CODES FOR REPRESENTATION, ITEMS FOR PREDICTION: RETHINKING SEMANTIC-ID RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic IDs (SIDs) represent items through compositions of shared discrete codes, yet generative recommenders typically also use these codes as autoregressive prediction targets. We observe that, under the same backbone configuration, direct item prediction can outperform autoregressive SID prediction, motivating the hypothesis that representation sharing and prediction factorization need not be coupled. To examine this hypothesis, we analyze what changes when a complete-item decision is factorized into an autoregressive SID path. At the distribution level, a locally normalized legal SID tree can represent any strictly positive catalog distribution, while controls over invalid-continuation competition do not explain the observed ranking gap. This directs attention to the supervision structure: under teacher forcing, SID prediction directly supervises branch logits along the target path, progressively refining nested target subtrees, whereas non-target subtrees compete through shared entry branches without direct supervision of their internal conditionals from the current example. Progressively controlled comparisons show that direct item scoring remains stronger under increasingly matched designs and exact catalog ranking. Based on this decoupling principle, we propose Next-item Representation Prediction (NRP), which composes shared SID embeddings into complete-item representations and directly optimizes item-level scores without an item-specific embedding table. Across three public datasets, NRP achieves competitive performance against strong sequential and generative baselines and improves transductive cold-start recommendation.

1 INTRODUCTION

Semantic-ID recommendation represents items through combinations of shared discrete codes, enabling parameter sharing across items (Shu & Nakayama, 2018; Shi et al., 2020) while preserving semantic relatedness (Rajput et al., 2023; Hou et al., 2023). In generative recommendation, however, these same codes are typically assigned a second role: they are also used as the sequential prediction targets of autoregressive models (Rajput et al., 2023; Wang et al., 2024b). This coupling is natural from a language-modeling perspective, but it is not dictated by the recommendation task itself, whose target is ultimately the next item. Indeed, under the same backbone configuration, we observe that direct item prediction with independent embeddings outperforms autoregressive SID prediction on both Beauty and Toys (Table 2(a)). This observation raises a fundamental question: should Semantic IDs be used both to represent items and to define the units of prediction? We hypothesize that these two roles need not be coupled—that the benefit of Semantic IDs may lie in compositional representation sharing, while recommendation supervision can remain at the complete-item level. Figure 1 contrasts these representation and prediction choices and previews the different supervision structures they induce.

To examine this hypothesis, we analyze autoregressive SID factorization from two perspectives: the catalog distributions it can represent and the supervision it induces during training (Section 4). A locally normalized legal SID tree with unrestricted node conditionals can represent any strictly positive distribution over the catalog, so tree factorization alone imposes no inherent expressivity gap. Standard next-token prediction (NTP) further normalizes over invalid continuations, but controls over this competition do not account for the observed prediction-formulation gap. These obser-

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

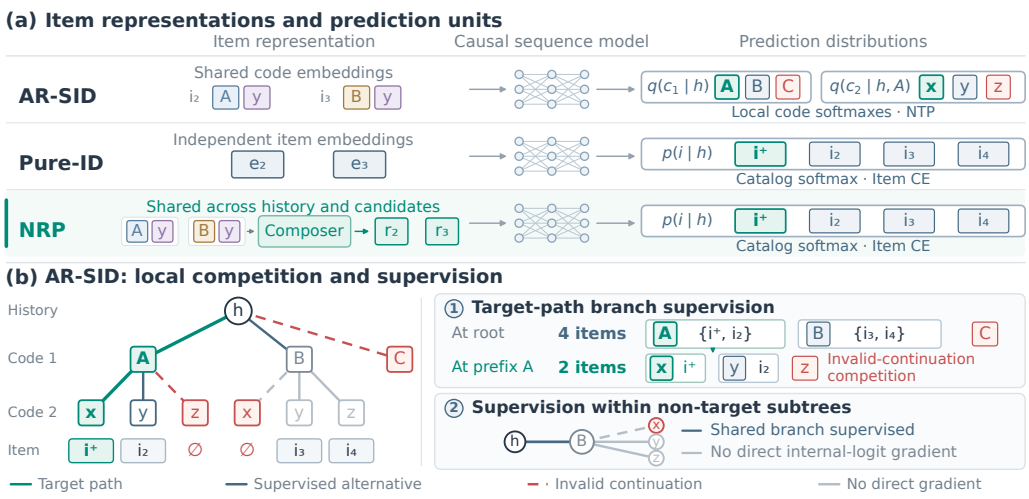


Figure 1: Decoupling representation and prediction in Semantic-ID recommendation. (a) AR-SID predicts codes; Pure-ID and NRP score complete items using independent and SID-composed embeddings, respectively. (b) Teacher-forced branch supervision refines nested target subtrees, while non-target subtrees compete through shared entry branches. Gray edges receive no direct logit gradient from this example, although shared-parameter updates may affect their predictions. Dashed edges denote invalid continuations; $\emptyset$ denotes no corresponding catalog item.

vations motivate examining how autoregressive factorization organizes supervision. Under teacher forcing, direct supervision acts on branch logits at target-path prefixes. As the target prefix grows, later decisions refine nested target subtrees; once a non-target subtree branches off, its items compete through a shared entry branch, without direct supervision of the subtree's internal conditionals from the current example. In contrast, direct item scoring makes complete items the units of competition and directly supervises their individual scores across the full catalog.

We next test the hypothesis under progressively stronger controls. With the SID-token history encoding design held fixed, direct scorers using either independent item embeddings or SID-composed representations outperform legal-path prediction, with composition performing best. We then compare the two formulations using the same composed-item history encoding design, composer architecture, causal backbone, and training budget, with exact catalog ranking for both. Direct item scoring remains substantially stronger under closely matched parameter budgets. Together, these results support a simple design principle: use Semantic IDs for representation sharing, but optimize recommendation at the item level. We instantiate this principle as Next-item Representation Prediction (NRP), which composes shared SID embeddings into complete-item representations for both history encoding and candidate scoring and trains them with full-catalog item cross-entropy, without an item-specific embedding table. Across three public datasets, NRP achieves competitive performance against strong sequential and generative baselines, with additional benefits in transductive cold-start recommendation.

In summary, our main contributions are as follows:

- We disentangle SID representation sharing from autoregressive prediction factorization, establish the distributional expressivity of legal SID trees with unrestricted node conditionals, and distinguish target-path branch supervision from catalog-wide item supervision.
- We propose Next-item Representation Prediction (NRP) following this decoupling principle. NRP composes shared SID embeddings into complete-item representations and directly optimizes item-level scores, retaining code-level parameter sharing without an item-specific embedding table.
- We provide progressively controlled comparisons showing that direct item scoring remains stronger under increasingly matched representation designs and parameter budgets, with exact catalog ranking for both formulations. NRP is competitive with strong baselines, improves transductive cold-start recommendation, and is evaluated on a 64K-item catalog.

2 Related Work

Sequential recommendation. Attention-based models such as SASRec and BERT4Rec encode interaction histories for item prediction (Kang & McAuley, 2018; Sun et al., 2019). S3-Rec and CL4SRec enhance representation learning through self-supervised objectives (Zhou et al., 2020; Xie et al., 2022). UniSRec and Recformer further explore transferable representations based on item text (Hou et al., 2022; Li et al., 2023).

Autoregressive Semantic-ID recommendation. TIGER constructs quantized SIDs for autoregressive recommendation (Rajput et al., 2023). Subsequent work integrates collaborative semantics (Zheng et al., 2024), learns item tokenization (Wang et al., 2024a), structures identifiers as semantic trees (Si et al., 2024), and jointly optimizes tokenization and recommendation (Liu et al., 2025). EAGER combines behavioral and semantic generation streams (Wang et al., 2024b). Within this formulation, SID codes serve both as shared representation components and as autoregressive prediction targets.

Alternative prediction formulations. LIGER and COBRA connect generative and dense retrieval (Yang et al., 2025a;b). RPG predicts discrete SIDs in parallel, SETRec generates sets of continuous tokens, and DiffGRM uses discrete diffusion (Hou et al., 2025; Lin et al., 2025; Liu et al., 2026). VQ-Rec and DiGR compose shared code embeddings for item scoring, focusing on transferability and ranking recovery, respectively (Hou et al., 2023; Zhao et al., 2026). Our work examines the coupling between SID representation sharing and autoregressive prediction factorization. We characterize the supervision induced by branch-level and item-level parameterizations and test the decoupling hypothesis through progressively controlled comparisons, including a comparison under closely matched representation designs and exact catalog ranking (Appendix B.4).

3 Preliminaries

Sequential recommendation. Let $\mathcal{I}$ denote the item catalog and $h = (i_1, \ldots, i_t)$ a user's interaction history. Given h, the task is to rank candidate items for predicting the next interaction $i^+ \in \mathcal{I}$ (Hidasi et al., 2016; Tang & Wang, 2018). We study two ways to parameterize this prediction: autoregressive prediction of an item's Semantic ID and direct scoring of complete items.

Semantic IDs and autoregressive prediction. Each item $i \in \mathcal{I}$ is assigned a fixed-length Semantic ID (Rajput et al., 2023) $c(i) = (c_1(i), \ldots, c_L(i))$, where $c_\ell(i) \in \mathcal{V}_\ell$ and $\mathcal{V}_\ell$ is the code vocabulary at position ℓ. Throughout our analysis, we assume that distinct catalog items have distinct complete code sequences, so catalog items correspond one-to-one to the leaves of the induced legal code tree. For a valid prefix u of length $\ell - 1$, define its valid continuations as

$$\mathcal{A}(u) = \{a \in \mathcal{V}_\ell \mid \exists i \in \mathcal{I} : c_{<\ell}(i) = u,\ c_\ell(i) = a\}. \tag{1}$$

Validity is therefore prefix-dependent: a code is invalid at u if it does not belong to $\mathcal{A}(u)$.

An autoregressive model assigns a code sequence probability

$$q_\theta(c(i) \mid h) = \prod_{\ell=1}^{L} q_\theta(c_\ell(i) \mid h, c_{<\ell}(i)). \tag{2}$$

Standard next-token prediction (NTP) normalizes each conditional over the full vocabulary $\mathcal{V}_\ell$ and minimizes

$$\mathcal{L}_{\mathrm{NTP}}(\theta; h, i^+) = -\sum_{\ell=1}^{L} \log q_\theta(c_\ell(i^+) \mid h, c_{<\ell}(i^+)). \tag{3}$$

This model can assign probability to sequences outside the catalog's valid code set. Restricting each local normalization to $\mathcal{A}(u)$ instead defines a locally normalized legal code tree, which we analyze in Section 4.

Direct item scoring. Alternatively, a model assigns a score $s_\phi(h, i)$ to each complete item and defines a distribution over the catalog:

$$p_\phi(i \mid h) = \frac{\exp s_\phi(h, i)}{\sum_{j \in \mathcal{I}} \exp s_\phi(h, j)}, \qquad \mathcal{L}_{\mathrm{item}}(\phi; h, i^+) = -\log p_\phi(i^+ \mid h). \tag{4}$$

Training minimizes this full-catalog cross-entropy. The scoring formulation does not prescribe how item representations are constructed: it can use either independent item embeddings or representations composed from shared Semantic-ID embeddings.

4 RETHINKING SEMANTIC-ID PREDICTION

To examine the decoupling hypothesis, we analyze autoregressive SID factorization from two perspectives: how it parameterizes item probabilities and how practical prediction parameterizations organize supervision. We first study distributional expressivity and legal support, then compare target-path branch supervision with catalog-wide item supervision. Formal results and proofs are provided in Appendix A.

4.1 DISTRIBUTION PARAMETERIZATION AND LEGAL SUPPORT

We first examine whether legal SID-tree factorization restricts catalog-level expressivity, and then characterize the additional objective introduced by full-vocabulary normalization in standard NTP.

Distributional expressivity. Tree-structured probability factorization is also used in hierarchical language models (Morin & Bengio, 2005; Mnih & Hinton, 2008). A locally normalized legal code tree defines a complete item distribution and, for a fixed context, can represent any strictly positive item distribution given unrestricted node conditionals. Specifically, the probability assigned to item i is

$$p_{\mathrm{tree}}(i \mid h) = \prod_{\ell=1}^{L} r(c_\ell(i) \mid h, c_{<\ell}(i)), \tag{5}$$

where each conditional is normalized over $\mathcal{A}(u)$. With unique complete SIDs, these probabilities sum to one over $\mathcal{I}$. For any strictly positive distribution π, let $\mathcal{I}_u$ denote the items with prefix u and $M(u) = \sum_{i \in \mathcal{I}_u} \pi(i)$. Choosing $r(a \mid h, u) = M(ua)/M(u)$ recovers $\pi(i)$ through the telescoping product along each path. Thus, tree factorization alone does not impose a distributional expressivity gap under unrestricted conditionals. Practical learning nevertheless depends on how these conditionals are parameterized and trained.

Invalid-continuation competition. Standard NTP normalizes each conditional over the full code vocabulary, allowing probability mass on invalid continuations. To relate this model to a locally normalized legal tree, let $u_\ell = c_{<\ell}(i^+)$, $\alpha_\theta(h, u) = \sum_{a \in \mathcal{A}(u)} q_\theta(a \mid h, u)$, and $\bar{q}_\theta(a \mid h, u) = q_\theta(a \mid h, u)/\alpha_\theta(h, u)$ for $a \in \mathcal{A}(u)$. The NTP loss admits the exact decomposition

$$\mathcal{L}_{\mathrm{NTP}} = \underbrace{-\sum_{\ell=1}^{L} \log \bar{q}_\theta(c_\ell(i^+) \mid h, u_\ell)}_{\mathcal{L}_{\mathrm{tree}}} + \underbrace{-\sum_{\ell=1}^{L} \log \alpha_\theta(h, u_\ell)}_{\mathcal{L}_{\mathrm{support}}}. \tag{6}$$

The support term introduces competition between valid and invalid continuations at each target prefix. In the SID-token-history control, optimizing only $\mathcal{L}_{\mathrm{tree}}$ reduces ranking performance relative to standard NTP under the same constrained decoder (Section 6.3), suggesting that support supervision can be useful here.

Unrestricted legal-tree factorization does not limit catalog-level expressivity at a fixed context, while removing invalid-continuation competition does not improve ranking in the SID-token-history control. These observations motivate examining how practical predictors organize supervision. Restricting local normalization changes which continuations compete, but both objectives retain supervision through local decisions along the target path.

