RELDIFF: A REINFORCEMENT LEARNING GUIDED DIFFUSION FRAMEWORK FOR RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have demonstrated promising potential in recommender systems. Many Gaussian diffusion methods initiate the generation process by injecting random Gaussian noise, without conditioning the noise distribution on the user. Owing to the sparse nature of recommendation data, such perturbations can disturb interaction structure; our investigation also reveals generation collapse during training. To address the issue, we propose **RelDiff**, a novel reinforcement learning guided diffusion framework for recommendation. Rather than using a fixed, user-independent Gaussian distribution, we employ a Preference-Conditional Policy to generate user-conditioned Gaussian noise, learned from immediate reconstruction rewards, to guide the diffusion process. Besides, to tackle generation collapse, we design a Negative-aware Refinement strategy that transforms stagnant optimization into a push-pull process, using confidence-weighted attraction and repulsion around a reference model. Additionally, we decouple the diffusion process from the inference stage by injecting reconstruction information into item embeddings through contrastive learning, supporting recommendation without iterative denoising. Extensive experiments on five real-world datasets report higher ranking accuracy than the evaluated baselines. Our work explores the combination of user-conditioned perturbations and reference-based refinement for diffusion recommendation.

1 INTRODUCTION

With the growth of digital information, recommender systems have emerged as indispensable mechanisms for filtering information overload and efficiently distributing content. As user interaction records accumulate at an unprecedented scale, there is an escalating demand for advanced recommendation paradigms capable of accurately uncovering and modeling users' latent preferences. In recent years, diffusion models have gained attention within the research community, owing to their powerful generative capabilities and flexibility in modeling complex data distributions. Distinct from conventional discriminative approaches, diffusion models formulate recommendation as a conditional generative process, iteratively denoising corrupted signals to reconstruct user preferences. This paradigm allows them to capture the complex nature of user interests with high fidelity, making them particularly well-suited for addressing the data scarcity and noise inherent in real-world recommendation scenarios.

Diffusion models have emerged as a powerful generative paradigm in recommendation (Wang et al., 2023; Li et al., 2023; Yang et al., 2023; Zhao et al., 2024; Zhu et al., 2024; Mao et al., 2025; Ho et al., 2020; Song et al., 2021b; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021), enabling high-quality user preference generation. As an early diffusion-based collaborative filtering model, DiffRec (Wang et al., 2023) introduces the diffusion process to collaborative filtering, modeling user preferences by corrupting the interaction history vector with Gaussian noise and learning to reconstruct the original interactions. Extending this generative paradigm to sequential recommendation, DiffuRec (Li et al., 2023) proposes to diffuse on the latent item embedding space rather than discrete indices, aiming to model the uncertainty of user dynamic interests for future behavior prediction. Subsequently, DreamRec (Yang et al., 2023) fundamentally reshapes sequential recommendation as a learning-to-generate task. It employs a guided diffusion process to directly generate an oracle item representation that captures the user's true preference, effectively eliminating the need for negative sampling during training. To further enhance the recovery of preferences from noisy data,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

DDRM (Zhao et al., 2024) optimizes the generation quality by designing a specific denoising strategy tailored for the recommendation context. GiffCF (Zhu et al., 2024) instead uses graph-signal smoothing and user-history-conditioned reconstruction, with a noise-free reverse process. Additionally, numerous other models (Walker et al., 2022; Jiang et al., 2024; Xu et al., 2024; Li et al., 2025b; Han et al., 2025; Chen et al., 2025a; Ma et al., 2024c;b; Chen et al., 2025b; Ma et al., 2025; Hou et al., 2024; Ma et al., 2024a; Wang et al., 2024; Li et al., 2025a) explore diffusion-based and conditional representation learning for recommendation, including graph diffusion beyond generative denoising.

Motivations. Despite achieving impressive progress, many Gaussian diffusion-based recommendation models rely on user-independent noise and can exhibit generation collapse during training. First, as the standard Gaussian noise is completely independent of the authentic user preference, injecting such noise can perturb the sparse interaction structure used to infer connections between users and items. This motivates learning a perturbation distribution conditioned on the user rather than using the same noise distribution for everyone. Second, existing diffusion models can encounter a convergence bottleneck, which we name **Generation Collapse**. The quality of samples generated during noise injection and subsequent denoising can be uneven. We observe that extended reconstruction training yields only small performance improvements, although these curves do not isolate the cause of the slowdown. This motivates adapting the training strategy to different reconstruction outcomes.

Challenges. The key to alleviating these issues lies in transforming the diffusion process from stochastic generation into a controllable one, in which the injected noise is conditioned on the user, and in adapting the training strategy to different reconstruction outcomes. Nevertheless, achieving these goals presents two challenges: (i) **How to generate preference noise that can reliably guide diffusion**? Unlike user-independent standard Gaussian noise, preference noise is drawn from a learned user-conditioned Gaussian distribution. However, such noise lacks explicit supervision, and its effectiveness can only be assessed after it is used to generate representations. We therefore investigate a learned evaluator that predicts reconstruction reward from the user and the generated noise. Moreover, to incorporate personalized information aligned with user interests, it is essential to learn a user-conditioned continuous policy from scalar feedback while ensuring sufficient exploration. (ii) **How to efficiently refine the model to tackle the generation collapse**? As training progresses, the standard reconstruction objective weights all samples equally, failing to differentiate between well-learned and poorly-learned instances, which motivates exploring confidence-dependent refinement in the later stages of optimization. Simply discarding poorly reconstructed samples also eliminates the information they contain. We therefore retain these instances, while recognizing that a large reconstruction error does not identify an incorrect observed interaction.

Contributions. In this paper, we propose a novel ***Re****inforcement* ***L****earning guided* ***Diff****usion framework for recommendation*, called **RelDiff**. Our framework makes noise generation user-conditioned and combines it with confidence-dependent reconstruction training. Specifically, for the first challenge, we transform the forward process from random noise injection into a controllable one by learning a **Preference Noise Generation (PNG)** policy, which produces a user-conditioned Gaussian perturbation for the forward process. To overcome the absence of direct supervision for noise quality, we incorporate explicit feedback to learn a policy with sufficient exploration, using two critics to estimate immediate reconstruction rewards for policy optimization. For the second challenge, to avoid generation collapse, we devise a **Negative-aware Refinement (NAR)** strategy that re-weights training dynamics by actively leveraging low-quality generation outcomes, combining confidence-weighted attraction and repulsion around a historical reference rather than discarding those instances. In addition, we adopt a contrastive injection mechanism to distill the generative preference into item embeddings, allowing embedding-based recommendation without iterative denoising at inference. Experiments on five real-world datasets compare our framework with graph, contrastive, and diffusion baselines. Our main contributions are summarized as follows:

- We propose a novel diffusion-based generative recommendation framework that replaces standard Gaussian noise with controllable, user-conditioned preference noise, thereby making the forward perturbation distribution adaptive to the user representation.
- We develop a Reinforcement Learning based **Preference Noise Generation** policy that learns a user-conditioned control signal from explicit feedback, using immediate reconstruction rewards and entropy regularization to train the policy.

