# WHAT PROPERTIES MAKE A GOOD SID? GWT AS A PRINCIPLED PROXY FOR LEARNABILITY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Generative recommender systems commonly use semantic IDs (SIDs) to represent items as sequences of tokens that are in turn predicted autoregressively. When each item has a unique code, different SID mappings preserve the same item information but decompose item prediction into different sequences of token predictions. We study this output factorization: what properties of a code make its successive predictions easier for a model with restricted output rank? We describe atomic and SID mappings as filtrations and derive spectral approximation bounds for the best predictor in a specified class. These bounds motivate the gain-weighted spectral tail (GWT), a mapping-level proxy for learnability at a given head rank. Controlled synthetic experiments isolate spectral concentration at matched predictive gain and numerical rank, showing its consequences for learning. We then assess whether GWT remains useful after training rank-restricted SASRec on three Amazon datasets. With atomic histories fixed, GWT exhibits strong correlation with test negative log-likelihood (NLL) across seven SID mappings, including RQ-VAE and RQ-Kmeans. We also compare GWT with common SID diagnostics, including codebook utilization and the per-level token count Gini coefficient. These initial results support GWT as a principled proxy metric for SID learnability.

## 1 INTRODUCTION

Generative recommendation treats recommendation as a sequence-generation problem: given a user's past interactions, a model generates the identifier of the next item (Rajput et al., 2023). With one output symbol per item, the vocabulary grows with the catalog and can become very large, making prediction over the full vocabulary prohibitively expensive. Semantic IDs (SIDs) map each item to a short sequence of tokens drawn from smaller vocabularies. These identifiers are typically constructed from item content or interaction data, allowing the model to select an item through successive token predictions.

Different SIDs can preserve exactly the same item identities when the mappings are injective yet lead to different learning outcomes. Our central question is therefore: *What properties make a good SID?* Here, *good* means easier to learn for a specified prediction task, model class, and amount of training data. We seek to understand properties that help compare candidate mappings, rather than a new algorithm for constructing them.

We separate three questions about how the choice of mapping affects learning:

1. **Output factorization.** Each token prediction chooses among groups of items compatible with the preceding tokens. With the model's input fixed, how difficult are these successive choices to predict?
2. **Signal density.** How many training examples support each token prediction, and which examples update the same parameters?
3. **History encoding.** When past items are also represented by SIDs, how difficult is it for the model to extract the information needed for its next prediction?

This paper focuses on *output factorization*. Signal density and history encoding explain why output structure alone cannot determine trained performance and we have supporting experiments in



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.

the appendices. The present contribution concerns the conditional output laws induced by a SID mapping.

To study output factorization, we describe a mapping by the groups of items sharing each code prefix. Each additional token splits a group into smaller groups, until the complete code identifies one item. These nested partitions form an *item filtration*. This description lets us examine both the predictive information assigned to each step and the model capacity needed to represent its probabilities. Theorem 2 concerns the best predictor in a specified class. Our empirical question is whether its suggested diagnostic remains useful for comparing mappings after training.

Our four contributions are:

- **Mappings as filtrations.** We give a common mathematical description of atomic and SID outputs. A predictive-gain decomposition shows how mappings redistribute the same total predictive information across their token predictions.
- **Spectral bounds and a learnability diagnostic.** Theorem 2 bounds the best attainable excess item log-loss under explicit probability and projection-realizability conditions. It motivates GWT as a mapping-level learnability diagnostic at a specified head rank.
- **A controlled generator and synthetic experiments.** Our generator varies spectral concentration while matching predictive gain and numerical rank. We use it to study the effect of spectral shape and compare mappings on the same prediction task.
- **Real-data evidence.** We assess whether GWT orders mappings usefully for rank-restricted SASRec (Kang & McAuley, 2018). The Amazon comparison fixes atomic histories and one reference law, varies SID targets, and evaluates complete-item test NLL within each head rank.

## 2 Related Work

**Learned hierarchical predictors and label trees.** Hierarchical softmax predicts a categorical outcome through choices along a tree (Morin & Bengio, 2005). Tree construction has been studied for predictive quality as well as computational efficiency. For example, Jernite et al. (2017) jointly learn input representations and multiway label trees, encouraging balanced, separable partitions. Our objective is different: we compare the conditional laws induced by different trees for the same item-prediction task and ask what makes some trees better than others.

