OXYGENFLOWREC: SEMANTIC-ID RECOMMENDATION VIA ONE-STEP CONTINUOUS GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic-ID (SID) recommenders commonly predict discrete code tuples, coupling recommendation to token-level decoding and, in beam-based methods, a search budget. We ask whether SID-based semantic sharing and SID outputs can be retained without discrete code prediction. We propose OxygenFlowRec, a catalogue-grounded SID recommender based on one-step continuous generation. Fixed SID assignments define parameter sharing, while task-trained code embeddings and frequency-gated item residuals learn recommendation-specific geometry. Conditioned on user history, the model generates a continuous query in a single update, ranks catalogue items by similarity, and returns their SIDs through deterministic lookup while preserving exact item identities. Training combines target recovery with supervision of the actual inference endpoint. To accommodate the different gradient aggregation patterns of shared codes and item residuals, we assign separate learning-rate scales to these parameter groups. This design separates the continuous generation process from the discrete SID output interface, avoiding token-level decoding and beam search. Experiments on six Amazon Reviews 2023 categories demonstrate competitive recommendation accuracy, while inference benchmarks show substantially lower retrieval cost relative to our beam-based LLaDA-Rec reproduction.

1 INTRODUCTION

Generative recommendation recasts item retrieval as identifier generation rather than direct candidate scoring. A prominent approach represents each item with a semantic ID (SID), a short tuple of discrete codes obtained by quantising its content representation (Rajput et al., 2023; Wang et al., 2024). Shared codes capture semantic structure across catalogue items and replace a catalogue-sized token vocabulary with several smaller code vocabularies. Autoregressive approaches, including TIGER, LETTER-based models and LC-Rec, predict these codes sequentially (Rajput et al., 2023; Wang et al., 2024; Zheng et al., 2024). Parallel generation in RPG and masked discrete diffusion in LLaDA-Rec change the prediction procedure, but retain discrete codes as the variables generated by the model (Hou et al., 2025; Shi et al., 2025).

For beam-based SID recommenders, producing a recommendation list couples accuracy to the decoding budget. A narrow beam can discard promising candidates, whereas a wider beam increases computation; multiple decoding stages further increase the number of candidate states evaluated. Parallel prediction reduces sequential dependence, but the distinction between the output interface and the generation process remains important: returning a discrete identifier does not necessarily require generating in a discrete space. This motivates our central question:

> *Can a recommender retain SID-based semantic sharing and SID outputs while replacing stepwise code prediction with one-step continuous generation?*

Continuous recommenders such as DreamRec and DiffuRec already generate item representations instead of code tokens (Yang et al., 2023; Li et al., 2023), but do not explicitly organise those representations through SID-indexed parameter sharing. Their difference from SID-based methods is therefore not an absence of learned semantics, but the absence of this particular structural prior.

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.

Conversely, simply generating over frozen quantiser centroids would preserve geometry optimised for content reconstruction rather than recommendation. These choices need not be coupled. VQ-Rec separates discrete assignments from task-learned code embeddings (Hou et al., 2023), while MIRAGE studies one-step recovery in a learned item space (Fang et al., 2026). Building on these directions, we separate three roles that are often conflated: *semantic sharing, recommendation geometry and the SID output interface*.

We propose OxygenFlowRec, a catalogue-grounded SID recommender based on one-step continuous generation. Fixed SID assignments determine which parameters items share, while recommendation training learns the corresponding code embeddings and frequency-gated item residuals. A history-conditioned network learns to recover target item states from continuous interpolants, with additional supervision on the endpoint actually used at inference. At serving time, a single update produces a continuous query. Full-catalogue cosine similarity ranks the learned item representations, and deterministic lookup returns the selected items' SID tuples. The system therefore retains SID outputs without token-level prediction or beam search: the generation states and paths are continuous, while catalogue selection and the final identifiers remain discrete.

Separating item ranking from SID readout also clarifies the granularity of recommendation. When multiple items share a complete SID tuple, a tuple match alone does not establish an exact-item hit. OxygenFlowRec introduces item-specific capacity through residuals, allowing same-tuple items to be distinguished when their residuals are enabled. Its SID readout preserves the selected item identities, ranking and duplicate tuples. This does not eliminate tokenizer collisions; instead, it allows semantic sharing and item-level discrimination to coexist without requiring SID tuples to uniquely identify every item.

The shared-code and item-specific components also have different optimisation structures. A code embedding aggregates gradient contributions from multiple items, whereas each residual has an item-specific contribution. We therefore assign separate learning-rate scales to these parameter groups and evaluate the choice under controlled training budgets. Together with inference-aligned recovery training, this design provides a route to SID recommendation that avoids repeated code decoding while retaining task-adapted item representations.

Contributions.

- **Continuous generation with SID output.** We introduce a catalogue-grounded SID recommender combining task-learned shared-code embeddings, item residuals and inference-aligned recovery training. One-step continuous query generation followed by catalogue readout produces ranked SIDs without token-level decoding or beam search.
- **Representation-aware optimisation.** We characterise the different gradient aggregation patterns of shared codes and item residuals, and use separate learning-rate scales. Controlled experiments evaluate this optimisation choice alongside the contributions of SID sharing and recovery training.
- **Competitive accuracy with lower measured retrieval cost.** OxygenFlowRec improves Recall@20 over our LLaDA-Rec reproduction on five of six categories, with Recall@20 and NDCG@20 gains of 24.4% and 24.5% on the largest category. Against four-step, beam-50 LLaDA-Rec, cached top-20 retrieval on one B200 in FP32 is $21.9\times$ faster per user at batch 64 and $1.88\times$ faster at batch one.

2 Background and Related Work

Setting and SIDs. Given $S_u = (i_1, \ldots, i_L)$, $L \leq 20$, we rank N items in $\mathcal{I}$ and return top-K item IDs and SIDs, holding out final/penultimate interactions for test/validation. Quantisation yields $c_i = (c_{i1}, \ldots, c_{iM})$, $c_{im} \in \{1, \ldots, K_{\text{cb}}\}$ (van den Oord et al., 2017; Lee et al., 2022; Hou et al., 2023). Our shared frozen tokenizer (Shi et al., 2025) uses Sentence-T5-base (Ni et al., 2022), PCA-128 and 4×256 multi-head VQ codes.

SID generation and retrieval. TIGER is autoregressive (Rajput et al., 2023); LETTER and LC-Rec learn collaborative tokenisation and alignment (Wang et al., 2024; Zheng et al., 2024). LLaDA-Rec uses masked discrete diffusion (Nie et al., 2025; Shi et al., 2025); RPG combines parallel SID

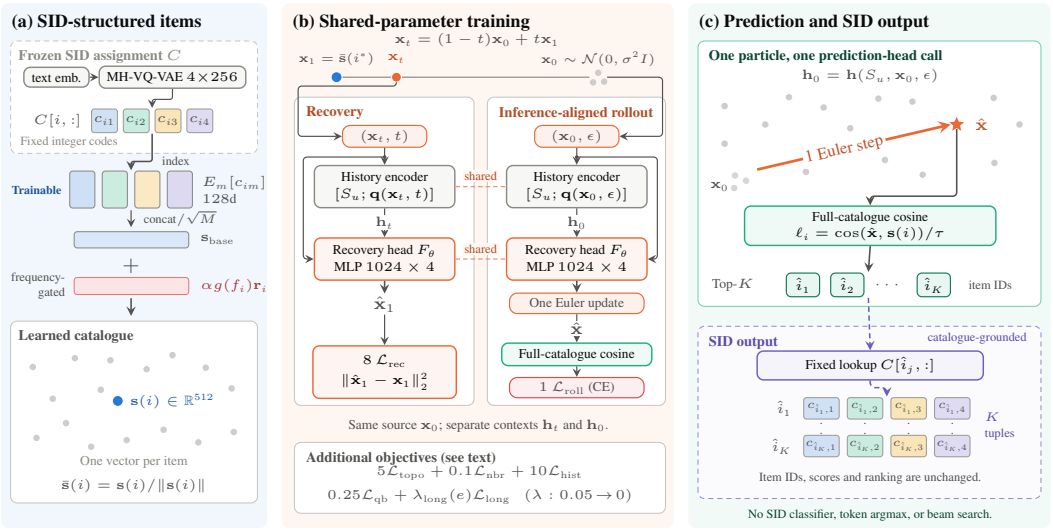


Figure 1: OxygenFlowRec. (a) Frozen four-code assignments index trainable SID embeddings; frequency-gated residuals yield item vectors. (b) Recovery and rollout reuse $\mathbf{x}_0$ and share the encoder and F_θ (dashed links), but recompute contexts at $(\mathbf{x}_t, t)$ and $(\mathbf{x}_0, \epsilon)$. Recovery supervises $\hat{\mathbf{x}}_1$; rollout CE scores $\hat{\mathbf{x}}$. (c) One prediction-head call and one Euler update yield a continuous query; full-catalogue cosine ranks item IDs. The purple readout returns their four-code tuples from the same frozen C, retaining IDs, scores, order and duplicate tuples. This is catalogue-grounded SID output, not token-level prediction or inverse flow; it differs from the nearest-code diagnostic in Appendix I.

prediction with graph-guided retrieval (Hou et al., 2025). LIGER combines generative/dense retrieval (Yang et al., 2024); VQ-Rec learns centroid-independent code embeddings (Hou et al., 2023). Our residuals add item-specific capacity but sacrifice code-only compression.

Continuous generation. DreamRec and DiffuRec use continuous diffusion (Yang et al., 2023; Li et al., 2023); FMRec and FlowRec (Liu et al., 2025; Li et al., 2025) adapt flow-based prediction (Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2023). MIRAGE combines recovery and geometric regularisation for one-step inference (Fang et al., 2026). We study SID–residual states, rollout-endpoint supervision and grouped learning rates within this family. Dense retrieval (Covington et al., 2016; Yi et al., 2019) and fixed lookup provide SID outputs without token decoding.

3 OxygenFlowRec

OxygenFlowRec retains SID sharing and output while generating a continuous query. Similarity ranking selects catalogue items, and fixed lookup returns their SIDs. Figure 1 summarises this pipeline. We describe the item manifold (§3.1), the history encoder (§3.2), the flow and its training objective (§3.3–§3.4), the learning-rate multipliers used for this representation (§3.5), and SID-output inference (§3.6).

3.1 SID-Structured, Task-Learned Item Representations

Each codebook m owns a trainable table $E_m \in \mathbb{R}^{K_{\text{cb}} \times d_c}$ with d_c=128, initialised randomly — we do *not* import the VQ-VAE centroids. The SID base of item i is the concatenation of its four code embeddings, rescaled so that its norm is independent of M:

$$\mathbf{s}_{\text{base}}(i) = \frac{1}{\sqrt{M}} \operatorname{concat}\big(E_1[c_{i1}],\, E_2[c_{i2}],\, E_3[c_{i3}],\, E_4[c_{i4}]\big) \in \mathbb{R}^D, \qquad D = M \cdot d_c = 512. \quad (1)$$

Items assigned an identical tuple have the same base vector. Such collisions are an empirical tokenizer property, not a consequence of the 256^4 possible tuples. We add a per-item residual $\mathbf{r}_i \in \mathbb{R}^D$, scaled

by a constant α=0.2 and gated by training frequency f_i:

$$\mathbf{s}(i) = \mathbf{s}_{\text{base}}(i) + \alpha\, g(f_i)\, \mathbf{r}_i, \qquad g(f_i) = \min\left(1,\ \sqrt{\frac{\max(f_i - 1, 0)}{8}}\right). \tag{2}$$

The gate (Figure 3a, Appendix A) assigns an item seen once $g = 0$, using only shared SID embeddings; an item seen at least nine times gets the full residual. This is a capacity-allocation rule, not a guarantee of tail improvement. The padding row has g=0 and is masked out of every score. Same-tuple items whose residuals are both disabled still have identical representations.

The code assignment determines parameter sharing, whereas recommendation training learns the vector geometry. We use "item manifold" only informally for this learned embedding space; no smooth-manifold constraint is imposed on the finite catalogue or on the interpolation path. The residual increases item-specific capacity, and its ablation cannot by itself distinguish collision resolution from this capacity effect.

3.2 State-Conditioned History Encoding

Unlike discrete models, which expand a length-L history into $M \cdot L$ code tokens, OxygenFlowRec represents each history item by its single continuous state, projected from 512 to d_h=256. A learned query token is appended and the sequence is encoded by a 6-layer, 8-head pre-norm Transformer (RMSNorm, SiLU, MLP ratio 4, dropout 0.1) — 21 tokens for a length-20 history rather than 81. We let the query token see the current flow state, so history attention can adapt to where the trajectory is, bounding the update to keep the state from overwhelming the history signal:

$$\Delta\mathbf{q} = \gamma \cdot \frac{\phi(\mathbf{x}_t, t)}{1 + \|\phi(\mathbf{x}_t, t)\|/\beta}, \qquad \gamma = 2.0,\ \beta = 0.5, \tag{3}$$

where ϕ combines learned state and time projections. The norm is bounded by $\gamma\beta = 1$, not by β alone. Gradients through this query update on the interpolant branch are scaled by 0.25; the rollout branch uses scale one. Removing this term costs 3.5% Recall@20 (Table 3).

3.3 One-Step Target-Recovery Training

Let $\bar{\mathbf{s}}(i) = \mathbf{s}(i)/\|\mathbf{s}(i)\|$ and $\mathbf{x}_1 = \bar{\mathbf{s}}(i_{L+1})$. We sample $\mathbf{x}_0 \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.03125$ and $t \sim \mathcal{U}(0, 1)$, clipped to $[\epsilon, 1 - \epsilon]$ for $\epsilon = 10^{-3}$. For $\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1$, the network predicts

$$\hat{\mathbf{x}}_1 = F_\theta\big(\mathbf{x}_t, t, \mathbf{h}(S_u, \mathbf{x}_t, t)\big). \tag{4}$$

F_θ is a width-1024, four-block residual MLP; each block expands to width 2048 before projecting back. Its source-conditioned update is $v_\theta(\mathbf{x}_t, t, \mathbf{h}; \mathbf{x}_0) = \hat{\mathbf{x}}_1 - \mathbf{x}_0$. For each fixed source–target pair, regressing this update to $\mathbf{x}_1 - \mathbf{x}_0$ is algebraically equivalent to endpoint regression. This does not establish a source-independent marginal vector field or a one-step density-transport guarantee.