4.2 SUPERVISION STRUCTURE UNDER PRACTICAL PARAMETERIZATIONS

Both legal-tree training and full-catalog item cross-entropy optimize complete-item likelihoods. Their practical parameterizations expose different output variables: local branch logits and complete-item scores, respectively. For the legal-tree model, each conditional is normalized over valid continuations and evaluated at the ground-truth target prefix under teacher forcing, yielding $\mathcal{L}_{\mathrm{tree}} = -\log p_{\mathrm{tree}}(i^+ \mid h)$. We compare how these objectives distribute direct supervision through

their output variables. Here, direct supervision refers to derivatives with respect to branch logits or item scores; shared-parameter updates can also affect outputs with zero direct derivative (Appendix A.4).

Target-path branch supervision. Under the legal-tree objective, direct supervision from each training example acts on branch logits at target-path nodes, including non-target alternatives. Let $z_{u,a}$ denote the logit of valid continuation a at prefix u, and let $r_\theta(a \mid h, u)$ be its local softmax probability. When constructed from the same logits as NTP, r_θ coincides with $\bar{q}_\theta$. Writing $\mathcal{P}(i^+)$ for the target's internal prefixes and a_u^+ for its continuation at u, we obtain

$$\frac{\partial \mathcal{L}_{\text{tree}}}{\partial z_{u,a}} = \begin{cases} r_\theta(a \mid h, u) - \mathbf{1}[a = a_u^+], & u \in \mathcal{P}(i^+), \\ 0, & u \notin \mathcal{P}(i^+). \end{cases} \tag{7}$$

Standard NTP shares this restriction of direct supervision to target-path nodes, although each local competition extends to the full code vocabulary. For the legal tree, the local softmax at each visited node compares the target continuation with all valid alternatives. As the target prefix grows, subsequent decisions operate over nested candidate sets. Let $\mathcal{C}_\ell = \mathcal{I}_{c_{1:\ell}(i^+)}$ denote the items sharing the target's first ℓ codes. Then

$$\mathcal{I} = \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \supseteq \mathcal{C}_L = \{i^+\}. \tag{8}$$

The decision at position ℓ partitions $\mathcal{C}_{\ell-1}$, while subsequent decisions further partition only the target subset $\mathcal{C}_\ell$. Direct supervision at deeper nodes therefore distinguishes among increasingly specific target-prefix subsets.

Supervision within non-target subtrees. This path restriction aggregates the direct supervision of a non-target subtree at its shared entry branch. Consider a non-target continuation $a \neq a_u^+$ at a visited prefix u. Items in $\mathcal{I}_{ua}$ participate in the local competition through the common branch logit $z_{u,a}$, whose gradient is $r_\theta(a \mid h, u)$. Internal branch logits within this subtree receive no direct gradient from the current example. For any $j, k \in \mathcal{I}_{ua}$,

$$\frac{p_{\text{tree}}(j \mid h)}{p_{\text{tree}}(k \mid h)} = \frac{p_{\text{tree}}(j \mid h, ua)}{p_{\text{tree}}(k \mid h, ua)}, \tag{9}$$

where conditioning on ua restricts prediction to that subtree. The common prefix probability cancels. Holding the subtree's internal conditionals fixed, changes to its ancestor branches cannot alter this probability ratio. The current example therefore directly supervises the probability allocated to the subtree, without directly supervising its internal allocation among items.

Catalog-wide item supervision. Full-catalog cross-entropy directly supervises each catalog item through its own score. For an explicit item scorer with $s_i = s_\phi(h, i)$,

$$\frac{\partial \mathcal{L}_{\text{item}}}{\partial s_i} = p_\phi(i \mid h) - \mathbf{1}[i = i^+], \qquad i \in \mathcal{I}. \tag{10}$$

Competition is defined over complete items, with each negative item's score gradient equal to its predicted probability. Items sharing a non-target SID prefix retain separate score coordinates. The distinction therefore concerns how complete-item likelihood supervision is organized through the model outputs.

Progressively controlled comparisons in Section 6.3 favor direct item scoring under the evaluated parameterizations. Together with the supervision analysis, these findings support decoupling SID representation sharing from autoregressive prediction and motivate the NRP formulation below.

5 Next-item Representation Prediction

The decoupling principle separates the representational role of Semantic IDs from the autoregressive factorization used for prediction. Next-item Representation Prediction (NRP) instantiates this principle: Semantic IDs are retained as shared compositional primitives for constructing item representations, while next-item prediction is performed directly over complete items. Accordingly, NRP consists of a shared SID-based item composer, a causal history encoder operating on composed item representations, and a full-catalog item-scoring objective. No independent item embedding table is introduced.

5.1 Shared Compositional Item Representations

Each SID position ℓ has a learnable embedding table $E_\ell \in \mathbb{R}^{|\mathcal{V}_\ell| \times d}$. For an item with fixed SID $c(i) = (c_1(i), \ldots, c_L(i))$, NRP composes its code embeddings into a complete-item representation:

$$z_i = g_\psi([E_1[c_1(i)]; \ldots; E_L[c_L(i)]]) \in \mathbb{R}^d, \tag{11}$$

where $g_\psi : \mathbb{R}^{Ld} \to \mathbb{R}^d$ is a shared residual MLP composer with hidden width w and D blocks (He et al., 2016). The code embeddings and composer are learned jointly through recommendation training. Items sharing a code at the same position reuse its embedding. The same embedding tables and composer encode both historical and candidate items, without an independent item embedding table. Composer details and representation analyses are provided in Appendices B.2 and D.

5.2 Direct Complete-item Prediction

On the prediction side of the decoupling principle, NRP operates on complete-item representations rather than autoregressively predicting their constituent SID codes.

The composed representations contain a catalog-wide mean component shared across items. We apply catalog centering to express each composed item representation relative to a common reference:

$$\mu = \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} z_k, \qquad r_i = z_i - \mu. \tag{12}$$

This differentiable transformation uses the current catalog mean, preserves pairwise representation differences, and introduces no trainable parameters. The centered representations are used for both history encoding and candidate scoring.

For a history $h = (i_1, \ldots, i_t)$, a causal sequence encoder f_ω processes one composed representation per historical item and returns the hidden state at the last valid position. A bias-free linear projection maps this state to a query, which is matched directly against complete candidate-item representations:

$$q_h = W_h f_\omega(r_{i_1}, \ldots, r_{i_t}), \qquad s(h, i) = q_h^\top r_i. \tag{13}$$

The resulting score is defined at the complete-item level rather than through a product of local SID decisions. NRP minimizes the full-catalog cross-entropy defined in Eq. (4), jointly learning the code embeddings, composer, history encoder, and query projection.

In this way, NRP directly instantiates the decoupling principle: Semantic IDs determine how item representations share parameters, whereas the recommendation objective compares and supervises complete items. The model therefore retains compositional representation sharing without coupling it to autoregressive SID prediction.

6 Experiments

Our experiments address the following research questions:

- **RQ1:** How does NRP compare with existing sequential and generative recommendation methods across datasets?
- **RQ2:** Does the decoupling hypothesis hold under progressively stronger controls on training objective, history representation, model parameterization, and ranking procedure?
- **RQ3:** Do shared SID representations benefit from semantic structure and support cold-start recommendation?

6.1 Experimental Setup

Datasets. We use the official 5-core Beauty, Sports and Outdoors (Sports), and Toys and Games (Toys) subsets of Amazon Reviews 2014 (McAuley et al., 2015), a collection of user reviews and product metadata. Reviews are treated as interactions; dataset statistics appear in Appendix B.1.

Table 1: Performance comparison of different methods. The best performance is highlighted in bold while the second-best performance is underlined. NRP results are reported as mean ± standard deviation over five random seeds.

	Beauty				Sports				Toys			
Method	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
SASRec	0.0385	0.0245	0.0615	0.0319	0.0221	0.0148	0.0330	0.0183	0.0455	0.0311	0.0673	0.0382
BERT4Rec	0.0277	0.0180	0.0454	0.0237	0.0161	0.0104	0.0271	0.0140	0.0265	0.0173	0.0396	0.0215
TIGER	0.0401	0.0270	0.0624	0.0341	0.0233	0.0154	0.0374	0.0193	0.0398	0.0270	0.0574	0.0327
EAGER	<u>0.0618</u>	**0.0451**	0.0836	<u>0.0525</u>	0.0281	0.0184	<u>0.0441</u>	0.0236	0.0584	<u>0.0464</u>	0.0714	0.0505
COBRA	0.0537	0.0395	0.0725	0.0456	0.0305	0.0215	0.0434	0.0257	0.0619	0.0462	0.0781	0.0515
RPG	0.0569	0.0405	0.0786	0.0475	0.0307	0.0211	<u>0.0441</u>	0.0254	<u>0.0627</u>	0.0450	<u>0.0871</u>	<u>0.0528</u>
SETRec	0.0405	0.0273	0.0626	0.0344	0.0233	0.0148	0.0348	0.0185	0.0415	0.0279	0.0633	0.0349
DiffGRM	0.0603	0.0414	<u>0.0876</u>	0.0502	<u>0.0363</u>	<u>0.0245</u>	**0.0550**	<u>0.0305</u>	0.0618	0.0455	0.0834	0.0524
NRP (ours)	**0.0638** ±0.0011	<u>0.0443</u> ±0.0008	**0.0928** ±0.0005	**0.0536** ±0.0006	**0.0365** ±0.0004	**0.0248** ±0.0003	**0.0550** ±0.0003	**0.0308** ±0.0002	**0.0664** ±0.0008	**0.0475** ±0.0009	**0.0944** ±0.0014	**0.0565** ±0.0008

Evaluation protocol. Chronological leave-one-out reserves each user's last interaction for testing, the penultimate for validation, and the remainder for training. Histories retain at most the 20 most recent items. We report user-averaged Recall@K and NDCG@K ($K \in \{5, 10\}$) and select checkpoints by validation NDCG@10. Test-time filtering excludes all items in the complete pre-target history except the held-out target.

Implementation details. We construct fixed SIDs from 768-dimensional Sentence-T5-base features (Ni et al., 2022) using three residual k-means levels with 256 codes each and a disambiguation suffix. NRP uses 128-dimensional code embeddings, a six-block composer (width 256, catalog centering), and a randomly initialized GPT-2 backbone (four layers, four heads, hidden size 128). We train with AdamW (Loshchilov & Hutter, 2019) (learning rate 5×10^{-5}, batch size 64) and report NRP's main results as mean ± standard deviation over five seeds.

6.2 Overall Performance (RQ1)

Baselines. We compare NRP with two conventional sequential recommendation methods, SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019), and six generative methods: TIGER (Rajput et al., 2023), EAGER (Wang et al., 2024b), COBRA (Yang et al., 2025b), RPG (Hou et al., 2025), SETRec (Lin et al., 2025), and DiffGRM (Liu et al., 2026). Results for EAGER, COBRA, and DiffGRM are taken from their original papers, which use the same Amazon benchmark categories and chronological leave-one-out evaluation setting as our main comparison; the remaining baselines are reimplemented under our protocol. Detailed baseline descriptions and implementation settings are provided in appendix B.3.

Results. Table 1 shows that NRP outperforms the conventional sequential baselines SASRec and BERT4Rec across all datasets and metrics. Compared with the generative baselines, NRP achieves the highest NDCG@10 on all three datasets, although EAGER remains slightly stronger on Beauty NDCG@5. On Beauty and Toys, NRP improves Recall@10 over the strongest respective baselines, DiffGRM and RPG, by 5.9% and 8.4%. On Sports, NRP performs comparably to DiffGRM, matching its Recall@10 at the reported precision and slightly improving the other metrics. To focus on prediction parameterization, we use fixed SIDs from standard residual quantization without introducing a specialized SID construction method. On the larger CDs and Vinyl catalog with over 64,000 items, NRP also outperforms SASRec and RPG across all four metrics (Appendix F).

6.3 Progressively Controlled Tests of the Decoupling Hypothesis (RQ2)

We test the decoupling hypothesis by progressively controlling alternative explanations for the initial gap between autoregressive SID prediction and direct item scoring (Table 2). Configurations and further checks on capacity, decoding, backbone, and SID length appear in Appendix C.

Table 2: Controlled comparisons on Beauty and Toys. Panel (a) uses three seeds; panels (b)–(d) report means and sample standard deviations over three paired seeds. R and N denote Recall and NDCG, respectively. Best results within each panel are bold.

	Beauty				Toys			
Variant	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
(a) Initial prediction-formulation gap								
AR-SID	0.0392	0.0254	0.0599	0.0326	0.0352	0.0234	0.0555	0.0300
Pure-ID	**0.0485**	**0.0367**	**0.0651**	**0.0420**	**0.0477**	**0.0373**	**0.0622**	**0.0419**
(b) Control I: support-normalization control								
Standard CE	**0.0393** ±0.0005	**0.0258** ±0.0004	**0.0602** ±0.0007	**0.0326** ±0.0003	**0.0351** ±0.0006	**0.0233** ±0.0004	**0.0551** ±0.0007	**0.0298** ±0.0004
Legal-branch CE	0.0362 ±0.0008	0.0236 ±0.0001	0.0571 ±0.0010	0.0303 ±0.0001	0.0332 ±0.0004	0.0221 ±0.0006	0.0513 ±0.0008	0.0279 ±0.0006
(c) Control II: fixed-history prediction control								
Legal SID prediction	0.0362 ±0.0008	0.0236 ±0.0001	0.0571 ±0.0010	0.0303 ±0.0001	0.0332 ±0.0004	0.0221 ±0.0006	0.0513 ±0.0008	0.0279 ±0.0006
Free-item scoring	0.0420 ±0.0006	0.0279 ±0.0005	0.0648 ±0.0010	0.0352 ±0.0006	0.0407 ±0.0012	0.0279 ±0.0005	0.0616 ±0.0011	0.0346 ±0.0006
Composed-item scoring	**0.0450** ±0.0007	**0.0291** ±0.0007	**0.0708** ±0.0006	**0.0374** ±0.0003	**0.0439** ±0.0003	**0.0296** ±0.0005	**0.0684** ±0.0003	**0.0374** ±0.0005
(d) Control III: closely matched prediction formulations								
Legal SID prediction	0.0391 ±0.0014	0.0275 ±0.0015	0.0544 ±0.0013	0.0324 ±0.0015	0.0328 ±0.0011	0.0236 ±0.0010	0.0487 ±0.0019	0.0287 ±0.0013
NRP	**0.0640** ±0.0012	**0.0446** ±0.0010	**0.0927** ±0.0005	**0.0538** ±0.0008	**0.0658** ±0.0005	**0.0471** ±0.0003	**0.0939** ±0.0021	**0.0561** ±0.0004

Table 3: NRP with different SID constructions under exact full-catalog ranking with history filtering. R: Recall; N: NDCG. Best results in bold.