**Softmax rank restrictions and spectral approximation.** Restricting a linear-softmax head's rank can prevent it from representing the true conditional probabilities (Yang et al., 2018). Ganea et al. (2019) give a singular-value lower bound on log-probability matrix approximation and characterize minimum cross-entropy with fixed output embeddings. Godey et al. (2024) connect approximation of an ideal language-model head to a loss gap and evaluate heads of restricted rank. Our addition is to apply spectral approximation to the *conditional child laws induced by each mapping*. The work here yields a diagnostic for comparing item factorizations, extending the flat-output question studied by these works.

**SID construction and generative recommendation.** Differentiable search indices and TIGER retrieve documents or recommend items by generating identifiers (Tay et al., 2022; Rajput et al., 2023). LETTER combines semantic, collaborative, and code-diversity objectives (Wang et al., 2024); DIGER allows recommendation gradients to update SID assignments (Fu et al., 2026). GRID compares tokenization, architecture, and training choices in a common evaluation framework (Ju et al., 2025). These works show how important well-structured identifiers are for the downstream recommendation performance. We investigate properties that could explain why one SID mapping is easier to learn than another.

**Tokenization.** Tokenization can change which source dependencies a restricted predictor can express (Rajaraman et al., 2024); frequency-based criteria have also been proposed and challenged (Zouhar et al., 2023; Cognetta et al., 2024). We study fixed injective SID mappings and their conditional output spectra.

# 3 FILTRATIONS AND OUTPUT LEARNABILITY

A SID replaces one item prediction with a sequence of token predictions. We want to know which mappings make those predictions easier for a model with limited output capacity. Two quantities will matter: how much a prediction can improve by using context, and how much of its changing probability distribution a rank-restricted head can represent. We define these quantities before relating them to item-prediction error.

## 3.1 PREDICTION TASK AND INJECTIVE CODES

Let $H = (Y_1, \dots, Y_T)$ be an observed item history and $Y$ the next item, with finite catalog $\mathcal{V} = \mathrm{supp}(Y)$. Let $C$ describe the generating context that determines the next-item probabilities. We assume it is predictively sufficient:

$$\mathbb{P}(Y \in \cdot \mid C, H) = \mathbb{P}(Y \in \cdot \mid C) \qquad \text{almost surely.} \tag{1}$$

Thus, once $C$ is known, history provides no further information about the next item. The context itself may be hidden: this assumption does not say that a model observing $H$ can recover $C$.

For level $i$, let $\mathcal{A}_i$ be a finite token alphabet. An $m$-level item code is an injective map

$$\phi : \mathcal{V} \longrightarrow \mathcal{A}_1 \times \cdots \times \mathcal{A}_m, \qquad \phi(y) = (a_1(y), \dots, a_m(y)). \tag{2}$$

Each symbol $a_i(y) \in \mathcal{A}_i$ is called a token, and $i$ denotes its position, or level, in the item's code. Given the preceding tokens, predicting the next token selects a subgroup of the items compatible with that prefix. Let $A_i = a_i(Y)$ and $P_i = (A_1, \dots, A_i)$, with $P_0$ the empty prefix. For a fixed item $y$, write $p_i(y) = (a_1(y), \dots, a_i(y))$ and $p_0(y) = P_0$. At each level, items sharing a prefix form a group. The groups are successively refined, giving the filtration

$$\sigma(P_0) \subseteq \sigma(P_1) \subseteq \cdots \subseteq \sigma(P_m) = \sigma(Y). \tag{3}$$

The final equality follows from injectivity. An atomic ID predicts the item in one step: $m = 1$ and $P_1 = Y$. An injective code preserves the optimal item log-loss, since $\mathrm{H}(\phi(Y) \mid H) = \mathrm{H}(Y \mid H)$. Differences in learned performance therefore depend on the model, data, training procedure, or decoding rule.

## 3.2 CONDITIONAL PREDICTION LAWS AND PREDICTIVE GAINS

A parent at level $i$ is a prefix $p$ in the support of $P_{i-1}$. The tokens that can follow it form a subset of the level's alphabet:

$$\mathcal{A}_{i,p} = \{a_i(y) : y \in \mathcal{V},\ p_{i-1}(y) = p\} \subseteq \mathcal{A}_i, \qquad \rho_i(p) = \mathbb{P}(P_{i-1} = p). \tag{4}$$

Write $B_{i,p} = |\mathcal{A}_{i,p}|$ and $B_i = |\mathcal{A}_i|$. Tokens outside $\mathcal{A}_{i,p}$ have probability zero at this parent.