At inference we use this network as a one-step conditional predictor. Small per-coordinate noise should not be confused with negligible vector noise: $\sqrt{\mathbb{E}\|\mathbf{x}_0\|^2} = \sqrt{512}\sigma \simeq 0.707$, while $\|\mathbf{x}_1\| = 1$. One-step performance is an empirical result, not a consequence of this noise scale or of the straightness of individual training interpolants.

3.4 Training Objective

Catalogue logits are cosine similarities at temperature $\tau = 0.07$:

$$\ell_j(\mathbf{x}) = \frac{\mathbf{x}^\top \mathbf{s}(j)}{\tau \|\mathbf{x}\| \|\mathbf{s}(j)\|}, \qquad j = 1, \ldots, N. \tag{5}$$

The padding item is masked. The recorded configuration optimises

$$\begin{aligned}\mathcal{L} = 8\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{roll}} + 0.25\mathcal{L}_{\text{qb}} + 5\mathcal{L}_{\text{topo}} \\ + 0.1\mathcal{L}_{\text{nbr}} + 10\mathcal{L}_{\text{hist}} + \lambda_{\text{long}}(e)\mathcal{L}_{\text{long}}.\end{aligned} \tag{6}$$

Recovery uses $(\mathbf{x}_t, t)$, whereas rollout recomputes the shared network and state-conditioned history at $(\mathbf{x}_0, \epsilon)$ using the same sampled source. Rollout CE scores the inference endpoint of Equation 8, not

$\hat{\mathbf{x}}_1$, against all catalogue items; target and catalogue states receive gradients. The query correction is frequency-weighted minus unweighted rollout CE, with the catalogue detached only in that correction. Topology, neighbour, history and long-history terms regularise training; Appendix L specifies their reductions, negative sampling and gradient routes. The long-history weight starts at 0.05 and anneals to zero over epochs 20–24, after the selected mainline checkpoints, so that anneal does not explain the reported results.

3.5 Separate Learning Rates for Shared and Item-Specific Parameters

Both shared codes and residuals receive dense catalogue gradients: $\partial \mathcal{L}_{\mathrm{CE}} / \partial \ell_j = p_j - \mathbf{1}[j = y]$. A nonzero-gated residual can therefore update without appearing in the history or target. Shared-code gradients aggregate across items; item residuals do not. Aggregation may reinforce or cancel gradients, not imply universally larger updates (Appendix L).

We tune separate multipliers as an empirical optimisation choice:

$$\eta_{\mathrm{SID}} = \eta(e)\kappa(e)\rho_{\mathrm{SID}}, \quad \eta_{\mathrm{res}} = \eta(e)\kappa(e)\rho_{\mathrm{res}}, \quad \rho_{\mathrm{SID}} = 0.125, \quad \rho_{\mathrm{res}} = 1. \tag{7}$$

Here $\eta(e)$ follows the cosine schedule with warm-up and base rate 3×10^{-4}; $\kappa(e)$ falls linearly from 1.5 to 0.25 over epochs 15–18. The flow and history groups use multipliers one and 0.25, respectively. At seed 13, restoring equal item-group multipliers reduces Recall@20 by 18.3%. This supports the selected recipe but does not establish the proposed update mechanism; the wider sweep also contains unequal training budgets (Section 4.6).

3.6 One-Step Prediction and Semantic-ID Output

A deterministic hash of the history seeds the Gaussian source, making evaluation reproducible. With $\mathbf{h}_0 = \mathbf{h}(S_u, \mathbf{x}_0, \epsilon)$, one update gives

$$\hat{\mathbf{x}} = \mathbf{x}_0 + (1-\epsilon)\big(F_\theta(\mathbf{x}_0, \epsilon, \mathbf{h}_0) - \mathbf{x}_0\big) = \epsilon \mathbf{x}_0 + (1-\epsilon) F_\theta(\mathbf{x}_0, \epsilon, \mathbf{h}_0). \tag{8}$$

We score $\hat{\mathbf{x}}$ against every catalogue item to determine the top-K ranking. There is one source and one integration step, with no token-level decoding, beam search, candidate prefiltering or checkpoint ensembling. This is *beam-free*, not search-free: catalogue scoring costs $O(ND)$ per query, plus top-K selection. With $\epsilon = 10^{-3}$, history-seeded $\mathbf{x}_0$ and one particle, $\hat{\mathbf{x}} \approx F_\theta(\mathbf{x}_0, \epsilon, \mathbf{h}_0)$. Inference is a deterministic query plus dense retrieval, not a density sampler. The relevant differences from a direct retriever are training and representation (§4.5).

Catalogue-grounded SID readout. Let $C[i,:] = (c_{i1}, c_{i2}, c_{i3}, c_{i4})$ store the frozen assignments from Section 3.1. The ranked item IDs and final four-code SID output are

$$(\hat{i}_1, \ldots, \hat{i}_K) = \mathrm{TopK}_{i \in \mathcal{I}} \cos(\hat{\mathbf{x}}, \mathbf{s}(i)), \qquad \widehat{C}_u = \big(C[\hat{i}_1,:], \ldots, C[\hat{i}_K,:]\big). \tag{9}$$

The $O(4K)$ lookup adds no learned parameters, preserves scores and order, and retains duplicate tuples with item IDs. It neither predicts tokens nor inverts the flow; it provides catalogue-valid SIDs for downstream code-based consumers. States and generation paths are continuous, whereas top-K selection and the final SID output remain discrete. Readout correctness is separate from recommendation accuracy. Appendix I studies blockwise nearest-code projection instead: a geometric diagnostic that need not recover the original tuple after item residuals change the representation.

4 Experiments

4.1 Setup

Data. We use six Amazon Reviews 2023 categories (Hou et al., 2024). Scientific, Instrument and Game follow the benchmark preprocessing of Shi et al. (2025); our reproduction and OxygenFlowRec share the cache, split, history length and tokenizer artefact. Three additional categories probe a small catalogue (All Beauty), a sparse 0-core split (Digital Music), and a larger catalogue (Toys and Games). Counts appear in Table 6.

Evaluation. We retain exact-item identities: a hit requires the target item, not merely its SID tuple. Reproduced LLaDA-Rec results use the original SID-level evaluator, which can count collisions

Table 1: Overall performance. †LLaDA-Rec is reproduced with the released code on the same cache, split, sequence length and tokenizer artefact (seed 2024, up to 150 epochs, four generation steps, beam 50); published and reproduced values are both given in Appendix C. The other thirteen baseline rows are quoted from Shi et al. (2025), not rerun. The reproduction uses SID-level evaluation; quoted methods retain their original evaluators, whereas OxygenFlowRec uses exact items, one particle and no beam search. **Bold** and underline denote the best and second distinct values at the displayed precision; ties share a rank.

	Scientific					Instrument					Game				
Method	R@1	R@5	R@10	N@5	N@10	R@1	R@5	R@10	N@5	N@10	R@1	R@5	R@10	N@5	N@10
GRU4Rec	0.0071	0.0184	0.0272	0.0128	0.0156	0.0094	0.0297	0.0453	0.0196	0.0246	0.0149	0.0461	0.0712	0.0307	0.0387
SASRec	0.0063	0.0240	0.0379	0.0152	0.0197	0.0089	0.0331	0.0525	0.0211	0.0273	0.0128	0.0516	0.0823	0.0323	0.0421
BERT4Rec	0.0045	0.0157	0.0264	0.0100	0.0134	0.0065	0.0255	0.0412	0.0160	0.0211	0.0082	0.0315	0.0530	0.0199	0.0267
FMLP-Rec	0.0046	0.0181	0.0300	0.0113	0.0151	0.0086	0.0299	0.0496	0.0193	0.0257	0.0099	0.0395	0.0649	0.0246	0.0328
LRURec	0.0049	0.0169	0.0267	0.0110	0.0141	0.0071	0.0272	0.0431	0.0172	0.0223	0.0134	0.0480	0.0753	0.0308	0.0396
DreamRec	0.0052	0.0184	0.0299	0.0118	0.0155	0.0069	0.0245	0.0423	0.0157	0.0214	0.0125	0.0381	0.0611	0.0253	0.0326
DiffuRec	0.0050	0.0190	0.0310	0.0119	0.0158	0.0077	0.0283	0.0465	0.0179	0.0237	0.0111	0.0425	0.0709	0.0268	0.0359
VQ-Rec	0.0076	0.0248	0.0385	0.0162	0.0206	0.0099	0.0345	0.0532	0.0222	0.0282	0.0150	0.0497	0.0769	0.0325	0.0412
TIGER	0.0084	0.0282	0.0446	0.0183	0.0236	0.0105	0.0359	0.0566	0.0233	0.0300	0.0166	0.0529	0.0823	0.0348	0.0442
TIGER-SAS	0.0067	0.0221	0.0356	0.0144	0.0187	0.0102	0.0342	0.0521	0.0223	0.0280	0.0170	0.0548	0.0847	0.0360	0.0457
LETTER	0.0082	0.0273	0.0423	0.0179	0.0227	0.0114	0.0362	0.0562	0.0239	0.0303	0.0169	0.0552	0.0863	0.0362	0.0462
LC-Rec	**0.0091**	0.0280	0.0434	0.0186	0.0235	**0.0119**	0.0379	0.0587	0.0251	0.0318	0.0165	0.0567	0.0891	0.0366	0.0471
RPG	0.0087	0.0257	0.0395	0.0174	0.0218	0.0118	0.0362	0.0545	0.0241	0.0300	**0.0209**	0.0579	0.0853	0.0397	0.0485
LLaDA-Rec†	**0.0091**	0.0288	0.0444	0.0191	0.0241	**0.0119**	0.0381	0.0580	0.0250	0.0314	0.0195	**0.0610**	0.0938	0.0404	0.0509
OxygenFlowRec	0.0090	**0.0308**	**0.0484**	**0.0198**	**0.0255**	0.0116	**0.0392**	**0.0628**	**0.0252**	**0.0326**	0.0195	0.0603	**0.0940**	**0.0408**	**0.0520**

Table 2: The three categories outside the public benchmark, against the same LLaDA-Rec reproduction (†, Appendix C). Superscripts are relative change (%); both sides use seed 13 for OxygenFlowRec and seed 2024 for the baseline. **Bold** marks the better of the two; the last column counts the seven metrics OxygenFlowRec wins. Toys and Games is the largest category in the paper at 432K users and 162K items.

		Recall@K				NDCG@K			
Data	Method	1	5	10	20	5	10	20	wins
TG	LLaDA-Rec†	0.005691	0.020467	0.031559	0.045516	0.013115	0.016685	0.020200	
	OxygenFlowRec	**0.007192**$^{+26.4}$	**0.025378**$^{+24.0}$	**0.038872**$^{+23.2}$	**0.056634**$^{+24.4}$	**0.016331**$^{+24.5}$	**0.020668**$^{+23.9}$	**0.025139**$^{+24.5}$	**7/7**
DM	LLaDA-Rec†	**0.003425**	0.005847	0.006432	0.006766	0.004832	0.005029	0.005115	
	OxygenFlowRec	0.002757$^{-19.5}$	**0.008103**$^{+38.6}$	**0.010776**$^{+67.5}$	**0.013867**$^{+105.0}$	**0.005264**$^{+8.9}$	**0.006125**$^{+21.8}$	**0.006894**$^{+34.8}$	**6/7**
AB	LLaDA-Rec†	**0.019763**	**0.031621**	**0.043478**	**0.075099**	**0.026727**	**0.030467**	**0.038471**	
	OxygenFlowRec	0.000000$^{-100.0}$	0.007905$^{-75.0}$	0.031621$^{-27.3}$	0.051383$^{-31.6}$	0.004470$^{-83.3}$	0.011988$^{-60.7}$	0.016971$^{-55.9}$	**0/7**

as hits (Appendix F). On identical OxygenFlowRec lists, SID scoring adds at most 7.2×10^{-5} Recall@20 and 31 hit rows per category. This within-model audit does not quantify the change for LLaDA-Rec's lists. Quoted baselines retain their original protocols. We report Recall, NDCG and MRR at small cutoffs and Recall/NDCG at 64 and 512. Each archived OxygenFlowRec run uses one raw checkpoint selected by validation item Recall@20, one deterministic source, one integration step and exact catalogue ranking, without ensembling. Scientific averages seeds 13/42/7; the other categories use seed 13.

Comparison boundaries. Table 1 combines thirteen quoted baseline rows — sequential recommenders (Hidasi et al., 2016; Kang & McAuley, 2018; Sun et al., 2019; Zhou et al., 2022; Yue et al., 2024), diffusion-based recommenders (Yang et al., 2023; Li et al., 2023), and SID-based generative recommenders (Hou et al., 2023; Rajput et al., 2023; Wang et al., 2024; Zheng et al., 2024; Hou et al., 2025) — with our reproduction. LLaDA-Rec uses released defaults: six layers, seed 2024, up to 150 epochs, validation NDCG@1 with beam one, and four generation steps. Test beam is 50 for $K \leq 20$ and 512 for extended cutoffs. The original paper uses four layers on Scientific/Instrument; the reproduced and published values are kept separate (Appendix C). Thus neither layer count nor checkpoint selection is fully matched to our model. Table 1 ends at $K = 10$, the depth of the quoted rows; Tables 14 and 2 give Recall@20.