	Beauty				Toys			
SID construction	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
RQ-KMeans	0.0642	0.0442	0.0929	0.0535	0.0656	0.0468	0.0959	0.0566
RQ-VAE	0.0612	0.0434	0.0870	0.0517	0.0628	0.0447	0.0879	0.0528
OPQ	**0.0659**	**0.0459**	**0.0944**	**0.0551**	**0.0744**	**0.0523**	**0.1049**	**0.0622**
Random-Permuted	0.0524	0.0369	0.0759	0.0444	0.0515	0.0384	0.0699	0.0443
IID-Uniform	0.0529	0.0373	0.0747	0.0444	0.0483	0.0353	0.0682	0.0417

Initial observation. With the same backbone configuration and historical interactions, Pure-ID outperforms AR-SID across all four metrics on both datasets (Table 2(a)). However, history representation, output parameterization, and parameter count still differ, motivating the following controls.

Control I: support normalization. To test whether invalid-continuation competition accounts for the initial gap (Section 4.1), we replace full-vocabulary cross-entropy with legal-branch cross-entropy, keeping the SID-token history design and constrained decoder fixed. Legal-branch training reduces all four metrics on both datasets (Table 2(b)), showing that removing this competition does not improve ranking in this setting.

Control II: fixed history representation. We fix the SID-token history design, backbone configuration, and paired encoder initialization, and compare legal SID prediction with two direct scorers: Free-item scoring uses independent item vectors, whereas Composed-item scoring constructs candidate representations from shared SID embeddings. Both direct scorers outperform legal SID prediction across all four metrics on both datasets, with Composed-item scoring performing best (Table 2(c)). Thus, direct item prediction can retain SID-based representation sharing. Output architectures, parameter budgets, and training configurations are not yet fully matched, motivating Control III.

Table 4: Cold-start recommendation on Beauty with 10% item holdout. R: Recall; N: NDCG. Best results in bold.

	Warm targets				Cold targets			
Method	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
Pure-ID	0.0655	0.0411	0.0896	0.0472	0.0000	0.0000	0.0000	0.0000
AR-SID	0.0624	0.0333	0.0938	0.0412	0.0005	0.0002	0.0005	0.0002
SASRec	0.0621	0.0329	0.0914	0.0402	0.0000	0.0000	0.0000	0.0000
TIGER	0.0620	0.0343	0.0899	0.0413	0.0015	0.0005	0.0020	0.0007
RPG	0.0744	0.0439	0.1037	0.0512	0.0119	0.0042	0.0218	0.0068
NRP	**0.0900**	**0.0517**	**0.1274**	**0.0611**	**0.0154**	**0.0063**	**0.0342**	**0.0110**

Control III: closely matched prediction formulations. NRP and the legal-tree predictor use the same composed-item history design, composer architecture, and causal backbone, with paired initialization of shared components. Both use the same learning rate and epoch limits, introduce no independent item embedding table, and rank the full catalog exactly. NRP remains stronger across all four metrics on both datasets (Table 2(d)): NDCG@10 increases from 0.0324 to 0.0538 on Beauty and from 0.0287 to 0.0561 on Toys. This advantage also persists when the matched autoregressive predictor uses standard full-vocabulary cross-entropy under exact catalog ranking (Appendix C.2). Together, these controls support the decoupling hypothesis under the evaluated parameterizations: SID representation sharing can be retained while direct item scoring improves ranking.

6.4 SID Representation and Generalization (RQ3)

We next examine the representation side of decoupling: whether shared SID representations benefit from semantic structure and support cold-start recommendation under direct item scoring.

SID construction and semantic structure. We compare three content-based SID constructions—RQ-KMeans, RQ-VAE (Lee et al., 2022), and OPQ (Ge et al., 2013)—with two random controls, Random-Permuted and IID-Uniform, holding NRP's architecture fixed and keeping SID assignments unchanged during training. All content-based constructions outperform both random controls on both datasets, with OPQ performing best (Table 3). The random controls retain nontrivial ranking performance, indicating that useful representations can also be learned from shared discrete combinations without content-based SID construction.

Cold-start recommendation. On Beauty, we withhold all interactions of 10% of items from training and input histories, retaining their content for SID construction in a transductive cold-start setting. NRP leads all four metrics for both warm and cold targets (Table 4), improving cold-target Recall@10 and NDCG@10 over RPG by approximately 29% and 50%, respectively. Appendix E provides evaluation protocols and additional holdout ratios.

7 Conclusion

We revisit whether Semantic-ID representation sharing needs to be coupled with autoregressive SID prediction. Our analysis shows that legal SID-tree factorization does not inherently restrict catalog-level expressivity, while its practical parameterization organizes direct supervision along target-path branches. Progressively controlled experiments further show that the advantage of complete-item scoring persists as alternative explanations related to support normalization, history representation, model parameterization, and ranking procedure are increasingly controlled. These findings support the decoupling principle. NRP instantiates this principle by composing shared SID embeddings into complete-item representations and optimizing item-level scores directly, without an item-specific embedding table. Across three public datasets, NRP achieves competitive performance against strong sequential and generative baselines, while additional analyses support the value of semantic SID structure and compositional sharing for cold-start recommendation.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and English-language editing, translation, figure preparation, refinement of the research hypothesis and experimental framing, and discussions of the theoretical analysis and interpretation of experimental results. We also used these tools to help check mathematical derivations, cross-check tabulated values, and verify bibliographic information. The authors evaluated and revised all AI-assisted suggestions and take full responsibility for the final content of this work, including its mathematical arguments, empirical results, figures, and references.

ETHICS STATEMENT

This study evaluates recommendation models using existing public benchmark datasets in an offline setting. We have not identified specific ethical concerns arising from the proposed method or the reported experiments. As with other recommendation methods, potential deployment should account for the privacy of interaction data and biases that may be reflected in recommendation outcomes.

REPRODUCIBILITY STATEMENT

The assumptions and derivations underlying our analysis are presented in Section 4. The NRP architecture and training objective are described in Section 5, while experimental settings and evaluation protocols are provided in Section 6.1 and the corresponding appendices. Section 6.3 describes the controlled comparisons, and Appendix E provides additional cold-start protocols and results. We specify the number of random seeds for the corresponding experiments and distinguish our reimplementations from results taken from prior publications. We plan to release the source code publicly.

REFERENCES

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2946–2953. IEEE Computer Society, 2013. doi: 10.1109/CVPR.2013.379. URL https://doi.org/10.1109/CVPR.2013.379.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.06939.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 585–593. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539381. URL https://doi.org/10.1145/3534678.3539381.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pp. 1162–1171. Association for Computing Machinery, 2023. doi: 10.1145/3543507.3583434. URL https://doi.org/10.1145/3543507.3583434.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. Generating long semantic IDs in parallel for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 956–966. ACM, 2025. doi: 10.1145/3711896.3736979. URL https://doi.org/10.1145/3711896.3736979.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11523–11532, 2022. URL https://arxiv.org/abs/2203.01941.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1258–1267. Association for Computing Machinery, 2023. doi: 10.1145/3580305.3599519. URL https://doi.org/10.1145/3580305.3599519.

Xinyu Lin, Haihan Shi, Wenjie Wang, Fuli Feng, Qifan Wang, See-Kiong Ng, and Tat-Seng Chua. Order-agnostic identifier for large language model-based generative recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 1923–1933. ACM, 2025. doi: 10.1145/3726302.3730053. URL https://doi.org/10.1145/3726302.3730053.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 729–739. ACM, 2025. doi: 10.1145/3726302.3729989. URL https://doi.org/10.1145/3726302.3729989.

Zhao Liu, Yichen Zhu, Yiqing Yang, Xiao Lv, Guoping Tang, Rui Huang, Qiang Luo, Ruiming Tang, and Guorui Zhou. DiffGRM: Diffusion-based generative recommendation model. In *Proceedings of the ACM Web Conference 2026*, pp. 5853–5864. Association for Computing Machinery, 2026. doi: 10.1145/3774904.3792156. URL https://doi.org/10.1145/3774904.3792156.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52. Association for Computing Machinery, 2015. doi: 10.1145/2766462.2767755. URL https://doi.org/10.1145/2766462.2767755.

Andriy Mnih and Geoffrey E. Hinton. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, volume 21, pp. 1081–1088. Curran Associates, Inc., 2008. URL https://papers.nips.cc/paper_files/paper/2008/hash/1e056d2b0ebd5c878c550da6ac5d3724-Abstract.html.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pp. 246–252. PMLR, 2005. URL https://proceedings.mlr.press/r5/morin05a.html. Reissued by PMLR on 30 March 2021.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1864–1874. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.146. URL https://aclanthology.org/2022.findings-acl.146/.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula,

Ed Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10299–10315. Curran Associates, Inc., 2023. doi: 10.52202/075280-0452. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/20dcab0f14046a5c6b02b61da9f13229-Paper-Conference.pdf.

Hao-Jun Michael Shi, Dheevatsa Mudigere, Maxim Naumov, and Jiyan Yang. Compositional embeddings using complementary partitions for memory-efficient recommendation systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 165–175. ACM, 2020. doi: 10.1145/3394486.3403059. URL https://doi.org/10.1145/3394486.3403059.

Raphael Shu and Hideki Nakayama. Compressing word embeddings via deep compositional code learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJRZzFlRb.

Zihua Si, Zhongxiang Sun, Jiale Chen, Guozhang Chen, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, Jun Xu, and Kun Gai. Generative retrieval with semantic tree-structured identifiers and contrastive learning. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 154–163. ACM, 2024. doi: 10.1145/3673791.3698408. URL https://doi.org/10.1145/3673791.3698408.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450. Association for Computing Machinery, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Jiaxi Tang and Ke Wang. Personalized Top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573. Association for Computing Machinery, 2018. doi: 10.1145/3159652.3159656. URL https://doi.org/10.1145/3159652.3159656.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409. ACM, 2024a. doi: 10.1145/3627673.3679569. URL https://doi.org/10.1145/3627673.3679569.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and Zhenhua Dong. EAGER: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3245–3254. Association for Computing Machinery, 2024b. doi: 10.1145/3637528.3671775. URL https://doi.org/10.1145/3637528.3671775.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1259–1273. IEEE, 2022. doi: 10.1109/ICDE53745.2022.00099. URL https://doi.org/10.1109/ICDE53745.2022.00099.

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. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=jxdnFIsjCb.

Yuhao Yang, Zhi Ji, Zhaopeng Li, Yi Li, Zhonglin Mo, Yue Ding, Kai Chen, Zijian Zhang, Jie Li, Shuanglong Li, and Lin Liu. Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025b. doi: 10.52202/

085713-3135. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/86ba836d4c5dd859d795a172911745e2-Paper-Conference.pdf.

Yuanbo Zhao, Ruochen Liu, Senzhang Wang, Jun Yin, Yuxin Dong, Huan Gong, Hao Chen, Shirui Pan, and Chengqi Zhang. From token generation to item ranking: Direct generative recommendation with Semantic IDs. *arXiv preprint arXiv:2602.07847*, 2026. doi: 10.48550/arXiv.2602.07847. URL https://arxiv.org/abs/2602.07847.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1435–1448. IEEE, 2024. doi: 10.1109/icde60146.2024.00118. URL https://doi.org/10.1109/icde60146.2024.00118.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 1893–1902. Association for Computing Machinery, 2020. doi: 10.1145/3340531.3411954. URL https://doi.org/10.1145/3340531.3411954.

A Theoretical Analysis and Proofs

This appendix formalizes the two theoretical components used to examine the decoupling hypothesis in Section 4. We first characterize the distributional expressivity of legal SID-tree factorization and the additional support objective induced by standard NTP. We then analyze how practical branch-level and item-level parameterizations organize direct supervision, and clarify how shared parameters mediate these output-level gradients. These results characterize differences in parameterization and supervision; they do not by themselves imply that one formulation must achieve better recommendation performance. The corresponding empirical hypothesis tests are provided in Section 6.3.

A.1 Legal-Tree Likelihoods and Expressivity

Setting and notation. Fix a history h and a nonempty finite catalog $\mathcal{I}$. The fixed assignment $c : \mathcal{I} \to \prod_{\ell=1}^{L} \mathcal{V}_\ell$ is injective, with $L \geq 1$ and finite vocabularies. Let ϵ denote the empty prefix and $\mathcal{U}$ the set of valid internal prefixes, including ϵ. For each valid prefix u, define $\mathcal{I}_u = \{i \in \mathcal{I} : c_{1:|u|}(i) = u\}$, with $\mathcal{I}_\epsilon = \mathcal{I}$. For $u \in \mathcal{U}$, its valid continuations $\mathcal{A}(u)$ are nonempty, and the sets $\{\mathcal{I}_{ua} : a \in \mathcal{A}(u)\}$ partition $\mathcal{I}_u$; ua denotes concatenation. For a target $y = i^+$, let $\mathcal{P}(y) = \{c_{<\ell}(y) : 1 \leq \ell \leq L\}$ and let a_u^+ be its continuation at $u \in \mathcal{P}(y)$.

At each internal prefix, let $r(a \mid h, u) > 0$ for $a \in \mathcal{A}(u)$, with $\sum_{a \in \mathcal{A}(u)} r(a \mid h, u) = 1$. These conditions include finite-logit softmax models; a node with one valid child has conditional probability one. For $i \in \mathcal{I}_u$, define the continuation probability

$$p_r(i \mid h, u) = \prod_{\ell=|u|+1}^{L} r(c_\ell(i) \mid h, c_{<\ell}(i)), \qquad p_r(i \mid h) = p_r(i \mid h, \epsilon), \tag{14}$$

where an empty product equals one.