For $a \in \mathcal{A}_{i,p}$, compare the probability of the next token when both context and prefix are known with its probability given the prefix alone:

$$q_{i,p}(a \mid c) = \mathbb{P}(A_i = a \mid C = c, P_{i-1} = p), \qquad \bar{q}_{i,p}(a) = \mathbb{P}(A_i = a \mid P_{i-1} = p). \tag{5}$$

The *predictive gain* at level $i$ is the improvement in optimal next-token log-loss from using that additional context:

$$G_i = I(A_i; C \mid P_{i-1}) = \mathbb{E}_{P_{i-1},C}\left[\mathrm{KL}\big(q_{i,P_{i-1}}(\cdot \mid C) \,\|\, \bar{q}_{i,P_{i-1}}\big)\right]. \tag{6}$$

Gain measures the benefit available from context and only depends on the mapping and the joint distribution $(Y, C)$.

**Proposition 1 (conservation of predictive gain).** Every injective item filtration allocates the same total predictive information across its refinements:

$$\sum_{i=1}^{m} G_i = I(Y; C). \tag{7}$$

This follows from the chain rule for mutual information (Appendix C.2). Changing the mapping can move gain between levels, but cannot change its total.

### 3.3 PARENTWISE LOGIT SPECTRA

For a fixed distribution and SID mapping, the conditional token probabilities define the *true logits* at each parent. We measure the spectrum of how much these logits vary with context.

Fix the mapping and the law of $(C, Y)$. Assume $q_{i,p}(a \mid c) \geq \epsilon > 0$ for every possible child on the support of $(C, P_{i-1})$. This keeps the log probabilities bounded. Because softmax is unchanged by adding a constant to every score, we represent the true probabilities by logits centered across children:

$$\ell_{i,p,a}(c) = \log q_{i,p}(a \mid c) - \frac{1}{B_{i,p}} \sum_{a' \in \mathcal{A}_{i,p}} \log q_{i,p}(a' \mid c). \tag{8}$$

The vector $\boldsymbol{\ell}_{i,p}(c) = (\ell_{i,p,a}(c))_{a \in \mathcal{A}_{i,p}}$ has $B_{i,p}$ coordinates and its softmax is $q_{i,p}(\cdot \mid c)$. To study its variation with context, let

$$\boldsymbol{\mu}_{i,p} = \mathbb{E}[\boldsymbol{\ell}_{i,p}(C) \mid P_{i-1} = p], \qquad \boldsymbol{u}_{i,p}(c) = \boldsymbol{\ell}_{i,p}(c) - \boldsymbol{\mu}_{i,p}. \tag{9}$$

The covariance

$$K_{i,p} = \mathbb{E}[\boldsymbol{u}_{i,p}(C)\boldsymbol{u}_{i,p}(C)^\top \mid P_{i-1} = p] \tag{10}$$

describes the directions of this variation. We order its eigenvalues as $\lambda_{i,p,1} \geq \cdots \geq \lambda_{i,p,B_{i,p}-1} \geq 0$, with orthonormal eigenvectors $v_{i,p,j}$. Let $E_{i,p,r}$ be the span of the first $\min\{r, B_{i,p}-1\}$ eigenvectors. Projecting onto this space retains the largest possible logit variation in $r$ directions.

Zero-pad each eigenvalue list to length $B_i - 1$ and define

$$\lambda_{i,j} = \sum_p \rho_i(p)\lambda_{i,p,j}, \qquad T_i(r) = \sum_{j>r} \lambda_{i,j} = \sum_p \rho_i(p) \sum_{j>r} \lambda_{i,p,j}. \tag{11}$$

Thus $T_i(r)$ is the smallest mean squared-logit error when each parent has an affine output subspace of dimension at most $r$. It is attained by the mean plus projection onto $E_{i,p,r}$. We average the ordered eigenvalues of the parent covariances, not the eigenvalues of an averaged covariance.