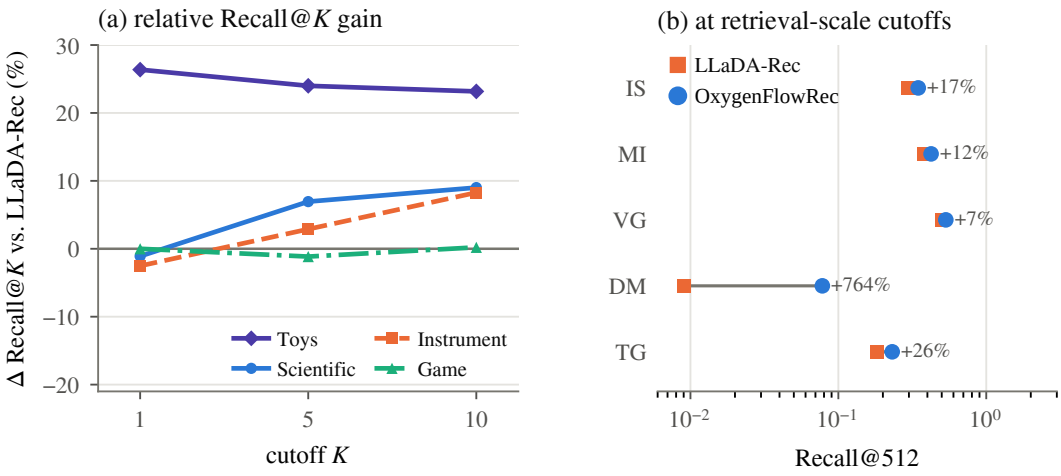


Figure 2: **(a)** Relative Recall@K against the reproduction, with its original SID-level evaluator, at the three cutoffs of the updated results in Table 1; the baseline uses beam 50. Toys uses Table 2. **(b)** Exact-item Recall@512 for OxygenFlowRec and SID-level Recall@512 for the baseline on the five evaluated categories, using archived checkpoints and beam 512 for LLaDA-Rec. Values are in Table 15; panel (b) is a separate evaluation record, not an extension of panel (a)'s curves. On Digital Music the baseline is near-degenerate at this depth (0.0090 at K=512 versus 0.0068 at K=20), so its large relative gain is not comparable with the other categories.

4.2 Overall Performance

OxygenFlowRec has the best displayed value in 11 of the 15 columns in Table 1. Replacing only the reproduced LLaDA-Rec row with its published values (Table 8) reduces this count to three. These are descriptive ranks within different comparison sets, not significance tests or a unified state-of-the-art claim. Table 1 reports the updated evaluations at the reported cutoffs. Wider-cutoff measurements and geometric diagnostics use the separately identified archived checkpoints in Tables 14 and 21; these records are not combined into a single checkpoint result. Ablation and sweep variants are separately trained.

On the archived checkpoints, Recall@20 against our *reproduction* improves by 12.9%, 11.0% and 2.8% on the three public categories (Table 14). Table 2 adds the other categories: Toys improves across all seven metrics, Digital Music on six, and All Beauty on none. Thus the system-level evidence is strongest on Toys; metric wins are correlated observations, not independent tests of significance. Selection and evaluator differences remain relevant to the cutoff-dependent pattern.

At retrieval-scale cutoffs, the frozen OxygenFlowRec checkpoints outperform the beam-512 reproduction on all five evaluated categories (Figure 2). Recall@512 improves by 16.7%, 12.1%, 6.6% and 26.2% on Scientific, Instrument, Game and Toys. These are not the beam-50 configurations timed in Section 4.7; we make no speed-up claim for the beam-512 comparison. Increasing K reuses the catalogue scores but still incurs top-K selection costs.

4.3 Beyond the Standard Benchmark

Three of the six categories are outside the public benchmark. Toys and Games is the strongest evidence in the paper: on 432,264 test users every Recall and NDCG cutoff improves, from +26.4% at Recall@1 to +24.4% at Recall@20 (4,806 additional users with a top-20 hit), with the margin between 23.2% and 26.4% across all seven metrics.

On Digital Music, a 0-core split with more items than users and mean history length 3.3, Recall@20 more than doubles from a small absolute baseline. All Beauty has only 253 test users: the Recall@20 difference corresponds to 19 baseline hits versus 13 for OxygenFlowRec. Insufficient data for the larger model is a plausible explanation, but is not established by a capacity sweep or a significance test. We report this as an observed failure case, not a proven minimum-data threshold.

Table 3: Component ablation (Scientific, seed 13). All values are exact item metrics recomputed from raw logs. Removing the gate or history reconstruction slightly increases Recall@20 while decreasing NDCG@20.

Variant	Epoch	R@10	R@20	N@20	ΔR@20	ΔN@20
OxygenFlowRec (seed 13)	13	0.048524	0.073865	0.032236	—	—
− per-item residual	19	0.023870	0.036050	0.015830	−51.2%	−50.9%
− rollout catalogue CE	12	0.027400	0.040679	0.017232	−44.9%	−46.6%
− split learning rates	17	0.038698	0.060371	0.025259	−18.3%	−21.6%
velocity parameterisation	23	0.045288	0.070138	0.029488	−5.1%	−8.5%
− state-aware query	21	0.045837	0.071295	0.030145	−3.5%	−6.5%
− topology/neighbour losses	15	0.046837	0.071825	0.031117	−2.8%	−3.5%
− history reconstruction	15	0.048524	0.074257	0.032171	+0.5%	−0.2%
− frequency gate	16	0.048858	0.074473	0.032033	+0.8%	−0.6%

4.4 Ablation Study

Table 3 uses Scientific, seed 13, 24 training epochs and a validation-selected checkpoint for each variant. The reference is the same-seed mainline, not the three-seed mean. Removing the residual, rollout CE or learning-rate split reduces Recall@20 by 51.2%, 44.9% or 18.3%, respectively. Residual removal changes both parameter count and item-specific capacity; it does not isolate SID collisions. Removing rollout CE also removes its query-balancing correction, so the row measures the combined rollout-supervision branch.

Removing the gate or history reconstruction gives mixed changes below one percent in aggregate. Broken down by target frequency (Appendix H), the residual trades tail recall for head recall: without it, recall on targets seen two or three times rises from 9 to 33 hits, while targets seen nine or more times lose heavily. The velocity-parameterisation comparison is confounded by different output-head initialisation.

These component ablations quantify dependencies within the full system. The matched controls below separately examine recovery and SID sharing relative to direct catalogue prediction.

4.5 Matched Representation and Training Controls

Table 4 separates representation and training effects on Scientific and Toys. The first four rows cross SID-plus-residual versus item-only representations with the full recipe versus direct catalogue CE. No-recovery and shuffled-SID variants test the recovery loss and code assignments separately. Appendix M gives the protocols.

Table 4: Matched representation and training controls on Scientific and Toys. Full uses the seed-13 reference. The first four rows form the representation × training study. Relative changes use the displayed Full value; all metrics are reported to four decimal places.

	Scientific			Toys		
Variant	R@20	N@20	ΔR@20	R@20	N@20	ΔR@20
Full, SID + residual	0.0739	0.0322	–	0.0566	0.0251	–
Item-only, full recipe	0.0713	0.0311	−3.5%	0.0541	0.0240	−4.4%
Direct CE, SID + residual	0.0713	0.0311	−3.5%	0.0549	0.0244	−3.0%
Direct CE, item-only	0.0694	0.0303	−6.1%	0.0530	0.0235	−6.4%
No recovery	0.0720	0.0314	−2.6%	0.0555	0.0246	−1.9%
Shuffled SID, full recipe	0.0694	0.0303	−6.1%	0.0538	0.0239	−4.9%

Design. The item-only representation is a free $D = 512$ embedding table. The matched variants share the history encoder, query/head, source, data split, training budget and validation item Recall@20 selection rule. Direct CE retains catalogue CE and its query-balancing correction but removes

recovery and the auxiliary objectives. No recovery changes only the recovery coefficient. Shuffled SID permutes complete four-code tuples across items, preserving code-use counts and collision sizes.

SID sharing. With direct CE, the hybrid representation increases Recall@20 from 0.0694 to 0.0713 on Scientific and from 0.0530 to 0.0549 on Toys: relative gains of 2.7% and 3.6%. Under the full recipe, the corresponding gains over the item-only representation are 3.6% and 4.6%. Shuffling the assignments lowers Full Recall@20 by 6.1% and 4.9%, respectively. Together, the direct-CE and full-recipe comparisons show a benefit from SID-indexed sharing beyond a free item table; the shuffled control also shows that the assignment structure matters.

Recovery and the training recipe. Full improves Recall@20 over the no-recovery variant by 2.6% on Scientific and 2.0% on Toys. This contrast isolates recovery within the full recipe. Against direct CE using the same hybrid representation, the gains are 3.6% and 3.1%, which measure the combined recipe rather than recovery alone. Full has the highest Recall@20 and NDCG@20 among these controls on both datasets, although the direct predictors remain close. Textual gains use the named comparator; the table reports changes relative to Full, explaining the different percentage magnitudes.

4.6 Sensitivity and Representation

The equal-budget comparison of learning-rate ratios 0, 0.0625, 0.125 and 0.25 places 0.125 highest, with internal-validation Recall@20 of 0.0930 (Table 24). This is an interior best point among the tested 24-epoch settings. The other runs in Table 7, stopped after 5–13 epochs, remain partial-budget comparisons. Shared-only alternatives underperform the item-residual state (Figure 8a). The depth comparison fixes tokenizer width, not recommender dimension, capacity or cost; $M = 4$ was fixed in advance (Figure 9b). Across Scientific seeds $13/42/7$, Recall@20 is 0.073865, 0.074061 and 0.074296 (sample s.d. 0.000216), all above the baseline's 0.065627.

4.7 Efficiency

On Toys, OxygenFlowRec has 110.9M parameters versus 6.85M for LLaDA-Rec, primarily because of its item table. On one B200 in FP32, cached top-20 retrieval at batch 64 takes 0.133 versus 2.906 ms/user ($21.9\times$); LLaDA-Rec uses four stages and beam 50 (Table 16). This is about 8.5 ms per batch, not single-request latency; batch-one acceleration is $1.88\times$. Appendix G reports memory and uncached costs. These measurements cover retrieval, not a separately timed end-to-end SID-output pipeline, and do not establish an advantage over capacity-matched dense retrieval or other hardware/catalogue settings.

5 Limitations

The matched controls cover Scientific and Toys. Their gains over direct predictors are modest, and paired uncertainty is not reported for these contrasts. Repeated-seed statistics for OxygenFlowRec are available only on Scientific, and All Beauty is a negative result. Item residuals and exact scoring scale with catalogue size; million-item deployment is unvalidated. One-step recovery does not establish conditional-density modelling, diverse generation or multi-step transport. Scratch training refers to the recommender, not its external tokenizer.

6 Conclusion

SID output need not require token-level generation. OxygenFlowRec combines shared SID embeddings, item residuals and one-step recovery with catalogue readout. Its strongest gains over our LLaDA-Rec reproduction occur on Toys at lower batched retrieval cost. Scientific and Toys controls support SID sharing under both objectives and recovery within the full recipe, which improves Recall@20 over hybrid direct CE by 3.6% and 3.1%. Minimal recovery-plus-CE retains most accuracy; alternative sources produce small changes. Near-parity with MIRAGE and failure on All Beauty delimit the gains, supporting this design rather than universal superiority.

AI USE STATEMENT

During manuscript preparation, generative AI tools assisted with drafting, grammar check and polishing. They also assisted with LaTeX and document-processing code, figure-label updates and layout checks. AI-generated suggestions are not treated as independent scientific evidence. The authors retain responsibility for independently validating AI-assisted text, code, figures, citations and scientific claims, and for the final content of the submission.

ETHICS STATEMENT

The reported evaluation is an offline study of existing Amazon Reviews 2023 product-interaction data (Hou et al., 2024), rather than an intervention with users. Public availability does not remove privacy or data-use obligations: reuse and redistribution should respect the source terms and avoid exposing user-level histories.

REPRODUCIBILITY STATEMENT

Section 3 specifies the representation, training and SID readout; Appendices A and L provide the hyperparameters, objective reductions and gradient routes. Data, splits and evaluation conventions are described in Section 4.1 and Appendices B and C. Checkpoint selection, seeds, tokenizer/checkpoint digests and saved-ranking consistency checks are documented in Appendices D and E. Appendix F distinguishes exact-item evaluation from SID matching, while Appendix K separates the reported and archived evaluation records. Hardware, caching and timing conditions appear in Section 4.7 and Appendix G; matched-control budgets, selection rules and comparison boundaries are given in Appendix M.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *ACM Conference on Recommender Systems (RecSys)*, 2016.

Dengzhao Fang, Jingtong Gao, Yu Li, Xiangyu Zhao, and Yi Chang. Escaping the euclidean void: Manifold-informed flow matching for sequential recommendation. *arXiv preprint arXiv:2607.23762*, 2026.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *The Web Conference (WWW)*, 2023.

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. *arXiv preprint arXiv:2403.03952*, 2024.

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 *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, 2018.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Li Li, Mingyue Cheng, Yuyang Ye, Zhiding Liu, and Enhong Chen. FlowRec: Prior-informed flow matching for efficient sequential recommendation generation. *arXiv preprint arXiv:2508.17618*, 2025.

Zihao Li, Aixin Sun, and Chenliang Li. DiffuRec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Feng Liu, Lixin Zou, Xiangyu Zhao, Min Tang, Liming Dong, Dan Luo, Xiangyang Luo, and Chenliang Li. Flow matching based sequential recommender model. *arXiv preprint arXiv:2505.16298*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

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.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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.

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 *ACM International Conference on Information and Knowledge Management (CIKM)*, 2019.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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 *ACM International Conference on Information and Knowledge Management (CIKM)*, 2024.

Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, Bo Long, Robert D. Nowak, Xiaoli Gao, and Hamid Eghbalzadeh. 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *ACM Conference on Recommender Systems (RecSys)*, 2019.

Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2024.

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 *IEEE International Conference on Data Engineering (ICDE)*, 2024.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced MLP is all you need for sequential recommendation. In *The Web Conference (WWW)*, 2022.

A HYPERPARAMETERS

Table 5 lists the full official configuration. All datasets use the same values except where noted.

Table 5: OxygenFlowRec configuration. The optimiser is AdamW (Loshchilov & Hutter, 2019).