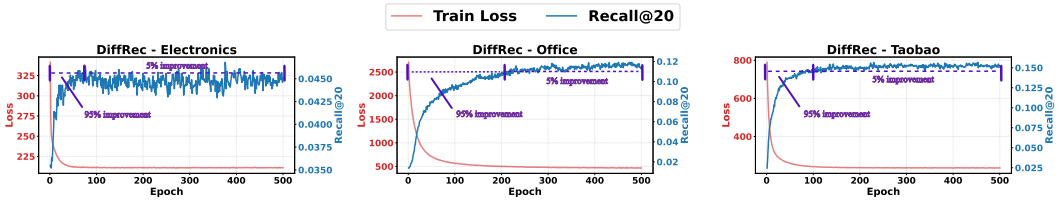


Figure 1: Training dynamics of diffusion model (DiffRec) across three datasets.

- We introduce a **Negative-aware Refinement** strategy to tackle generation collapse through confidence-dependent attraction and repulsion around a reference model, while retaining poorly reconstructed instances.
- We conduct extensive evaluations on five real-world datasets, and the reported results exceed the evaluated baselines. We further examine PNG/NAR ablations, training dynamics, per-epoch training cost, and signal-to-noise power ratios.

2 Investigation: Generation Collapse Analysis

We track **DiffRec** on Amazon-Electronics, Amazon-Office, and Taobao over 500 epochs. Figure 1 presents reconstruction loss and Recall@20: the plotted runs show comparatively rapid early improvement followed by smaller gains. On Office, Recall@20 rises steeply at first and then changes more gradually; Electronics and Taobao also fluctuate around slower-changing later trends. Continued reconstruction optimization therefore does not produce equally large ranking improvements throughout training. We call this convergence bottleneck **generation collapse**: extended training yields only limited performance improvements, with much of the later optimization producing small gains. Exact thresholds behind the original percentage annotations cannot be independently checked without logs.

This generation collapse motivates examining how a uniform reconstruction objective uses samples with different errors. **Negative-aware Refinement (NAR)** retains poorly reconstructed samples and uses relative confidence and a reference model to form an attraction–repulsion objective. High error does not identify an incorrect interaction, and repulsion is not necessarily corrective for observed positives. Section 5.4 examines the reported training behavior; Appendix A states the limits of the theoretical interpretation.

3 Problem Definition

• **User-Item Graph.** Let $\mathcal{U} = \{u_1, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, \ldots, i_N\}$ denote users and items. Observed interactions form a bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ and a binary matrix $\boldsymbol{A} \in \{0,1\}^{M \times N}$. An entry $A_{mn} = 1$ indicates an observed interaction. Missing edges indicate unobserved interactions.

• **Problem Statement.** Given $\mathcal{G}$, we learn user and item representations to rank items absent from each user's training interactions. User-conditioned noise $\boldsymbol{w}$ guides representation learning; inference uses the learned embeddings.

4 Methodology

Figure 2 presents RelDiff. PNG makes the forward perturbation distribution depend on the user and learns from reconstruction feedback. Contrastive injection then transfers reconstructed information into item embeddings, while NAR tackles generation collapse through confidence-weighted attraction and repulsion during reconstruction training. These modules operate during training; graph propagation and dot-product ranking use the resulting embeddings at inference. The following subsections specify their complementary roles.

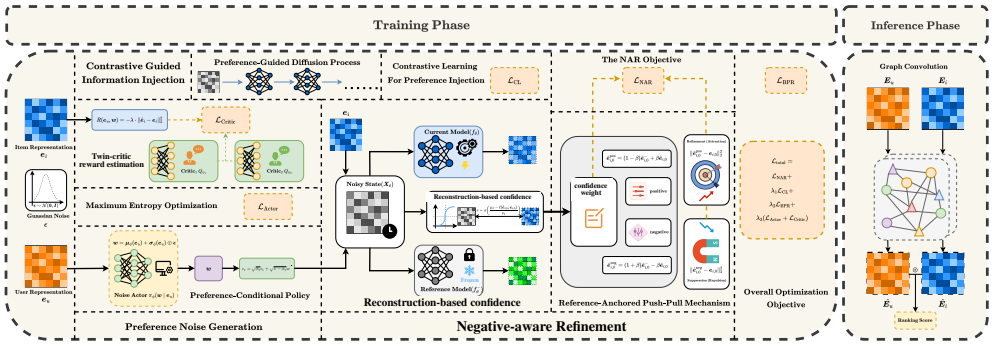


Figure 2: The overall framework of the proposed RelDiff. The architecture comprises the training phase and inference phase, incorporating the Preference Noise Generation Module for user-conditioned noise generation, Contrastive Guided Information Injection module for information extraction, and the Negative-aware Refinement strategy to tackle the generation collapse.

4.1 Preference Noise Generation

Standard Gaussian noise is independent of user intent. PNG instead treats noise injection as a one-step decision problem, using a user-conditioned continuous policy to choose perturbations. Since the noise itself has no ground-truth label, reconstruction provides scalar feedback; two critics learn to evaluate the generated noise, and maximum-entropy optimization encourages exploration.

4.1.1 Preference-Conditional Policy

The policy $\pi_\phi(\boldsymbol{w} \mid \boldsymbol{e}_u)$ conditions noise on the user representation $\boldsymbol{e}_u$. We use a diagonal Gaussian and sample with reparameterization:

$$\boldsymbol{w} = \boldsymbol{\mu}_\phi(\boldsymbol{e}_u) + \boldsymbol{\sigma}_\phi(\boldsymbol{e}_u) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{1}$$

where $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}_\phi$ are the actor's mean and standard deviation networks. The sampled $\boldsymbol{w}$ replaces standard Gaussian noise in the forward diffusion process.

4.1.2 Twin-Critic Reward Estimation

Given a user state $\boldsymbol{e}_u$ and a noise action $\boldsymbol{w}$, we define the reward $R(\boldsymbol{e}_u, \boldsymbol{w})$ based on the diffusion model's reconstruction of the target item $\boldsymbol{e}_i$. Let $\hat{\boldsymbol{e}}_i$ denote the predicted reconstruction, then:

$$R(\boldsymbol{e}_u, \boldsymbol{w}) = -\lambda \cdot \|\hat{\boldsymbol{e}}_i - \boldsymbol{e}_i\|_2^2, \tag{2}$$

Here λ scales the immediate reward; larger reward means smaller reconstruction error. This proxy need not measure ranking quality or semantic alignment directly.