The corresponding *tail fraction* is

$$\tau_i(r) = \begin{cases} T_i(r)/T_i(0), & T_i(0) > 0, \\ 0, & T_i(0) = 0. \end{cases} \tag{12}$$

It measures the fraction of context-dependent logit variation missed at rank $r$.

### 3.4 RANK-RESTRICTED PREDICTORS AND APPROXIMATION ERROR

Let $X$ be the model's input: either the supplied context $C$ or a fixed function of the observed history $H$. We keep this input unchanged when comparing output mappings.

A *rank-$r$ categorical predictor* assigns probabilities by applying softmax to centered logits of the form

$$\widehat{\boldsymbol{\ell}}(x) = \boldsymbol{\beta} + Vh(x), \qquad h(x) \in \mathbb{R}^r, \tag{13}$$

where $\boldsymbol{\beta}$ and $V$ are fixed across inputs. The feature function $h$ may be nonlinear. Rank $r$ means *at most* $r$ throughout.

We apply this restriction to the output factorization induced by $\phi$. At each level $i$ and parent prefix $p$, the model predicts the next token among the valid children $\mathcal{A}_{i,p}$ using

$$\widehat{\boldsymbol{\ell}}_{i,p}(x) = \boldsymbol{\beta}_{i,p} + V_{i,p}h_{i,p}(x), \qquad h_{i,p}(x) \in \mathbb{R}^r, \tag{14}$$

Here $V_{i,p} \in \mathbb{R}^{B_{i,p} \times r}$. Write $\widehat{q}_{i,p}(\cdot \mid x)$ for the resulting softmax distribution. These conditional distributions together define an item predictor $f$ through

$$\widehat{q}_f(y \mid x) = \prod_{i=1}^{m} \widehat{q}_{i,p_{i-1}(y)}(a_i(y) \mid x). \tag{15}$$

For a model family $\mathcal{M}$, let $\mathcal{M}_r$ be the class of predictors with this factorization and rank restriction, retaining the family's feature and parameter-sharing constraints.

Its *excess log-loss* is

$$\mathcal{R}_\phi(f) = \mathbb{E}_{X,Y}[-\log \widehat{q}_f(Y \mid X)] - \mathrm{H}(Y \mid C). \tag{16}$$

Here $\mathrm{H}(Y \mid C)$ is the smallest expected log-loss achievable if the generating context is known. This comparison counts both restrictions of the model and any predictive information in $C$ unavailable from $X$.

The class's *approximation error* is its best excess log-loss:

$$\mathrm{App}_r(\mathcal{M}) = \inf_{f \in \mathcal{M}_r} \mathcal{R}_\phi(f). \tag{17}$$

This is a property of the predictor class, not the output of training. Let $\mathcal{R}_i(f)$ denote the expected conditional KL between the true and predicted token distributions at level $i$, averaged over context, model input, and parent prefix. The item factorization and context sufficiency give $\mathcal{R}_\phi(f) = \sum_i \mathcal{R}_i(f)$. The approximation error therefore minimizes this sum over $f \in \mathcal{M}_r$, subject to the model family's shared constraints. Appendix C.3 gives the decomposition explicitly and shows where sufficiency is used.

### 3.5 Spectral bounds and the gain-weighted tail

We can now connect the mapping to prediction error. A large tail fraction means that a rank-$r$ head misses much of a level's logit variation, but the importance of that level also depends on its predictive gain. Theorem 2 makes this connection precise: under the conditions below, the best item-prediction error is bounded by constant multiples of $\sum_i G_i \tau_i(r)$. The lower bound describes the error forced by limited output rank. The upper bound describes what is achievable when the model can produce the optimal spectral projections.

**Theorem 2(a) (informal: unavoidable output-rank cost).** Suppose the true child probabilities are at least $\epsilon > 0$ and the class optimum is attained by a predictor whose child probabilities are also at least $\epsilon$, over each parent's possible children. Then

$$\mathrm{App}_r(\mathcal{M}) \geq 2\epsilon \sum_{i=1}^{m} G_i \tau_i(r). \tag{18}$$

This bound holds regardless of how the model computes its features.

