GENERATIVE RECOMMENDATIONS VIA LATENT DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation models commonly represent items using sequences of semantic tokens. However, these representations are typically learned independently of the recommendation task, potentially discarding information relevant to next-item predictions. We introduce a latent diffusion framework that jointly learns item representations from both content and interaction history and a masked denoising model for next-item recommendation. Our variational formulation supports discrete, continuous, and hybrid latent variables while encouraging discriminative representations and reducing collisions. Using a straight-through estimator, we allow the denoiser to update the item encoder, while our catalog-retrieval objective reduces collisions. We additionally propose beam-free retrieval based on denoising scores. In our experiments on three Amazon Reviews datasets, our method outperforms comparable two-stage methods. Further, our ablations demonstrate that joint training improves recommendation performance and reduces collisions.

1 INTRODUCTION

Recommendation systems are the keystone of many online platforms, helping users navigate large item catalogs by suggesting items that match their preferences. In sequential recommendation, these suggestions are based on a user's interaction history and are modeled as a next-item prediction task, which itself is commonly formulated as a classification or ranking task over the catalog (Hidasi et al., 2015; Kang & McAuley, 2018; Sun et al., 2019). While effective, this setup requires a separate output class for every item, limiting its scalability and its ability to share information between semantically related items (Hebert et al., 2025). Recently, generative recommenders have emerged to address these limitations by representing each item as a sequence of discrete semantic tokens, referred to as a semantic ID, and generating those autoregressively (Rajput et al., 2023).

In these two-stage semantic-ID approaches, the tokenizer is typically trained in a separate pre-training stage, e.g., on item metadata, and remains frozen while the recommendation model is trained. Consequently, the recommendation loss typically updates only the prediction model, not the tokenizer (Rajput et al., 2023).

Masked discrete diffusion models offer various advantages over autoregressive semantic-ID generation. Instead of committing to a fixed left-to-right factorization, they learn to recover clean tokens from randomly masked sequences (Shi et al., 2025). This enables bidirectional information flow among all tokens within the semantic ID and allows their generation order to be dynamically selected. LLaDA-Rec recently demonstrated the effectiveness of masked diffusion for semantic-ID recommendation (Shi et al., 2025). Nevertheless, its tokenizer and recommendation model remain separate: semantic IDs are constructed before denoiser training and are not influenced by the recommendation objective.

In this work, we introduce a latent diffusion framework that jointly learns item representations and recommendations via a discrete diffusion model. An encoder maps each catalog item to a latent representation, while a conditional diffusion model learns to recover the latent target representation from a noised representation given the interaction history. We derive the model from a variational item-classification perspective that naturally supports both discrete and continuous latent variables. By using the straight-through estimator (Bengio et al., 2013), the diffusion gradients can shape the

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.

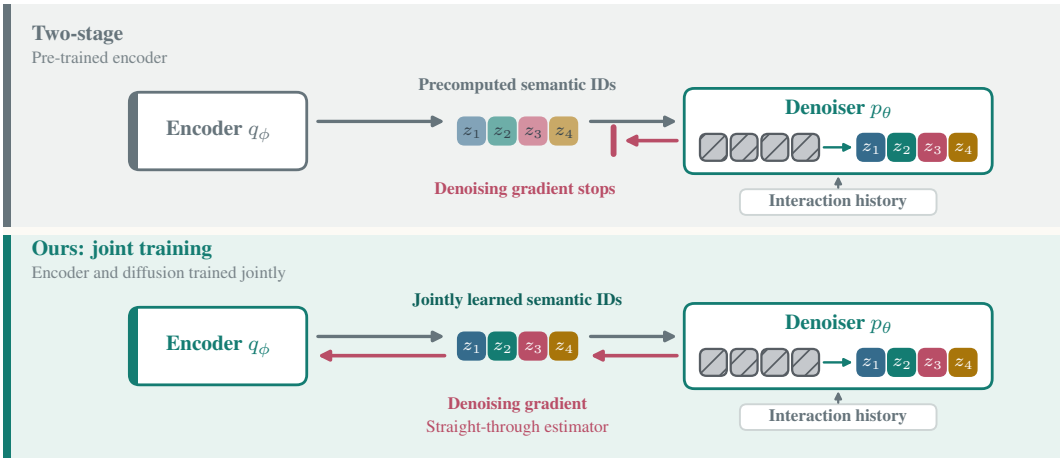

Figure 1: Our model encodes item content into semantic IDs and trains a masked denoiser conditioned on the interaction history. Unlike typical two-stage training, our joint model trains the encoder and denoiser jointly via a straight-through estimator, enabling gradient flow through discrete assignments.

item encoder. We reduce collisions through a retrieval objective that enables each latent representation to retrieve its original item.

Furthermore, we propose beam-free catalog retrieval. An all-mask prediction first proposes catalog-supported candidates, which are then ranked using expected denoising scores.

We evaluate our approach on the Industrial and Scientific, Musical Instruments, and Video Games categories of Amazon Reviews 2023 (Hou et al., 2026). Our ablations highlight two central findings. First, joint representation learning improves retrieval performance and reduces catalog collisions compared to a fully frozen tokenizer. Second, discrete representations outperform continuous ones.

Our key contributions can be summarized as:

- **Unified latent diffusion for recommendation.** We introduce a generative recommendation framework that models items through latent representations naturally supporting discrete, continuous, and hybrid latent domains within a unified variational formulation.
- **Joint semantic representation learning.** By jointly optimizing the tokenizer and diffusion model, we allow recommendation gradients to shape the semantic IDs. Our classification and reconstruction objectives further help prevent representational collapse.
- **Beam-free catalog retrieval.** We propose an inference procedure that ranks candidates based on expected denoising scores, thereby avoiding sequential beam expansion.
- **Comprehensive empirical evaluation.** Across three Amazon Reviews datasets, our approach achieves state-of-the-art recommendation performance. Our ablations further demonstrate the benefits of discrete representations and joint training of the tokenizer and denoiser.

2 Background

2.1 Semantic IDs

As large item vocabularies can limit scalability in recommendation systems, these are often represented using semantic IDs. Rather than representing an item using a fixed ID y, it is represented by a sequence of discrete variables $\mathbf{z} = (z_1, \ldots, z_H) \in \mathcal{V}^H$, where z_h is the token at position h, enabling meaningful discrete representations. Here, $\mathcal{V} = \{1, \ldots, K\}$ denotes the vocabulary of discrete code indices, and $K = |\mathcal{V}|$ is the vocabulary size. Often, semantic IDs are obtained via residual or hierarchical quantization (Lee et al., 2022), making them well-suited to autoregressive

recommendation models (Rajput et al., 2023). The quantization aims to obtain unique representations for each item, enabling the mapping of semantic IDs back to the items. However, as these approaches construct semantic IDs before training the recommendation model, they cannot refine these initial representations (Tay et al., 2022; Rajput et al., 2023; Shi et al., 2025).