Lemma A.1 (Normalization and pointwise expressivity). For every valid prefix u, $\sum_{i \in \mathcal{I}_u} p_r(i \mid h, u) = 1$. Moreover, every strictly positive distribution π on $\mathcal{I}$ can be represented exactly as $p_r(\cdot \mid h)$ when the node conditionals are unrestricted.

Proof. At a leaf, injectivity of c makes $\mathcal{I}_u$ a singleton, and its continuation probability is one. If the assertion holds for the children of an internal prefix u, the partition of $\mathcal{I}_u$ gives

$$\sum_{i \in \mathcal{I}_u} p_r(i \mid h, u) = \sum_{a \in \mathcal{A}(u)} r(a \mid h, u) \sum_{i \in \mathcal{I}_{ua}} p_r(i \mid h, ua) = 1. \tag{15}$$

Backward induction proves normalization, including at the root. For expressivity, set $M_\pi(u) = \sum_{i \in \mathcal{I}_u} \pi(i) > 0$ and choose $r(a \mid h, u) = M_\pi(ua)/M_\pi(u)$. These conditionals are normalized by the same partition property. Since $M_\pi(\epsilon) = 1$ and $M_\pi(c(i)) = \pi(i)$,

$$p_r(i \mid h) = \prod_{\ell=1}^{L} \frac{M_\pi(c_{1:\ell}(i))}{M_\pi(c_{<\ell}(i))} = \pi(i). \tag{16}$$

Finite branch logits $z_{u,a} = \log M_\pi(ua)$ realize these conditionals through a local softmax. □

Likelihood and the comparison target. This construction follows the probability decomposition underlying hierarchical language models (Morin & Bengio, 2005; Mnih & Hinton, 2008). For any strictly positive π, normalization also yields

$$\mathbb{E}_{y \sim \pi}[-\log p_r(y \mid h)] = H(\pi) + D_{\mathrm{KL}}(\pi \| p_r(\cdot \mid h)). \tag{17}$$

Thus, unrestricted legal-tree likelihood and unrestricted item softmax have the same optimum π as an item distribution. This establishes that legal-tree factorization itself imposes no pointwise expressivity gap for strictly positive catalog distributions. Consequently, an inherent lack of catalog-level expressivity cannot by itself explain the empirical gap between autoregressive SID prediction and complete-item scoring. The remaining question concerns how practical parameterizations represent and learn these distributions.

A.2 NTP Decomposition and Support Supervision

Let q_θ be a normalized autoregressive distribution on $\prod_{\ell=1}^{L} \mathcal{V}_\ell$, with sequence probabilities given by the product of its next-code conditionals. Only conditionals at catalog-valid prefixes are needed below. For a valid prefix u of length $\ell - 1$, let $z_{u,a}$ be finite logits for all $a \in \mathcal{V}_\ell$ and define

$$\begin{aligned} Z_{\mathcal{V}}(u) &= \sum_{a \in \mathcal{V}_\ell} e^{z_{u,a}}, \qquad Z_{\mathcal{A}}(u) = \sum_{a \in \mathcal{A}(u)} e^{z_{u,a}}, \\ q_\theta(a \mid h, u) &= \frac{e^{z_{u,a}}}{Z_{\mathcal{V}}(u)}, \qquad \alpha_\theta(h, u) = \frac{Z_{\mathcal{A}}(u)}{Z_{\mathcal{V}}(u)}. \end{aligned} \tag{18}$$

For $a \in \mathcal{A}(u)$, write $\bar{q}_\theta(a \mid h, u) = q_\theta(a \mid h, u)/\alpha_\theta(h, u) = e^{z_{u,a}}/Z_{\mathcal{A}}(u)$. Let $\bar{p}_\theta$ be the legal-tree item distribution obtained by using $\bar{q}_\theta$ at every valid prefix. All quantities below are constructed from the same logits at the same parameter value θ. The per-example NTP loss is

$$\mathcal{L}_{\mathrm{NTP}}(\theta; h, y) = -\sum_{u \in \mathcal{P}(y)} \log q_\theta(a_u^+ \mid h, u).$$

Proposition A.2 (Exact loss decomposition and support gradients). For every $y \in \mathcal{I}$,

$$\begin{aligned} \mathcal{L}_{\mathrm{NTP}}(\theta; h, y) &= \mathcal{L}_{\mathrm{tree}}(\theta; h, y) + \mathcal{L}_{\mathrm{support}}(\theta; h, y), \\ \mathcal{L}_{\mathrm{tree}}(\theta; h, y) &= -\log \bar{p}_\theta(y \mid h), \\ \mathcal{L}_{\mathrm{support}}(\theta; h, y) &= \sum_{u \in \mathcal{P}(y)} \ell_{\mathrm{support}}(u) \geq 0, \qquad \ell_{\mathrm{support}}(u) = -\log \alpha_\theta(h, u). \end{aligned} \tag{19}$$

At a visited prefix u, the output-coordinate derivatives are

$$\frac{\partial \ell_{\mathrm{support}}(u)}{\partial z_{u,a}} = \begin{cases} q_\theta(a \mid h, u) - \bar{q}_\theta(a \mid h, u), & a \in \mathcal{A}(u), \\ q_\theta(a \mid h, u), & a \notin \mathcal{A}(u). \end{cases} \tag{20}$$

Proof. For each target continuation, $q_\theta(a_u^+ \mid h, u) = \alpha_\theta(h, u)\bar{q}_\theta(a_u^+ \mid h, u)$. Taking negative logarithms and summing over $\mathcal{P}(y)$ gives the decomposition. Nonnegativity follows from $0 < \alpha_\theta(h, u) \leq 1$. Furthermore, $\ell_{\mathrm{support}}(u) = \log Z_{\mathcal{V}}(u) - \log Z_{\mathcal{A}}(u)$. The derivative of the first term is $q_\theta(a \mid h, u)$; the derivative of the second is $\bar{q}_\theta(a \mid h, u)$ for a valid child and zero otherwise, proving the gradient formula. □

For a valid child, the support gradient is equivalently $-(1-\alpha_\theta(h,u))\bar{q}_\theta(a \mid h,u)$. The term therefore introduces competition between the legal set and invalid continuations at visited prefixes. Although its value depends on the total legal mass, a gradient step on these logits can also change relative legal logits. Thus, standard NTP includes an explicit valid-support learning objective in addition to the legal-tree likelihood. Legal-branch training removes this term while retaining target-path branch supervision. This motivates the support-normalization control in Section 6.3. If invalid-continuation competition were responsible for the observed disadvantage of autoregressive SID prediction, removing this term should improve ranking. Instead, legal-branch training reduces performance under the same constrained decoder. The main RQ2 test therefore does not support invalid-continuation competition as an explanation for the prediction-formulation gap.

Local versus catalog-level renormalization. For each catalog item i, let $\gamma_\theta(i \mid h)$ denote the product of $\alpha_\theta(h,u)$ over $u \in \mathcal{P}(i)$. The same factorization gives

$$\bar{p}_\theta(i \mid h) = \frac{q_\theta(c(i) \mid h)}{\gamma_\theta(i \mid h)}, \qquad q_{\theta,\mathcal{I}}(i \mid h) = \frac{q_\theta(c(i) \mid h)}{\sum_{j \in \mathcal{I}} q_\theta(c(j) \mid h)}. \tag{21}$$

Here $q_{\theta,\mathcal{I}}$ is the original sequence distribution conditioned on producing a catalog-valid sequence. Since all catalog-path probabilities are positive, the two distributions are equal if and only if $\gamma_\theta(i \mid h)$ is constant across catalog items: equality forces each $\gamma_\theta(i \mid h)$ to equal the common catalog normalizer, and the converse follows by normalization of $\bar{p}_\theta$. Consequently, valid-child renormalization can change rankings within the legal catalog. Restricting admissible paths, changing their local normalization, and approximating search are separate operations, as reflected in Appendix C.4.

A.3 Logit Gradients and Subtree Supervision

We first regard valid branch logits $z = (z_{u,a})$ as output coordinates, with $r(a \mid h,u) = e^{z_{u,a}} / \sum_{b \in \mathcal{A}(u)} e^{z_{u,b}}$. An output-coordinate derivative holds the other output coordinates fixed; it does not assume that a neural model has independent parameters for these outputs. For the item scorer, let $s = (s_i)_{i \in \mathcal{I}}$ be finite scores and $p_s(i \mid h) = e^{s_i} / \sum_j e^{s_j}$.

Proposition A.3 (Target-path and catalog-wide derivatives). For $\mathcal{L}_{\text{tree}} = -\log p_r(y \mid h)$ and $\mathcal{L}_{\text{item}} = -\log p_s(y \mid h)$,

$$\frac{\partial \mathcal{L}_{\text{tree}}}{\partial z_{u,a}} = \begin{cases} r(a \mid h,u) - \mathbf{1}[a = a_u^+], & u \in \mathcal{P}(y), \\ 0, & u \notin \mathcal{P}(y), \end{cases} \qquad \frac{\partial \mathcal{L}_{\text{item}}}{\partial s_i} = p_s(i \mid h) - \mathbf{1}[i = y]. \tag{22}$$

Proof. The tree loss is $\sum_{u \in \mathcal{P}(y)} [\log \sum_{b \in \mathcal{A}(u)} e^{z_{u,b}} - z_{u,a_u^+}]$. Differentiation gives the first case; logits at other prefixes do not occur in this expression. Differentiating $\log \sum_j e^{s_j} - s_y$ gives the item-score formula. For standard NTP, the same calculation uses $\mathcal{V}_{|u|+1}$ instead of $\mathcal{A}(u)$ and q_θ instead of r; the restriction to visited prefixes remains. □

Nested target sets. Let $\mathcal{C}_\ell = \mathcal{I}_{c_{1:\ell}(y)}$, with $\mathcal{C}_0 = \mathcal{I}$. The prefix partitions and injectivity give $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \supseteq \mathcal{C}_L = \{y\}$. The decision at position ℓ partitions $\mathcal{C}_{\ell-1}$; later visited decisions further partition its target child $\mathcal{C}_\ell$. These inclusions need not be strict. At a node with one valid child, the local legal-tree loss and all its logit derivatives are zero, despite the node being visited.

Proposition A.4 (Non-target subtrees and internal odds). Take $u \in \mathcal{P}(y)$ and $a \in \mathcal{A}(u)$ with $a \neq a_u^+$, and write $S = \mathcal{I}_{ua}$. The current tree loss is independent, in output coordinates, of every internal logit at a prefix extending ua. For any $j,k \in S$,

$$\frac{p_r(j \mid h)}{p_r(k \mid h)} = \frac{p_r(j \mid h,ua)}{p_r(k \mid h,ua)}. \tag{23}$$

Thus any change that holds the internal conditionals of S fixed preserves these odds, while it may change the total probability assigned to S.

Proof. No prefix extending ua belongs to $\mathcal{P}(y)$, so the loss expression in Proposition A.3 contains none of its internal logits. Let $\rho_r(v \mid h)$ denote the product of branch probabilities on the path from

the root to a valid prefix v, with $\rho_r(\epsilon \mid h) = 1$. For any $i \in S$, $p_r(i \mid h) = \rho_r(ua \mid h)p_r(i \mid h, ua)$. The common, strictly positive factor cancels in the ratio. Lemma A.1 also gives $\sum_{i \in S} p_r(i \mid h) = \rho_r(ua \mid h)$. Consequently, the derivative at the entry branch is

$$\frac{\partial \mathcal{L}_{\text{tree}}}{\partial z_{u,a}} = r(a \mid h, u) = \frac{\sum_{i \in S} p_r(i \mid h)}{\sum_{i \in \mathcal{I}_u} p_r(i \mid h)}. \tag{24}$$

It acts on the subtree's conditional mass through its shared entry branch, while the current loss has no direct derivatives with respect to its internal logits. □

The proposition expresses the derivative at a non-target subtree's shared entry branch in terms of its conditional mass, with no direct derivatives for its internal branch logits on this example. Examples whose targets lie inside S supervise the branching nodes on their own target paths within S. Shared-parameter effects on unvisited nodes are analyzed in Section A.4.

Corollary A.5 (One update in independent output coordinates). Suppose a legal tree and an item softmax initially induce the same strictly positive catalog distribution p. In each model, treat its output coordinates as independent optimization variables and take one ordinary gradient-descent step of size $\eta > 0$ on the current example's loss, with no regularization or optimizer state. Let $p^{\text{new}}_{\text{tree}}$ and $p^{\text{new}}_{\text{item}}$ be the resulting distributions. For j, k in the same non-target subtree S of Proposition A.4,

$$\begin{aligned} \log \frac{p^{\text{new}}_{\text{tree}}(j \mid h)}{p^{\text{new}}_{\text{tree}}(k \mid h)} - \log \frac{p(j \mid h)}{p(k \mid h)} &= 0, \\ \log \frac{p^{\text{new}}_{\text{item}}(j \mid h)}{p^{\text{new}}_{\text{item}}(k \mid h)} - \log \frac{p(j \mid h)}{p(k \mid h)} &= -\eta\big(p(j \mid h) - p(k \mid h)\big). \end{aligned} \tag{25}$$

Proof. All internal coordinates of S have zero tree-loss derivative, so they remain fixed and Proposition A.4 applies. For the item model, both j and k are negatives, and $s_i^{\text{new}} = s_i - \eta p(i \mid h)$ for $i \in \{j, k\}$. The softmax log odds equal $s_j - s_k$, giving the second identity exactly. □

For example, let the probabilities of (A, x), (A, y), (B, x), and (B, y) be 0.2, 0.1, 0.6, and 0.1, respectively, with target (A, x). The odds between (B, x) and (B, y) stay at 6 after the tree-coordinate update and become $6e^{-0.5\eta}$ after the item-coordinate update. The example therefore exhibits different gradient-descent dynamics despite identical initial item distributions and likelihood values. In these independent coordinates, the tree update preserves within-subtree odds, whereas the item-score update changes them whenever the two negative probabilities differ. This result characterizes a difference in update geometry under independent output coordinates; it is not a claim that the tree parameterization is universally less expressive or that every shared-parameter implementation must exhibit the same update.

