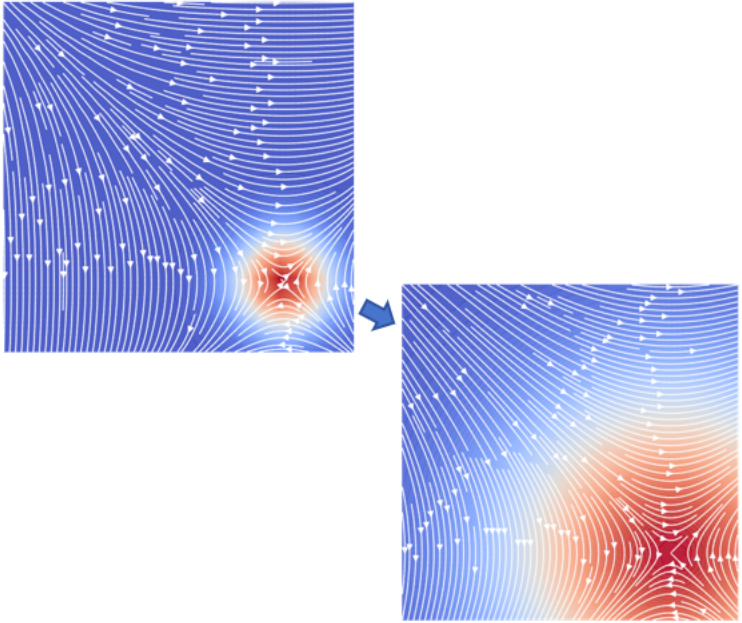
Q1: Why Choose Diffusion Models for Recommendation?

- **Data Sparsity leads to Gradient Collapse**

  Directly modeling sparse data distributions is inherently challenging.

- **Injecting Noise smooths and densifies data**

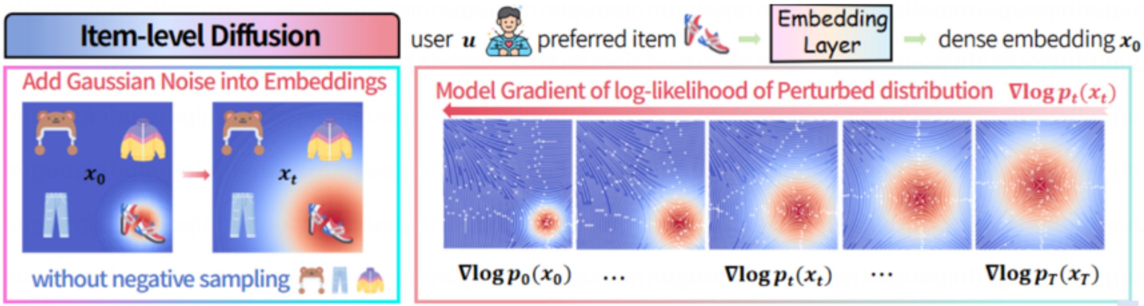
  The injection of noise offers an indirect pathway to tackle data sparsity.



Recent advances in generative modeling (e.g., VAE, Flow, Diffusion, and Consistency models) hold promise for effectively alleviating data sparsity.