Let f_ϕ denote the item encoder, $\mathbf{e}_y = f_\phi(y) \in \mathbb{R}^d$ the embedding of item y, and $\mathcal{E}_h = \{\mathbf{c}_{h,v} \in \mathbb{R}^d\}_{v=1}^K$ the codebook containing K code vectors at depth $h \in \{1, \ldots, H\}$. Starting with $\mathbf{r}_y^{(0)} = \mathbf{e}_y$, residual quantization constructs the semantic ID following a coarse-to-fine approach:

$$z_h(y) = \arg\min_{v \in \mathcal{V}} \left\| \mathbf{r}_y^{(h-1)} - \mathbf{c}_{h,v} \right\|_2^2, \qquad \mathbf{r}_y^{(h)} = \mathbf{r}_y^{(h-1)} - \mathbf{c}_{h,z_h(y)}. \tag{1}$$

Although this factorization provides up to $|\mathcal{V}|^H$ distinct tuples, it does not guarantee that the catalog encoder provides unique representations, which is necessary for item retrieval.

2.2 Discrete Diffusion

Unlike standard Gaussian diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020), discrete diffusion operates on discrete states and corrupts categorical variables (Austin et al., 2021; Sahoo et al., 2024). The reverse process recovers the clean sequence and is parametrized using a neural network. Conditioned on an interaction history $\mathbf{x}$, a discrete diffusion model is a latent-variable model of the form

$$p_\theta(\mathbf{z}_0 \mid \mathbf{x}) = \sum_{\mathbf{z}_{1:T}} p_\theta(\mathbf{z}_{0:T} \mid \mathbf{x}), \tag{2}$$

where $\mathbf{z}_0 \sim q(\mathbf{z})$ denotes the clean semantic ID and $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ are latent categorical states. The latent variables are generated by a Markovian noising process,

$$q(\mathbf{z}_{1:T} \mid \mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t \mid \mathbf{z}_{t-1}). \tag{3}$$

In absorbing-state diffusion, tokens transition to a dedicated mask token $m \notin \mathcal{V}$ determined by β_t, the probability of replacing an unmasked token at step t (Austin et al., 2021; Sahoo et al., 2024). Here, δ_a denotes the Dirac delta centered at state a. The transition of a token is given by

$$q(z_t \mid z_{t-1}) = \mathrm{Cat}\big(\alpha_t\, \delta_{z_{t-1}} + (1 - \alpha_t)\, \delta_m\big), \qquad \alpha_t = 1 - \beta_t. \tag{4}$$

The fixed forward process allows direct sampling at an arbitrary timestep given z_0,

$$q(z_t \mid z_0) = \mathrm{Cat}(\bar{\alpha}_t\, \delta_{z_0} + (1 - \bar{\alpha}_t)\, \delta_m), \qquad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{5}$$

Since the mask state is absorbing, this transition results in $q(\mathbf{z}_T) \approx \delta_{(m,\ldots,m)}$.

The reverse process is derived via Bayes' rule and modeled by the factorization

$$p_\theta(\mathbf{z}_{0:T} \mid \mathbf{x}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}, t), \tag{6}$$

where a denoising network parameterizes the categorical transitions. The model is trained to approximate the true reverse process by maximizing an evidence lower bound on $\log p_\theta(\mathbf{z}_0 \mid \mathbf{x})$ (Austin et al., 2021). In the $\mathbf{z}_0$ parameterization, the network predicts the semantic ID, i.e., the clean tokens, through a categorical distribution at each position and is trained using the weighted cross-entropy loss on the masked tokens (Sahoo et al., 2024):

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x},\mathbf{z}_0,t,\mathbf{z}_t} \left[-w(t) \sum_{h=1}^{H} \mathbf{1}[z_t^{(h)} = m] \, \log p_\theta\Big(z_0^{(h)} \mid \mathbf{z}_t, \mathbf{x}, t\Big) \right]. \tag{7}$$

Generation starts from the fully masked state and iteratively applies the learned reverse transitions. In contrast to autoregressive models, the denoising model does not require a fixed left-to-right factorization and can recover multiple semantic-ID positions in parallel.

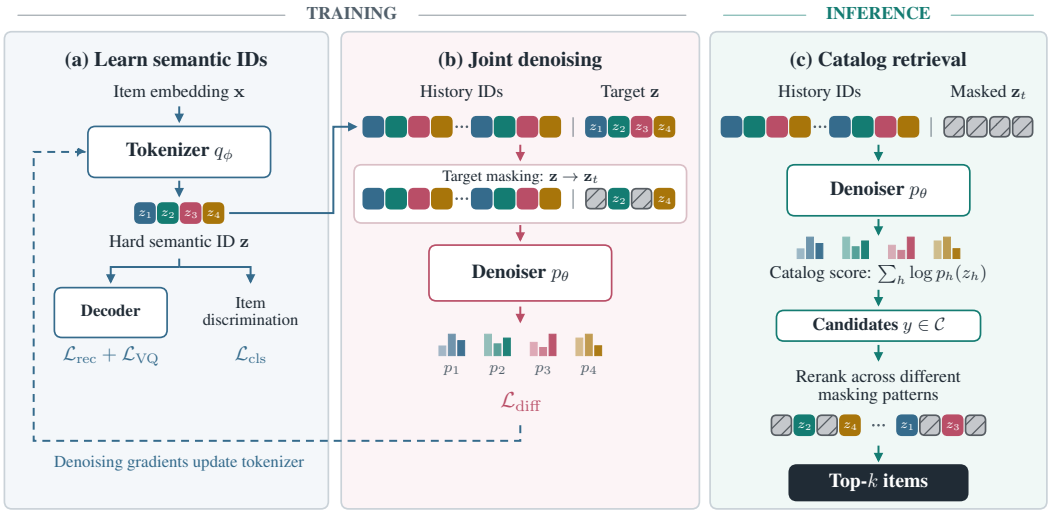

Figure 2: Overview of the discrete recommendation pipeline. (a) An encoder learns semantic ID tokenization jointly with the denoiser. (b) Token masking trains the denoiser model. After joint training, we finetune the model. During inference, (c) an initial target prediction proposes catalog-supported candidates, which are ranked using expected denoising scores to extract the top-k recommendations.