Item manifold		**Flow**	
manifold dim D	512	parameterisation	target recovery
code dim d_c	128	field hidden / depth	1024 / 4
codebooks $M \times K_{\mathrm{cb}}$	4×256	field dropout	0.1
residual scale α	0.2	time embedding dim	256
gate floor / ref / power	1.0 / 8.0 / 0.5	source σ	0.03125
gate max	1.0	ϵ	10^{-3}
		train rollout steps	1
		inference ODE steps	1
History encoder		**Optimisation**	
layers / heads / d_h	6 / 8 / 256	optimiser	AdamW
MLP ratio	4	base LR η_0	3×10^{-4}
norm / activation	RMSNorm / SiLU	weight decay	10^{-3}
dropout	0.1	schedule	cosine, warm-up 0.1
max history L	20	epochs	24
history mode	item concat	train batch	256
state query scale γ	2.0	eval / test batch	128 / 64
query saturation scale η	0.5	flow core LR scale	1.0
field gradient scale	0.25	history LR scale	0.25
last-item dropout	0.1	item group κ	$1.5 \rightarrow 0.25$ (ep. 15–18)
		ρ_{SID} / ρ_{res}	0.125 / 1.0
Loss weights		**Retrieval**	
recovery	8.0	temperature τ	0.07
rollout catalogue CE	1.0	candidate restriction	none
query-balancing correction	0.25		
topology (k=20)	5.0	mask seen items	no
neighbour CE (k=8)	0.1	particles	1
history recon. (64 neg., τ=0.1)	10.0	source mode	history hash
long-history (L>10)	0.05, $\rightarrow$0 ep. 20–24	selection metric	item Recall@20
target balance α_{bal} / cap	0.1 / 2.0		

Parameter breakdown (Scientific, 41.19M total). Flow field, time embedding and state-query modules 20.87M; per-item residual table 13.23M; history input projection, 6-layer encoder and norm 6.43M; geometry head 0.39M (frozen throughout); history reconstruction head 0.13M; SID code embeddings 0.13M. Parameters actually updated during flow training: 40.80M.

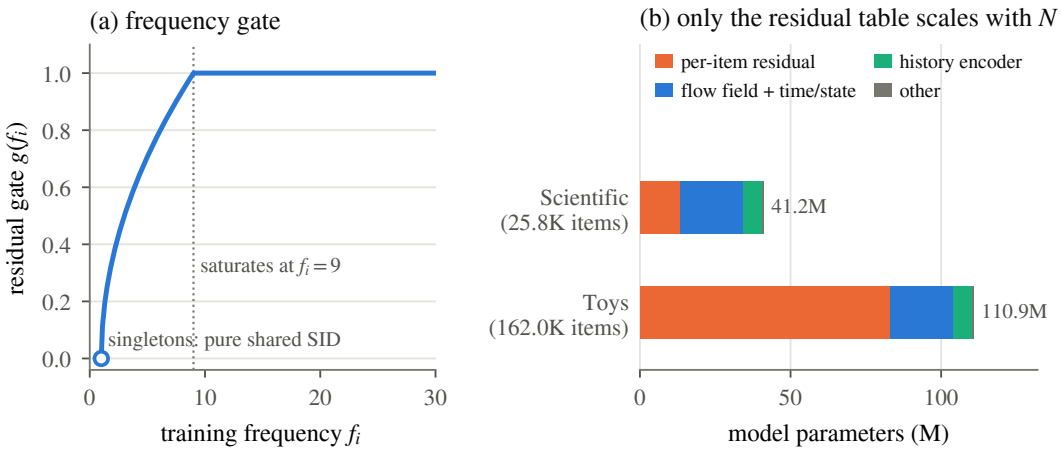


Figure 3: **(a)** The frequency gate of Equation 2. An item seen once carries no residual and uses only shared SID embeddings; the gate saturates at nine occurrences. **(b)** Parameter composition. Only the per-item residual table grows with catalogue size; every other module is identical across datasets.

B Datasets and Learning-Rate Sweep

Table 6: Dataset statistics. The first three rows reproduce the benchmark of Shi et al. (2025) exactly. "Params" is the total number of OxygenFlowRec parameters, which grows with N because of the per-item residual table.

Dataset	k-core	#Users	#Items	#Inter.	Avg. len	Params
Scientific (IS)	5	50,985	25,848	412,947	8.10	41.19M
Instrument (MI)	5	57,439	24,587	511,836	8.91	40.55M
Game (VG)	5	94,762	25,612	814,586	8.60	41.07M
Beauty (AB)	5	253	356	2,535	10.02	28.14M
Music (DM)	0	11,971	30,431	39,782	3.32	43.54M
Toys (TG)	5	432,264	162,035	3,861,886	8.93	110.92M

Table 7: Shared-SID learning-rate ratio (Scientific, peak Recall@20). Δ for the † columns compares runs of unequal length — ρ=0.75 ran only five epochs — and is not a like-for-like comparison. The sweep is scored on an *internal offset-2* split: the validation target is taken two positions from the end of each sequence rather than one, so the official validation and test items are both held out and neither is touched while tuning. † stopped early; peak is a lower bound.

ρ_{SID}	0.125	0.1875†	0.25	0.375†	0.50†	0.75†
Epochs run	24	13	24	13	12	5
Peak R@20	**0.093040**	0.089114	0.086615	0.080874	0.078108	0.068174
Δ	—	−4.2%	−6.9%	−13.1%	−16.1%	−26.7%

C Reproducing LLaDA-Rec

We report our LLaDA-Rec reproduction in Tables 1, 2 and 14, and retain both published and reproduced values in Table 8. The run uses the released repository unmodified: seed 2024 (the repository default), 150 epochs with patience 50, validation metric NDCG@1, 4 generation steps, beam 50 at test and beam 1 at validation, learning rate 3×10^{-3}, train batch 512, 6 layers, $d_{\text{model}} = 256$, 8 heads. Data is the same Amazon Reviews 2023 cache, the same 5-core `last_out` split, the same maximum sequence length of 20, and the same multi-head VQ-VAE artefact that OxygenFlowRec consumes, so tokenisation is identical on both sides.

Table 8 compares the two sets of values. Our reproduction is below the published numbers at every cutoff on all three datasets, by 5.9–7.3% on Scientific, 6.3–6.9% on Instrument and 0.4–3.8% on Game. Shi et al. (2025) state four layers for Scientific and Instrument and six for Game, whereas the released code defaults to six, which is what we ran — and Game, the one dataset where the paper's own configuration is six layers, is also the one where our numbers come closest. The layer/selection study in Table 23 finds higher exact-item Recall@20 with four layers under a common selection rule on Scientific and Instrument. These controlled results identify a layer-count effect in that protocol, but do not fully account for the difference from the published scores. Published and reproduced values remain separate records.

Table 8: LLaDA-Rec as published by Shi et al. (2025) versus our reproduction with the released code. Δ is relative.

		Recall@K			NDCG@K	
Data	Source	1	5	10	5	10
IS	published	0.0098	0.0310	0.0474	0.0203	0.0256
	reproduced	0.0091	0.0288	0.0444	0.0191	0.0241
	Δ	-7.3%	-7.1%	-6.3%	-6.1%	-5.9%
MI	published	0.0128	0.0406	0.0623	0.0268	0.0337
	reproduced	0.0119	0.0381	0.0580	0.0250	0.0314
	Δ	-6.8%	-6.3%	-6.9%	-6.8%	-6.9%
VG	published	0.0203	0.0623	0.0942	0.0415	0.0517
	reproduced	0.0195	0.0610	0.0938	0.0404	0.0509
	Δ	-3.8%	-2.0%	-0.4%	-2.7%	-1.5%

Seed sensitivity of the reproduction. On Instrument and Game we ran the baseline at four seeds (Table 9). The spread is small — a coefficient of variation of 0.8% on Instrument Recall@20 and 0.3% on Game Recall@20 — and seed 2024, the value we report, is the strongest of the four at Recall@5 and Recall@20 on both datasets, so reporting it rather than the four-seed mean is the conservative choice. Only seed 2024 was available on Scientific and at the extended cutoffs of Figure 7, so seed 2024 is used for the baseline throughout the paper.

Table 9: Our LLaDA-Rec reproduction across four seeds. Recall only; the paper reports the seed-2024 row.

Data	Seed	R@1	R@5	R@10	R@20
MI	2024 (reported)	0.011926	**0.038058**	**0.058044**	**0.085952**
	42	0.011682	0.037257	0.056982	0.085465
	100	0.011038	0.036595	0.057679	0.085360
	2025	**0.012030**	0.036822	0.056495	0.084211
	mean	0.011669	0.037183	0.057300	0.085247
VG	2024 (reported)	0.019533	**0.061037**	0.093835	**0.138347**
	42	0.019269	0.060889	0.093708	0.138199
	100	**0.019618**	0.060710	0.092928	0.137618
	2025	0.019438	0.060657	**0.093951**	0.137502
	mean	0.019465	0.060823	0.093606	0.137917

D Training Dynamics and Checkpoint Selection

Every run peaks well before the budget ends. All six official runs reach their best validation Recall@20 between epoch 9 and epoch 16 of 24 and then decline: -4.7% on Scientific, -3.6% on Instrument, -3.1% on Game, -9.9% on Toys, -20.8% on All Beauty and -27.6% on Digital Music by epoch 24. This is why every reported model is the validation-selected checkpoint rather than the last epoch, and why the selected epochs in Table 10 differ across datasets. We have not run an equal-budget, multi-seed comparison of early-stopping strategies, so we do not present this as an additional algorithmic contribution — only as the selection rule we followed.

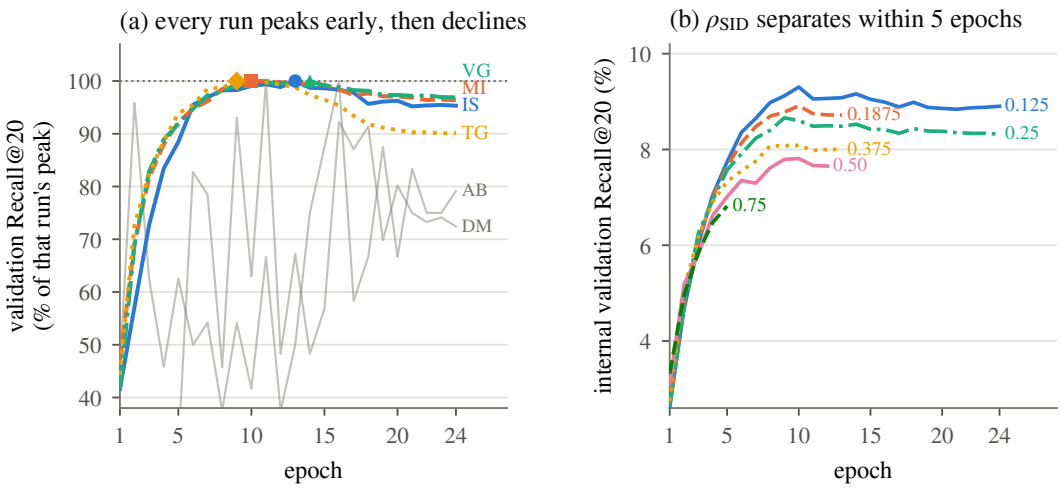


Figure 4: Per-epoch validation Recall@20, read directly from the training logs. **(a)** Official validation, normalised to each run's own peak; markers show the selected epoch. The four larger categories are drawn in colour, All Beauty and Digital Music in grey. Their validation sets contain 253 and 11,971 users; we do not separate finite-sample effects from optimisation variability in these curves. **(b)** Internal offset-2 validation for the shared-SID learning-rate sweep; four of the six runs were stopped early, so the curves end at different epochs.

Learning-rate sensitivity under partial training budgets. Four of the six sweep configurations in Table 7 stop before the full budget, so their observed peaks are lower bounds. Figure 4b shows separation over these observed training intervals. The completed 24-epoch comparison in Table 24 adds the lower ratios 0 and 0.0625 and places 0.125 above both, as well as above 0.25. This equal-budget result supports the selected ratio; the partial trajectories do not establish an ordering for the configurations stopped early.

E Protocol and Integrity Checks

Table 10: The eight frozen mainline checkpoints. SHA-256 digests are truncated to twelve hexadecimal characters; full digests will be released with the checkpoints. Each dataset has its own tokenizer artefact, which is shared with the LLaDA-Rec reproduction on that dataset.

Dataset	Seed	Epoch	Checkpoint SHA-256	SID artefact SHA-256
Industrial and Scientific	13	13	`8a8d83be62e8`	`c6375a3d0523`
Industrial and Scientific	42	11	`7fc264bb3dee`	`c6375a3d0523`
Industrial and Scientific	7	11	`dcb93caa9c60`	`c6375a3d0523`
Musical Instruments	13	10	`95a4898e3236`	`600ae1d970bb`
Video Games	13	14	`a1b8fbb07f15`	`6b742f5dcefd`
All Beauty	13	16	`fda9d913fdac`	`6c3d60b7f05c`
Digital Music	13	11	`3e4186b72299`	`efe576e6c4f0`
Toys and Games	13	9	`4f5c66585ed8`	`16e2b40aa1da`

In Table 3 the history-reconstruction row matches the mainline Recall@10 exactly (2,474 hits). We re-counted from the saved predictions: the two models share only 1,832 of those hits and agree on 18,637 of 50,985 top-1 items, so the equal totals are a coincidence, not a duplicated run.

The archived test numbers come from an independent rerun of frozen checkpoints in which no checkpoint, hyperparameter or decoding parameter was selected. For every artefact we verified checkpoint SHA-256, SID-artefact SHA-256, checkpoint provenance, the test split identifier, single-particle inference and exact catalogue ranking.

Saved item labels match the raw official test targets exactly: $50{,}985/50{,}985$ per seed on Scientific, $57{,}439/57{,}439$ on Instrument, $94{,}762/94{,}762$ on Game. All twelve metrics were independently recomputed from saved rankings, with a maximum deviation from the evaluator of 3.73×10^{-9}. Five reruns reproduced every Recall value exactly and NDCG/MRR to within 2.83×10^{-7}.

We note one accounting point for reproducibility: our logs contain retries and pipeline duplicates, so the number of log files is not the number of independent experiments; the mainline comprises eight frozen checkpoints (three Scientific seeds plus one each for the other five datasets).

F SID Collisions, Metric Semantics and Fixed SID Output

Four-code catalogue audit. We directly counted tuple multiplicities in the six mainline tokenizer artifacts, verifying each file's SHA-256 against its frozen checkpoint provenance. An item is counted as colliding if at least one other item has the identical full four-code tuple. Table 11 reports this item-weighted catalogue quantity, not an interaction-weighted test collision rate. No model was retrained or re-evaluated to produce these counts.

Table 11: Exact four-code SID collisions in the mainline catalogues. "Affected" counts all items belonging to a non-singleton tuple group; it does not count only the extra members beyond the first item.