The two critics Q_{ψ_1} and Q_{ψ_2} regress directly on the immediate reconstruction reward:

$$\mathcal{L}_{\text{Critic}} = \sum_{i=1,2} \mathbb{E}_{(\boldsymbol{e}_u, \boldsymbol{w}) \sim \mathcal{D}} \left[\left(Q_{\psi_i}(\boldsymbol{e}_u, \boldsymbol{w}) - R(\boldsymbol{e}_u, \boldsymbol{w}) \right)^2 \right]. \tag{3}$$

There is no next-state term, Bellman bootstrap, or discounted long-term return in this objective. The formulation is therefore a one-step contextual-bandit-style actor–critic objective, rather than Double Q-learning. The actor uses the smaller of the two predicted rewards.

4.1.3 Maximum Entropy Optimization

Optimizing only the currently highest predicted reward may restrict the noise explored by the policy. An entropy term encourages stochastic exploration while the actor learns to maximize predicted reconstruction reward:

$$\mathcal{L}_{\text{Actor}} = \mathbb{E}_{\boldsymbol{e}_u \sim \mathcal{D}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\alpha \log \pi_\phi(\boldsymbol{w} \mid \boldsymbol{e}_u) - \min_{i=1,2} Q_{\psi_i}(\boldsymbol{e}_u, \boldsymbol{w}) \right], \tag{4}$$

Here α balances the estimated reward and policy entropy.

4.2 Contrastive Guided Information Injection

The diffusion reconstruction and final ranking use different representations. Contrastive learning connects them by drawing an item embedding toward its reconstructed view and contrasting it with other in-batch views. This injects the generation signal into the item embeddings during training, allowing inference to use graph-based embeddings without running the diffusion process.

4.2.1 Preference-Guided Diffusion Process

For item embedding $\boldsymbol{e}_i$, the forward step uses the user-conditioned noise from Section 4.1:

$$\boldsymbol{z}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{e}_i + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{w}, \quad \boldsymbol{w} \sim \pi_\phi(\cdot|\boldsymbol{e}_u), \tag{5}$$

Here $\bar{\alpha}_t$ is the cumulative diffusion coefficient. The denoiser reconstructs $\hat{\boldsymbol{e}}_i$ from this conditioned input; below we write $\boldsymbol{e}_{i,0} = \boldsymbol{e}_i$ and $\boldsymbol{e}_{i,t} = \boldsymbol{z}_t$. Conditioning on users does not by itself establish that the learned perturbation direction encodes their preferences.

4.2.2 Contrastive Learning for Preference Injection

We align each item embedding $\boldsymbol{e}_i$ with its reconstructed representation $\hat{\boldsymbol{e}}_i$, treating other in-batch items as negatives:

$$\mathcal{L}_{\mathrm{CL}} = -\sum_{i=1}^{B} \log \frac{\exp(\mathrm{sim}(\boldsymbol{e}_i, \hat{\boldsymbol{e}}_i)/\tau)}{\sum_{j=1}^{B} \exp(\mathrm{sim}(\boldsymbol{e}_i, \hat{\boldsymbol{e}}_j)/\tau)}, \tag{6}$$

Here B is the batch size (distinct from the number of items N), τ is the temperature, and sim denotes embedding similarity. Minimizing $\mathcal{L}_{\mathrm{CL}}$ transfers reconstruction information into the embeddings used for ranking; the precise similarity and masking implementation remains to be documented.

4.3 Negative-aware Refinement

Standard reconstruction can exhibit **generation collapse**: rapid early learning is followed by small performance gains despite continued training. Its squared-error objective is:

$$\mathcal{L}(\theta) = \mathbb{E}\big[\|f_\theta(\boldsymbol{e}_{i,t}) - \boldsymbol{e}_{i,0}\|_2^2\big], \tag{7}$$

To alleviate generation collapse, NAR retains samples of differing reconstruction quality and combines a dynamic confidence score with a reference-anchored push-pull objective.

4.3.1 Reconstruction-based Confidence

NAR maps reconstruction error to a relative confidence score $r \in [0, 1]$. For $\hat{\boldsymbol{e}}_{i,0} = f_\theta(\boldsymbol{e}_{i,t})$, the score is:

$$r = \sigma\left(\frac{\mu_\ell - \ell(\hat{\boldsymbol{e}}_{i,0}, \boldsymbol{e}_{i,0})}{\nu_\ell}\right), \tag{8}$$

Here ℓ measures reconstruction error and (μ_ℓ, ν_ℓ) are batch statistics. Larger r indicates lower error relative to the batch. It is not a calibrated probability of a true preference, and high error does not identify an incorrect observed interaction. The exact scale statistic and numerical stabilization are not specified in the available records.

4.3.2 Reference-Anchored Push-Pull Mechanism

The reference model f_θ^- is described as a previously best-performing checkpoint. Its output $\hat{\boldsymbol{e}}_{i,0}^- = f_\theta^-(\boldsymbol{e}_{i,t})$ anchors two affine combinations. The checkpoint-selection criterion and update schedule require confirmation; we do not assume that this historical output equals the current conditional mean.

$$\hat{\boldsymbol{e}}_{i,0}^{\mathrm{pos}} = (1-\beta)\hat{\boldsymbol{e}}_{i,0}^- + \beta\hat{\boldsymbol{e}}_{i,0}, \tag{9}$$

$$\hat{\boldsymbol{e}}_{i,0}^{\mathrm{neg}} = (1+\beta)\hat{\boldsymbol{e}}_{i,0}^- - \beta\hat{\boldsymbol{e}}_{i,0}, \tag{10}$$

The coefficient $\beta > 0$ controls these combinations. Reference anchoring alone does not guarantee stable updates or prevent drift.

Table 1: Statistics of the datasets

Datasets	ML-1M	Office	Baby	Taobao	Electronics
#Users	6040	4,905	19,445	12,539	32,886
#Items	3706	2,420	7,050	8,735	52,974
#Int.	1,000,209	53,258	159,669	83,648	337,837
Sparsity	95.53%	99.55%	99.88%	99.92%	99.98%

4.3.3 THE NAR OBJECTIVE

We unify refinement and suppression into a single loss:

$$\mathcal{L}_{\text{NAR}} = \frac{1}{\beta}\,\mathbb{E}\Big[\underbrace{r \cdot \|\hat{\boldsymbol{e}}^{\text{pos}}_{i,0} - \boldsymbol{e}_{i,0}\|_2^2}_{\text{Refinement (Attraction)}} + \underbrace{(1-r) \cdot \|\hat{\boldsymbol{e}}^{\text{neg}}_{i,0} - \boldsymbol{e}_{i,0}\|_2^2}_{\text{Suppression (Repulsion)}}\Big]. \tag{11}$$

The two terms weight attraction and repulsion relative to the reference, aiming to mitigate generation collapse without discarding poorly reconstructed samples. Low-confidence instances remain observed positive interactions: repulsion can move a prediction away from such a target, so it is not inherently corrective. Appendix A gives conditional algebra and its limitations. Whether gradients flow through r, the batch statistics, the reference output, and the target embedding is unverified without implementation records.