3 Methodology

3.1 Problem Statement

Let $\mathcal{C}$ denote the item catalog. Given an interaction sequence $(y_1, \ldots, y_L) \in \mathcal{C}^L$, we use $\mathbf{x} = (y_1, \ldots, y_{L-1}) \in \mathcal{C}^{L-1}$ as the context to predict the target item $\mathbf{y} := y_L$. Rather than modeling the item space directly, we map each item to a semantic ID $\mathbf{z} = (z^{(1)}, \ldots, z^{(H)}) \in \mathcal{V}^H$ using an encoder $q_\phi(\mathbf{z} \mid \mathbf{y})$, where $\mathcal{V}$ is the codebook vocabulary. In our discrete model, we use $H = 4$ codebooks with $K = 256$ entries each.

This representation replaces one catalog-wide prediction with H smaller token predictions. To recover catalog items, however, the semantic IDs must remain discriminative, i.e., distinct items have to be assigned different tuples $\mathbf{z}$. A *collision* occurs when distinct items are assigned the same $\mathbf{z}$.

Following Section 2.2, we introduce a noisy semantic ID at diffusion time t by $\mathbf{z}_t \in (\mathcal{V} \cup \{\text{MASK}\})^H$, with its respective denoiser $p_\theta(\mathbf{z} \mid \mathbf{z}_t, \mathbf{x}, t)$, which predicts the clean semantic ID $\mathbf{z}$ conditioned on the noisy state $\mathbf{z}_t$ and interaction history $\mathbf{x}$.

Figure 2 summarizes our training and retrieval pipeline. Here, we assume that the semantic ID is sufficient for identifying the item, thus $\mathbf{y}$ and $\mathbf{x}$ are conditionally independent given $\mathbf{z}$. This leads to the factorization $p_\theta(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) = p(\mathbf{y} \mid \mathbf{z})\, p_\theta(\mathbf{z} \mid \mathbf{x})$. Note that this assumption holds when the semantic IDs are unique. When collisions occur, however, $p(\mathbf{y} \mid \mathbf{z})$ represents the likelihood of items sharing the same discrete representation. Further, we introduce the variational distribution $q_\phi(\mathbf{z} \mid \mathbf{y})$, i.e., our encoder, to obtain the following evidence lower bound (ELBO) (Kingma & Welling, 2013):

$$\log p(\mathbf{y} \mid \mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}\left[\log \frac{p_\theta(\mathbf{y}, \mathbf{z} \mid \mathbf{x})}{q_\phi(\mathbf{z} \mid \mathbf{y})}\right] \tag{8}$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\log p(\mathbf{y} \mid \mathbf{z})]}_{\text{classification term}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\log p_\theta(\mathbf{z} \mid \mathbf{x})]}_{\text{diffusion term}} - \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\log q_\phi(\mathbf{z} \mid \mathbf{y})]}_{\text{negative entropy}}. \tag{9}$$

3.2 Joint Representation and Diffusion Training

In the following, we describe the model components and training objectives in detail and provide an intuition of their purpose.

Tokenizer and denoising model In Eq. (9), we have introduced two distributions parametrized by neural networks. First, we have the encoder $q_\phi(\mathbf{z} \mid \mathbf{y})$, which encodes the target item given metadata embeddings and yields a semantic ID. Further, we have a conditional denoising model $p_\theta(\mathbf{z} \mid \mathbf{z}_t, \mathbf{x}, t)$ that aims to reconstruct the same semantic ID from the history and a corrupted state inducing $p_\theta(\mathbf{z} \mid \mathbf{x})$. The objectives aim to align these models while preserving the ability to distinguish and retrieve items.

Classification loss. The negative classification term in Eq. (9) serves as the item-retrieval objective. Under a uniform prior over catalog items, applying Bayes' rule yields the following loss:

$$\mathcal{L}_{\text{cls}} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}\left[\log \frac{q_\phi(\mathbf{z} \mid \mathbf{y})}{\sum_{\mathbf{y}' \in \mathcal{C}} q_\phi(\mathbf{z} \mid \mathbf{y}')}\right]. \tag{10}$$

Minimizing $\mathcal{L}_{\text{cls}}$ makes the semantic IDs predictive. More specifically, the loss maximizes $q_\phi(\mathbf{z} \mid \mathbf{y})$ relative to $q_\phi(\mathbf{z} \mid \mathbf{y}')$ for all other items $\mathbf{y}' \neq \mathbf{y}$. Thus, the loss enables the retrieval of the originating catalog item and therefore reduces collisions.

Diffusion loss. We model the conditional distribution $p_\theta(\mathbf{z} \mid \mathbf{x})$ using a discrete diffusion model. More specifically, we randomly mask tokens and train the denoiser network to recover the clean tokens. We first draw $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{y})$ from the encoder, a diffusion time t, and $\mathbf{z}_t \sim q(\mathbf{z}_t \mid \mathbf{z})$ from the diffusion forward process. The resulting objective is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{z},t,\mathbf{z}_t}[-w(t) \log p_\theta(\mathbf{z} \mid \mathbf{z}_t, \mathbf{x}, t)]. \tag{11}$$

We apply the noising process to the target tokens and introduce an additional noising pass for the history as done by Shi et al. (2025). We denote this additional diffusion loss by $\mathcal{L}_{\text{hist}}$. The diffusion model enables us to perform inference by sampling semantic IDs given the interaction history.

Joint objective. To enable joint training of the tokenizer and diffusion model, we use hard semantic IDs in the forward pass and soft assignments via the straight-through estimator (Bengio et al., 2013) in the backward pass. Thus, the variational encoding distribution is a point mass, and the entropy term in equation 9 is zero and can be omitted. For the classification term, however, we use the soft assignment probabilities as a differentiable relaxation.

To stabilize tokenizer training, we further include a reconstruction and vector-quantization loss, denoted as $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{vq}}$, respectively (see Appendix A.1 for more details). The overall objective is:

$$\mathcal{L} = \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} + \lambda_{\text{hist}}\mathcal{L}_{\text{hist}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{vq}}\mathcal{L}_{\text{vq}}. \tag{12}$$

This path lets recommendation gradients, i.e., the diffusion gradients, shape the semantic representation. Batch estimators, masking rules, tokenizer losses, optimization, and collision repair are explained in Appendix A.1.