A.4 Shared Parameters and Practical Prediction Parameterization

From output derivatives to parameter updates. Let $z_\theta(h)$ collect all valid branch logits for a fixed history, and let $J_z = \partial z_\theta(h)/\partial\theta$. For item scores $s_\phi(h)$, define $J_s = \partial s_\phi(h)/\partial\phi$. Assuming differentiability, the chain rule gives

$$\nabla_\theta \mathcal{L}_{\text{tree}} = J_z^\top g_z, \qquad \nabla_\phi \mathcal{L}_{\text{item}} = J_s^\top g_s, \tag{26}$$

where g_z and g_s are the output derivatives in Eq. (22). These Jacobians include every dependence through the history encoder, code embeddings, and prediction head. For stochastic networks, the differential statements are conditional on a fixed realization of their stochastic operations.

For an ordinary parameter-space gradient step $\theta^{\text{new}} = \theta - \eta J_z^\top g_z$, first-order expansion at the current θ yields

$$z_{\theta^{\text{new}}}(h) - z_\theta(h) = -\eta J_z J_z^\top g_z + o(\eta) \quad \text{as } \eta \to 0. \tag{27}$$

The matrix $J_z J_z^\top$ can couple visited and unvisited coordinates, so shared parameters can change an unvisited prediction even when its output-coordinate derivative is zero. Per-item score derivatives are likewise propagated through shared parameters in NRP, which shares both the composer and the code embeddings across items. The output gradients specify which coordinates receive direct supervision, and the Jacobians determine how that supervision updates shared parameters. Practical

learning thus depends on both the supervision structure and the parameterization through which it is propagated. The expansion describes ordinary gradient descent; the experiments evaluate the resulting models trained with AdamW.

Proposition A.6 (A score-induced tree preserves loss and gradients). Let $s_\phi(h, i)$ be any finite, differentiable item scores. For every valid prefix u, define $Z_\phi(u) = \sum_{i \in \mathcal{I}_u} \exp s_\phi(h, i)$ and, for $a \in \mathcal{A}(u)$, set

$$r^{\mathrm{ind}}_\phi(a \mid h, u) = \frac{Z_\phi(ua)}{Z_\phi(u)}. \tag{28}$$

Then the induced legal tree and the original item softmax have identical item distributions, losses, and gradients with respect to the same underlying parameters ϕ:

$$\begin{aligned} p^{\mathrm{ind}}_{\mathrm{tree},\phi}(i \mid h) &= \frac{\exp s_\phi(h, i)}{Z_\phi(\epsilon)} = p_\phi(i \mid h), \\ \mathcal{L}^{\mathrm{ind}}_{\mathrm{tree}}(\phi; h, y) &= \mathcal{L}_{\mathrm{item}}(\phi; h, y), \\ \nabla_\phi \mathcal{L}^{\mathrm{ind}}_{\mathrm{tree}} &= \nabla_\phi \mathcal{L}_{\mathrm{item}}. \end{aligned} \tag{29}$$

Proof. The child subtrees partition $\mathcal{I}_u$, so $Z_\phi(u) = \sum_{a \in \mathcal{A}(u)} Z_\phi(ua)$. Thus Eq. (28) defines positive normalized conditionals; they are also local softmaxes with logits $\log Z_\phi(ua)$. Along any complete item path, the partition functions telescope:

$$\prod_{\ell=1}^{L} \frac{Z_\phi(c_{1:\ell}(i))}{Z_\phi(c_{<\ell}(i))} = \frac{Z_\phi(c(i))}{Z_\phi(\epsilon)} = \frac{\exp s_\phi(h, i)}{\sum_{j \in \mathcal{I}} \exp s_\phi(h, j)}. \tag{30}$$

The losses are therefore identical functions of ϕ, not merely equal at a single matching distribution. Differentiability then gives equality of their parameter gradients. □

This construction realizes equivalence by deriving local logits from subtree partition sums of the same underlying item scores. Their Jacobians propagate the target-path loss through those aggregates and recover the full item-score gradient. These are coupled functions of item scores, rather than the independent optimization coordinates in Corollary A.5. The autoregressive heads in our controlled comparisons directly predict branch logits; they are not constrained to be the score-induced conditionals of the compared item scorer. The proposition therefore identifies the parameter dependence that makes this construction equivalent and clarifies why a common likelihood interpretation does not by itself equate the learning dynamics of separately parameterized predictors.

Implications for the decoupling hypothesis. The analysis therefore does not identify autoregressive factorization itself as inherently deficient. Rather, it identifies a distinction in how the practical parameterizations studied here expose and propagate supervision. The progressively controlled experiments in Section 6.3 test whether this distinction remains consequential after alternative explanations are increasingly controlled. Their results support the decoupling principle under the evaluated parameterizations. NRP instantiates this principle by retaining shared SID composition while directly optimizing complete-item scores.

B Additional Experimental Details

B.1 Data preparation and evaluation

We use the Beauty, Sports and Outdoors (Sports), and Toys and Games (Toys) categories from the official Amazon Reviews 2014 5-core subsets. Each review is treated as an interaction without additional rating filtering or deduplication. Records with missing identifiers are discarded, and users with at least five interactions are retained, without further iterative 5-core filtering. Table 5 summarizes the datasets before sequence splitting and history truncation.

Interactions are ordered by timestamp, with ties resolved by original record order. The penultimate and final interactions are reserved for validation and testing, respectively. Each evaluation history contains the last 20 interactions preceding its target, including the validation interaction at test time.

Table 5: Dataset statistics before sequence splitting and history truncation. Density is the number of interactions divided by the product of the user and item counts.

Dataset	Users	Items	Interactions	Density	Sequence length	
					Mean	Median
Beauty	22,363	12,101	198,360	0.00073	8.87	6
Sports	35,598	18,357	296,175	0.00045	8.32	6
Toys	19,412	11,924	167,526	0.00072	8.63	6

NRP trains on the last 20 items of each training prefix, prepended with a beginning-of-sequence token, and minimizes mean cross-entropy over all valid next-item positions.

Evaluation uses the full catalog without sampled negatives. Exact scorers rank all items, while retrieval-based baselines use their respective search procedures. Test filtering excludes items from the complete, untruncated pre-target history, except the target itself. Standard validation and training softmax denominators apply no history filtering; cold-start validation follows Appendix E. Recall and NDCG are averaged equally over users, with unretrieved targets contributing zero. We report $K \in \{5, 10\}$ for standard evaluation and $K \in \{10, 20\}$ for cold-start experiments.

B.2 NRP IMPLEMENTATION DETAILS

The main NRP configuration uses fixed SIDs (c_1, c_2, c_3, u) derived from 768-dimensional Sentence-T5-base item features. The first three positions are obtained by residual k-means with 256 entries per level. The suffix u disambiguates items sharing the semantic positions; its cardinality is 6, 15, and 14 on Beauty, Sports, and Toys, respectively. Complete SIDs uniquely identify catalog items, and their discrete assignments remain fixed during recommendation training. The embedding parameters associated with these assignments are learned.

Composer architecture. The composer g_ψ in Eq. (11) first concatenates the code embeddings and projects them to width w:

$$\begin{aligned} x_i &= [E_1[c_1(i)]; \ldots; E_L[c_L(i)]] \in \mathbb{R}^{Ld}, \\ v_i^{(0)} &= \sigma(W_{\text{in}} x_i + b_{\text{in}}) \in \mathbb{R}^w, \end{aligned} \tag{31}$$

where σ denotes GELU. It then applies D residual blocks with LayerNorm before each feed-forward transformation:

$$\begin{aligned} v_i^{(b)} &= v_i^{(b-1)} + \mathrm{FFN}_b\left(\mathrm{LN}_b(v_i^{(b-1)})\right), \quad b = 1, \ldots, D, \\ \mathrm{FFN}_b(v) &= \mathcal{D}_p\left(W_{b,2}\, \mathcal{D}_p\left(\sigma(W_{b,1} v + b_{b,1})\right) + b_{b,2}\right), \end{aligned} \tag{32}$$

where $\mathcal{D}_p$ denotes dropout with probability p. The two affine layers expand and contract the hidden dimension as $w \to 2w \to w$. A final LayerNorm and affine projection produce the complete-item representation:

$$z_i = W_{\text{out}}\, \mathrm{LN}_{\text{out}}(v_i^{(D)}) + b_{\text{out}} \in \mathbb{R}^d. \tag{33}$$

Four position-specific embedding tables produce 128-dimensional vectors. Their concatenation is processed by a residual MLP composer with width 256 and six blocks, followed by final LayerNorm and a projection to dimension 128. Catalog mean centering is used for history representations and candidate scoring; output ℓ_2 normalization is not used. The causal backbone is a randomly initialized GPT-2 architecture with four layers, four heads, hidden size 128, feed-forward width 512, and dropout 0.1. Each historical item occupies one backbone position. Training uses AdamW with learning rate 5×10^{-5}, weight decay 0.01, batch and effective batch size 64, and gradient clipping at 1.0. Validation is performed every epoch, with a minimum of 20 and a maximum of 250 epochs and early-stopping patience 10. Checkpoints are selected by validation NDCG@10, with lower item negative log-likelihood breaking ties.

Random seeds and result aggregation. Unless stated otherwise, parameter counts in the appendix refer to trainable parameters participating in the forward computation. Composer-only counts are explicitly identified where used for matched-capacity analyses. The same precomputed SID assignments are reused across recommendation-training seeds within each experimental configuration. Main NRP results use seeds 2020–2024. The three-seed RQ2 controlled comparisons use paired seeds 2020–2022. For five-seed NRP results, we report the population standard deviation (denominator n), whereas the three-seed RQ2 comparisons use the sample standard deviation (denominator $n-1$). Standard deviations describe variation across random seeds rather than confidence intervals or significance tests.

B.3 Baseline details

Conventional sequential recommendation.

- **SASRec** (Kang & McAuley, 2018) uses causal self-attention to encode interaction histories and predicts the next item by matching the resulting history representation against learned item embeddings.
- **BERT4Rec** (Sun et al., 2019) learns bidirectional sequence representations through masked-item prediction. At inference, it appends a masked position to the interaction history to predict the next item.

Generative recommendation.

- **TIGER** (Rajput et al., 2023) constructs Semantic IDs by quantizing item content embeddings with RQ-VAE. An encoder–decoder Transformer autoregressively generates the next item's Semantic ID.
- **EAGER** (Wang et al., 2024b) uses a shared encoder and separate autoregressive decoders for behavioral and semantic codes. Auxiliary contrastive and transfer objectives support collaboration between the two streams, whose predictions are combined through confidence-based ranking.
- **COBRA** (Yang et al., 2025b) combines sparse semantic IDs and trainable dense representations through cascaded generation. It first predicts sparse IDs, then generates conditioned dense vectors, combining beam-search and similarity scores for item retrieval.
- **RPG** (Hou et al., 2025) represents items with long Semantic IDs and predicts their constituent codes in parallel. It retrieves candidate items through iterative graph-constrained decoding over similar identifiers.
- **SETRec** (Lin et al., 2025) represents items as sets of continuous tokens encoding collaborative and semantic information. It generates these tokens simultaneously and grounds them to catalog items through similarity-based scoring.
- **DiffGRM** (Liu et al., 2026) generates Semantic IDs using masked discrete diffusion. Its denoising model uses bidirectional context across code positions and confidence-guided parallel decoding to produce candidate identifiers.
- **VQ-Rec** (Hou et al., 2023) quantizes item text representations into discrete codes using optimized product quantization and constructs item representations by mean pooling shared code embeddings. A self-attentive sequence encoder models interaction histories and scores candidates using these representations. Its original framework further supports transfer through contrastive pre-training and code-embedding alignment.

Baseline implementations and result sources. We reimplement SASRec, BERT4Rec, TIGER, RPG, and SETRec following their original model formulations. We additionally evaluate a backbone-adapted implementation of VQ-Rec (Hou et al., 2023), denoted VQ-Rec (GPT-2). Our reimplemented baseline results in Tables 1 and 6 are averaged over five random seeds. All locally evaluated models follow the same evaluation protocol as NRP, described in Appendix B.1. Our reproduction attempts for EAGER, COBRA, and DiffGRM did not achieve performance comparable to their published results under our experimental setup. We therefore use their originally reported results in Table 1.

Table 6: Published reference results and our reimplementations. Ours denotes mean performance over five random seeds; Reported denotes values from the cited sources. R and N denote Recall and NDCG, respectively. Reference sources and the SETRec configuration are specified in the accompanying text.

		Beauty				Sports				Toys			
Method	Source	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
SASRec	Reported	0.0387	0.0249	0.0605	0.0318	0.0233	0.0154	0.0350	0.0192	0.0463	0.0306	0.0675	0.0374
	Ours	0.0385	0.0245	0.0615	0.0319	0.0221	0.0148	0.0330	0.0183	0.0455	0.0311	0.0673	0.0382
BERT4Rec	Reported	0.0203	0.0124	0.0347	0.0170	0.0115	0.0075	0.0191	0.0099	0.0116	0.0071	0.0203	0.0099
	Ours	0.0277	0.0180	0.0454	0.0237	0.0161	0.0104	0.0271	0.0140	0.0265	0.0173	0.0396	0.0215
TIGER	Reported	0.0454	0.0321	0.0648	0.0384	0.0264	0.0181	0.0400	0.0225	0.0521	0.0371	0.0712	0.0432
	Ours	0.0401	0.0270	0.0624	0.0341	0.0233	0.0154	0.0374	0.0193	0.0398	0.0270	0.0574	0.0327
RPG	Reported	0.0550	0.0381	0.0809	0.0464	0.0314	0.0216	0.0463	0.0263	0.0592	0.0401	0.0869	0.0490
	Ours	0.0569	0.0405	0.0786	0.0475	0.0307	0.0211	0.0441	0.0254	0.0627	0.0450	0.0871	0.0528
SETRec	Reported	0.0106	0.0083	0.0161	0.0103	0.0114	0.0101	0.0185	0.0126	0.0110	0.0089	0.0189	0.0118
	Ours	0.0405	0.0273	0.0626	0.0344	0.0233	0.0148	0.0348	0.0185	0.0415	0.0279	0.0633	0.0349
VQ-Rec (GPT-2)	Ours	0.0557	0.0393	0.0798	0.0470	0.0292	0.0195	0.0450	0.0246	0.0554	0.0393	0.0791	0.0469

Table 6 presents published reference results alongside our reimplementations. For SASRec and BERT4Rec, reference values are taken from the benchmark results reported by Rajput et al. (2023). For TIGER, RPG, and SETRec, we use results from their respective original papers. The published reference results use the same Amazon benchmark categories and chronological leave-one-out setting where applicable, but may differ in preprocessing, history truncation, and method-specific inference settings. They are therefore included as published references rather than results obtained under our common implementation protocol. The SETRec reference corresponds to its T5-small configuration under the chronological split used in the original paper and is included only as a reference to the reported result. Our reimplemented SETRec result under the common evaluation protocol is used in the main comparison.