4.4 OVERALL OPTIMIZATION OBJECTIVE

Reconstruction and contrastive alignment shape representations but do not directly optimize candidate ordering. We therefore add Bayesian Personalized Ranking (BPR) to connect these representations to the recommendation objective.

For each user u, an observed item i should score above an unobserved item j:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,i,j)\in\mathcal{D}} \ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right), \tag{12}$$

where $\hat{y}_{ui}$ represents the predicted score between user u and item i.

• **Joint Training.** The overall objective combines reconstruction, contrastive injection, ranking, and policy optimization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NAR}} + \lambda_1\mathcal{L}_{\text{CL}} + \lambda_2\mathcal{L}_{\text{BPR}} + \lambda_3(\mathcal{L}_{\text{Actor}} + \mathcal{L}_{\text{Critic}}), \tag{13}$$

Here $\lambda_1, \lambda_2, \lambda_3$ weight contrastive injection, ranking, and policy optimization. This sum lists the loss terms; parameter-specific updates and gradient blocking are not specified here.

4.5 INFERENCE PHASE

Inference uses learned embeddings, LightGCN-style propagation, and dot-product ranking without iterative diffusion denoising. To state one propagation step with consistent dimensions, let $E_u^{(0)} \in \mathbb{R}^{M\times d}$ and $E_i^{(0)} \in \mathbb{R}^{N\times d}$ be the learned embeddings and let $\bar{A} \in \mathbb{R}^{M\times N}$ be the normalized interaction matrix:

$$E_u^{(k+1)} = \bar{A}E_i^{(k)}, \qquad E_i^{(k+1)} = \bar{A}^{\mathsf{T}}E_u^{(k)}, \qquad \bar{A}_{ui} = \frac{A_{ui}}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_i|}}. \tag{14}$$

The final graph embeddings g_u, g_i yield scores $\hat{y}_{ui} = \langle g_u, g_i\rangle$. The draft reports exploring 0–3 propagation layers; dataset-specific choices and layer aggregation are not recoverable from the supplied materials. The absence of iterative denoising is an architectural property, not a measured latency result.

Table 2: Transposed performance comparison of different methods on the five datasets. The best and second-best performances are indicated in bold and underlined font, respectively.

	Baby		Office		Taobao		Electronics		ML-1M	
Model	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
	0.0532	0.0226	0.0928	0.0400	0.1223	0.0523	0.0368	0.0163	0.0877	0.0347
NGCF	0.1002	0.0320	0.1684	0.0563	0.1902	0.0658	0.0593	0.0209	0.1785	0.0525
	0.0715	0.0298	0.1215	0.0558	0.1502	0.0681	0.0394	0.0178	0.0790	0.0278
LightGCN	0.1255	0.0409	0.2064	0.0702	0.2250	0.0830	0.0645	0.0229	0.1666	0.0451
	0.0651	0.0269	0.0778	0.0363	0.1303	0.0621	0.0398	0.0178	0.1057	0.0377
UltraGCN	0.1271	0.0393	0.1479	0.0515	0.2361	0.0843	0.0668	0.0233	0.2040	0.0570
	0.0772	0.0313	0.1268	0.0529	0.1496	0.0725	0.0417	0.0199	0.0867	0.0297
MGDCF	0.1321	0.0435	0.2017	0.0634	0.2290	0.0891	0.0652	0.0250	0.1829	0.0489
	0.0656	0.0297	0.1151	0.0549	0.1555	0.0748	0.0359	0.0175	0.0778	0.0304
SGL	0.1090	0.0384	0.1838	0.0697	0.2107	0.0859	0.0561	0.0217	0.1719	0.0489
	0.0742	0.0321	0.0966	0.0463	0.1558	0.0717	0.0435	0.0199	0.0868	0.0310
NCL	0.1305	0.0433	0.1595	0.0594	0.2372	0.0880	0.0679	0.0249	0.1740	0.0480
	0.0782	0.0332	0.1305	0.0562	0.1611	0.0762	0.0423	0.0192	0.1192	0.0417
SimGCL	0.1324	0.0443	0.2073	0.0733	0.2189	0.0898	0.0655	0.0230	0.2167	0.0581
	0.0726	0.0338	0.1254	0.0534	0.1459	0.0713	0.0409	0.0182	0.0858	0.0321
RecDCL	0.1142	0.0431	0.1969	0.0689	0.2114	0.0844	0.0614	0.0213	0.1709	0.0487
	0.0765	0.0318	0.1206	0.0572	0.1458	0.0720	0.0415	0.0192	0.1020	0.0371
CoGCL	0.1289	0.0427	0.1986	0.0737	0.1957	0.0820	0.0648	0.0239	0.1977	0.0560
	0.0713	0.0327	0.1159	0.0511	0.1492	0.0715	0.0236	0.0123	0.0794	0.0316
DiffRec	0.1181	0.0422	0.1867	0.0704	0.2013	0.0824	0.0451	0.0189	0.1761	0.0468
	0.0118	0.0051	0.0133	0.0058	0.0139	0.0057	0.0033	0.0020	0.0141	0.0059
DDRM	0.0178	0.0063	0.0277	0.0088	0.0228	0.0075	0.0044	0.0022	0.0182	0.0068
	0.0725	0.0323	0.1252	0.0537	0.1524	0.0659	0.0343	0.0138	0.1044	0.0379
GiffCF	0.1253	0.0449	0.2084	0.0719	0.2084	0.0786	0.0509	0.0181	0.2006	0.0569
	0.0845	**0.0361**	**0.1373**	**0.0633**	**0.2009**	**0.0905**	**0.0462**	**0.0216**	**0.1781**	**0.0509**
RelDiff	**0.1434**	**0.0476**	**0.2198**	**0.0807**	**0.2847**	**0.1071**	**0.0726**	**0.0271**	**0.2629**	**0.0673**
	8.06%	6.80%	5.21%	10.66%	24.71%	18.77%	6.21%	8.54%	49.41%	22.06%
Imprv.	8.31%	6.01%	5.47%	9.50%	20.03%	19.27%	6.92%	8.40%	21.32%	15.83%

5 EXPERIMENT

5.1 EXPERIMENTAL SETTINGS

5.1.1 DATASETS

We evaluate on Amazon-Office, Amazon-Electronics, Amazon-Baby, ML-1M, and Taobao; Table 1 reports their statistics.

5.1.2 EVALUATION METRICS

We report **Recall@*K*** and **NDCG@*K*** for $K \in \{20, 50\}$. The manuscript describes full ranking over candidate items and averaging scores across the test set.