3.3 Catalog retrieval

At inference time, our goal is to predict and rank catalog items given the interaction history $\mathbf{x}$. As the diffusion model generates semantic IDs, we convert those to item scores that quantify how likely an item is to produce the semantic ID. First, by marginalizing over the semantic ID and applying Bayes' rule, we obtain

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{y} \mid \mathbf{z}) p_\theta(\mathbf{z} \mid \mathbf{x}) \propto p(\mathbf{y}) \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[q_\phi(\mathbf{z} \mid \mathbf{y})]. \tag{13}$$

As we are only interested in the ordering of items, normalization constants can be omitted.

Beam-free catalog proposal. Rather than generating items autoregressively with beam search, we directly retrieve items from the catalog based on expected denoising scores. Let $\mathbf{z}(y)$ be the semantic ID of item y. First, we make a single prediction from the fully masked target, producing a categorical distribution per token. We use these log-likelihoods to score valid semantic IDs:

$$s_{\text{prop}}(y \mid \mathbf{x}) = \sum_{h=1}^{H} \log p_{\theta,h}\left(z^{(h)}(y) \mid \text{MASK}^H, \mathbf{x}\right). \tag{14}$$

We then select the top-scoring items from the training catalog. A second, denoising evaluation measures how well each candidate is recovered from different partially masked targets:

$$s_{\text{den}}(y \mid \mathbf{x}) = \sum_{\varnothing \neq M \subseteq \{1,\dots,H\}} w_M \sum_{h \in M} \log p_{\theta,h}\left(z^{(h)}(y) \mid \mathbf{z}_M(y), \mathbf{x}\right), \tag{15}$$

where $\mathbf{z}_M(y)$ denotes the semantic ID with coordinates in M masked, and w_M weights each masking pattern. This score measures how well the model recovers the candidate and serves as a proxy for the item's log-likelihood under the diffusion model. Finally, a reranker fitted on the validation set combines the proposal and denoising scores with training-derived transition and popularity features. Appendix A.1 describes additional details.

3.4 Continuous and Hybrid Representations

The discrete semantic ID imposes a bottleneck that may discard fine-grained item information required for reconstruction. Therefore, we consider an additional continuous latent $\mathbf{z}^c$ alongside the discrete ID $\mathbf{z}^d$. The hybrid inference model becomes

$$q_\phi(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{y}) = q_{\phi_d}(\mathbf{z}^d \mid \mathbf{y}) q_{\phi_c}(\mathbf{z}^c \mid \mathbf{y}). \tag{16}$$

Here, q_d remains a masking diffusion process and $q_c(\mathbf{z}_t^c \mid \mathbf{z}^c) = \mathcal{N}(\alpha_t \mathbf{z}^c, \sigma_t^2 \mathbf{I})$ applies Gaussian diffusion. We model their joint generative distribution $p_\theta(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{x})$ with a shared denoising model. Note that the reverse process is coupled through the network, allowing the discrete and continuous variables to inform one another during generation.

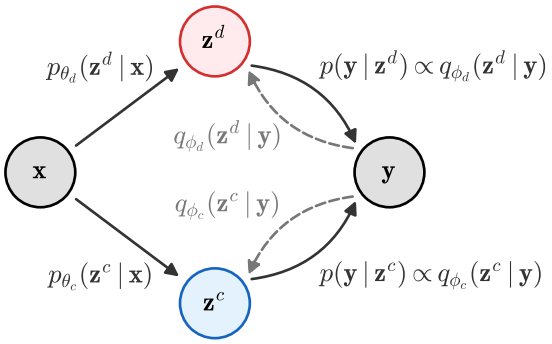

Figure 3: Graphical model of the discrete and continuous latent-variable formulations. The encoders map the target item $\mathbf{y}$ to $\mathbf{z}^d$ or $\mathbf{z}^c$, while the conditional denoisers predict each latent from the interaction history $\mathbf{x}$; both paths define a likelihood over catalog items.

At inference, both variables are generated jointly. Catalog candidates are retrieved solely via discrete semantic IDs, while the continuous variable contributes to the likelihood scores used to rank those candidates. Removing $\mathbf{z}^c$ recovers the discrete model, whereas removing $\mathbf{z}^d$ yields the continuous latent diffusion ablation. We provide the full variational derivation in Appendix A.2.

4 Experiments

4.1 Experimental Setup

We evaluate our model on the Amazon Reviews 2023 dataset (Hou et al., 2026). More specifically, we use the categories *Industrial and Scientific*, *Musical Instruments*, and *Video Games*. Each sample consists of the interaction history and the next-item target. We truncate the history to the 20 most recent items. We concatenate the available metadata fields and encode them once using the `sentence-t5-base` transformer (Vaswani et al., 2017; Ni et al., 2022), retaining only the resulting 768-dimensional embeddings. All our Transformer models use a hidden width of 256, four-coordinate semantic IDs with a codebook size of 256, and dropout 0.1. Dataset-specific architecture, optimization settings, and evaluation details are given in Appendix A.1.

4.2 Recommendation Performance

We compare our results with the baselines in Table 1. Both variants of our model outperform LLaDA-Rec across all datasets and evaluation metrics. Further, beam-free inference achieves the best NDCG on all three datasets.

Compared to native beam decoding, beam-free inference also improves Recall@10 on Scientific and Video Games. On Musical Instruments, native beam decoding achieves a marginally higher

Recall@10, whereas beam-free inference performs better at the smaller cutoffs and on both NDCG metrics. Overall, these results show that beam-free inference improves the ranking of relevant items while retaining the retrieval performance of beam sampling.

Table 1: Recommendation performance across the three Amazon Reviews datasets. Baseline results are taken from Shi et al. (2025). The "Best item-ID" column denotes the strongest result among the seven item-ID baselines reported. Best scores are in **bold** and second-best scores are underlined.

Dataset	Metric	Best item-ID	TIGER	LC-Rec	RPG	LLaDA-Rec	Ours (native beam)	Ours (beam-free)
Scientific	Recall@1	0.0071	0.0084	0.0091	0.0087	0.0098	0.0117	**0.0122**
	Recall@5	0.0240	0.0282	0.0280	0.0257	0.0310	0.0328	**0.0338**
	Recall@10	0.0379	0.0446	0.0434	0.0395	0.0474	0.0479	**0.0493**
	NDCG@5	0.0152	0.0183	0.0186	0.0174	0.0203	0.0225	**0.0232**
	NDCG@10	0.0197	0.0236	0.0235	0.0218	0.0256	0.0273	**0.0282**
Musical Instruments	Recall@1	0.0094	0.0105	0.0119	0.0118	0.0128	0.0152	**0.0156**
	Recall@5	0.0331	0.0359	0.0379	0.0362	0.0406	0.0438	**0.0440**
	Recall@10	0.0525	0.0566	0.0587	0.0545	0.0623	**0.0648**	0.0645
	NDCG@5	0.0211	0.0233	0.0251	0.0241	0.0268	0.0297	**0.0300**
	NDCG@10	0.0273	0.0300	0.0318	0.0300	0.0337	0.0364	**0.0366**
Video Games	Recall@1	0.0149	0.0166	0.0165	0.0209	0.0203	0.0250	**0.0254**
	Recall@5	0.0516	0.0529	0.0567	0.0579	0.0623	0.0710	**0.0722**
	Recall@10	0.0823	0.0823	0.0891	0.0853	0.0942	0.1041	**0.1051**
	NDCG@5	0.0323	0.0348	0.0366	0.0397	0.0415	0.0484	**0.0492**
	NDCG@10	0.0421	0.0442	0.0471	0.0485	0.0517	0.0590	**0.0598**

4.3 Representation and Inference Ablations

In the following, we provide various ablation results.