VQ-Rec configuration. Our VQ-Rec implementation uses the same GPT-2 backbone architecture as NRP: four layers, four attention heads, hidden dimension 128, feed-forward dimension 512, and dropout 0.1. Its discrete codes are constructed by fitting OPQ separately to each dataset's Sentence-T5 item features, using 32 codebooks with 256 entries each. Training uses a fixed learning rate of 10^{-3}, validation after every epoch, and early-stopping patience ten. Checkpoints are selected by validation NDCG@10, with ties resolved by item negative log-likelihood. This comparison aligns the backbone architecture and evaluation protocol, while retaining different tokenization and optimization settings. The reported results correspond to this backbone-adapted variant of VQ-Rec.

B.4 Analytical Contribution and Relation to VQ-Rec and DiGR

From separation analysis to a design principle. The central contribution of this work is an analysis of SID-based recommendation that separates representation sharing from prediction parameterization. We examine this separation at two levels: the catalog distributions a predictor can represent and the supervision its practical parameterization induces during training. A legal SID tree with unrestricted node conditionals can represent any strictly positive catalog distribution. Under teacher forcing, direct branch-logit supervision acts at target-path prefixes: later decisions refine nested target subtrees, while off-path subtrees compete through shared entry branches without direct supervision of their internal conditionals from the current example. Full-catalog item cross-entropy instead directly supervises individual complete-item scores. This analysis distinguishes distributional expressivity from the organization of learning (Section 4; Appendix A). It motivates a design principle: use SIDs to share representational components across items while organizing direct supervision and competition at the complete-item level.

Progressively controlled examination. We examine this principle through a sequence of comparisons that progressively controls alternative explanations for the observed prediction-formulation gap (Table 2). These comparisons address support normalization, history representation, architectural differences, and model capacity. The closest comparison uses the same composed-item history design, composer architecture, and causal backbone, with paired initialization of shared components and exact catalog ranking for both formulations. Appendix C.2 further evaluates standard full-vocabulary cross-entropy alongside the legal-branch objective used in the main comparison. Direct item scoring remains stronger under both evaluated autoregressive training objectives. The combined evidence shows that the performance gap persists under increasingly matched designs, including exact ranking and alternative training normalizations.

NRP as an instantiation and composition as a design space. NRP is one concrete instantiation of this principle. It constructs complete-item representations from shared SID embeddings and reuses a learned composer for both history encoding and candidate scoring. The principle itself leaves the composition function open. We therefore also investigate Mean, Scaled Sum, Sum, shallow MLP, and residual MLP composition (Table 19), together with composer depth, width, and component choices (Appendix D). Here, Mean serves as an aggregation control within NRP, rather than an end-to-end reproduction of DiGR. These studies examine how to construct effective shared representations once complete-item prediction is fixed. They characterize the available composition choices and support the residual MLP adopted in our implementation.

Relation to existing code-based item scoring. VQ-Rec (Hou et al., 2023) constructs item representations from shared code embeddings and performs item-level prediction within a transferable recommendation framework. DiGR (Zhao et al., 2026) explicitly separates SID-based representation from token prediction, optimizes item-level likelihoods, and analyzes ranking recovery, including exhaustive catalog-scoring comparisons. Our study identifies how prediction parameterization organizes direct supervision even when unrestricted catalog-level expressivity is preserved. This separation provides the analytical basis for the design principle, which we examine through progressively matched comparisons under exact catalog ranking and both autoregressive training objectives.

The empirical comparison also includes VQ-Rec (GPT-2) in Table 6. Under the aligned backbone architecture and evaluation protocol, NRP outperforms VQ-Rec (GPT-2) across all reported metrics on all three datasets (Tables 1 and 6). The remaining tokenization and optimization differences are documented in Appendix B.3.

C Additional Controls for the Decoupling Hypothesis

This section supplements the progressively controlled hypothesis tests in Section 6.3. The following analyses examine residual alternative explanations for the observed advantage of complete-item scoring, including differences in training normalization, model capacity, decoding approximation, backbone architecture, and SID length. These experiments are supplementary robustness checks rather than independent sources of the decoupling hypothesis.

C.1 Design and Matching of the RQ2 Controls

Table 7 details how the progressively controlled comparisons in Section 6.3 are constructed. The controls should be read sequentially: the SID-history comparison holds the history-encoding design fixed while changing the prediction interface, whereas the composed-history comparison further aligns the composer, causal backbone, initialization of common components, parameter budget, and ranking procedure.

The SID-history variants retain the same expanded history format, including beginning-of-sequence and item-separator tokens, and use paired encoder initialization. Free-item and composed-item scoring predict at item boundaries: the beginning-of-sequence state predicts the first retained item, and each separator state predicts the following item. The target's SID is not provided before making that item prediction.

Table 7: RQ2 architectures. SID and item histories denote expanded SID tokens and composed item vectors. Parameter counts are in millions; the item-specific column counts parameters in independent candidate embedding/classifier tables. The two legal SID predictors use different history representations.

Variant	History	Item-specific Beauty/Toys	Total Beauty/Toys	Objective	Ranking
Legal SID prediction	SID	0 / 0	0.982 / 0.983	Branch CE	Beam20
Free-item scoring	SID	1.549 / 1.526	2.531 / 2.509	Item CE	Exact
Composed-item scoring	SID	0 / 0	2.743 / 2.744	Item CE	Exact
Legal SID prediction	Item	0 / 0	2.743 / 2.744	Branch CE	Exact
NRP	Item	0 / 0	2.743 / 2.744	Item CE	Exact

Free-item scoring uses a bias-free $N \times 128$ classifier, while composed-item scoring ties its composer inputs to the SID-history embedding rows and introduces no independent item vectors. The common training windows provide 138,622 and 116,679 next-item events on Beauty and Toys, respectively.

In the composed-history comparison, the autoregressive predictor and NRP use the same composer architecture, causal backbone architecture, and training examples, with paired initialization of common components. During training, historical items are represented by composed item vectors, and the ground-truth target SID prefix is appended as individual SID-token embeddings. The causal backbone processes this sequence to predict the next SID token. Each conditional softmax is normalized only over the valid continuations of the current prefix, yielding the teacher-forced legal-branch cross-entropy objective. NRP instead optimizes cross-entropy over complete catalog items. Both models compute exact full-catalog rankings. The autoregressive predictor evaluates each catalog item's legal path log-probability using that item's own SID prefixes, whereas NRP ranks explicit item scores. Both use the same learning rate, epoch limits, and patience, and select checkpoints by validation NDCG@10. Ties are broken by Recall@10 for the autoregressive predictor and by item negative log-likelihood for NRP. For the three SID-history variants, Recall@10 also breaks validation NDCG@10 ties.

For the training-normalization comparison, standard and legal-branch CE retain the SID-history architecture and use the same decoder with valid-child normalization. Legal-branch CE changes the training denominator at target prefixes. This comparison is distinct from changing only the decoder of an already trained checkpoint, examined in Appendix C. The controls support conclusions about the evaluated prediction parameterizations.

C.2 Training-Objective Robustness under the Matched Composed-History Design

The strongest main-text control compares NRP with an autoregressive SID predictor trained using legal-branch cross-entropy under closely matched composed-item designs. To examine whether the observed advantage depends on this training normalization, we repeat the autoregressive experiment using standard full-vocabulary cross-entropy (Standard CE), retaining the same composed-item history design, composer architecture, causal backbone, and training configuration.

As in the legal-branch variant, historical items are represented by composed item vectors. During training, the ground-truth target SID prefix is appended as individual SID-token embeddings, and the causal backbone processes this sequence to predict the next SID token. Standard CE uses teacher-forced next-token prediction with each conditional softmax normalized over the full SID-token vocabulary at that position, including continuations that are invalid for the current prefix. Thus, the training denominator is not restricted to legal children.

For each of seeds 2020–2022, the Standard-CE checkpoint is selected by validation NDCG@10 under Beam20 decoding. We then evaluate both Beam20 decoding and exact catalog ranking using the same selected checkpoint. Exact evaluation scores every catalog-valid item path using that item's own SID prefixes. Table 8 reports the resulting three-seed means and sample standard deviations, together with the legal-branch and NRP reference results.

Table 8: Training-objective robustness under the composed-history design. Results are means with sample standard deviations over three paired seeds (2020–2022). Standard CE uses full-vocabulary token-level normalization during training, whereas Legal-branch CE normalizes over valid continuations. Exact evaluation ranks all catalog-valid item paths. Standard CE Beam20 and Exact use the same selected checkpoint for each seed. R and N denote Recall and NDCG, respectively.

		Beauty				Toys			
Training	Ranking	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
Standard CE	Beam20	0.0348 ±0.0017	0.0242 ±0.0018	0.0506 ±0.0025	0.0293 ±0.0020	0.0287 ±0.0017	0.0203 ±0.0017	0.0429 ±0.0028	0.0248 ±0.0021
Standard CE	Exact	0.0352 ±0.0017	0.0244 ±0.0018	0.0516 ±0.0027	0.0297 ±0.0021	0.0290 ±0.0017	0.0204 ±0.0017	0.0434 ±0.0026	0.0251 ±0.0020
Legal-branch CE	Exact	0.0391 ±0.0014	0.0275 ±0.0015	0.0544 ±0.0013	0.0324 ±0.0015	0.0328 ±0.0011	0.0236 ±0.0010	0.0487 ±0.0019	0.0287 ±0.0013
NRP	Exact	**0.0640** ±0.0012	**0.0446** ±0.0010	**0.0927** ±0.0005	**0.0538** ±0.0008	**0.0658** ±0.0005	**0.0471** ±0.0003	**0.0939** ±0.0021	**0.0561** ±0.0004

For the selected Standard-CE checkpoints, exact catalog ranking yields only small improvements over Beam20 in the reported recommendation metrics. NDCG@10 increases from 0.0293 to 0.0297 on Beauty and from 0.0248 to 0.0251 on Toys. Standard CE nevertheless remains substantially below NRP under exact ranking: NRP achieves NDCG@10 of 0.0538 versus 0.0297 on Beauty and 0.0561 versus 0.0251 on Toys. The same ordering holds across all four metrics on both datasets. These results show that NRP's advantage in the composed-history comparison persists under both evaluated autoregressive training objectives, and replacing Beam20 with exact catalog ranking does not close the gap.

C.3 Parameter-Capacity Control for AR-SID

Another possible explanation for the initial prediction-formulation gap is the smaller parameter budget of the AR-SID reference. We therefore vary AR-SID's feed-forward width over $\{512, 1024, 2224, 3072\}$ with four backbone layers, and its backbone depth over $\{2, 4, 6, 8\}$ with feed-forward width 512. Hidden size 128 and four attention heads remain fixed. For each new configuration, learning rates 10^{-4} and 3×10^{-4} are compared using validation NDCG@10; the standard four-layer configuration is reused as the reference. Tables 9 and 10 report the selected learning rate and checkpoint epoch. The width-2224 predictors have approximately the same parameter counts as NRP: 2.742M versus 2.743M on Beauty, and 2.743M versus 2.744M on Toys.

Increasing capacity yields modest gains, but does not close the ranking gap to NRP in these sweeps. The best Beauty AR-SID variant has width 2224. On Toys, eight backbone layers give the highest Recall@10, with nearly identical NDCG@10 to the width-2224 variant. Larger models are not uniformly better, and these sweeps therefore do not support model capacity alone as an explanation for the observed gap.

C.4 Decoding, Filtering, and Normalization Controls

Post-filtering truncation. A fixed beam of width B retains at most B paths at every SID position, then filters previously interacted items after decoding, without replenishing the list. Table 11 reports both ranking quality and returned-list coverage: Mean returned is the average number of retained items, and Coverage@10 is the fraction of users with at least ten retained items. All test users remain in metric averages, including those with fewer than ten recommendations. At width 10, only about half the users receive ten items; widths 20 and 30 largely restore coverage and have similar ranking scores. This control measures the combined effects of search and post-filtering list truncation.

Search approximation. The reference decoder reserves additional search capacity for paths that may be removed by history filtering, targeting B retained items rather than fixing every intermediate beam to B paths. Table 12 compares this procedure with exhaustive ranking over catalog-valid SID paths (Full). Beam20 is close to Full on both datasets, suggesting that search approximation contributes little to the observed reference gap. Compensated Beam10 also avoids the shortage of ten returned items, but is not an equal-computation counterpart of fixed Beam10.

Table 9: AR-SID capacity on Beauty (seed 2020). LR is in units of 10^{-4}. Params denotes trainable parameters participating in the forward computation.

Configuration	Params (M)	LR	Epoch	R@10	N@10
Feed-forward width (four backbone layers)					
512	0.982	3	60	0.0601	0.0322
1024	1.508	3	60	0.0611	0.0327
2224	2.742	1	90	0.0625	0.0339
3072	3.613	1	100	0.0580	0.0315
Backbone layers (feed-forward width 512)					
2	0.585	1	220	0.0595	0.0324
4	0.982	3	60	0.0601	0.0322
6	1.378	3	60	0.0603	0.0326
8	1.775	1	110	0.0571	0.0312
NRP	2.743	0.5	95	0.0929	0.0535

Table 10: AR-SID capacity on Toys (seed 2020). LR is in units of 10^{-4}. NRP is included as a reference with its fixed composer and training configuration.