Q2: Existing Diffusion Models for Recommendation

Item-level Operation on Euclidean Space: $\mathbb{R}^d$

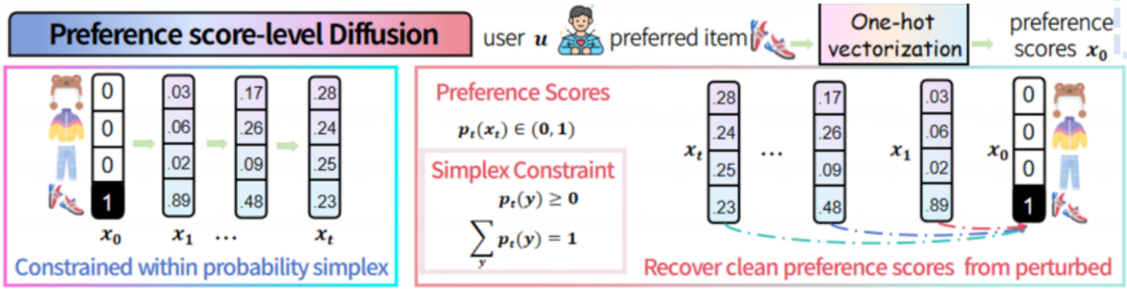


prior noise

complex representation

Preference score-level Operation on Prob. Vector: $\vec{p}$

discrete and ranking nature of recommendation



complex target

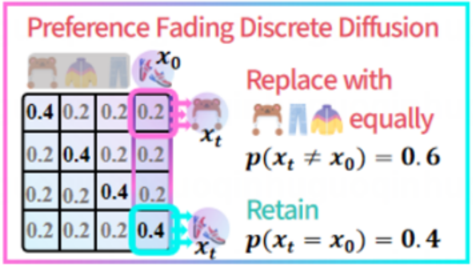
simplex constraints

Q2: WhyDiscrete Diffusion for Recommendation

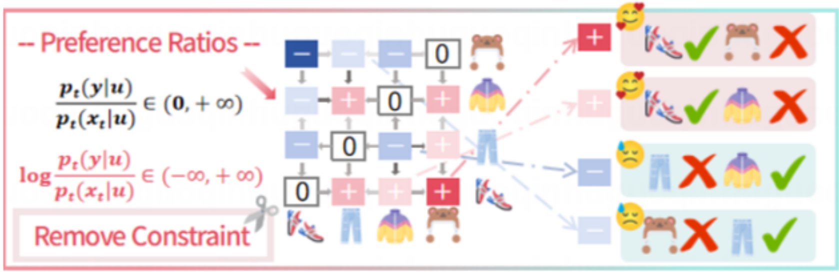
Operation on Discrete item corpus: $\mathcal{X}$

PreferGrow : Discrete Diffusion Modeling Preference Ratios user u preferred item x_0

Preference Fading Perturbation



Modeling Preference ratios



- ➢ **Perturb with other items to align with the discrete nature**
- ➢ **Model preference ratios to reflect the ranking natrue**

PreferGrow: Q3: How Discrete Diffusion for Rec?

-- Preference Fading Forms Reference Ratios --

♡ Forward Perturbation: $\mathbf{P}_{t|0} = \alpha_t \mathbf{I} + (1-\alpha_t)\mathbf{E}$

Retain $p(x_t = x_0) = \alpha_t$ or Replace $p(x_t \neq x_0) = 0.6$

-- Model Preference Ratios via Score Entropy --

♡ Preference Ratios: log concrete score $\log \frac{p_t(y|u)}{p_t(x_t|u)} \in (-\infty, +\infty)$

♡ Score Entropy: advanced loss for discrete diffusion

$$l_{SE}(x_0, x_t, y) = e^{s_\Theta(y,x_t)} - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} s_\Theta(y, x_t) + \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} [\log \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} - 1]$$

-- Preference Growing from Preference Ratios --

♡ Backward Generation: $\mathbf{P}_{s|t} = \mathbf{P}_{t|s}^{-1} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t}\right)^\top\right] \odot \mathbf{P}_{t|s}^\top,$

PreferGrow: Step1: Preference Fading Forms Reference Ratios.

Input: target item ID x_0

$$P_{t|0} = \alpha_t I + (1-\alpha_t) E$$

Output: perturbed item ID x_t

Fading Matrix E:

$$\mathbf{E} = \frac{\vec{p}_T \vec{1}^\top}{\vec{1}^\top \vec{p}_T}.$$

$$\mathbf{E} = \begin{pmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 1 \end{pmatrix}$$

Pointwise

$$\mathbf{E} = \begin{pmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \ddots & \vdots \\ \frac{1}{N} & \cdots & \frac{1}{N} \end{pmatrix}$$