Latent representation. As discussed in Section 3.4, our variational formulation supports discrete, continuous, and hybrid latent representations. We show results of an empirical comparison in Table 2. We observe that the discrete and hybrid variants achieve comparable Recall rates and NDCG scores across all cutoffs on Musical Instruments. The continuous model, however, falls behind these and does not achieve competitive results. Overall, these results highlight the effectiveness of discrete semantic IDs for representing items and modeling interactions.

Table 2: Test performance for Musical Instruments for different latent representations, i.e., discrete, continuous, and hybrid.

Latent representation	Recall@1	Recall@5	Recall@10	NDCG@5	NDCG@10
Continuous	0.0035	0.0114	0.0177	0.0074	0.0094
Discrete	**0.015512**	0.043907	**0.064364**	0.029916	**0.036485**
Hybrid	**0.015512**	**0.043942**	0.064347	**0.029927**	0.036478

Joint representation learning. Further, we investigate whether recommendation interactions provide a useful learning signal for the semantic-ID encoder. We compare the joint optimization of the encoder and denoiser to a two-stage approach in which the representation is learned separately and kept frozen during diffusion training.

Table 3: Joint versus fully frozen representation learning on Musical Instruments.

	Collision	R@1	R@5	R@10	N@5	N@10
Frozen	0.0112	0.0107	0.0345	0.0541	0.0227	0.0290
Joint	**0.0045**	**0.0112**	**0.0354**	**0.0560**	**0.0234**	**0.0300**

As shown in Table 3, joint representation learning consistently improves over the frozen approach across all Recall and NDCG metrics while also achieving a lower catalog collision rate. These results show that item interactions provide a valuable signal for learning discriminative discrete representations that improve downstream recommendation performance.

Further examine whether the learned semantic IDs preserve locality in the original observation space. For this, we compare the distances of the top-ranked item for each test prediction to the ground truth items.

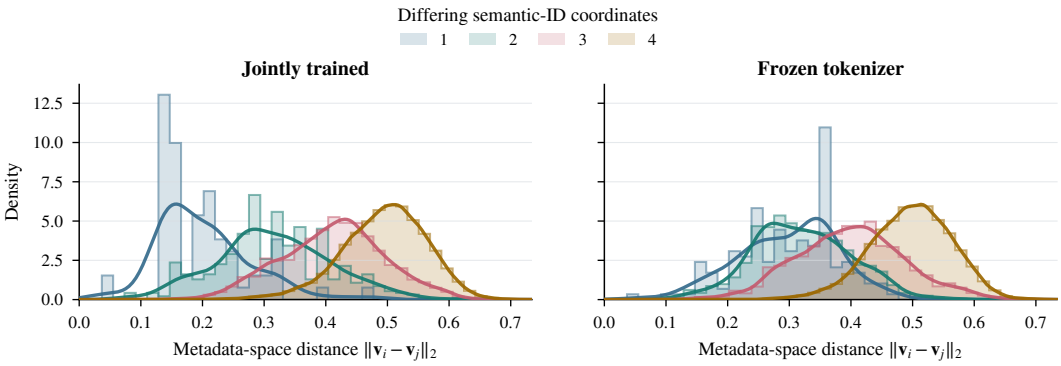

Figure 4: Errors in metadata space across different semantic-ID distances for the jointly trained and frozen models on Musical Instruments. We evaluate the top-ranked item on the test set and group predictions by the number of semantic-ID coordinates that differ from the target. The x-axis is $\|\mathbf{v}_i - \mathbf{v}_j\|_2$, where $\mathbf{v}_i$ and $\mathbf{v}_j$ are the `sentence-t5-base` embeddings of the target and recommended items.

As shown in Fig. 4, larger semantic-ID distances generally correspond to larger errors in input space, i.e., their `sentence-t5-base` embeddings. This relationship is stronger in joint training: predictions that differ in only one coordinate remain closer to the target than with a frozen tokenizer.

Loss ablations. We analyze the impact of the classification and reconstruction objectives in Table 4. Including both objectives achieves the best recommendation performance and the lowest collision rate. While removing either individually leads to only a small decrease in performance, removing both substantially increases collisions and degrades all recommendation metrics. This shows that the two objectives provide important supervision to learn discriminative and meaningful semantic IDs.

Table 4: 20-epoch Musical Instruments loss ablation.

Class.	Recon.	Collision	R@1	R@5	R@10	N@5	N@10
✓	✓	**0.0456**	**0.0100**	**0.0334**	**0.0514**	**0.0217**	**0.0275**
✗	✓	0.0538	0.0094	0.0319	0.0508	0.0206	0.0266
✓	✗	0.0466	0.0098	0.0325	0.0509	0.0211	0.0270
✗	✗	0.1957	0.0084	0.0254	0.0380	0.0171	0.0211

Table 5: Various ablations on the Musical Instruments validation set.

Factor	Setting	R@10	N@10
Steps	1	0.0052	0.0045
	2	0.0106	0.0082
	4	**0.0620**	**0.0327**
Order	Left-to-right	0.0609	0.0324
	Right-to-left	0.0611	**0.0328**
	Adaptive	**0.0620**	0.0327
λ_{hist}	0	0.0300	0.0155
	1	0.0300	0.0155
	4	0.0359	0.0182
	8	**0.0366**	**0.0187**
	16	0.0362	0.0185

Generation dynamics and history masking. In Table 5, we analyze the effect of the number of denoising steps, unmasking order, and the history-loss weight. Increasing the number of denoising steps from one to four is crucial for performance and substantially improves the predictions. The unmasking order, however, has a much smaller effect. We assume this may result from the fact that each semantic ID consists of only four tokens. Finally, increasing λ_{hist} from 0 to 8 improves both metrics, while a further increase to 16 slightly reduces them. This implies that the history loss provides an important supervision signal.