**Theorem 2(b) (informal: attainable spectral approximation).** Suppose the true child probabilities are at least $\epsilon > 0$. If one predictor in $\mathcal{M}_r$ produces the optimal rank-$r$ logit projections at every parent and level from its available input $X$, then

$$\mathrm{App}_r(\mathcal{M}) \leq \frac{1}{2\epsilon} \sum_{i=1}^{m} G_i \tau_i(r). \tag{19}$$

This ability to produce the projections is the *realizability* condition. The result asserts that a suitable predictor exists, not that training will find it. Formal statements and proofs are in Appendix C.3.

The bounds suggest favouring mappings that place little predictive gain at levels with large rank-$r$ tails.

**The normalized diagnostic.** For positive total gain, define the *gain-weighted spectral tail*

$$\mathrm{GWT}_\phi(r) = \frac{\sum_{i=1}^{m} G_i \tau_i(r)}{\sum_{i=1}^{m} G_i}. \tag{20}$$

For a fixed task, Proposition 1 makes the denominator the same for every mapping and thus normalization preserves mapping orderings.

**Using observed histories.** For models that observe history rather than the generating context, we can instead analyze $\mathbb{P}(Y \mid H)$. Both gains and spectra must then be computed using $H$ in place of $C$, and excess log-loss is measured relative to $\mathrm{H}(Y \mid H)$. Appendix D develops this version and its application to SASRec. It studies output approximation given history, not the difficulty of computing features from that history.

# 4 CONTROLLED SYNTHETIC EXPERIMENTS

We use two controlled experiments to study SID prediction difficulty. First, we vary spectral concentration while matching predictive gain and numerical rank. Second, we change the mapping while holding the item distribution and code-tree structure fixed. Both experiments use known conditional item probabilities and predictors that receive the generating context directly.

## 4.1 CONTROLLED SETTING AND EVALUATION

A complete $B$-ary tree of depth $m$ assigns one item to each leaf; its path is the item's *oracle* code. At parent $p$ on level $i$, context $c$ determines the child logits

$$\boldsymbol{z}_{i,p}(c) = \boldsymbol{b}_{i,p} + \alpha_i \boldsymbol{P}_{i,p} c, \qquad \boldsymbol{P}_{i,p} \in \mathbb{R}^{B \times d}. \tag{21}$$

Softmax gives the child probabilities. Each $\boldsymbol{P}_{i,p}$ has rank $R$, columns summing to zero, and prescribed singular values. The biases control average child frequencies; the scales control predictive gain. Appendix A gives construction and calibration details.

Each predictor, or *student*, receives $X = C \in \mathbb{R}^d$ and uses

$$\widehat{\boldsymbol{\ell}}_{i,p}(c) = \boldsymbol{\beta}_{i,p} + V_{i,p} U_{i,p}^{\top} c, \qquad U_{i,p} \in \mathbb{R}^{d \times r}, \quad V_{i,p} \in \mathbb{R}^{B_{i,p} \times r}. \tag{22}$$

Every parent has its own matrices and bias, with no parameter sharing across parents or levels. Training, validation, and test contexts are disjoint and separate from calibration contexts. Training uses sampled labels, with 1,280 labels per parent and three student initializations. Evaluation uses the exact KL from the generator's item distribution to the student's, averaged over test contexts. Appendix B.1 gives the full protocol.

## 4.2 SPECTRAL CONCENTRATION AND PREDICTION ERROR

Under the oracle mapping, we vary the eigenvalues of the generator matrices while keeping the same directions and full numerical rank. Five profiles range from concentrated to flat spectra, with concentration constant or varying across levels (Table 10). Each level is calibrated to a predictive-gain target of 0.30 nats and uniform child marginals. We use three generator seeds, sharing the context sets across spectral profiles within each seed.