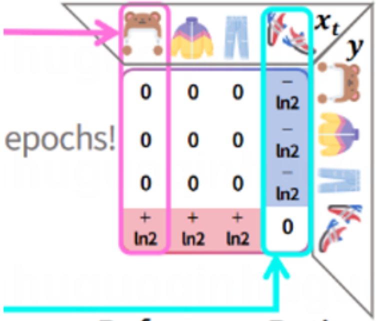
Pairwise

$$\mathbf{E} = \begin{pmatrix} \frac{\lambda}{N} & \cdots & \frac{\lambda}{N} & \frac{\lambda}{N} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\lambda}{N} & \cdots & \frac{\lambda}{N} & \frac{\lambda}{N} \\ 1-\lambda & \cdots & 1-\lambda & 1-\lambda \end{pmatrix}$$

Hybridwise

Input: item ID x_0, x_t, y

$$r_t = \log \frac{p_{t|0}(y|x_0, u)}{p_{t|0}(x_t|x_0, u)}$$



Output: preference $y \succ x_t | x_0$

PreferGrow: Q4: How to Design a Feasible Fading Matrix E ?

- Simplex Constrains ⇒ Non-negativity & Normalization
- Invertibility & Markov Property ⇒ Idempotence

Rank-1 Solution: $\mathbf{E} = \dfrac{\vec{p}_T \vec{1}^\top}{\vec{1}^\top \vec{p}_T}.$

- Example 1: Pointwise Masked Diffusion

$$\vec{p}_T = \vec{e}_{-1} \in \mathbb{R}^{N+1} \qquad \mathbf{E} = \begin{pmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 1 \end{pmatrix}$$

- Example 2: Pairwise Uniform Diffusion

$$\vec{p}_T = \vec{1} \in \mathbb{R}^{N} \qquad \mathbf{E} = \begin{pmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \ddots & \vdots \\ \frac{1}{N} & \cdots & \frac{1}{N} \end{pmatrix}$$

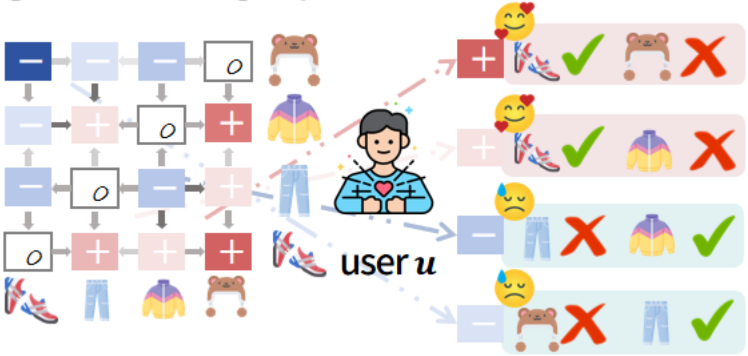
Rank-r Solution: $\mathbf{E} = \sum_{i=1}^{r} \dfrac{\vec{p}^{\,i} \vec{1}^\top_{\mathrm{supp}(\vec{p}^{\,i})}}{\vec{1}^\top_{\mathrm{supp}(\vec{p}^{\,i})} \vec{p}^{\,i}}$

- Partition N items into r clusters
- Independent rank-1 solution for each cluster

PreferGrow: Q5: Why is it called Preference Ratios?

- Preference Ratios: log concrete score $\log \frac{p_t(y|u)}{p_t(x_t|u)} \in (-\infty, +\infty)$

- Illustration: the finest-grained ranking objective



- Bradley–Terry model: Direct Preference Optimization

$$p(i_p \succ i_d \mid u) = \frac{p(i_p \mid u)}{p(i_p \mid u) + p(i_d \mid u)} = \frac{1}{1 + \exp\left(-\log \frac{p(i_p|u)}{p(i_d|u)}\right)} = \sigma(\log \frac{p(i_p \mid u)}{p(i_d \mid u)}).$$

PreferGrow: Step2: Modeling Preference Ratios with Reference

Input: reference ratios r_t

const

Loss: $\mathcal{L}_{SE}(x_0, x_t, y) = e^{s_\Theta(x_t,t,u)_y} - s_\Theta(x_t, t, u)_y \cdot e^{r_t(x_0,x_t,y)} + \boxed{e^{r_t(x_0,x_t,y)}(r_t(x_0, x_t, y) - 1)}.$

compute $\boxed{s_\Theta(x_t, t, u)_y = \mathbf{y}^\top \mathbf{MLP}(\text{concat}(\mathbf{x}_t, \mathbf{t}, \mathbf{u})),}$