Score ranking. We analyze the impact of the individual ranking components in Table 6. Expected denoising scores provide the strongest individual ranking signal, while N-gram statistics substantially improve the initial neural proposal. Combining the model-based and interaction-based components achieves the best performance across all metrics.

Table 6: Score decomposition on 100 Musical Instruments candidates.

Candidate score	R@10	N@5	N@10
Neural proposal only	0.0298	0.0120	0.0156
Train transition only	0.0551	0.0235	0.0298
Neural + transition	0.0573	0.0259	0.0319
Denoising scores only	0.0601	0.0261	0.0327
Complete selected ranker	**0.0645**	**0.0300**	**0.0366**

5 Related Work

Sequential recommendation. Sequential recommendation systems predict the next item from an interaction history. Earlier neural approaches use recurrent neural networks (Hidasi et al., 2015), self-attention (Kang & McAuley, 2018), or transformer architectures (Sun et al., 2019) to directly predict or rank over the whole item catalog. While effective, the output parameters of the models remain independent and are not shared across items. Further, the output space grows with catalog size. Content-aware methods further incorporate item metadata into the input representations (Hou et al., 2023; Yang et al., 2024).

Generative recommendation with semantic IDs. Generative recommenders, on the other hand, first map items to semantic IDs and then generate the tokens of the next item conditioned on the interaction history. TIGER constructs these using residual quantization in combination with autoregressive generation (Rajput et al., 2023). Other approaches extended generative recommendation with collaborative tokenization, language-model alignment, and parallel semantic IDs (Wang et al., 2024; Zheng et al., 2024; Hou et al., 2025). Sharing a token vocabulary enables information transfer between similar items and reduces the network's output space. However, this representation learning can suffer from representation collapse, necessitating collision handling or repair (Rajput et al., 2023; Wang et al., 2024; Zheng et al., 2024; Shi et al., 2025).

Most semantic-ID pipelines construct item tokens before training the generative recommender, creating a potential mismatch between the learned representations and item interactions. ETEGRec alternates between training an item tokenizer and an autoregressive recommender using alignment objectives (Liu et al., 2025). DIGER uses differentiable RQ-VAE assignments and an autoregressive recommendation loss to train the tokenizer (Fu et al., 2026). Finally, UniSID jointly learns representations and their discrete identifiers from raw advertising data (Jiang et al., 2026). Our framework similarly makes the item tokenization and denoising jointly trainable, while combining the denoising objective with an explicit retrieval objective that encourages collision-free semantic IDs.

Diffusion-based recommendation. Earlier diffusion recommenders are primarily based on Gaussian transitions on interaction vectors (Wang et al., 2023) or item representations, recovering a continuous target (Li et al., 2023; Du et al., 2023; Yang et al., 2023).

Another line of work applies discrete diffusion to semantic-ID recommendation (Liu et al., 2026; Shi et al., 2025; Shah et al., 2025; Mu et al., 2026). They progressively mask semantic-ID tokens and train denoisers to reconstruct them, enabling flexible non-autoregressive generation orders and parallel token updates. These methods typically train the tokenizer separately before training the diffusion model. Our framework, in contrast, jointly trains the semantic-ID encoder and the diffusion model through denoising and catalog-retrieval objectives via a variational formulation that also supports continuous and hybrid latent representations.

6 Conclusion

In this work, we introduced a latent diffusion model for generative recommendation that jointly learns semantic representations and a denoising model. By allowing gradients of the item interactions to shape the tokenizer, our approach shapes our representations, i.e., semantic IDs, to be predictive and discriminative across the item catalog. We further proposed a beam-free retrieval procedure that generates candidates by ranking items using their expected denoising likelihood.

Our experiments on three Amazon Reviews datasets demonstrate that the proposed model consistently outperforms comparable baselines. More specifically, beam-free inference achieves the best NDCG across all three datasets, while remaining competitive with native beam search in Recall. In our ablations, we show that discrete semantic IDs perform substantially better than continuous latent variables and that joint representation learning improves over a fully frozen tokenizer.

Limitations. Our current framework relies on a post-training collision-repair step and a ranking model, leading to computational overhead. Moreover, our expected denoising scoring ranking increases inference cost, even though it can be evaluated in parallel, unlike beam sampling.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models and conceptual frameworks and to provide feedback on the research methodology and experiments. We have not used generative AI tools to generate synthetic datasets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, or propose or refine hypotheses. Additionally, we used generative AI tools to create or modify figures and illustrations, suggest experimental parameters, create or edit code, and rephrase parts of the paper. We have reviewed all AI-assisted work, i.e., we checked generated ideas and written text, and verified and tested code for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Hanwen Du, Huanhuan Yuan, Zhen Huang, Pengpeng Zhao, and Xiaofang Zhou. Sequential recommendation with diffusion models. *arXiv preprint arXiv:2304.04541*, 2023.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. *arXiv preprint arXiv:2601.19711*, 2026.

Liam Hebert, Marialena Kyriakidi, Hubert Pham, Krishna Sayana, James Pine, Sukhdeep Sodhi, and Ambarish Jash. Flare: Fusing language models and collaborative architectures for recommender enhancement. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 2235–2244, 2025.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pp. 1162–1171, 2023.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. Generating long semantic ids in parallel for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 956–966, 2025.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking llms as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, 2026.

Jie Jiang, Xinxun Zhang, Enming Zhang, Yuling Xiong, Jun Zhang, Jingwen Wang, Huan Yu, Yuxiang Wang, Hao Wang, Xiao Yan, et al. End-to-end semantic id generation for generative advertisement recommendation. *arXiv preprint arXiv:2602.10445*, 2026.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11513–11522. IEEE, 2022.

Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42(3):1–28, 2023.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 729–739, 2025.

Zhao Liu, Yichen Zhu, Yiqing Yang, Xiao Lv, Guoping Tang, Rui Huang, Qiang Luo, Ruiming Tang, and Guorui Zhou. Diffgrm: Diffusion-based generative recommendation model. In *Proceedings of the ACM Web Conference 2026*, pp. 5853–5864, 2026.

Lingyu Mu, Hao Deng, Haibo Xing, Jinxin Hu, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. Masked diffusion generative recommendation. *arXiv preprint arXiv:2601.19501*, 2026.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pp. 1864–1874, 2022.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Kulin Shah, Bhuvesh Kumar, Neil Shah, and Liam Collins. Masked diffusion for generative recommendation. *arXiv preprint arXiv:2511.23021*, 2025.