5.1.3 BASELINE MODELS

We compare three groups of baselines: (1) graph models: **NGCF** (Wang et al., 2019), **LightGCN** (He et al., 2020), **UltraGCN** (Mao et al., 2021), and **MGDCF** (Hu et al., 2024); (2) contrastive models: **SGL** (Wu et al., 2021), **NCL** (Lin et al., 2022), **SimGCL** (Yu et al., 2022), **RecDCL** (Zhang et al., 2024), and **CoGCL** (Zheng et al., 2025); (3) diffusion models: **DiffRec** (Wang et al., 2023), **DDRM** (Zhao et al., 2024), and **GiffCF** (Zhu et al., 2024).

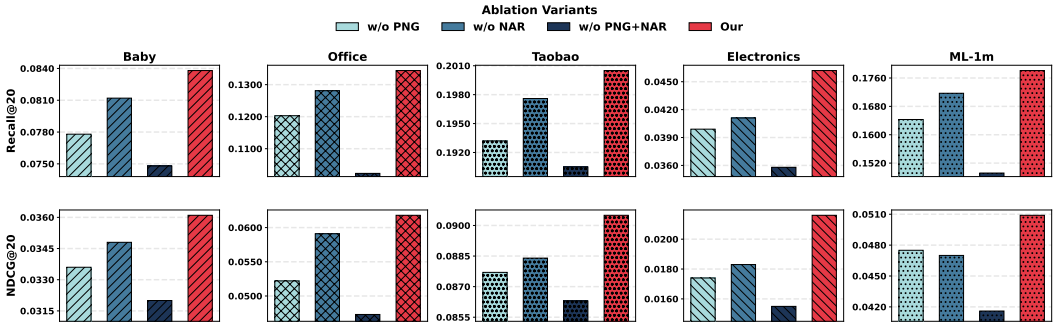


Figure 3: Ablation analysis across five datasets

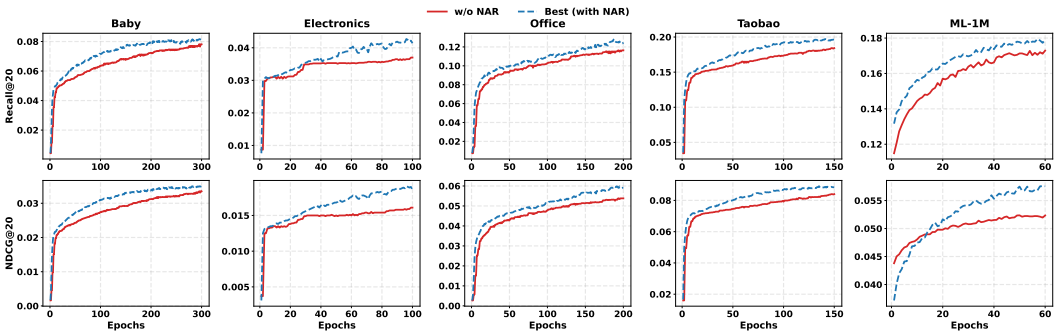


Figure 4: Convergence curves comparison.

5.1.4 Implementation Details

The available manuscript reports embedding dimension 64, Xavier initialization, a learning-rate search from 10^{-6} to 5×10^{-1}, and batch sizes of 1024 for ML-1M and 2000 for Office. It reports searching $\lambda_1, \lambda_2 \in [0, 1]$, $\lambda_3 \in [10^{-3}, 1]$, 0–3 propagation layers, 100–500 diffusion steps, and Adam/AdamW. These are reported search ranges, not verified final settings. Source code, configurations, and run logs were not supplied. Dataset versions, filtering and split rules, random seeds, repeat counts, hardware, early stopping, and validation-based selection are therefore not independently verified. Appendix C lists the remaining details needed to reproduce the results.

5.2 Performance Comparison

Table 2 reports the highest scores for RelDiff across all datasets and metrics. Each cell lists @20 above @50; Imprv. is relative to the strongest baseline. Relative Recall@20 gains range from 5.21% on Office to 49.41% on ML-1M, using the strongest baseline for each metric. These are point estimates; run variability and statistical significance cannot be assessed from the available records.

Baseline rankings vary across metrics. On Baby, SimGCL leads in Recall, while RecDCL and GiffCF lead at NDCG@20 and NDCG@50. On Electronics, NCL leads in Recall and ties MGDCF at NDCG@20, while MGDCF leads at NDCG@50. Thus, the table does not establish a uniform ordering of graph, contrastive, and diffusion methods. RelDiff combines learned perturbations, contrastive injection, and reference refinement; the following ablations examine PNG and NAR within the tested configurations.

5.3 Ablation Analysis

Figure 3 compares RelDiff with three variants. **w/o PNG** replaces learned preference noise with standard Gaussian noise; **w/o NAR** uses the standard reconstruction objective; **w/o PNG+NAR** removes both modules.

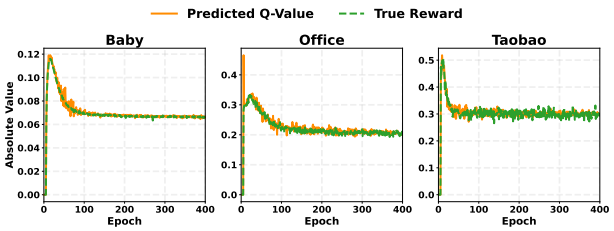


Figure 5: Comparison between predicted Q-value and true reward.

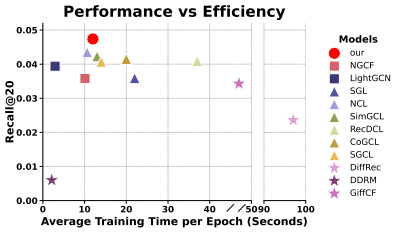


Figure 6: Performance versus efficiency analysis on Amazon-Electronics.

The full model has higher reported scores than the tested variants. These ablations do not isolate contrastive injection, establish critic necessity, or separate user conditioning from perturbation magnitude and covariance; these controls remain missing.

5.4 Convergence Acceleration Analysis

Figure 4 compares epoch-wise **Recall@20** (top row) and **NDCG@20** (bottom row) on five benchmarks. NAR (blue dashed line) generally has higher later scores than **w/o NAR** (red solid line), although ML-1M NDCG@20 initially trails the variant without NAR.

On Taobao and Office, the persistent later gap suggests benefits beyond earlier convergence, consistent with alleviating generation collapse. The curves do not establish an exact epoch threshold, time to fixed quality, or the responsible gradient mechanism.

5.5 Aggregate Critic–Reward Comparison

We compare critic predictions and reconstruction rewards on Amazon-Baby, Amazon-Office, and Taobao. Both curves decrease sharply early in training and then change more slowly, with a similar scale within each dataset; Taobao retains later fluctuations.