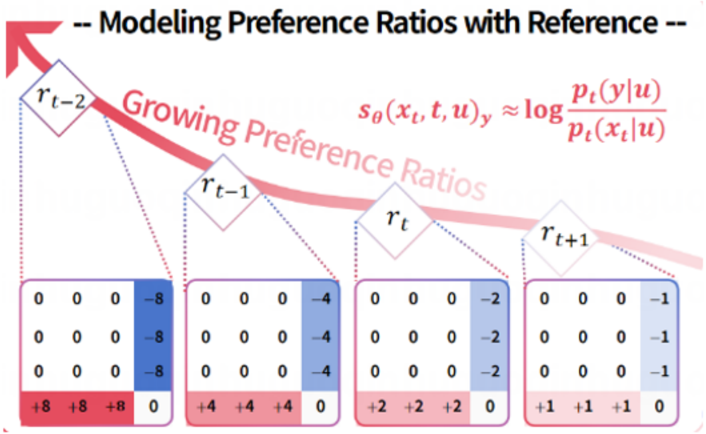
Connection to Ranking Loss:

$$\pi_{y \succ x_t | x_0} = p(y \succ x_t \mid x_0) = \sigma(r_t(x_0, x_t, y))$$

$$\mathcal{L}_{sBCE}(x_0, x_t, y) = -\pi_{y \succ x_t | x_0} \log \sigma(s_\Theta(x_t, t, u)_y) - (1 - \pi_{y \succ x_t | x_0}) \log(1 - \sigma(s_\Theta(x_t, t, u)_y)).$$

$$\nabla_{s_\Theta} \mathcal{L}_{SE} = (1 + e^{s_\Theta(x_t,t,u)_y})(1 + e^{r_t(x_0,x_t,y)}) \nabla_{s_\Theta} \mathcal{L}_{sBCE}.$$

Output: score entropy loss l_{SE}



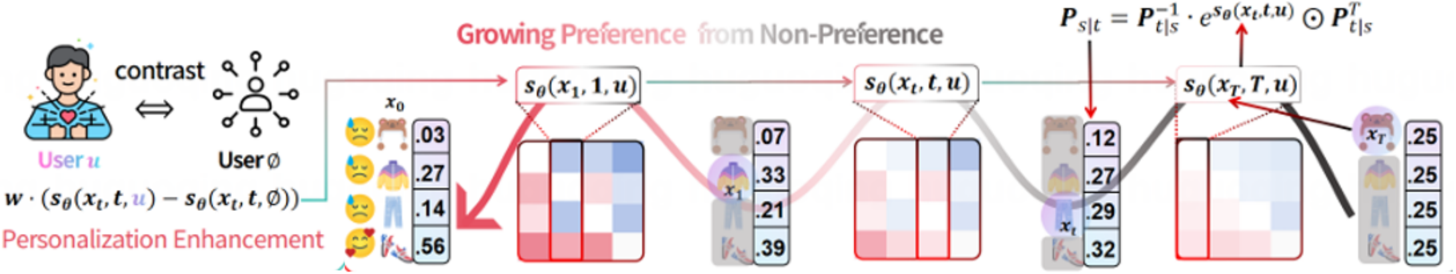
NeurIPS

PreferGrow: Step3: Preference Growing from Reference Ratios

♡ Backward Generation: $\mathbf{P}_{s|t} = \mathbf{P}_{t|s}^{-1} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^{\top} \right] \odot \mathbf{P}_{t|s}^{\top},$

Input: user condition u, random item ID $x_T \sim \vec{p}_T$

$$\mathbf{E} = \frac{\vec{p}_T \vec{1}^{\top}}{\vec{1}^{\top} \vec{p}_T}.$$



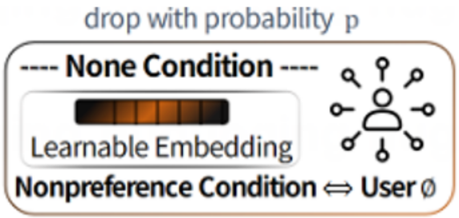
Output: preference score $p(x_0|u), x \in \mathcal{X}$

PreferGrow: Q6: How to condition on the user?