Teng Shi, Chenglei Shen, Weijie Yu, Shen Nie, Chongxuan Li, Xiao Zhang, Ming He, Yan Han, and Jun Xu. Llada-rec: Discrete diffusion for parallel semantic id generation in generative recommendation. *arXiv preprint arXiv:2511.06254*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. Transformer memory as a differentiable search index. *Advances in neural information processing systems*, 35:21831–21843, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 832–841, 2023.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024.

Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, et al. Unifying generative and dense retrieval for sequential recommendation. *arXiv preprint arXiv:2411.18814*, 2024.

Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Advances in Neural Information Processing Systems*, 36:24247–24261, 2023.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1435–1448. IEEE, 2024.

A IMPLEMENTATION DETAILS

In this section, we describe the dataset specifics and implementation details of the discrete, continuous, and hybrid latent models.

A.1 DISCRETE MODEL

Dataset splits. We use the pre-split Amazon Reviews 2023 dataset (Hou et al., 2026). The dataset statistics are summarized in Table 7.

Table 7: Number of interactions in each split and items in the training catalog.

Dataset	Train	Validation	Test	Training Items
Industrial and Scientific	310,977	50,985	50,985	25,754
Musical Instruments	396,958	57,439	57,439	24,556
Video Games	625,062	94,762	94,762	25,527

Training catalog and sequences. The training catalog contains all items that occur in the interactions of the training set. We represent each item via its semantic ID. Further, we only include items that were observed in the training set to prevent leakage. We truncate each interaction sequence to the 20 most recent items and the target. Shorter histories are padded accordingly. Therefore, each sequence consists of at most $21 \times 4 = 84$ tokens. We use each training interaction once per epoch.

Collision metrics. The catalog collision rate is the fraction of catalog items whose semantic ID is shared by another item across all tokens. The batch collision rate uses the same definition but is computed within the batch.

Tokenizer architecture. We use the `sentence-t5-base` model (Ni et al., 2022) to obtain a fixed 768-dimensional embedding for each item. Then, we normalize these by subtracting the mean computed over the training items. Our encoder is an MLP with dimensions $768 \to 512 \to 256 \to 1024$, ReLU activations, and LayerNorm. The output is split into four parallel 256-dimensional blocks, each with an independent codebook of size 256. The decoder maps $1024 \to 256 \to 512 \to 768$ without a LayerNorm.

Denoiser architecture. We use the bidirectional Transformer architecture of LLaDA-Rec (Shi et al., 2025). We flatten the semantic IDs of the history and target items into a sequence of at most 84 tokens. The model has a hidden width of 256 and uses an independent classification head for each semantic-ID position. Dataset-specific numbers of layers and attention heads are reported in Table 8.

Quantizer and codebook updates. Let $\mathbf{h}_i^{(m)}$ denote encoder block m for item i and $\mathbf{e}_k^{(m)}$ the kth vector in its K-entry codebook. Its soft assignment vector is

$$\mathbf{q}_{\phi,i}^{(m)} := \left(q_\phi(z_i^{(m)} = k \mid y_i)\right)_{k=1}^{K} = \mathrm{softmax}\left(\left(\frac{s(\mathbf{h}_i^{(m)}, \mathbf{e}_k^{(m)})}{\tau_q}\right)_{k=1}^{K}\right). \tag{17}$$

We take $s(\mathbf{h}, \mathbf{e}) = -\|\mathbf{h} - \mathbf{e}\|_2^2$ for the Euclidean quantizer and $s(\mathbf{h}, \mathbf{e}) = \cos(\mathbf{h}, \mathbf{e})$ for cosine–EMA. We use nearest-cosine identifiers in the forward pass. Codebook centroids are updated from detached, Sinkhorn-balanced soft assignments with temperature 0.1 and three balancing iterations. Writing $\bar{e}_k^{(n)}$ for the balanced batch centroid of code k at update n, we use the EMA update

$$e_k^{(n)} = \rho e_k^{(n-1)} + (1 - \rho)\bar{e}_k^{(n)}. \tag{18}$$

This affects centroid statistics, but does not replace the hard nearest-code identifiers. The stage-specific decay ρ is reported in Table 10.

Straight-through batch estimator. To enable gradient flow from the denoising loss to the encoder, we use the straight-through estimator (Bengio et al., 2013), which replaces gradients of hard assignments with their continuous counterparts. More specifically, we use

$$\widetilde{A}_i = A_i + Q_i - \text{stopgrad}(Q_i), \tag{19}$$

where A_i denotes the hard one-hot variable and Q_i the continuous one. We estimate the catalog denominator in Eq. (10) over batch items with:

$$S_{ij} = \sum_{m=1}^{4} \sum_{k=1}^{256} \widetilde{A}_{imk} \log q_\phi(z^{(m)} = k \mid y_j). \tag{20}$$

This enables gradient flow from the classification loss through the hard assignments to the encoder rather than just from q.

Target and history corruption. We use different noising strategies for the target and context. For the tokens of the target, we replace $r \sim U(\{1, 2, 3, 4\})$, tokens with a mask. As done by Shi et al. (2025), a second denoising pass samples a sequence-level count $r \in \{0, \ldots, 4\}$, converts it to masking probability, and noises all coordinates independently. We evaluate the cross-entropy loss only at masked positions. We denote the loss from the target-only pass by $\mathcal{L}_{\text{diff}}$ and the loss from the sequence-wide pass by $\mathcal{L}_{\text{hist}}$.

Auxiliary tokenizer losses. The decoder reconstructs the normalized embeddings from the quantized representation. We use the normalized mean squarred error as loss:

$$\mathcal{L}_{\text{rec}} = \frac{|\mathcal{C}|}{|\mathcal{B}|} \frac{\sum_{i \in \mathcal{B}} \| D_\phi(\widehat{z}_i) - \widetilde{\mathbf{v}}_i \|_2^2}{\sum_{j \in \mathcal{C}} \| \widetilde{\mathbf{v}}_j \|_2^2}. \tag{21}$$

Here $\mathbf{v}_i$ represents the metadata embedding of item y_i, and $\widetilde{\mathbf{v}}_i$ the normalized embeddings. Further, we include a VQ loss for stabilization:

$$L_{\text{vq}} = \beta \, \mathbb{E}_{i,m} \left\| \mathbf{h}_i^{(m)} - \text{stopgrad}\left(\mathbf{e}^{(m)}_{z_i^{(m)}} \right) \right\|_2^2, \tag{22}$$

with $\beta = 0.25$.