Configuration	Params (M)	LR	Epoch	R@5	N@5	R@10	N@10
Feed-forward width (four backbone layers)							
512	0.983	3	50	0.0352	0.0234	0.0550	0.0298
1024	1.509	1	100	0.0353	0.0235	0.0561	0.0302
2224	2.743	1	105	0.0367	0.0251	0.0580	0.0320
3072	3.614	1	90	0.0355	0.0234	0.0561	0.0300
Backbone layers (feed-forward width 512)							
2	0.586	1	180	0.0336	0.0226	0.0524	0.0286
4	0.983	3	50	0.0352	0.0234	0.0550	0.0298
6	1.379	1	120	0.0367	0.0244	0.0559	0.0306
8	1.776	1	110	0.0382	0.0254	0.0587	0.0320
NRP	2.744	0.5	137	0.0656	0.0468	0.0959	0.0566

Inference-time normalization. Catalog-valid search and valid-child normalization are distinct: the former restricts admissible paths, whereas the latter also renormalizes conditional probabilities over valid children at each prefix. For each of seeds 2020–2022, we reuse the same standard-CE checkpoint and compare its original conditional probabilities with valid-child-normalized probabilities, using history-compensated Beam20 in both cases. Table 13 shows small, dataset-dependent changes. This inference-only intervention differs from the training-loss comparison in Section 6.3, where both training variants are evaluated with valid-child normalization.

Taken together, these controls show that beam width, post-decoding history filtering, and inference-time valid-child normalization affect AR-SID ranking only modestly once adequate search capacity is used. They therefore do not provide an alternative explanation for the substantially larger prediction-formulation gap examined in RQ2.

C.5 Backbone Robustness

This experiment is not part of the progressive matching chain in Table 2; instead, it tests whether the observed NRP advantage is specific to the GPT-2-style backbone. We replace the GPT-2 backbone with a LLaMA-style architecture using four layers, four heads, hidden size 128, rotary positional embeddings, RMSNorm, and a SwiGLU intermediate size of 403. The backbone counts are closely matched: 882,432 parameters for LLaMA and 882,176 for GPT-2. This matches backbone capacity across architectures, not total parameter counts across Pure-ID, AR-SID, and NRP. Table 14 reports all four metrics and total model parameter counts. NRP outperforms both references under either

Table 11: Fixed-width beam search with post-decoding history filtering and no replenishment (seed 2020). Coverage@10 is a fraction rather than a recommendation recall metric.

Dataset	B	R@5	N@5	R@10	N@10	Mean returned	Coverage @10
Beauty	10	0.0384	0.0250	0.0531	0.0298	9.1620	0.4705
	20	0.0394	0.0255	0.0606	0.0323	18.7160	0.9978
	30	0.0393	0.0254	0.0606	0.0323	28.4050	0.9996
Toys	10	0.0346	0.0232	0.0488	0.0279	9.2800	0.5180
	20	0.0351	0.0233	0.0550	0.0298	18.9040	0.9998
	30	0.0352	0.0234	0.0552	0.0298	28.6270	0.9999

Table 12: History-compensated beam search and exhaustive catalog-path ranking for the same AR-SID reference checkpoints (seed 2020).

	Beauty				Toys			
Decoder	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
Beam10	0.0390	0.0253	0.0602	0.0321	0.0351	0.0234	0.0549	0.0297
Beam20	0.0392	0.0254	0.0601	0.0322	0.0352	0.0234	0.0550	0.0298
Beam30	0.0394	0.0255	0.0605	0.0323	0.0352	0.0234	0.0550	0.0298
Full	0.0394	0.0256	0.0606	0.0324	0.0350	0.0234	0.0552	0.0298

architecture on both datasets. Under LLaMA, Pure-ID and AR-SID have mixed Recall/NDCG rankings; we therefore limit the robustness claim to NRP's performance relative to these two references in the tested architectures.

C.6 SID-Length Sensitivity of Autoregressive Prediction

We test whether the AR-SID reference is unusually disadvantaged by the default SID length by varying the number of autoregressive prediction steps. To vary AR-SID's identifier length, we construct a shared five-level residual k-means chain with 256 codes per level. An SID of length $L \in \{3, 4, 5, 6\}$ uses the first $L - 1$ residual levels and a disambiguation suffix, preserving a one-to-one mapping between complete SIDs and items. Table 15 shows a nonmonotonic dependence on SID length: Beauty performs best at $L = 5$, whereas Toys performs best at $L = 6$. No single length dominates across datasets, so the default four-position SID is not uniquely responsible for the behavior of the autoregressive predictor.

D Additional Representation Analyses

Having tested the prediction side of the decoupling principle in RQ2, we further examine its representation side. We study which properties of shared SID-based representations contribute to NRP after SID codes are no longer used as autoregressive prediction targets. We analyze catalog centering, SID construction, and composer architecture and capacity.

D.1 Effect of Catalog Centering

Catalog centering is an implementation choice within NRP rather than a requirement of the decoupling principle. We compare NRP with and without catalog centering over five random seeds, using the same composer architecture. Table 16 reports mean performance and population standard deviations.

On Sports, centered NRP achieves higher mean performance and substantially lower variability across all four metrics. In particular, the standard deviations of Recall@10 and NDCG@10 decrease from 0.0054 to 0.0003 and from 0.0034 to 0.0002, respectively, indicating more consistent

Table 13: Inference-only normalization control. Each row evaluates the same standard-trained AR-SID checkpoint with the two decoding scores.

		Original normalization		Valid-child normalization	
Dataset	Seed	R@10	N@10	R@10	N@10
Beauty	2020	0.0601	0.0322	0.0607	0.0323
	2021	0.0604	0.0330	0.0604	0.0329
	2022	0.0593	0.0325	0.0595	0.0325
Toys	2020	0.0550	0.0298	0.0544	0.0295
	2021	0.0551	0.0297	0.0551	0.0295
	2022	0.0564	0.0306	0.0558	0.0303

Table 14: Backbone comparison using three seeds. Params denotes trainable parameters participating in the forward computation; backbone-only counts are reported in the text.

Backbone	Method	Params (M)	R@5	N@5	R@10	N@10
Beauty						
LLaMA	Pure-ID	3.981	0.0409	0.0297	0.0568	0.0348
LLaMA	AR-SID	0.982	0.0377	0.0257	0.0576	0.0320
LLaMA	NRP	2.743	0.0611	0.0431	0.0887	0.0520
GPT-2	Pure-ID	3.980	0.0485	0.0367	0.0651	0.0420
GPT-2	AR-SID	0.982	0.0392	0.0254	0.0599	0.0326
GPT-2	NRP	2.743	0.0640	0.0446	0.0927	0.0538
Toys						
LLaMA	Pure-ID	3.935	0.0366	0.0278	0.0487	0.0317
LLaMA	AR-SID	0.983	0.0367	0.0244	0.0567	0.0308
LLaMA	NRP	2.744	0.0619	0.0444	0.0891	0.0531
GPT-2	Pure-ID	3.935	0.0477	0.0373	0.0622	0.0419
GPT-2	AR-SID	0.983	0.0352	0.0234	0.0555	0.0300
GPT-2	NRP	2.744	0.0658	0.0471	0.0939	0.0561

performance across random seeds. On Beauty and Toys, uncentered NRP achieves slightly higher mean scores, suggesting that the benefit is dataset-dependent.

D.2 SID Construction under Autoregressive and Decoupled Prediction

Table 17 compares AR-SID, NRP, and NRP without catalog centering under five SID constructions. Both NRP variants use the same composer architecture, and SID assignments remain fixed during recommendation training. For each dataset, RQ-VAE and OPQ are fitted to its item features, using three codebooks of size 256 and a uniqueness suffix; OPQ retains its native subspace order. All results use seed 2020.

Random SID controls. Random-Permuted randomly permutes the complete RQ-KMeans SID assignments across catalog items. It therefore preserves the set of identifiers and their code-frequency and tree statistics while breaking the association between item content and SID assignment. IID-Uniform instead samples each semantic code independently and uniformly from the corresponding 256-code vocabulary. Collisions are resolved using a disambiguation suffix so that complete SIDs remain unique across catalog items. The resulting random assignments are generated once and kept fixed throughout recommendation training.

These results provide two complementary observations. First, for centered NRP, content-based SID constructions consistently outperform random controls, supporting the value of semantic structure on the representation side of the decoupling principle. Second, NRP outperforms AR-SID under every SID construction tested, indicating that the advantage of complete-item prediction is not specific

Table 15: AR-SID with different SID lengths (seed 2020). L includes the disambiguation suffix; each semantic level contains 256 codes.

Dataset	L	Epoch	R@5	N@5	R@10	N@10
Beauty	3	60	0.0369	0.0239	0.0568	0.0303
	4	60	0.0392	0.0254	0.0601	0.0322
	5	130	0.0386	0.0255	0.0618	0.0329
	6	80	0.0357	0.0229	0.0570	0.0297
Toys	3	60	0.0363	0.0245	0.0569	0.0311
	4	50	0.0352	0.0234	0.0550	0.0298
	5	100	0.0361	0.0242	0.0561	0.0306
	6	190	0.0369	0.0253	0.0575	0.0319

Table 16: Effect of catalog centering. Each cell reports the mean with the population standard deviation below it over five seeds. Higher means within each dataset are highlighted in bold. R and N denote Recall and NDCG, respectively.

Dataset	Centering	R@5	N@5	R@10	N@10
Beauty	Without	**0.0647** ±0.0010	**0.0450** ±0.0006	**0.0931** ±0.0010	**0.0541** ±0.0005
	With	0.0638 ±0.0011	0.0443 ±0.0008	0.0928 ±0.0005	0.0536 ±0.0006
Sports	Without	0.0334 ±0.0044	0.0224 ±0.0031	0.0514 ±0.0054	0.0282 ±0.0034
	With	**0.0365** ±0.0004	**0.0248** ±0.0003	**0.0550** ±0.0003	**0.0308** ±0.0002
Toys	Without	**0.0675** ±0.0002	**0.0476** ±0.0005	**0.0962** ±0.0010	**0.0569** ±0.0005
	With	0.0664 ±0.0008	0.0475 ±0.0009	0.0944 ±0.0014	0.0565 ±0.0008

to the default RQ-KMeans tokenizer. The random-SID results are nevertheless sensitive to catalog centering, so they should not be interpreted as a general claim about arbitrary SID assignments.

D.3 Composer Depth and Parameter Budget

Having fixed the decoupled prediction formulation, we next examine architectural choices within its compositional representation mechanism. Table 18 reports all four metrics and total parameter counts for the formal depth sweeps. Depth denotes the number of composer blocks; the causal backbone remains a four-layer, four-head GPT-2 with hidden size 128 and FFN width 512. All runs use composer width 256, final hidden LayerNorm, no output ℓ_2 normalization, learning rate 5×10^{-5}, patience ten, validation every epoch, and exact history-filtered ranking. Complete sweeps are available for centered Beauty and Toys and uncentered Beauty.

Additional depth improves performance over one block, but the best depth depends on the dataset and centering configuration. On centered Beauty, four blocks give the highest NDCG scores and essentially the same Recall@10 as eight blocks. On centered Toys, eight blocks perform best, with only small @10 gains over six blocks. Uncentered Beauty peaks at six blocks and declines at eight. Six blocks are used as a common configuration in the main experiments, rather than selecting a separate depth from each dataset's test scores. Because total parameters increase with depth, these sweeps measure the combined effect of depth and capacity.

D.4 Composer Architecture and Components

These experiments characterize alternative implementations of the representation side of NRP after the prediction formulation has already been fixed to complete-item scoring. They are architectural studies rather than tests of the decoupling hypothesis itself. All comparisons use fixed four-position SIDs, a four-layer GPT-2 backbone with hidden size 128, four attention heads, and FFN width 512,

Table 17: SID construction and prediction performance. NRP w/o centering denotes the uncentered variant. The best result among the three methods for each SID construction and dataset is highlighted in bold. R and N denote Recall and NDCG, respectively.

		Beauty		Toys	
SID construction	Method	R@10	N@10	R@10	N@10
RQ-KMeans	AR-SID	0.0601	0.0322	0.0550	0.0298
	NRP w/o centering	**0.0944**	**0.0547**	**0.0970**	**0.0577**
	NRP	0.0929	0.0535	0.0959	0.0566
RQ-VAE	AR-SID	0.0494	0.0265	0.0447	0.0249
	NRP w/o centering	**0.0889**	**0.0522**	**0.0881**	**0.0529**
	NRP	0.0870	0.0517	0.0879	0.0528
OPQ	AR-SID	0.0607	0.0325	0.0657	0.0354
	NRP w/o centering	0.0928	0.0528	**0.1055**	0.0608
	NRP	**0.0944**	**0.0551**	0.1049	**0.0622**
Random-Permuted	AR-SID	0.0324	0.0182	0.0147	0.0096
	NRP w/o centering	0.0176	0.0084	0.0072	0.0043
	NRP	**0.0759**	**0.0444**	**0.0699**	**0.0443**
IID-Uniform	AR-SID	0.0339	0.0193	0.0174	0.0116
	NRP w/o centering	0.0310	0.0153	0.0090	0.0042
	NRP	**0.0747**	**0.0444**	**0.0682**	**0.0417**

Table 18: Complete composer-depth sweeps (seed 2020). Parameters are total trainable parameters in millions. All sweeps use validation every epoch.

Depth	Params (M)	R@5	N@5	R@10	N@10
(a) Beauty, centered					
1	1.426	0.0547	0.0388	0.0803	0.0470
2	1.689	0.0597	0.0423	0.0862	0.0508
4	2.216	0.0653	0.0459	0.0947	0.0554
6	2.743	0.0642	0.0442	0.0929	0.0535
8	3.270	0.0639	0.0443	0.0948	0.0542
(b) Toys, centered					
1	1.427	0.0545	0.0386	0.0799	0.0467
2	1.690	0.0616	0.0438	0.0871	0.0520
4	2.217	0.0644	0.0463	0.0937	0.0558
6	2.744	0.0656	0.0468	0.0959	0.0566
8	3.271	0.0666	0.0472	0.0962	0.0567
(c) Beauty, uncentered					
1	1.426	0.0567	0.0408	0.0831	0.0493
2	1.689	0.0609	0.0428	0.0857	0.0508
4	2.216	0.0629	0.0438	0.0905	0.0526
6	2.743	0.0658	0.0455	0.0944	0.0547
8	3.270	0.0650	0.0449	0.0933	0.0540

and a maximum history length of 20. Results use seed 2020 and exact full-catalog ranking with history filtering, without catalog centering.