Input: user infomation u

$$\mathbf{u} = \mathrm{SeqRec}(u)$$

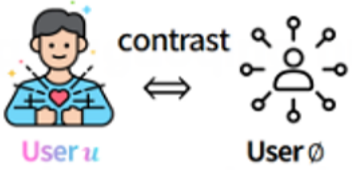
-- Non-preference User Modeling --



Output:

pos user condition u
neg user condition ϕ

Personalized Contrast
-- against Non-preference User --



$$w \cdot (s_\theta(x_t, t, u) - s_\theta(x_t, t, \emptyset))$$

Personalization Enhancement

Experiment: Overall Performance

Table 1: Performance comparison across different datasets and baselines.

Dataset		MoviesLens		Steam		Beauty		Toys		Sports	
Method		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
SASRec	@5	0.0905	0.0502	0.0321	0.0192	0.0326	0.0221	0.0340	0.0258	0.0154	0.0101
	@10	0.1709	0.0760	0.0572	0.0272	0.0438	0.0257	0.0438	0.0289	0.0230	0.0126
	@20	0.2899	0.1059	0.0957	0.0368	0.0595	0.0297	0.0499	0.0305	0.0298	0.0143
Caser	@5	0.0925	0.0576	0.0258	0.0163	0.0170	0.0106	0.0093	0.0072	0.0051	0.0030
	@10	0.1605	0.0794	0.0446	0.0223	0.0219	0.0122	0.0129	0.0083	0.0093	0.0043
	@20	0.2592	0.1042	0.0736	0.0295	0.0295	0.0141	0.0185	0.0098	0.0149	0.0057
GRURec	@5	0.0892	0.0551	0.0255	0.0156	0.0130	0.0082	0.0124	0.0078	0.0070	0.0048
	@10	0.1534	0.0757	0.0441	0.0216	0.0188	0.0101	0.0180	0.0092	0.0107	0.0059
	@20	0.2501	0.1000	0.0769	0.0298	0.0313	0.0132	0.0304	0.0117	0.0171	0.0076
DreamRec	@5	0.0676	0.0437	0.0109	0.0073	0.0300	0.0247	0.0381	0.0304	0.0095	0.0084
	@10	0.1083	0.0568	0.0157	0.0088	0.0353	0.0265	0.0412	0.0314	0.0112	0.0089
	@20	0.1610	0.0701	0.0218	0.0104	0.0402	0.0277	0.0463	0.0327	0.0149	0.0098
PreferDiff	@5	0.0538	0.0349	0.0167	0.0105	0.0335	0.0272	0.0386	0.0308	0.0168	0.0130
	@10	0.0852	0.0450	0.0297	0.0145	0.0434	0.0304	0.0494	0.0343	0.0211	0.0144
	@20	0.1270	0.0555	0.0526	0.0203	0.0577	0.0340	0.0644	0.0380	0.0256	0.0155
DiffRec	@5	0.0266	0.0121	0.0268	0.0143	0.0095	0.0055	0.0053	0.0028	0.0063	0.0035
	@10	0.0889	0.0320	0.0657	0.0268	0.0218	0.0094	0.0153	0.0059	0.0117	0.0052
	@20	0.1768	0.0541	0.1159	0.0394	0.0355	0.0128	0.0226	0.0078	0.0194	0.0072
DDSR	@5	0.0871	0.0533	0.0252	0.0157	0.0291	0.0217	0.0386	0.0302	0.0146	0.0109
	@10	0.1523	0.0742	0.0437	0.0216	0.0434	0.0262	0.0479	0.0332	0.0213	0.0130
	@20	0.2450	0.0975	0.0736	0.0291	0.0608	0.0306	0.0618	0.0367	0.0298	0.0151
PreferGrow	Impr.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Hybrid	@5	0.1409	0.0911	**0.0615**	**0.0406**	**0.0420**	**0.0322**	**0.0419**	**0.0311**	**0.0207**	**0.0142**
	@10	0.2229	0.1174	**0.0935**	**0.0508**	**0.0532**	**0.0358**	0.0480	0.0331	**0.0267**	**0.0162**
	@20	0.3355	0.1458	**0.1413**	**0.0628**	**0.0708**	**0.0402**	0.0625	0.0367	**0.0343**	**0.0181**
Adaptive	@5	**0.1413**	**0.0912**	0.0583	0.0375	0.0396	0.0310	0.0413	0.0304	0.0168	0.0127
	@10	**0.2240**	**0.1177**	0.0914	0.0481	0.0508	0.0347	**0.0513**	**0.0337**	0.0207	0.0140
	@20	**0.3362**	**0.1460**	0.1387	0.0600	0.0610	0.0373	**0.0642**	**0.0370**	0.0267	0.0155