Figure 5 shows absolute values of the averages. Their agreement does not establish sample-wise calibration, unbiased estimates, negligible prediction error, or critic necessity: averaging can hide individual errors, and absolute values remove sign information.

5.6 Efficiency Analysis

Figure 6 compares Recall@20 and average training time per epoch on Amazon-Electronics.

RelDiff combines relatively low per-epoch training time with high Recall@20. Diffusion baselines have varied costs: DDRM is near the low-time end, whereas DiffRec and GiffCF have larger costs. Without profiling records, the cause of these cost differences remains unverified.

The plot measures neither online latency nor time to fixed quality. Inference denoising and total epochs cannot explain per-epoch cost. Several Recall values differ from Table 2; absent run records, a controlled speedup or matched-run comparison is not established.

6 Conclusion

In this paper, we propose RelDiff, a novel framework addressing the limitations of user-independent Gaussian noise and generation collapse in diffusion-based recommendation. By employing an Actor-Critic policy to generate preference noise, we transform stochastic diffusion into a process with user-conditioned Gaussian perturbations. Furthermore, our Negative-aware Refinement (NAR) strategy combines reconstruction-based confidence with a historical reference to form attraction and repulsion terms, retaining both well-reconstructed and poorly reconstructed samples during training. Experiments report higher ranking accuracy than the evaluated baselines across five datasets. Future work will explore multi-dimensional objectives, such as diversity and fairness, as well as applications in cross-domain scenarios, including transfer to new item pools and cold-start settings.

AI USE STATEMENT

Generative AI assisted with language polishing, LaTeX formatting, consistency checks, and review of the mathematical arguments. The authors remain responsible for reviewing and verifying the manuscript.

REFERENCES

Jialei Chen, Yuanbo Xu, and Yiheng Jiang. Unlocking the power of diffusion models in sequential recommendation: A simple and effective approach. In *KDD*, pp. 155–166, 2025a.

Mengru Chen, Lianghao Xia, Yong Xu, and Ronghua Luo. Collaborative diffusion models for recommendation. In *SIGIR*, 2025b. doi: 10.1145/3726302.3729929.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pp. 8780–8794, 2021.

Gwangseok Han, Wonbin Kweon, Minsoo Kim, and Hwanjo Yu. Controlling diversity at inference: Guiding diffusion recommender models with targeted category preferences. In *KDD*, pp. 425–435, 2025. doi: 10.1145/3690624.3709211.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pp. 639–648. ACM, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

Yu Hou, Jin-Duk Park, and Won-Yong Shin. Collaborative filtering based on diffusion models: Unveiling the potential of high-order connectivity. In *SIGIR*, pp. 1360–1369. ACM, 2024.

Jun Hu, Bryan Hooi, Shengsheng Qian, Quan Fang, and Changsheng Xu. Mgdcf: Distance learning via markov graph diffusion for neural collaborative filtering. *IEEE Trans. on Knowl. and Data Eng.*, 36(7):3281–3296, 2024. ISSN 1041-4347.

Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. Diffkg: Knowledge graph diffusion model for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 313–321, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

Kaibei Li, Yihao Zhang, Xiaokang Li, Meng Yuan, and Wei Zhou. Mask diffusion-based contrastive learning for knowledge-aware recommendation. *IEEE Trans. Knowl. Data Eng.*, 37(9):5407–5419, 2025a. doi: 10.1109/TKDE.2025.3582767.

Xiaodong Li, Hengzhu Tang, Jiawei Sheng, Xinghua Zhang, Li Gao, Suqi Cheng, Dawei Yin, and Tingwen Liu. Exploring preference-guided diffusion model for cross-domain recommendation. In *KDD*, pp. 719–728, 2025b.

Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42:1 – 28, 2023.

Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, pp. 2320–2329. ACM, 2022.

Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang. Plug-in diffusion model for sequential recommendation. In *AAAI*, pp. 8886–8894. AAAI Press, 2024a. doi: 10.1609/aaai.v38i8.28736.

Haokai Ma, Ruobing Xie, Lei Meng, Yimeng Yang, Xingwu Sun, and Zhanhui Kang. Seedrec: Sememe-based diffusion for sequential recommendation. In *IJCAI*, pp. 2270–2278, 2024b. doi: 10.24963/ijcai.2024/251.

Haokai Ma, Yimeng Yang, Lei Meng, Ruobing Xie, and Xiangxu Meng. Multimodal conditioned diffusion model for recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, 2024c. doi: 10.1145/3589335.3651956.

Wenze Ma, Chenyu Sun, Yanmin Zhu, Zhaobo Wang, Xuhao Zhao, Mengyuan Jing, Jiadi Yu, and Feilong Tang. Generating difficulty-aware negative samples via conditional diffusion for multimodal recommendation. In *SIGIR*, 2025.

Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. Ultragcn: Ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1253–1262, New York, NY, USA, 2021. Association for Computing Machinery.

Wenyu Mao, Shuchang Liu, Haoyang Liu, Haozhe Liu, Xiang Li, and Lantao Hu. Distinguished quantized guidance for diffusion-based sequence recommendation. In *WWW*, pp. 425–435, 2025. doi: 10.1145/3696410.3714955.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1928–1937. JMLR.org, 2016.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, pp. 10684–10695. IEEE, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1889–1897. JMLR.org, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021b.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3: 9–44, 1988.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, pp. 1057–1063. The MIT Press, 1999.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pp. 2094–2100. AAAI Press, 2016.

Joojo Walker, Ting Zhong, Fengli Zhang, Qiang Gao, and Fan Zhou. Recommendation via collaborative diffusion generative model. In *Knowledge Science, Engineering and Management*, 2022.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *SIGIR*, pp. 832–841. ACM, 2023.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural Graph Collaborative Filtering. In *SIGIR*, pp. 165–174. ACM, 2019. doi: 10.1145/3331184.3331267.

Zhaobo Wang, Yanmin Zhu, Chunyang Wang, Xuhao Zhao, Bo Li, Jiadi Yu, and Feilong Tang. Graph diffusion-based representation learning for sequential recommendation. *IEEE Trans. Knowl. Data Eng.*, 36(12):8395–8407, 2024. doi: 10.1109/TKDE.2024.3477621.

Christopher J. C. H. Watkins and Peter Dayan. Technical note q-learning. *Mach. Learn.*, 8:279–292, 1992.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *SIGIR*, pp. 726–735. ACM, 2021.

Yiyan Xu, Wenjie Wang, Fuli Feng, Yunshan Ma, Jizhi Zhang, and Xiangnan He. Diffusion models for generative outfit recommendation. In *SIGIR*, pp. 1350–1359, 2024.

Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *NeurIPS*, 36, 2023.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary?: Simple graph contrastive learning for recommendation. In *SIGIR*. ACM, 2022.

Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. Recdcl: Dual contrastive learning for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pp. 3655–3666. ACM, 2024.

Jujia Zhao, Wenjie Wang, Yiyan Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. Denoising diffusion recommender model. In *SIGIR*, pp. 1370–1379. ACM, 2024.

Bowen Zheng, Junjie Zhang, Hongyu Lu, Yu Chen, Ming Chen, Wayne Xin Zhao, and Ji-Rong Wen. Enhancing graph contrastive learning with reliable and informative augmentation for recommendation. In *SIGKDD*, pp. 2101–2112. ACM, 2025.

Yunqin Zhu, Chao Wang, Qi Zhang, and Hui Xiong. Graph signal diffusion model for collaborative filtering. In *SIGIR*, pp. 1380–1390. ACM, 2024.

A Conditional Analysis and Limitations of NAR

NAR is motivated by alleviating generation collapse, the convergence bottleneck identified in Section 2. The original argument treated reconstruction confidence as fixed during differentiation and replaced the historical reference by the current conditional mean. Neither condition follows from the stated method. We therefore withdraw the claims of guaranteed convergence acceleration, strictly moving closer to a positive posterior mean, non-vanishing gradients, and drift-free updates. The identities below describe a restricted surrogate; they are not claims that the implementation satisfies these conditions.

A.1 A confidence-weighted decomposition

At a fixed conditioning state, let Y denote the target embedding with distribution p, and let $r(Y) \in [0, 1]$ be a frozen weighting function. Define $c = \mathbb{E}_p[r(Y)]$. For $0 < c < 1$, set

$$p^+(Y) = \frac{r(Y)p(Y)}{c}, \qquad p^-(Y) = \frac{(1 - r(Y))p(Y)}{1 - c}, \qquad p = cp^+ + (1 - c)p^-. \tag{15}$$

This follows by substitution; it is a reweighting identity, not evidence for distinct true and false preference populations. The notation $+$ and $-$ refers only to weights. With $m = \mathbb{E}[Y]$ and $m^\pm = \mathbb{E}_{p^\pm}[Y]$, we have $m = cm^+ + (1 - c)m^-$. The score in Eq. 8 also depends on predictions and batch statistics, so freezing it here is an analytical condition, not a verified implementation detail. The symbol c avoids confusion with the diffusion schedule and actor entropy coefficient.

A.2 Fixed-reference quadratic surrogate

Let f be the prediction, a a fixed reference output, y a fixed target, and $\Delta = f - a$. For $\beta > 0$, the pointwise form of Eq. 11 is

$$L(f) = \frac{1}{\beta}\left[r\|a + \beta\Delta - y\|^2 + (1 - r)\|a - \beta\Delta - y\|^2\right]. \tag{16}$$

If r, a, y are held fixed, expansion gives

$$L(f) = \frac{\|a - y\|^2}{\beta} + \beta\|\Delta\|^2 + 2(2r - 1)\langle a - y, \Delta\rangle, \tag{17}$$

and therefore

$$\nabla_f L = 2\beta(f - a) + 2(2r - 1)(a - y). \tag{18}$$

For a fixed target distribution and frozen weighting function, the unconstrained minimizer in prediction space is

$$\begin{aligned} f^* &= a - \frac{1}{\beta}\mathbb{E}[(2r(Y) - 1)(a - Y)] \\ &= a + \frac{2c(m^+ - a) + (a - m)}{\beta}. \end{aligned} \tag{19}$$

This follows by taking expectations in Eq. 18 and setting the result to zero. Only under the additional equality $a = m$ does it reduce to the expression used in the original analysis:

$$f^* = m + \frac{2c}{\beta}(m^+ - m). \tag{20}$$

A historical best checkpoint need not satisfy $a = m$. Even in this special case, $2c/\beta$ can exceed one and overshoot m^+. The identity does not prove that an optimizer step improves ranking, that neural-network training reaches this prediction-space optimum, or that NAR accelerates convergence.

A.3 Gradient paths and hard positive interactions

If $r = r(f)$ is differentiated while a and y remain fixed, the pointwise gradient additionally contains

$$\frac{\|a + \beta\Delta - y\|^2 - \|a - \beta\Delta - y\|^2}{\beta}\nabla_f r = 4\langle a - y, \Delta\rangle\nabla_f r. \tag{21}$$

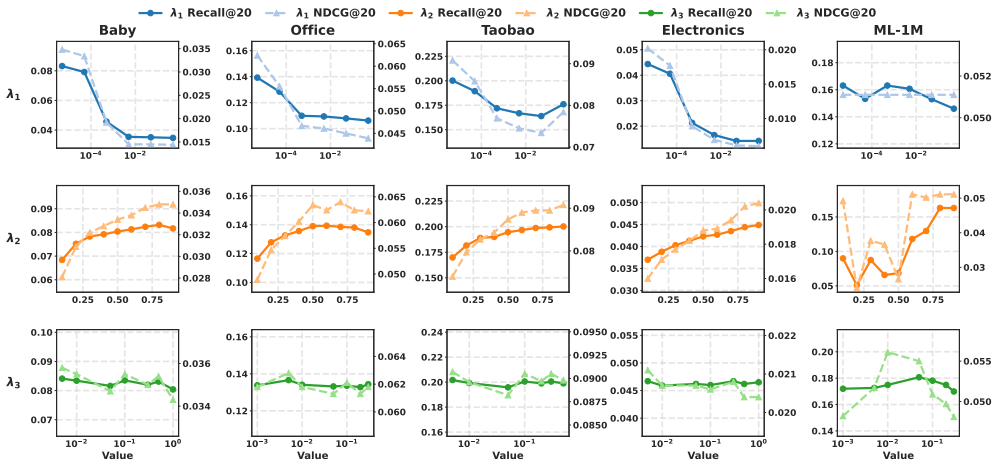


Figure 7: Hyperparameter sensitivity analysis of the three key balancing coefficients: λ_1 (contrastive injection weight), λ_2 (ranking alignment weight), and λ_3 (RL policy optimization weight). We report the performance trends of Recall@20 and NDCG@20 across five datasets.

Batch statistics introduce cross-sample dependencies, and trainable reference or target embeddings introduce further terms. A prediction-space partial derivative alone is therefore insufficient to describe the full parameter gradient. Whether these paths are detached, the value of β, and the reference update rule remain unverified.

Equation 18 also allows zero gradients, for example when $f = a = y$, or when $f = a$ and $r = 1/2$. Thus it provides no non-vanishing-gradient guarantee. With scalar values $a = f = 0$, $y = 1$, $\beta = 1$, and fixed $r = 0$, the gradient is $+2$: gradient descent moves f below zero, away from the observed positive target. This counterexample concerns the limiting low-confidence surrogate and shows why repulsion is not automatically corrective. High reconstruction error alone cannot establish that the interaction is wrong. The empirical NAR results remain as reported, but these theoretical and implementation questions remain unresolved.