Dataset-specific settings. The selected settings are summarized in Table 8. The prediction temperature rescales denoiser logits before the output softmax.

Table 8: Dataset-specific denoiser and optimization settings.

Setting	Industrial and Scientific	Musical Instruments	Video Games
Layers	4	4	6
Attention heads	4	8	8
λ_{hist}	3.5	4.0	5.0
τ_{pred}	0.3	0.7	1.0
Denoiser weight decay	0.05	0.05	0.001

Objective weights and optimization. Hyperparameters and settings are given in Tables 8 to 10.

Table 9: Hyperparameters and settings.

Setting	Value
λ_{diff}	1
λ_{cls}	80
λ_{rec}	100
λ_{vq}	1
β	0.25
Optimizer	AdamW
Batch size	256
Encoder/decoder decay	10^{-4}
Codebook decay	0

Table 10: Optimization stages.

Setting	Stabilization	Joint training	Frozen continuation
Duration	2,000 updates	150 epochs	50 epochs
Optimized parameters	Denoiser + tokenizer	Denoiser + tokenizer	Denoiser only
Learning rate	10^{-3}	3×10^{-3} / 9×10^{-8}	10^{-4}
Schedule	Constant	Cosine	Cosine
Latent normalization	On	Off	Off
Codebook EMA ρ	0.999	0.9999999	Frozen

For joint training, the denoiser and tokenizer entries are the peak rates, respectively. The tokenizer rate is 9×10^{-8}. These are followed by a cosine decay with a 10% linear warm-up. The stabilization uses a constant rate. The final stabilization checkpoint is used for joint training. After 150 epochs, the repaired joint checkpoint initializes the frozen finetuning described below.

Collision repair and frozen finetuning. After joint training, we encode the complete training catalog and reassign the last code of colliding items using Sinkhorn-balanced prototype scores. We use $\epsilon = 0.003$, ten Sinkhorn iterations, and at most 30 repair rounds without validation or test targets. Then, we freeze the tokenizer and finetune the denoiser for 50 epochs with peak learning rate 10^{-4} and a cosine schedule.

Native beam decoding. Our native inference starts from four masks, performs four denoising evaluations, reveals the unresolved coordinate with the largest confidence, expands code values, and retains the best partial tuples. Complete tuples are mapped to training-catalog items and are deduplicated before ranking.

Beam-free proposal and diffusion scores. The final denoising model first performs an all-mask prediction and scores every valid catalog tuple by the sum of its four head log-probabilities. We retain 100 candidates for each prediction. For each candidate, we then score all $2^4 - 1 = 15$ non-empty target masking combinations in parallel. We set $(w_M = 1/(15|M|))$, giving equal weight to each pattern after averaging over its masked coordinates. Thus, the proposal and likelihood stages require two sequential neural evaluations.

Final ranking. We fit a linear interaction model on validation that combines the diffusion scores, training-only transition features, popularity, and their pairwise and normalized interactions. It only reorders the fixed candidate set and does not modify the diffusion weights or semantic IDs.

A.2 Continuous and hybrid representation ablations

The continuous and hybrid systems are representation ablations rather than components of the selected discrete pipeline. Let $\mathbf{z}^d$ denote the discrete semantic ID and $\mathbf{z}^c$ a continuous item latent. The hybrid encoder factorizes as

$$q_\phi(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{y}) = q_{\phi_d}(\mathbf{z}^d \mid \mathbf{y}) q_{\phi_c}(\mathbf{z}^c \mid \mathbf{y}), \tag{23}$$

$$p_\theta(\mathbf{y}, \mathbf{z}^d, \mathbf{z}^c \mid \mathbf{x}) = p(\mathbf{y} \mid \mathbf{z}^d, \mathbf{z}^c) p_\theta(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{x}). \tag{24}$$

Removing $\mathbf{z}^c$ recovers the discrete model, removing $\mathbf{z}^d$ gives continuous latent diffusion with distance-based catalog retrieval, and retaining both variables gives the hybrid.

The common variational objective is

$$\log p(\mathbf{y} \mid \mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{y})} \left[\log \frac{p(\mathbf{y} \mid \mathbf{z}^d, \mathbf{z}^c) p_\theta(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{x})}{q_{\phi_d}(\mathbf{z}^d \mid \mathbf{y}) q_{\phi_c}(\mathbf{z}^c \mid \mathbf{y})} \right] \tag{25}$$

$$= \underbrace{\mathbb{E}_q[\log p(\mathbf{y} \mid \mathbf{z}^d, \mathbf{z}^c)]}_{\text{joint classification term}} + \underbrace{\mathbb{E}_q[\log p_\theta(\mathbf{z}^d, \mathbf{z}^c \mid \mathbf{x})]}_{\text{joint diffusion term}} \tag{26}$$

$$- \mathbb{E}_{q_{\phi_d}}[\log q_{\phi_d}(\mathbf{z}^d \mid \mathbf{y})] - \mathbb{E}_{q_{\phi_c}}[\log q_{\phi_c}(\mathbf{z}^c \mid \mathbf{y})]. \tag{27}$$

For the hybrid, the classification score combines the discrete score with continuous distance,

$$\mathcal{L}_{\text{hyb-cls}} = -\log \frac{\exp\big(s_\phi(\mathbf{y}, \mathbf{z}^d) - \frac{1}{2\sigma^2}\|\mathbf{z}^c - \boldsymbol{\mu}_\phi(\mathbf{y})\|^2\big)}{\sum_{\mathbf{y}'} \exp\big(s_\phi(\mathbf{y}', \mathbf{z}^d) - \frac{1}{2\sigma^2}\|\mathbf{z}^c - \boldsymbol{\mu}_\phi(\mathbf{y}')\|^2\big)}. \tag{28}$$

The shared network applies Gaussian corruption to $\mathbf{z}^c$ and masking to $\mathbf{z}^d$. Its denoising objective is

$$\mathcal{L}_{\text{hyb-diff}} = \mathbb{E}_{t,\mathbf{z}_t,\mathbf{z}}\Big[w_c(t)\|\mathbf{z}^c - f_{\theta_c}(\mathbf{z}_t^d, \mathbf{z}_t^c, \mathbf{x}, t)\|^2 \tag{29}$$

$$- w_d(t) \log p_{\theta_d}(\mathbf{z}^d \mid \mathbf{z}_t^d, \mathbf{z}_t^c, \mathbf{x}, t)\Big]. \tag{30}$$

The ablation minimizes $\mathcal{L}_{\text{hyb-cls}} + \mathcal{L}_{\text{hyb-diff}}$.