Category	Items	Distinct tuples	Affected	Largest group
Scientific	25,848	25,792	108	3
Instrument	24,587	24,525	120	3
Game	25,612	25,413	361	7
All Beauty	356	356	0	1
Digital Music	30,431	30,172	417	37
Toys and Games	162,035	160,892	2,080	11

Only $108/25{,}848 \simeq 0.42\%$ of Scientific catalogue items are in colliding four-code groups. This does not bound the interaction-weighted metric effect, since these items can have different target frequencies. It does, however, show why the large residual ablation cannot simply be identified with collision resolution: item-specific capacity can change the ranking of noncolliding items as well.

The item-only and shuffled-assignment controls in Section 4.5 separately examine the contribution of SID sharing and assignment structure; Appendix M specifies their capacity and training boundaries.

Discrete SID recommenders are conventionally evaluated by whether the generated code tuple matches the target's tuple. When several catalogue items share a tuple, this counts as a hit something that an item-level evaluation would not. The gap is measurable: on the same 2-SID OxygenFlowRec run on Scientific, SID-level Recall@20 is 0.074669 while exact item-level Recall@20 is 0.070452, a 6.0% inflation. Comparing the SID-level number of a shallow tokenizer against the item-level number of a deeper one would reverse the conclusion of Table 17. That 6.0% is specific to a two-code tokenizer: under the four-code mainline assignment the same inflation is 0.04–0.13% where it is non-zero (Table 12), so the size of the metric-semantics gap is itself a function of tokenizer depth.

Throughout this paper OxygenFlowRec is scored at the item level and LLaDA-Rec at the SID level, i.e. in the direction that disadvantages our method.

Catalogue-grounded SID output: protocol. The final SID output in Section 3.6 reads $C[\hat{i}_j, :]$ from the frozen assignments while retaining item IDs and their ranking (Figure 1). We verified this preservation, and then asked what happens if the SID tuples are instead allowed to *modify* the list. Eight frozen mainline runs were replayed — Scientific at seeds $13/42/7$ and the other five categories at seed 13 — covering $749{,}644$ query-seed evaluation rows and $14{,}992{,}880$ top-20 item positions. The audit reads archived predictions and the frozen SID assignment only: no training, no model forward pass, no GPU. Four rules were fixed in a protocol file before the run: ORIGINAL (keep the item list, attach the lookup), DEDUP (stable-deduplicate by tuple, keeping the first item, without back-filling from rank 21 onward), and two BUCKET RE-RETRIEVAL rules that discard item identity, visit the tuples in order of first appearance, and re-expand each bucket — breaking ties by item ID in one variant and by training frequency in the other — truncating back to 20 items. Bucket ordering

uses only tuples, item IDs and training frequencies recomputed from the training prefix of each sequence (validation and test items excluded); it never consults the test target and never recomputes the flow cosine.

Readout preserves the item ranking. This holds by construction; the replay confirms the implementation matches it, with recomputed item metrics within 4.5×10^{-9} of the archive.

SID-level scoring is a looser criterion. Table 12 scores the *same* list two ways. Because several catalogue items can share a tuple (Table 11), recommending a neighbour that shares the target's tuple counts as a SID hit but not as an item hit. Across all eight runs this adds 45 query-seed rows in total, at most 31 in any run (Toys). For Scientific at seed 13 it adds exactly one row, which independently reproduces the single-row gap reported in Appendix I. Instrument shows the subtler form: HR@20 is unchanged, yet SID NDCG and MRR are marginally higher, because a same-tuple item sits above the target and moves the first tuple match earlier. None of this is a recommendation improvement, and we do not report it as one: our main comparisons score OxygenFlowRec at exact-item level, while SID-level diagnostics are reported separately. Exact-item scoring is the stricter convention and disadvantages us relative to the SID-level reproduction (§4.1).

Table 12: The same top-20 list scored at item level and at SID level. ΔR is the Recall/HR gap in units of 10^{-5}. The last column counts query-seed rows whose target tuple was retrieved while the target item was not (Scientific lists seeds $13/42/7$ separately, as counts do not average). Scientific metrics are the three-seed mean; the rest are seed 13. Digital Music retains its 0-core filtering and is not comparable to the 5-core categories.

		Recall/HR@20			NDCG@20		SID-only
Category	Rows	item	SID	ΔR	item	SID	rows
Scientific	50,985	0.074074	0.074133	+5.88	0.031904	0.031921	1 / 4 / 4
Instrument	57,439	0.095406	0.095406	+0.00	0.040778	0.040780	0
Game	94,762	0.142198	0.142251	+5.28	0.062112	0.062133	5
Toys	432,264	0.056634	0.056706	+7.17	0.025139	0.025171	31
All Beauty	253	0.051383	0.051383	+0.00	0.016971	0.016971	0
Digital Music	11,971	0.013867	0.013867	+0.00	0.006894	0.006894	0

Letting SIDs rewrite the list costs hits. Table 13 and Figure 5 report the three rules that do change the output. The mechanism is visible in the figure: the catalogue collision rate is small everywhere (0–1.4%), but bucket re-retrieval rewrites 15–16% of Game and Toys lists, because re-expanding a bucket displaces items that the continuous ranking had placed deliberately. The cost is 69 net item hits on Game and 157 on Toys (159 under the item-ID tie-break). Deduplication is gentler — it only removes — and costs at most 14 hits. All Beauty is the clean control: its catalogue has no four-code collisions (Table 11), so every rule is a bit-exact no-op. Scientific is the one case where Recall@20 rises, by 1.31×10^{-5}, but its NDCG@20 falls, so not even there is the rewrite a uniform improvement. Digital Music shows the converse: Recall@20 is unchanged while NDCG@20 moves, so an unchanged Recall alone does not license the conclusion that the list is unchanged.

Expanded candidate coverage is not a Recall improvement. A tempting variant is to expand every tuple bucket touched by the original top-K and report the resulting coverage. That number is an identity, not a capability: if the target's tuple appears in the original top-K, the expanded set necessarily contains the target item, so coverage equals the original SID HR@K exactly. The expansion is also small — mean expanded size 20.02 to 20.27 items, maximum 52 on Toys. Reporting it against a 20-item budget would compare different budgets, and we did not recompute the flow cosine over the expanded candidates, so no selection or ordering among them is justified.

Caveats. This is a post-hoc offline diagnostic on an already-exposed test split, not a new blind evaluation, and we report no significance tests; the 69- and 157-hit differences are raw counts on single seeds. The three Scientific seeds are repeated runs over the same test queries and are not three times as many users. The four rules were fixed in advance, but they are simple rules: these results bound what *these* post-processing schemes achieve and do not show that every SID-based

Table 13: Effect of SID-based post-processing at a fixed budget of 20 items. Δ columns are differences from the original list in units of 10^{-5}; "net hits" counts exact item hits gained minus lost at seed 13. Bucket re-retrieval uses the training-frequency tie-break; the item-ID variant differs only on Toys (ΔR −36.78, net −159) and Scientific (ΔN −0.18). Scientific metric columns are the three-seed mean.

		ΔRecall@20		ΔNDCG@20		net hits	
Category	R@20	dedup	bucket	dedup	bucket	dedup	bucket
Scientific	0.074074	−0.65	+1.31	−0.16	−0.19	−1	0
Instrument	0.095406	−1.74	−3.48	−0.49	−1.25	−1	−2
Game	0.142198	−1.06	−72.81	−0.27	−25.97	−1	−69
Toys	0.056634	−3.24	−36.32	−0.88	−14.52	−14	−157
All Beauty	0.051383	0	0	0	0	0	0
Digital Music	0.013867	0	0	0	−0.04	0	0

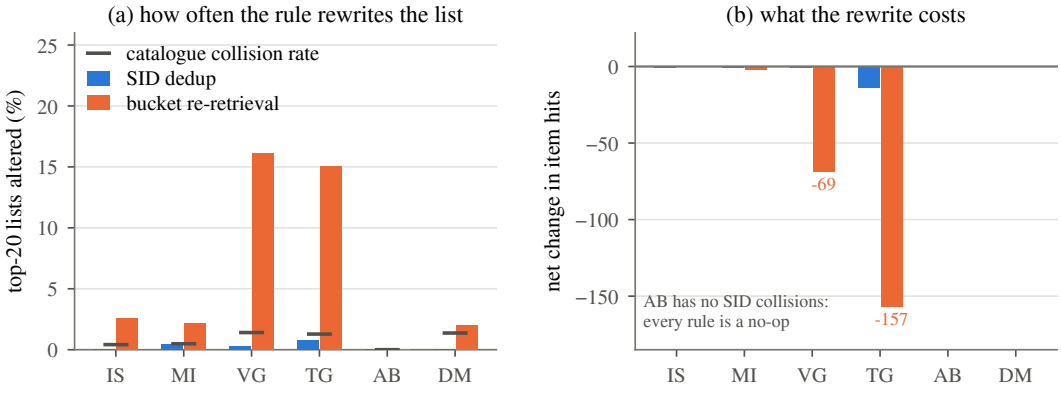


Figure 5: SID-based post-processing at a fixed budget of 20 items, seed 13. **(a)** Share of top-20 lists the rule rewrites; grey dashes are the catalogue four-code collision rate (Table 11). A collision rate near 1.4% produces a 16% rewrite rate on Game, because one collided item in a list triggers a bucket expansion that displaces others. **(b)** Net change in exact item hits. All Beauty has no collisions, so every rule is a no-op there. Bucket re-retrieval uses the training-frequency tie-break.

candidate-expansion or re-ranking method fails. In particular, a method that rescored expanded candidates with the model itself is untested here.

G Extended Cutoffs, Additional Categories, SID Depth and Latency

Tables 14 and 15 extend archived evaluations, not the updated Instrument/Game reported-cutoff records in Table 1. Tables 16 and 17 report the corresponding inference-cost and joint tokenizer/model-configuration comparisons. These scopes should be kept separate when interpreting accuracy, capacity and efficiency.

Table 14: Archived $K = 20$ comparison. These are completed frozen-checkpoint evaluations, not extensions of the updated Instrument/Game rows in Table 1. Scientific is the three-seed mean; Instrument and Game use seed 13. The reproduction uses its original SID evaluator. Δ uses the displayed six-decimal values.

Dataset	Method	Recall@20	NDCG@20
IS	LLaDA-Rec	0.065627	0.029394
	OxygenFlowRec	**0.074074**	**0.031904**
	Δ	$+12.9\%$	$+8.5\%$
MI	LLaDA-Rec	0.085952	0.038389
	OxygenFlowRec	**0.095406**	**0.040778**
	Δ	$+11.0\%$	$+6.2\%$
VG	LLaDA-Rec	0.138347	0.062115
	OxygenFlowRec	**0.142198**	0.062112
	Δ	$+2.8\%$	-0.0%

Table 15: Extended cutoffs on frozen checkpoints (no retraining, no ensembling). Our LLaDA-Rec uses four decoding stages and beam 512 in these runs, not the beam 50 configuration used for the top-20 latency measurements. OxygenFlowRec Recall/NDCG are exact item metrics; LLaDA-Rec values are SID-level, which favours the baseline. All Beauty is omitted (not evaluated at these cutoffs).

Data	Method	Recall@64	Recall@512	NDCG@64	NDCG@512
IS	LLaDA-Rec	0.118525	0.297146	0.039502	0.063042
IS	OxygenFlowRec	**0.138766**	**0.346808**	**0.044248**	**0.071645**
MI	LLaDA-Rec	0.157767	0.377809	0.052082	0.081210
MI	OxygenFlowRec	**0.174254**	**0.423632**	**0.055828**	**0.088830**
VG	LLaDA-Rec	0.240845	0.499947	0.081761	0.116354
VG	OxygenFlowRec	**0.252876**	**0.532724**	**0.083253**	**0.120619**
DM	LLaDA-Rec	0.007268	0.009022	0.005207	0.005447
DM	OxygenFlowRec	**0.022555**	**0.077938**	**0.008541**	**0.015672**
TG	LLaDA-Rec	0.081085	0.183205	0.027061	0.040619
TG	OxygenFlowRec	**0.098643**	**0.231238**	**0.033151**	**0.050623**

Table 16: Amortised inference latency per user on Toys and Games: one B200, FP32, top-20, 20 warm-up and 100 timed iterations. LLaDA-Rec uses 4 generation steps and beam 50; OxygenFlowRec uses one ODE step and one particle. "Cached" keeps the $D \times N$ catalogue matrix resident (316 MiB); "uncached" rebuilds it per batch.

Batch size	LLaDA-Rec (ms/user)	OxygenFlowRec cached (ms/user)	OxygenFlowRec uncached (ms/user)	Speed-up (cached)	Speed-up (uncached)
1	13.945	7.432	9.277	$1.88\times$	$1.50\times$
8	3.587	0.997	1.205	$3.60\times$	$2.98\times$
16	3.210	0.487	0.577	$6.59\times$	$5.56\times$
32	3.019	0.256	0.299	$11.81\times$	$10.11\times$
64	2.906	0.133	0.155	$\mathbf{21.85}\times$	$18.71\times$

Table 17: SID depth on Scientific, seed 13. LLaDA-Rec is scored at the SID level, OxygenFlowRec at the exact item level.