B Additional Empirical Analyses

B.1 Hyperparameter Sensitivity Analysis

To investigate the robustness of our model under different hyperparameter settings, we conducted a sensitivity analysis on three key balancing coefficients in the total objective function: λ_1 (contrastive injection weight), λ_2 (ranking alignment weight), and λ_3 (RL policy optimization weight). We tracked the performance trends of Recall@20 and NDCG@20 across five datasets via grid search, as illustrated in Figure 7.

For Baby, Office, Taobao, and Electronics, the highest plotted scores in the λ_1 panels occur at the smallest tested value. The λ_2 and λ_3 panels show dataset- and metric-dependent trends. Axis ticks do not specify the complete tested grids, which remain undocumented. These plots describe the reported settings only; they do not establish a universal optimum or diagnose the mechanism of performance changes. The validation metric, final selected values, and whether all choices used validation data rather than test results remain to be confirmed.

B.2 Signal-to-Noise Ratio Analysis

Figure 8 shows Amazon-Baby, Taobao, and ML-1M, the three datasets present in the supplied figure. For the forward state $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\eta$, define $S_t = \sqrt{\bar{\alpha}_t}x_0$ and $N_t = \sqrt{1-\bar{\alpha}_t}\eta$, where η

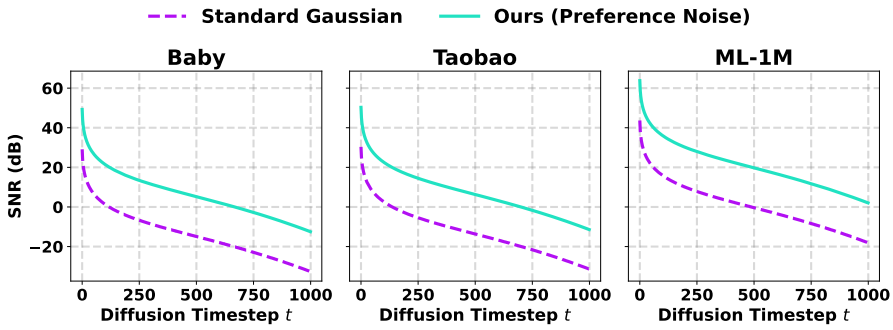


Figure 8: SNR comparison on Amazon-Baby, Taobao, and ML-1M datasets

denotes either standard Gaussian noise or the learned perturbation. The reported power ratio is

$$\mathrm{SNR}(t) = 10 \log_{10} \frac{\|S_t\|_2^2}{\|N_t\|_2^2}. \tag{22}$$

The learned-perturbation curve lies above the Gaussian curve in these panels. This is a signal-to-perturbation power comparison: reducing perturbation magnitude alone increases SNR. The statistic does not measure whether the perturbation direction encodes user preference, nor does it independently establish semantic preservation. No norm-matched, covariance-matched, or shuffled-user control is available. The figure spans approximately 0–1000 diffusion steps, whereas the implementation paragraph reports a 100–500 search range; the plotting schedule cannot be reconciled with the reported training settings without the original configuration.

C Reproducibility Details Requiring Confirmation

The supplied materials contain this manuscript, a bibliography, and figure PDFs, but no source code, configurations, or training logs. Consequently, the following details remain unspecified or unverified rather than being filled with conventional defaults:

- Dataset releases and download sources, filtering/preprocessing, split procedure, candidate-item exclusions, and validation metric.
- Random seeds, number of runs, uncertainty estimates, hardware/software, stopping rule, and how final checkpoints and hyperparameters were selected.
- Dataset-specific optimizer, learning rate, batch size, graph depth/aggregation, diffusion schedule and sampling steps, loss weights, reward scale, entropy coefficient, and contrastive temperature/similarity/masking.
- NAR's β, exact batch statistics and denominator stabilization, gradient handling for confidence/statistics/reference/target embeddings, and reference checkpoint update and selection rules.

The earlier claim of one negative per positive is not inferable from the in-batch denominator in Eq. 6, and its sampling interpretation needs confirmation. The reported score differences are descriptive; no statistical significance test is documented. We do not claim that source code, repeated-run statistics, or latency measurements have been provided. The efficiency plot includes SGCL, which has no active row in the main comparison table; its experimental configuration is likewise unavailable.

D Related Work

D.1 Diffusion Models

Diffusion models have become a dominant likelihood-based generative framework by learning to reverse a gradual noising process. DDPM formulates generation as iterative denoising of Gaussian-corrupted data via a learned reverse Markov chain (Ho et al., 2020). Score-based diffusion further

unifies diffusion and score matching by modeling the score function under stochastic differential equations (SDEs), enabling principled sampling and noise schedules (Song et al., 2021b). To reduce sampling cost, DDIM introduces a non-Markovian implicit diffusion process that allows fast deterministic or stochastic generation with fewer steps (Song et al., 2021a). Training and generation quality were subsequently improved through refined parameterizations and variance schedules in improved DDPM (Nichol & Dhariwal, 2021), and large-scale diffusion models demonstrated strong synthesis fidelity, surpassing GANs under standard evaluations (Dhariwal & Nichol, 2021). For high-resolution generation, latent diffusion performs the diffusion process in a compressed latent space to improve efficiency while preserving perceptual quality (Rombach et al., 2022). More recently, EDM clarifies key design choices (e.g., noise parameterization and sampling) and substantially improves sample quality and efficiency across diffusion variants (Karras et al., 2022).

D.2 Reinforcement Learning

Reinforcement learning (RL) includes sequential decision-making objectives that maximize long-term return. RelDiff uses the restricted immediate-reward setting described in the main text; the methods below provide broader background, not an equivalence to its training objective. Early foundations include temporal-difference (TD) learning, which learns predictions from incomplete sequences via bootstrapping (Sutton, 1988), and Q-learning, which enables off-policy learning of optimal action values without a model of the environment (Watkins & Dayan, 1992). Policy gradient methods provide a complementary line of work by directly optimizing a parameterized policy using gradient estimates, supporting stochastic policies and function approximation (Sutton et al., 1999). The emergence of deep RL combined RL objectives with deep neural networks; DQN achieved human-level Atari performance by stabilizing value learning with experience replay and target networks (Mnih et al., 2015). Subsequent advances improved stability and sample efficiency, including TRPO for constrained policy optimization with monotonic improvement guarantees (Schulman et al., 2015) and Double Q-learning to reduce overestimation bias in value-based methods (van Hasselt et al., 2016). In parallel, A3C demonstrated scalable on-policy actor-critic learning with asynchronous updates, substantially broadening the applicability of deep RL (Mnih et al., 2016).