Additive aggregation and learned composition. To assess whether simple aggregation can replace the learned composer, we compare three additive aggregators with shallow and residual MLP

composition on Beauty and Toys. The additive representations take the form

$$z_i = \alpha \sum_{\ell=1}^{L} E_\ell[c_\ell(i)], \qquad \alpha \in \left\{ \frac{1}{L}, \frac{1}{\sqrt{L}}, 1 \right\}, \tag{34}$$

corresponding to Mean, Scaled Sum, and Sum, respectively, with $L = 4$. These aggregation operations introduce no additional projections, normalization, or trainable parameters, while the code embeddings remain learnable. The shallow MLP ($D = 0$) concatenates the code embeddings, projects them from 512 to 256 dimensions, applies GELU and LayerNorm, and projects the result to 128 dimensions. It contains no residual blocks. All variants retain the same history encoder architecture and full-catalog item prediction objective.

The additive and shallow-MLP variants use learning rate 5×10^{-5}, validation after every epoch, and early-stopping patience ten, with checkpoints selected by validation NDCG@10. None of these runs reaches its maximum epoch budget. The six-block residual composer provides the reference configuration, using its existing uncentered results. All five variants omit output ℓ_2 normalization.

Table 19: Additive aggregation and learned composition without catalog centering. D denotes the number of residual composer blocks. Parameter counts are total model parameters in millions. The $D = 6$ results are reused from the corresponding uncentered reference runs. The best results within each dataset are highlighted in bold. R and N denote Recall and NDCG, respectively.

Composer	Params (M)	R@5	N@5	R@10	N@10
(a) Beauty					
Mean	0.998	0.0278	0.0183	0.0436	0.0233
Scaled Sum	0.998	0.0285	0.0192	0.0452	0.0246
Sum	0.998	0.0324	0.0217	0.0508	0.0276
Shallow MLP ($D = 0$)	1.162	0.0411	0.0273	0.0623	0.0341
Residual MLP ($D = 6$)	2.743	**0.0658**	**0.0455**	**0.0944**	**0.0547**
(b) Toys					
Mean	0.999	0.0212	0.0141	0.0366	0.0191
Scaled Sum	0.999	0.0265	0.0174	0.0436	0.0228
Sum	0.999	0.0278	0.0186	0.0483	0.0251
Shallow MLP ($D = 0$)	1.164	0.0336	0.0225	0.0544	0.0291
Residual MLP ($D = 6$)	2.744	**0.0676**	**0.0482**	**0.0970**	**0.0577**

Table 19 shows that both MLP composers outperform the additive aggregators across all four metrics on both datasets, with the six-block composer performing best. Thus, retaining direct item prediction does not make the choice of composition function inconsequential. Among the additive variants, Sum consistently outperforms Scaled Sum and Mean. Since these variants differ only by constant scaling, their performance differences indicate sensitivity to aggregation scale under the evaluated training configuration, rather than greater representational expressivity. The advantage of $D = 6$ over $D = 0$ supports more expressive composition in this setting, although the comparison also increases parameter count and does not isolate depth from capacity.

Composer width. We further examine width, residual connections, normalization, and depth under a comparable parameter budget on Beauty. These comparisons use validation NDCG@10 for checkpoint selection with early-stopping patience five.

Table 20 varies the width of a two-block residual composer with final hidden LayerNorm and no output ℓ_2 normalization. Increasing width from 128 to 256 improves all four metrics. Further increasing it to 512 improves NDCG but reduces Recall, while more than doubling the total parameter count. Larger width therefore does not uniformly improve ranking.

Residual connections and normalization. Table 21 examines individual components of a two-block composer with width 256. Removing either residual connections or the final hidden LayerNorm reduces both Recall@10 and NDCG@10. Removing output ℓ_2 normalization instead improves both metrics. Within this configuration, the results favor retaining residual connections and final hidden LayerNorm while omitting output ℓ_2 normalization.

Table 20: Composer width on Beauty without catalog centering. All variants use two residual blocks, final hidden LayerNorm, and no output ℓ_2 normalization. Parameter counts are total model parameters in millions.

Width	Params (M)	R@5	N@5	R@10	N@10
128	1.213	0.0566	0.0383	0.0848	0.0474
256	1.689	0.0582	0.0400	0.0859	0.0490
512	3.429	0.0578	0.0419	0.0821	0.0497

Table 21: Composer component analysis on Beauty without catalog centering. The reference uses two residual blocks with width 256, final hidden LayerNorm, and output ℓ_2 normalization. Each variant removes one component from the reference.

Variant	R@10	N@10
Reference	0.0838	0.0461
Without final hidden LayerNorm	0.0816	0.0451
Without output ℓ_2 normalization	0.0859	0.0490
Without residual connections	0.0801	0.0439

Depth under a comparable parameter budget. To examine depth while limiting differences in capacity, Table 22 adjusts composer width to maintain approximately 0.69 million composer parameters, with output ℓ_2 normalization. The two-block model achieves the highest scores, whereas the narrower four-block model performs worse. Increasing depth therefore does not consistently improve ranking under a comparable parameter budget.

Table 22: Composer depth under approximately matched parameter budgets on Beauty. All variants use output ℓ_2 normalization without catalog centering. Parameter counts refer to the composer only.

Blocks	Width	Composer params	R@10	N@10
1	344	696,384	0.0817	0.0453
2	256	691,584	0.0838	0.0461
4	188	690,276	0.0692	0.0368

E Additional Cold-Start Results

Evaluation protocol. This section further examines the generalization benefit of the representation side of NRP. Cold items never appear as positive training targets or in input histories. NRP constructs their representations from fixed content-derived SIDs using shared code embeddings and composition parameters learned from warm-item interactions. We extend the Beauty experiment in Section 6.4 to item holdout ratios of 5% and 20%, alongside the 10% setting reported in the main text. Each holdout setting is evaluated over three training seeds, using a fixed item-holdout split shared across all methods and training seeds. Cold-item interactions are removed from training and input histories, and each input retains at most the 20 most recent warm items. Validation uses warm targets and warm candidates only. Test users are grouped by whether their held-out target is warm or cold, and both groups are evaluated against the full warm-plus-cold catalog. Filtering uses the complete original pre-target history, retaining the target if it also occurs in that history. Recall and NDCG are averaged separately within each test group at cutoffs 10 and 20. In the 10% setting, the catalog contains 1,210 cold and 10,891 warm items, and the test groups contain 2,019 cold-target and 20,344 warm-target users. All methods use the same split within each holdout setting. Item content from the full catalog is available for content-only SID construction, but cold-item interactions provide no training supervision. This is therefore transductive, content-aware item cold start: the cold items and their content are known during SID construction.

Training candidates and catalog centering. For methods using item-level softmax, the training denominator contains only warm items; cold items are excluded from the candidate set used to compute the training loss. NRP computes the catalog-centering mean over the warm catalog only, and uses the centered representations for history encoding and candidate scoring. AR-SID and TIGER are trained with standard next-token prediction using SID sequences of warm training items. Their losses retain token-level normalization over their respective SID-token vocabularies. Consequently, although cold items never appear as positive training targets, their constituent tokens may still participate in token-level competition, and tokens shared with warm items can receive positive supervision through warm-item sequences. Cold-item interactions are excluded from training and input histories, while shared SID parameters are learned from warm-item interactions.

Table 23: Item cold-start recommendation on Beauty at three holdout ratios. Results are averaged over three random seeds. R and N denote Recall and NDCG, respectively. The best result within each holdout ratio and test group is highlighted in bold.

	Warm targets				Cold targets			
Method	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
(a) 5% cold items								
Pure-ID	0.0676	0.0427	0.0897	0.0482	0.0000	0.0000	0.0000	0.0000
AR-SID	0.0623	0.0333	0.0924	0.0409	0.0000	0.0000	0.0000	0.0000
NRP	**0.0905**	**0.0536**	**0.1262**	**0.0625**	**0.0163**	**0.0076**	**0.0388**	**0.0133**
(b) 10% cold items								
Pure-ID	0.0655	0.0411	0.0896	0.0472	0.0000	0.0000	0.0000	0.0000
AR-SID	0.0624	0.0333	0.0938	0.0412	0.0005	0.0002	0.0005	0.0002
NRP	**0.0900**	**0.0517**	**0.1274**	**0.0611**	**0.0154**	**0.0063**	**0.0342**	**0.0110**
(c) 20% cold items								
Pure-ID	0.0694	0.0431	0.0920	0.0488	0.0000	0.0000	0.0000	0.0000
AR-SID	0.0644	0.0348	0.0944	0.0424	0.0009	0.0004	0.0014	0.0005
NRP	**0.0885**	**0.0512**	**0.1242**	**0.0602**	**0.0139**	**0.0058**	**0.0297**	**0.0097**

Effect of the holdout ratio. Table 23 reports the complete ratio sweep for Pure-ID, AR-SID, and centered six-block NRP. NRP has the highest warm- and cold-target scores across all four metrics at each ratio. Pure-ID retrieves no held-out cold targets within the top 20, while AR-SID has zero or very low cold-target recall. NRP attains nonzero cold-target ranking performance while maintaining higher warm-target scores in these comparisons. Absolute cold-target recall remains much lower than warm-target recall, so the results demonstrate a relative benefit of shared SID composition under this protocol.

F Additional Evaluation on a Larger Catalog

We extend our evaluation to Amazon CDs and Vinyl to examine recommendation performance and inference efficiency on a larger item catalog.

Experimental setup. The Amazon Reviews 2014 CDs and Vinyl 5-core dataset contains 75,258 users, 64,443 items, and 1,097,592 interactions. We follow the chronological splitting and evaluation protocol in Appendix B.1, using at most 20 historical items as input. All methods are evaluated on the same test users, targets, and candidate catalog. SASRec and NRP perform exact full-catalog scoring, whereas RPG uses graph-based retrieval for both validation and testing. We report one run per method using seed 2020, with checkpoints selected by validation NDCG@10.

SASRec, RPG, and NRP use backbone depths of 2, 2, and 4, hidden dimensions of 64, 448, and 128, and effective batch sizes of 256, 256, and 64, respectively. Their learning rates are 10^{-3}, 10^{-3}, and 5×10^{-5}. NRP retains its six-block, width-256 composer and full-catalog cross-entropy objective with catalog centering. Its fixed SIDs comprise three RQ-KMeans codes and a uniqueness suffix, with vocabulary sizes $256/256/256/148$. BF16 training and activation recomputation allow

NRP to complete full-catalog training on a single 8-GB GPU. RPG uses 64-position SIDs and graph retrieval with beam width 20, 500 neighbors per item, and five propagation steps. These comparisons use method-specific architectures, training objectives, and supervision positions.

Recommendation performance. Table 24 shows that NRP achieves the highest scores across all four metrics. Its Recall@10 exceeds SASRec and RPG by 89.6% and 115.1%, respectively, under the evaluated configurations. These results extend the empirical evidence for NRP to a catalog of over 64,000 items while retaining exact full-catalog item scoring.

Table 24: Recommendation performance on CDs and Vinyl. RPG uses graph-based retrieval; SASRec and NRP use exact full-catalog scoring. R and N denote Recall and NDCG, respectively. The best results are highlighted in bold.

Method	R@5	N@5	R@10	N@10
SASRec	0.0311	0.0201	0.0498	0.0261
RPG	0.0319	0.0225	0.0439	0.0264
NRP	**0.0643**	**0.0433**	**0.0944**	**0.0529**

Inference efficiency. We benchmark inference on an RTX 5060 Ti 8-GB GPU, using BF16 autocast for all methods. After warmup, single-user latency is measured on 100 test users and batch-64 throughput on 1,024 test users, using the same sampled users across methods. Timing includes batch preparation, host–device transfers, history encoding, retrieval, history filtering, and top-10 selection. NRP's candidate representations and RPG's retrieval graph are precomputed and excluded from online timing. The NRP candidate cache occupies 31.47 MiB.

As shown in Table 25, NRP achieves a median single-user latency of 2.23 ms and a throughput of $5{,}752.1$ users/s. Its measured latency is lower and throughput higher than RPG's graph-based retrieval in this implementation, while SASRec remains faster. These measurements characterize inference with fixed checkpoints and precomputed retrieval structures.

Table 25: Inference performance on CDs and Vinyl. Latency is measured with batch size 1 and throughput with batch size 64. Offline candidate caching and graph construction are excluded from timing.

	Latency (ms)		Throughput
Method	P50	P95	(users/s)
SASRec	1.76	2.58	9,952.7
RPG	10.41	16.70	1,469.4
NRP	2.23	2.65	5,752.1

Training trajectories. Figure 2 shows validation NDCG@10 throughout training. The selected checkpoints occur at epochs 131, 104, and 141 for SASRec, RPG, and NRP, respectively. All three runs terminate after ten epochs without improvement, before reaching their maximum training budgets. The curves illustrate training trajectories and checkpoint selection; epoch counts do not represent equal computation across methods.

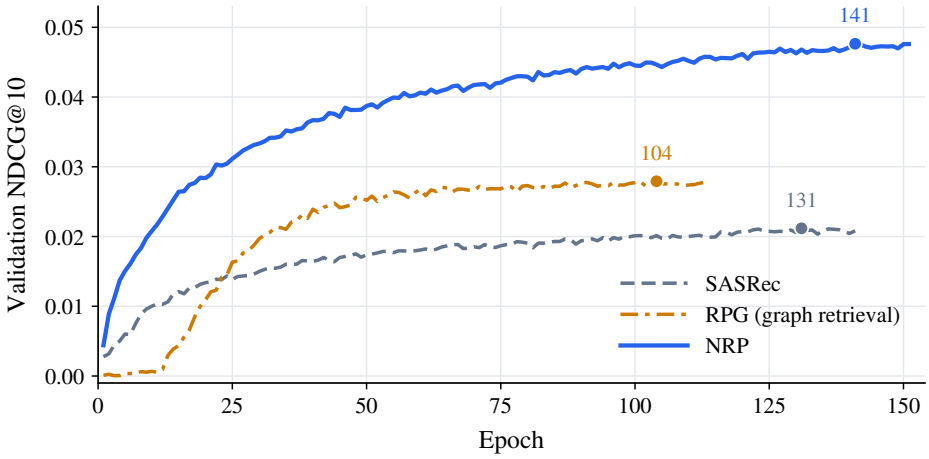


Figure 2: Validation NDCG@10 on CDs and Vinyl. Dots mark the checkpoints selected for testing, with labels indicating their epochs. Curves show the original per-epoch measurements without smoothing.