	LLaDA-Rec				OxygenFlowRec			
M	R@5	R@10	R@20	N@20	R@5	R@10	R@20	N@20
2	0.023850	0.038894	0.058900	0.025312	0.028871	0.044856	0.070452	0.030077
4	0.028812	0.044425	0.065627	0.029394	**0.031264**	**0.048524**	**0.073865**	**0.032236**
8	0.025772	0.040718	0.060371	0.026166	0.028714	0.045327	0.070648	0.030138
16	0.018299	0.028361	0.040365	0.018206	0.025302	0.040208	0.061979	0.026436

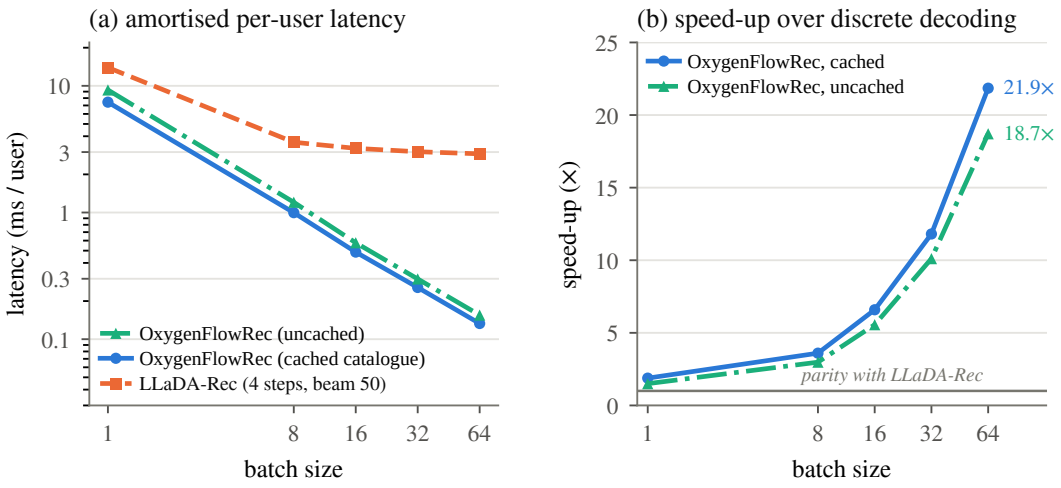


Figure 6: Inference cost on Toys and Games (one B200, FP32, top-20, 20 warm-up and 100 timed iterations). **(a)** Amortised per-user latency, log–log. **(b)** Speed-up over 4-step, beam-50 discrete decoding. Exact values in Table 16.

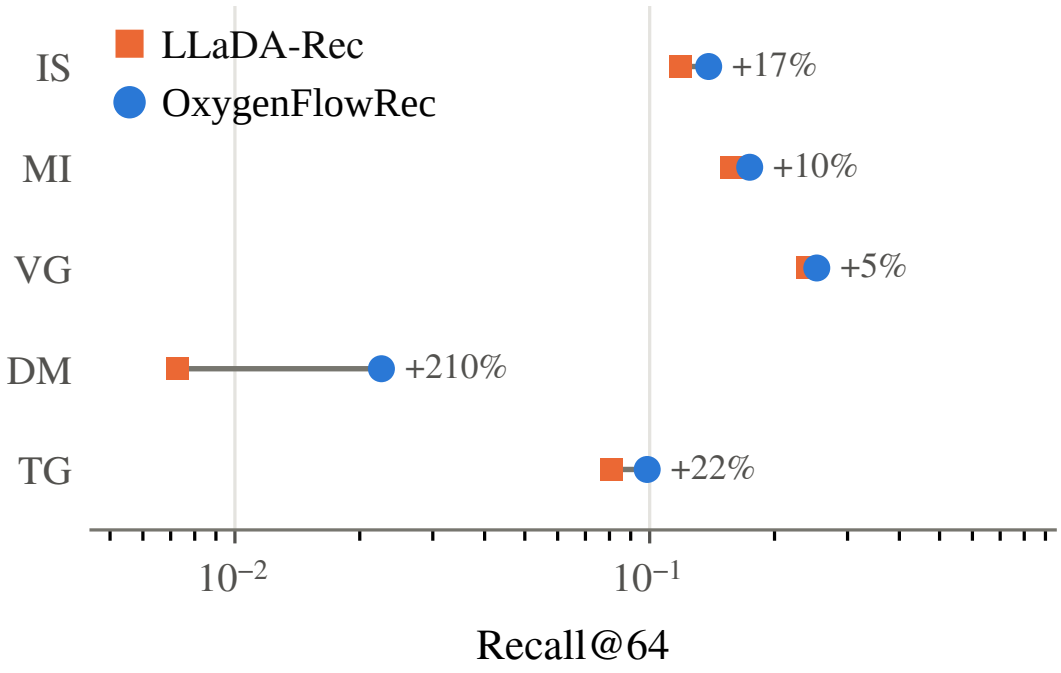


Figure 7: Recall@64 on the five datasets for which we have a reproduction (Recall@512 is in Figure 2b). OxygenFlowRec is higher on every dataset; percentages are relative. Digital Music is a 0-core split on which the discrete baseline is very weak in absolute terms. Exact values in Table 15, Appendix G.

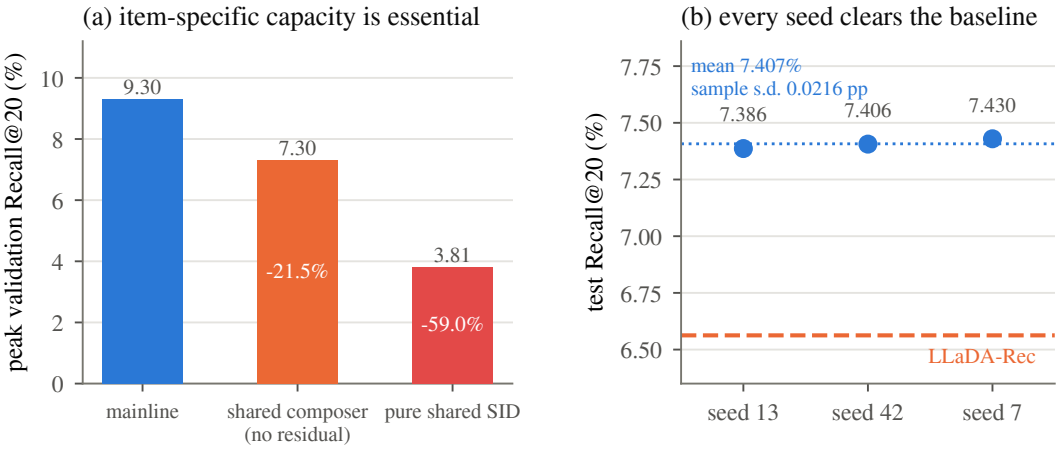


Figure 8: **(a)** Representation ablation on the internal validation split: the mainline (frequency-gated per-item residual, no composer) against a shared nonlinear SID composer with no residual, and a pure shared-SID manifold. **(b)** Scientific test Recall@20 for the three seeds, against our LLaDA-Rec reproduction.

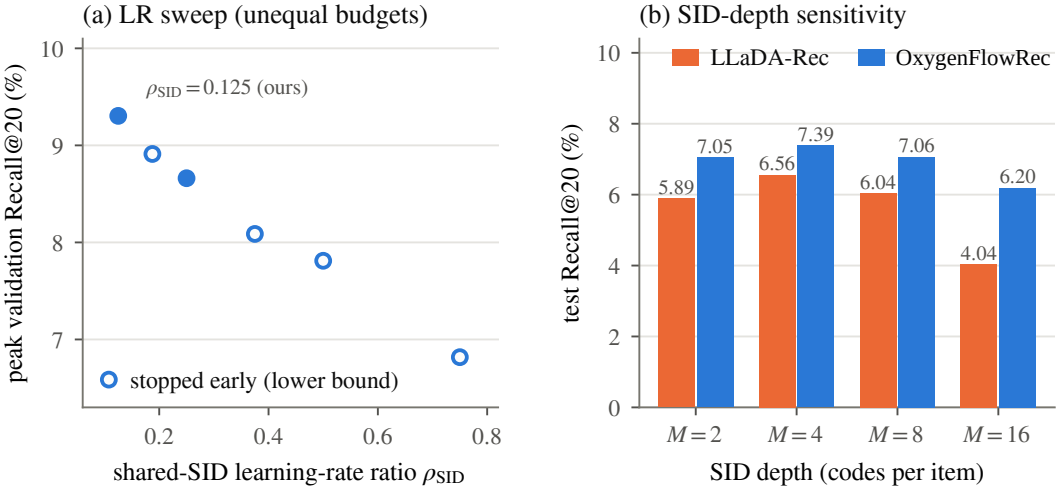


Figure 9: **(a)** Peak internal-validation Recall@20 against the shared-SID learning-rate ratio ρ_{SID} at fixed ρ_{res}=1.0; hollow markers ran fewer than 24 epochs, so those peaks are lower bounds. **(b)** Test Recall@20 against SID depth on Scientific, seed 13; OxygenFlowRec is scored at the item level and LLaDA-Rec at the SID level. Exact values in Tables 7 and 17.

H WHERE THE RANKING SUCCEEDS AND FAILS

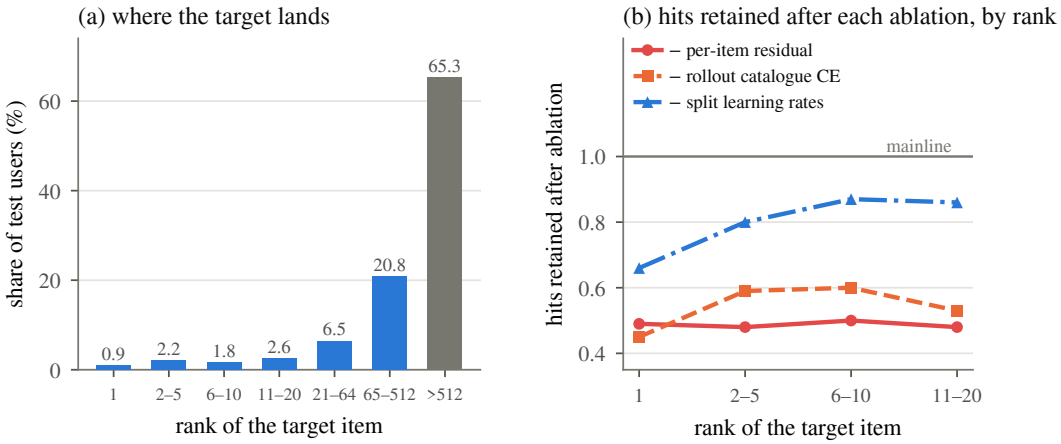


Figure 10: Decoded from the frozen Scientific predictions. **(a)** Where the target item lands in the ranked list, averaged over the three seeds; the top-512 predictions were saved, so everything beyond that is a single censored bucket. **(b)** For each rank bucket, the fraction of the mainline's hits that survive an ablation (seed 13). A rising curve means that a smaller fraction of rank-one hits survives than of lower-ranked hits; its shape alone does not identify the mechanism.

Panel (a) is the unflattering but useful view of the task. On Scientific the target is ranked first for 0.90% of users, reaches the top 20 for 7.4%, the top 512 for 34.7%, and is not recovered at all within 512 candidates for the remaining 65.3%. The bucket shares recompose exactly to the Recall values reported in the main text, which is how we checked the decoding.

In panel (b), removing the **per-item residual** retains 0.48–0.50 of the mainline's hits in every bucket. Removing the **rollout catalogue CE** retains 0.45–0.60 of the hits, lowest at rank 1 and without a clear trend over the remaining buckets. Restoring equal rather than **split learning rates** gives a different pattern: 86% of hits survive at ranks 11–20, versus 66% at rank 1. In this run, the optimisation change is associated with greater relative hit retention at ranks 11–20 than at rank 1. These conditional hit-retention statistics do not identify a capacity or geometric mechanism, and do not include hits newly acquired by an ablated model.

The ablations are single-seed, and the rank-1 bucket holds only about 480 of the 50,985 users. Without paired uncertainty estimates or repeated ablations, the apparent rank dependence is exploratory. Panel (a) is censored at 512 because that is the depth we saved, not because the model assigns no rank beyond it.

By target frequency. The SID partition is meant to help the items the residual cannot: rarely seen ones. We test this directly by splitting the strict-test users of each category by how often their target item occurs in the training prefix, using the saved predictions of the frozen mainline and of two Scientific ablations (Figure 11, Table 18). Recall concentrates heavily on popular targets — on Scientific it is 24.8% for targets seen more than 50 times and under 1% below nine — so these rates are small, and we report hit counts alongside them.

Removing the item residual, which leaves every item on its shared SID representation, *improves* recall on the rarest targets and costs it everywhere else. On targets seen two or three times it recovers 33 hits against the full model's 9 (z=3.7); on singletons 14 against 6 (z=1.8, suggestive only). From nine occurrences upward the full model is far ahead (z from −5.1 to −29.1). The residual therefore trades tail recall for head recall, with the crossover near four to eight occurrences. This is direct evidence that SID sharing carries real signal for rare items, and also that the full model does not keep it.

Panel (b) shows a dip at two or three occurrences in every large category, deepest on Toys, where such targets are recovered 7 times in 24,861 (0.03%) against 104 in 6,633 (1.57%) for singletons. Singletons have $g(f_i)$=0 and sit exactly on their SID representation; targets seen twice or three times

receive $g \in [0.35, 0.5]$ and a residual fitted to one or two examples, which on this evidence is worse than none in this single-seed comparison. This motivates a validation-set study of the gate threshold, not a replacement threshold chosen from test buckets. The frequency gate's own effect on singletons (6 hits against 1 without it) rests on too few hits to interpret.

All numbers are single-seed.

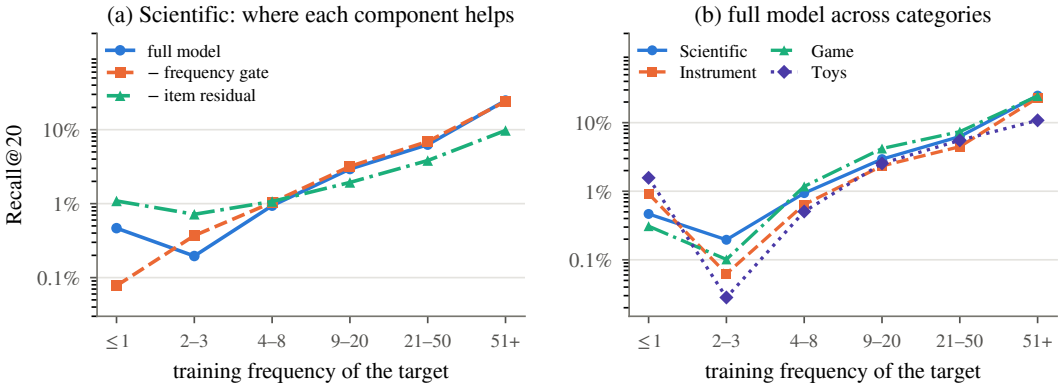


Figure 11: Recall@20 by the training frequency of the target item, log scale, seed 13. **(a)** Scientific: the full model against the ablations without the frequency gate and without the item residual. **(b)** The full model on the four categories large enough to populate every bucket.