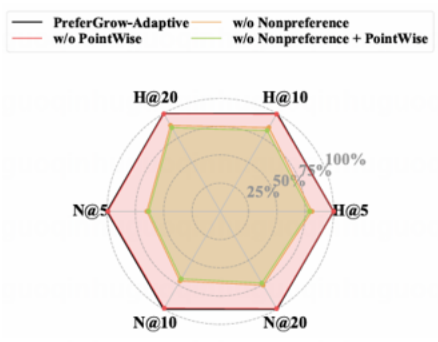
PreferGrow

(discrete diffusion-based recommender)

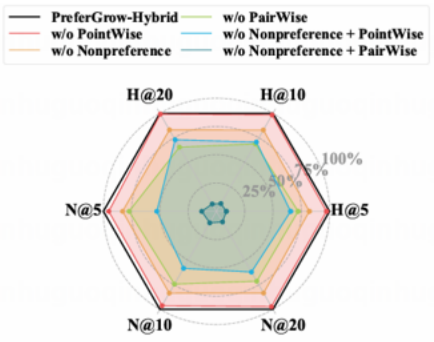
works well and outperform

other diffusion-based recommenders.

Experiment: Further Anylsis



(a) Abalation of Adaptive on Steam.

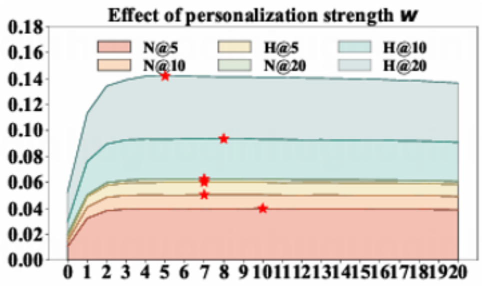


(b) Ablation of Hybrid on Steam.

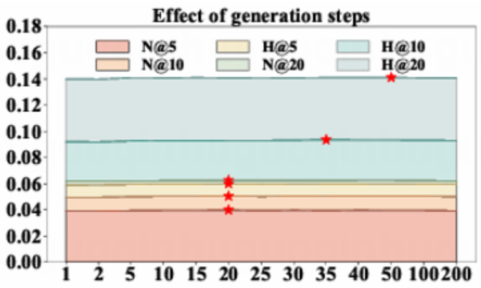
Pairwise ranking
(via Uniform Diffusion)

combined with personalized guidance
(via Classifier-Free Guidance)

is crucial for PreferGrow.



(e) w of Hybrid on Steam.



(f) Steps of Hybrid on Steam.

♡ **Personalization strength**

♡ **Multi-step v.s. Single-step**

Conclusions

♡ **Theoretical Foundation:** **a well-defined matrix-based formulation**, **flexible and interpretable closed-form solution** of this preference fading matrix and theoretically prove that its idempotent property is critical for ensuring both the Markov property and reversibility of the diffusion process.

♡ **Empirical validation:** **extensive experiments on five benchmark datasets** demonstrate that PreferGrow consistently **outperforms existing diffusion-based recommenders**, validating the practical effectiveness of its theoretical foundations.