Table 18: Hits and Recall@20 by target training frequency, Scientific, seed 13. n is the number of test users in the bucket.

Target frequency	≤ 1	2–3	4–8	9–20	21–50	51+
n	1,287	4,596	12,248	12,255	9,882	10,717
OxygenFlowRec	6 (0.47%)	9 (0.20%)	115 (0.94%)	360 (2.94%)	620 (6.27%)	2656 (24.8%)
− frequency gate	1 (0.08%)	17 (0.37%)	127 (1.04%)	391 (3.19%)	684 (6.92%)	2577 (24.0%)
− item residual	14 (1.09%)	33 (0.72%)	131 (1.07%)	237 (1.93%)	378 (3.83%)	1045 (9.75%)

I Nearest-Code Projection as a Geometric Diagnostic

The catalogue-grounded readout in Section 3.6 returns the frozen SIDs of ranked items. This appendix studies a different operation: blockwise nearest-code projection of a continuous vector. It is a geometric diagnostic, not an inverse of the full item representation or part of the retrieval pipeline. Since the base state concatenates M code vectors scaled by $1/\sqrt{M}$, split $\mathbf{x} \in \mathbb{R}^D$ into M contiguous blocks of $d_c = 128$ dimensions and match each block against the $K_{\text{cb}} = 256$ rows of its codebook:

$$\hat{c}_m(\mathbf{x}) = \arg\min_{k \le K_{\text{cb}}} \left\| \sqrt{M}\, \mathbf{x}_{[md_c:(m+1)d_c]} - E_m[k] \right\|_2, \tag{10}$$

This adds no learned parameters but requires $O(MK_{\text{cb}}d_c)$ work per vector. Unlike $C[i,:]$, the projected tuple need not equal the original SID after item-specific residuals change the geometry. Recovery percentages below measure agreement with the frozen assignments; they are not lookup success rates or recommendation metrics. All numbers are Scientific, seed 13, using the frozen mainline checkpoints of Appendix E.

Nearest-code agreement decreases during training. With the residual switched off, (10) recovers all four digits for 100% of the 25,848 items, in both ℓ_2 and cosine — the arithmetic identity holds, which also validates the block ordering and token offsets used throughout this section. On the full state $\mathbf{s}(i)$ at the selected checkpoint, it recovers all four digits for only 15.1% of items (17.9% under cosine). Figure 12(a) shows the transition: nearest-code agreement is 100.0% after two epochs, 82.7% at epoch 5, 15.1% at the selected epoch 13, and settles at 12.0%. Over the same epochs the median ratio $\|\alpha g(f_i)\mathbf{r}_i\| / \|\mathbf{s}_{\text{base}}(i)\|$ climbs from 0.17 to 0.51, and validation Recall@20 rises to its peak. The two move together over training; we do not claim that one causes the other.

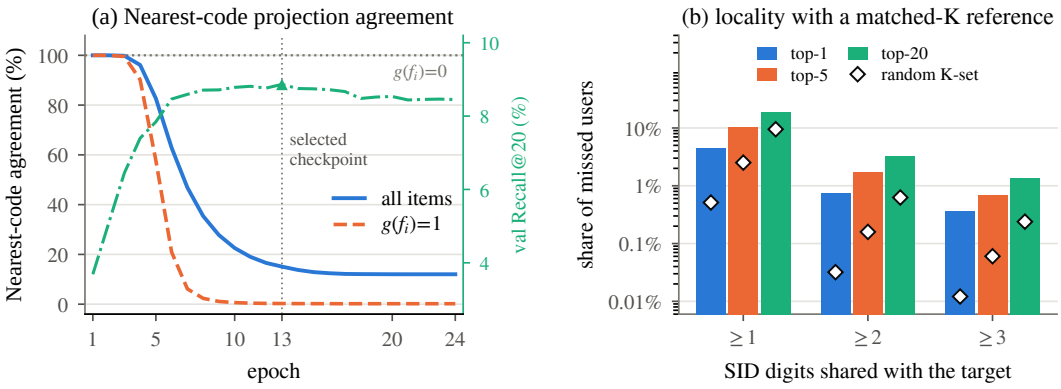


Figure 12: **(a)** Nearest-code projection agreement over training, Scientific, seed 13. Blue: all items. Orange: items at gate saturation $g(f_i)=1$. The dotted line at 100% is the $g(f_i)=0$ items, which stay exact at every epoch by construction. Green, right axis: validation Recall@20; the triangle marks the selected checkpoint. **(b)** For the model-miss users at each K, bars show the fraction with at least one retrieved item matching the target in the first 1, 2 or 3 codebook positions. Open diamonds show the exact expectation for K distinct uniform draws from the same eligible catalogue, averaged over the same users. Each bar has its own matched-K reference; the positions are parallel codebooks, not a coarse-to-fine hierarchy. Log scale.

Table 19: Nearest-code projection agreement by frequency gate, Scientific, seed 13, epoch 13, ℓ_2 matching. f_i is the training count; $g(f_i)$ is the gate of (2). Agreement is the fraction of items for which (10) returns all four true digits. The last column is the median of $\|\alpha g(f_i)\mathbf{r}_i\|/\|\mathbf{s}_{\text{base}}(i)\|$ within the bucket. The buckets partition the 25,848 non-padding items.

Gate bucket	Items	Median f_i	Projection agr.	Median residual ratio
$g = 0$ ($f_i \leq 1$)	437	1	100.00%	0.000
$0 < g \leq 0.5$	2,549	3	72.58%	0.357
$0.5 < g \leq 0.9$	11,206	5	14.03%	0.447
$0.9 < g < 1$	1,592	8	1.07%	0.531
$g = 1$ ($f_i \geq 9$)	10,064	14	0.26%	0.590

Projection agreement varies with the frequency gate. Table 19 stratifies by the frequency gate. Items the gate excludes keep perfect projection agreement at every epoch: they have no residual, so (10) recovers their shared-code indices by construction. Across the gate buckets, agreement decreases as the relative residual magnitude increases, reaching 0.26% at gate saturation. Residuals can therefore change nearest-code assignments without changing SID-indexed parameter sharing. This association does not establish a causal connection to recommendation gains.

Projection agreement varies across codebooks. At epoch 13 the per-codebook agreement rates are 70.5%, 51.2%, 55.4% and 19.3%. The tokenizer uses M *parallel* codebooks, so these positions carry no hierarchy and the ordering is not coarse-to-fine; the values are in fact not monotone in the index. The residual can vary across all 512 coordinates; it is not restricted to selected codebook blocks. This does not impose isotropy on the learned residuals. The recovery rates describe block-dependent learned displacements, but do not identify their cause.

Locality under a candidate-budget-matched random comparison. It does not follow that retrieval has left the partition. We ask, for the 50,985 strict-test users, how many leading SID digits the retrieved items share with the target, *excluding users where the target was retrieved*. Unconditional four-digit matches are dominated by exact-target hits, with a small additional contribution from SID

collisions, rather than isolating locality.[1] Figure 12(b) compares each top-K list with a uniformly sampled set of K distinct items from the same non-padding catalogue, using the same model-miss users at each K. The archived configuration does not mask history items. If $n_{u,d}$ of the N eligible items share the target's first d codebook positions, the random probability of at least one match is $1 - \binom{N-n_{u,d}}{K} / \binom{N}{K}$. We compute this probability for each user and then average; it is not a single-item match probability.

Among the $47{,}219$ model-miss users at $K = 20$, the observed first-position match rate is 18.68%, versus a random-set expectation of 9.51% ($1.96\times$). Matching the first two or three positions gives 3.28% versus 0.63% ($5.24\times$), and 1.35% versus 0.24% ($5.65\times$), respectively. Conditioning the random set additionally on missing the exact target gives $1.98\times$, $5.98\times$ and $8.35\times$; we retain the same-catalogue comparison as the primary reference. Across all users, the top-20 list occupies 11.89 distinct first-position cells on average, versus the exact random-set expectation of 19.13. These are descriptive locality statistics, not estimates of the causal benefit of SIDs.

What this means for the design. The item-state diagnostic motivates catalogue-grounded SID readout rather than interpreting independent nearest-code projection as an exact decoder. At the selected checkpoint, projected tuples disagree with the frozen assignments for about 85% of items and 99.7% at gate saturation. These are item-state reconstruction errors, not direct-query SID recommendation results. Retrieval neighbourhoods nevertheless show enrichment relative to matched-budget random sets, consistent with retained partition structure despite residual-induced shifts. After ranking the full SID–residual vectors, fixed lookup returns the original tuples regardless of these shifts. This single-run diagnostic does not establish that a separately trained discrete SID decoder would fail.

Caveats. This is one dataset and one seed; we did not repeat it on the other five categories. The projection rule in (10) is a natural one but not the only one — a learned probe, or a read-out that accounted for the residual subspace, could recover more. The locality statistics depend on the tokeniser's own partition quality, which we did not vary.

J Qualitative Examples

Table 20 shows what the retrieved lists look like. The three users are taken from the frozen Scientific checkpoint by rank bucket — target first, target inside the top 20, target outside it — and the saved rankings were aligned to the raw user sequences before selection (every one of the $50{,}985$ rows matches, so the examples correspond to the users shown).

Example (b) illustrates the distinction between semantic similarity and exact-item accuracy. The user has bought four Command adhesive hanging products, the target is a fifth, and all five retrieved items are Command hanging products — yet the exact variant sits at rank 13. This example is consistent with higher recall at broader cutoffs, but it does not establish that flow geometry explains the aggregate gains. In example (c), the retrieved set is broadly coherent with a history of electrical and hand tools, but the target — a brass pneumatic fitting — is not in it. These selected cases illustrate outcomes rather than provide a causal analysis.

[1] As a check, the unconditional four-digit rate at $k = 20$ is 7.3884% against a Recall@20 of 7.3865%; the gap is one user, an item sharing a colliding SID with its target. The independent replay in Appendix F recovers the same single row.

Table 20: Three qualitative examples from the frozen Scientific checkpoint (seed 13, epoch 13), one drawn from each of three rank buckets. Titles are truncated. Only the four most recent history items are shown. These are illustrations, not a systematic study: three hand-picked users cannot establish a behavioural claim.

(a) Target ranked first. History is entirely fasteners; the top of the list stays in that neighbourhood.

History	Racer's Speed Shop Mill 3/16" Large Head . . .
	1/4-20 x 1-1/2" Button Head Socket Cap . . .
	Vermont American 20068 1/4-Inch 20 High . . .
	3/8"-16 T-Nuts (100 Pack), Zinc-Plated Steel . . .
Target	Bolt Dropper 1/2" Stainless Flat Finish . . .
Top-1	Bolt Dropper 1/2" Stainless Flat Finish . . . ← target
Top-2	5/16"-18 Stainless Hex Nut (100 Pack), by . . .
Top-3	3/8-16 Nylon Insert Hex Lock Nuts, Stainless . . .
Top-4	No. 10-24 Nylon Insert Lock Nut (100 Pieces) . . .
Top-5	Hillman 830502 Flat Washer, Stainless-Steel . . .

(b) Target ranked 13th. Every one of the top five is a Command adhesive hanging product, i.e. the endpoint lands in the right region and the exact variant is what is missed.

History	Command Dorm Room Essentials - Decorate Your . . .
	Command Refill Strips, White, 8 Small, 4 . . .
	Command 17204-EF 07248000796 CMND PIC Hang . . .
	Command 17205-16ES 3M Small Picture Hanging . . .
Target	Command PH206-12NA Large Picture Hanging . . .
Top-1	Command Photo Hangers, White, Easy On, Easy . . .
Top-2	Command 17204-EF 07248000796 CMND PIC Hang . . .
Top-3	Command Poster Strips, Damage Free Hanging . . .
Top-4	Command Medium Picture Hanging Strips, Damage . . .
Top-5	Command Dorm Room Essentials - Decorate Your . . .

(c) Target outside the top 20. The retrieved set is still tools and electrical supplies, consistent with the history, but the brass pneumatic fitting is not recovered.

History	Etekcity Digital Multimeter, Auto-Ranging . . .
	Klein Tools VDV826-605 UR IDC Connectors UR . . .
	Sanuke 120Pcs AWG 8 6 4 2 Copper Wire Lugs . . .
	Gorilla Waterproof Patch & Seal Tape, 4" x . . .
Target	Pronese 4pcs Premium Solid Brass Pneumatics . . .
Top-1	CaRoller Drill Bit Set 29-Piece M35 Cobalt . . .
Top-2	haisstronica 620PCS Marine Grade Heat Shrink . . .
Top-3	AmazonCommercial 10 Gallon Commercial Office . . .
Top-4	Inspection Camera Borescope with 5" IPS . . .
Top-5	Wirefy 95 PCS Heat Shrink Tubing Kit - 4:1 . . .

K Reported Results and Archived Evaluation Records

Table 21: Results with separate evaluation scopes. Panel (a) repeats the updated measurements of Table 1 at their supplied four-decimal precision. Panel (b) contains archived wider-cutoff measurements used by the replay analyses; its Instrument/Game checkpoints precede the updated reported-cutoff evaluations. The panels are not merged into a single checkpoint record.

(a) Updated reported-cutoff results

Dataset	R@1	R@5	R@10	N@5	N@10
Scientific	0.0090	0.0308	0.0484	0.0198	0.0255
Instrument	0.0116	0.0392	0.0628	0.0252	0.0326
Game	0.0195	0.0603	0.0940	0.0408	0.0520

(b) Archived wider-cutoff records

Dataset	Selected epoch(s)	R@20	N@20	MRR@20
Scientific	13/11/11	0.074074	0.031904	0.020276
Instrument	10	0.095406	0.040778	0.025669
Game	14	0.142198	0.062112	0.039965
Toys	9	0.056634	0.025139	0.016362
All Beauty	16	0.051383	0.016971	0.007666
Digital Music	11	0.013867	0.006894	0.004909

L Objective Reductions and Gradient Routes

This appendix specifies the active objective using the recorded mainline configuration and the inspected model routines. Let $\mathcal{B}$ be a minibatch, y_u its target item, $\bar{\mathbf{s}}(j) = \mathbf{s}(j)/\|\mathbf{s}(j)\|$, and $U(a) = |\mathcal{B}|^{-1} \sum_{u \in \mathcal{B}} a_u$. For nonnegative weights w_u, write $W_w(a) = \sum_u w_u a_u / \sum_u w_u$. Reductions over a restricted set of valid rows use only those rows in both numerator and denominator; an empty set contributes zero. Numerical denominators are clamped away from zero in the implementation.

L.1 Frequency Weights and Endpoint Supervision

For training frequency f_j, define $a_j = (f_j + 1)^{-0.1}$ and $a_{\max} = \max_{j: f_j > 0} a_j$. The item weight is

$$w_j = \mathrm{clip}(a_j, a_{\max}/2, a_{\max}), \qquad w_u = w_{y_u}. \tag{11}$$

Unobserved items receive $a_{\max}$ and padding has weight zero. The recorded configuration uses frequency weighting for recovery and topology, but not for the primary rollout or neighbour averages. With q_u the one-step rollout endpoint and $p_{uj} = \mathrm{softmax}_j\, \ell(q_u)$,

$$\mathcal{L}_{\mathrm{rec}} = W_w\big(\|\hat{\mathbf{x}}_{1,u} - \bar{\mathbf{s}}(y_u)\|_2^2\big)\,, \tag{12}$$

$$\mathcal{L}_{\mathrm{roll}} = U(-\log p_{u,y_u}). \tag{13}$$

Recovery sums squared errors over the 512 coordinates; it does not average them. The target representation and the main rollout catalogue are not detached.

For the additional query-balancing branch, let $\ell_j^{\mathrm{sg}}(q_u)$ use $\mathrm{stopgrad}(\bar{\mathbf{s}}(j))$ as the catalogue, retaining gradients through q_u. If $a_u^{\mathrm{sg}} = -\log \mathrm{softmax}_{y_u}\, \ell^{\mathrm{sg}}(q_u)$, then

$$\mathcal{L}_{\mathrm{qb}} = W_w(a^{\mathrm{sg}}) - U(a^{\mathrm{sg}}). \tag{14}$$

Its coefficient is 0.25. This is a weighted-minus-unweighted correction, not an independent positive CE term, and its catalogue is detached only on this branch. Removing the rollout-supervision coefficient disables both the main rollout CE and this correction.

L.2 Neighbourhood and History Terms

Let $\mathcal{N}_{20}(y_u)$ be the valid top-20 training co-occurrence neighbours. The anchor index is

$$j_u^* = \arg\min_{j \in \mathcal{N}_{20}(y_u)} \|\mathbf{x}_{t,u} - \bar{\mathbf{s}}(j)\|_2^2.$$

The discrete anchor selection is detached; the selected anchor and path state remain differentiable. With $b_u = 4t_u(1-t_u)w_u$,

$$\mathcal{L}_{\text{topo}} = W_b\big(\|\mathbf{x}_{t,u} - \bar{\mathbf{s}}(j_u^*)\|_2^2\big)\,. \tag{15}$$

Thus the time weights are also included in the reduction denominator. For the valid first eight neighbours,

$$\mathcal{L}_{\text{nbr}} = U\left[-\frac{1}{|\mathcal{N}_8(y_u)|}\sum_{j \in \mathcal{N}_8(y_u)} \log p_{uj}\right]. \tag{16}$$

These probabilities are normalised over the complete catalogue, not over the eight neighbours. Padding and rows without valid neighbours are excluded.

History reconstruction uses the unconditioned history context. A decoder produces $d_u = \text{normalize}(D_\omega(\mathbf{h}_u))$. Candidates consist of observed history positions and 64 uniform nonpadding item draws. Padded history positions and negative draws colliding with a history item are masked; repeated valid draws retain their multiplicity. With binary labels z_{uj} and mask m_{uj},

$$\mathcal{L}_{\text{hist}} = \frac{\sum_{u,j} m_{uj}\left[\text{sigmoid}(d_u^\top \bar{\mathbf{s}}(j)/0.1) - z_{uj}\right]^2}{\sum_{u,j} m_{uj}}. \tag{17}$$

This is masked squared error on interaction probabilities, not embedding regression or a sampled softmax.

For histories longer than ten items, a separate auxiliary rollout uses a detached catalogue. Its CE is averaged with weights $w_{y_u}\max(L_u - 10, 0)/10$. The item inputs to its auxiliary history encoder are detached; history-encoder gradients are scaled by 0.5 while the prediction head remains trainable. Its coefficient is 0.05 until the decay interval, falls linearly over epochs 20–24, and is zero thereafter. The interpolant-conditioned query update uses gradient scale 0.25, whereas the main rollout-conditioned update uses one. These gradient routes are distinct from changing the forward query norm bound.

L.3 Why the Residual Gradient Is Not Sparse

Hold the query fixed and consider the direct catalogue branch of CE. Define $G_j = \partial\mathcal{L}_{\text{roll}}/\partial\mathbf{s}(j)$, including cosine normalisation. The representation in Equation 2 gives

$$\frac{\partial\mathcal{L}_{\text{roll}}}{\partial\mathbf{r}_j} = \alpha g(f_j) G_j, \qquad \frac{\partial\mathcal{L}_{\text{roll}}}{\partial E_m[a]} = \frac{1}{\sqrt{M}}\sum_{j:c_{jm}=a} [G_j]_m. \tag{18}$$

Here $[G_j]_m$ is the block corresponding to codebook m. Since $\partial\mathcal{L}_{\text{roll}}/\partial\ell_j = p_j - \mathbf{1}[j=y]$, negative catalogue items generally receive gradients even when not looked up as history or target items. Zero-gated residuals are an exception. Additional gradient routes through the query, recovery and local regularisers do not restore a general sparse-update interpretation. The sharing pattern motivates separate tuning, but the formula does not prove that shared updates always have greater magnitude.

M MATCHED-CONTROL PROTOCOLS AND ADDITIONAL RESULTS

This appendix details the matched study in Section 4.5 and reports comparisons with nearby baselines, learning-rate settings, a minimal training recipe and alternative inference sources. The tables report experimental results at their recorded display precision.

M.1 REPRESENTATION AND TRAINING

Figure 13 summarises Recall@20 differences from Full for the controls in Table 4. Full exceeds all five variants on both datasets. The no-recovery variant is closest, followed by item-only with the full recipe and direct CE with SID sharing on Scientific, and by the latter on Toys. The item-only direct predictor and shuffled-SID variant have the largest Scientific deficits. These results distinguish the smaller gains from recovery and semantic sharing from the larger component-removal losses in Table 3.

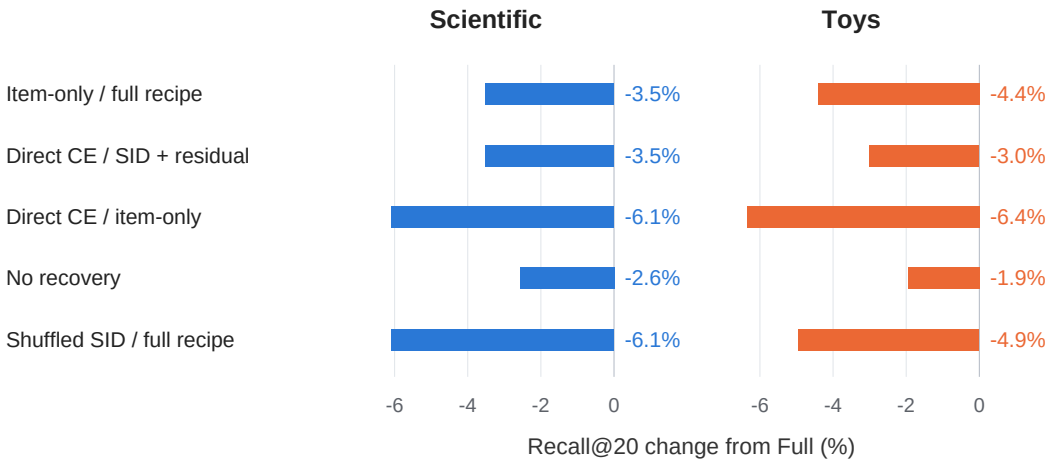


Figure 13: Recall@20 changes relative to Full for the matched controls in Table 4. Bars show point-result differences calculated from the displayed values; the vertical zero line denotes Full.

M.2 CAPACITY AND IMPLEMENTATION

The SID table has $1025 \times 128 = 131{,}200$ parameters, including its padding row. Removing it from an otherwise identical free-item-table model changes the reported total by approximately 0.32% on Scientific and 0.12% on Toys. The comparison is therefore near, rather than exactly, total-capacity matched. It also changes how rare items share parameters. The item-only variant retains a fixed SID lookup for output, without using codes as representation features.

No recovery retains the topology, neighbour, reconstruction and long-history losses; direct CE removes them along with recovery. These variants use the same history encoder, output-head initialisation and source convention as Full. The resulting 2×2 comparison tests the hybrid representation under both objectives, rather than relying only on the shuffled-SID result.

M.3 CLOSEST BASELINES AND CHECKPOINT SELECTION

The closest-baseline comparison uses the same cache, data split, sequence limit and exact-item evaluation protocol. Relative to SASRec with full-catalogue CE, Full improves Recall@20 by 6.5% on Scientific and 5.2% on Toys. MIRAGE, the manifold-informed flow method of Fang et al. (2026), matches Full on Scientific at the displayed precision (0.0739 Recall@20 and 0.0322 NDCG@20). On Toys, MIRAGE scores 0.0572 and 0.0254, versus Full's 0.0566 and 0.0251. Thus, the matched comparison shows an advantage over SASRec but near-parity with MIRAGE, with a small Toys advantage for MIRAGE. The timing study in Section 4.7 compares against LLaDA-Rec, not these two baselines.

Table 22: Closest-baseline comparison under the matched data and exact-item evaluation protocol. The corresponding OxygenFlowRec results are in Table 4.

	Scientific		Toys	
Method	R@20	N@20	R@20	N@20
SASRec with full-catalogue CE	0.0694	0.0303	0.0538	0.0239
MIRAGE, matched protocol	0.0739	0.0322	0.0572	0.0254

Table 23: LLaDA-Rec layer/selection study. The original six-layer reproduction retains its SID-level evaluator; the three controlled configurations use exact-item evaluation. Selection denotes the validation criterion.

Layers	Validation selection	Evaluation	Scientific R@20	Instrument R@20
6	original NDCG@1	SID	0.0656	0.0860
4	original NDCG@1	item	0.0683	0.0885
6	common item R@20	item	0.0679	0.0881
4	common item R@20	item	0.0696	0.0902

With validation item Recall@20 selection and exact-item evaluation fixed, reducing LLaDA-Rec from six to four layers raises Recall@20 from 0.0679 to 0.0696 on Scientific and from 0.0881 to 0.0902 on Instrument. Within the four-layer configurations, changing selection from NDCG@1 to item Recall@20 raises the scores from 0.0683 to 0.0696 and from 0.0885 to 0.0902. Both architecture and checkpoint selection affect the reproduced result. The original six-layer reference retains SID-level evaluation and is not combined with the controlled rows into a same-evaluator estimate of the published-to-reproduced gap.

M.4 Repeated-Seed Results and Evaluation Records

For the Scientific full model, seeds $13/42/7$ yield Recall@20 of 0.073865, 0.074061 and 0.074296: mean 0.074074 and sample standard deviation 0.000216. Table 4 uses the seed-13 Full reference, not this three-seed mean. The reported matched contrasts are point comparisons; the full-model seed variation does not quantify uncertainty for differences between separately trained variants.

The reported-cutoff evaluations in Table 1 and the archived wider-cutoff records in Table 21 retain their respective checkpoint scopes. The archived replays and geometric analyses refer to the listed frozen checkpoints rather than to a combined record of different evaluations.

M.5 Learning Rate, Minimal Recipe and Inference Source

Table 24: Shared-code learning-rate ratios on Scientific with a common 24-epoch budget. Values are peak Recall@20 on the internal offset-2 validation split, reported to four decimal places. Partial-budget runs are listed separately in Table 7.

ρ_{SID}	Internal validation R@20
0	0.0875
0.0625	0.0921
0.125	0.0930
0.25	0.0866

At a common 24-epoch budget, $\rho_{\mathrm{SID}} = 0.125$ achieves the highest internal-validation Recall@20 among the four tested ratios: 0.0930, compared with 0.0875 at 0, 0.0921 at 0.0625 and 0.0866 at 0.25. The selected ratio is therefore not the lower boundary of this completed comparison. Freezing the shared-code embeddings is worse than updating them at a reduced rate. The partial-budget runs in Table 7 remain separate from this equal-budget result.

Table 25: Minimal training recipe and inference-source ablations. The minimal recipe is trained separately. Source variants use a fixed full-model checkpoint on each dataset. Full-model references are in Table 4.

Variant	Scientific R@20	Toys R@20
Recovery + rollout CE only	0.0724	0.0558
Alternative deterministic hash	0.0738	0.0567
Zero source	0.0735	0.0565
Eight-particle average	0.0741	0.0567

The minimal recipe retains the hybrid representation, gate and grouped learning rates, but uses only recovery and rollout CE with its query correction. It reaches Recall@20 of 0.0724 on Scientific and 0.0558 on Toys, 2.0% and 1.4% below Full. Recovery and rollout supervision therefore retain most of the full model's accuracy; the remaining auxiliary objectives provide a smaller aggregate improvement.

The inference-source comparison keeps the full-model checkpoint fixed on each dataset. Across another deterministic hash, a zero source and eight-particle averaging, Recall@20 ranges from 0.0735 to 0.0741 on Scientific and from 0.0565 to 0.0567 on Toys. These small changes indicate limited sensitivity to the tested sources. Zeroing the source slightly lowers both scores, while eight-particle averaging raises them by only 0.3% and 0.2% relative to Full. The table compares accuracy, not latency or list overlap.

M.6 Frequency and SID-Depth Results

The frequency-bucket analysis in Appendix H compares Full, no-gate and no-residual variants on the same test users. The residual trades rare-target recall for head-target recall rather than improving every bucket. The aggregate item-only contrast in Table 4 does not localise its gain to particular frequency buckets.

The depth results in Figure 9b keep the tokenizer width fixed while recommender dimension and capacity change. They compare joint tokenizer/model configurations, not SID depth at fixed $D = 512$. Their interpretation remains separate from the matched representation study.