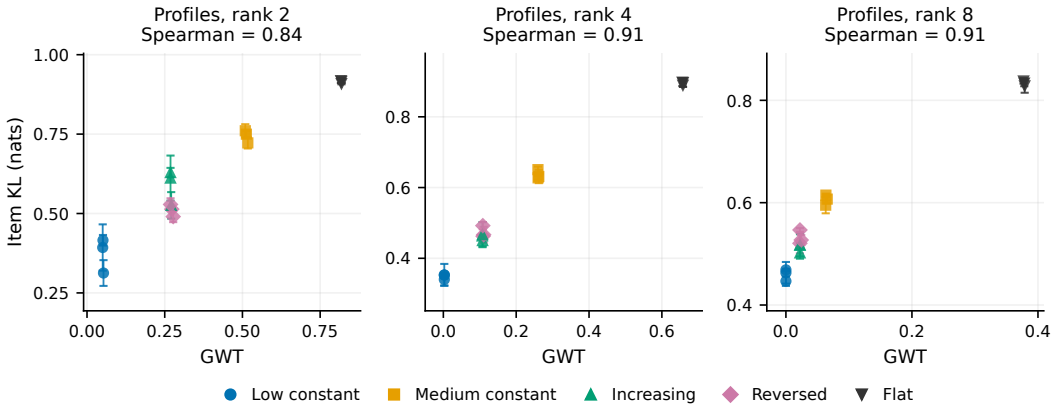


Figure 1: GWT and trained item KL across spectral profiles. Each point represents one profile and generator seed; error bars show sample standard deviations over three student initializations. Gains and numerical rank are matched.

With predictive gain and numerical rank held fixed, students generally achieve lower KL on more concentrated profiles (Figure 1). At head rank 2, mean KL is 0.374 for the low-constant profile against 0.915 for the flat profile. GWT tracks these performance differences across all profile–generator combinations, with Spearman correlations of 0.84/0.907/0.91 at ranks 2/4/8. Appendix B.4 gives the full results and paired comparisons.

### 4.3 Mapping performance at fixed code structure

We now fix the generating distribution and change only the item-to-code assignment. All mappings are bijections onto the same complete code tree, so their counting statistics are identical (Table 1).

Table 1: Counting-based statistics shared by all control mappings on a complete $B$-ary tree of depth $m$.

| Statistic at level $i$ | All mappings |
|---|---|
| Token utilization | 1 |
| Token-count Gini | 0 |
| Normalized token-count entropy | 1 |
| Effective vocabulary size | $B$ |
| Decoding-state count | $B^{i-1}$ |
| Valid-next-token fraction | 1 |
| Distinct-prefix count / mean prefix-group size | $B^i$ / $B^{m-i}$ |
| Mean valid-next-token count | $B$ |

We use one generator realization and four mapping families for comparison.

The *oracle* mapping retains the generator's codes. The *predictive* mapping recursively partitions items by their mean training contexts, using balanced splits along the first principal component. The *popularity* mapping assigns lexicographic codes in item-frequency order; *random* mappings permute the assignments. Predictive and popularity mappings use training observations only (Appendix A.4).

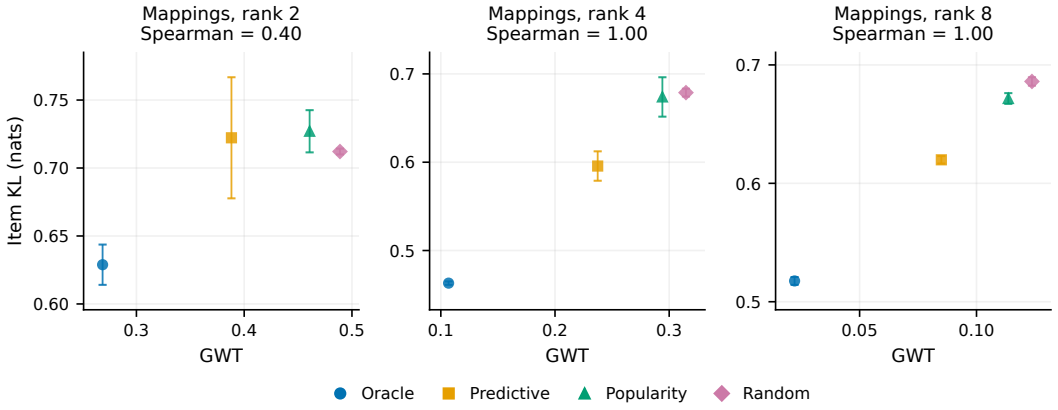


Figure 2: GWT and trained item KL for mappings with identical code structure. Points and error bars are means and sample standard deviations over three student initializations, after averaging random constructions within each initialization.

Across the four mapping families, Spearman correlation is 0.4 at head ranks 1 and 2 and 1.0 at ranks 4 and 8 (Table 6 in Appendix B.2). Counting statistics cannot distinguish these mappings, whereas GWT captures differences associated with trained error. Its ordering is imperfect at low ranks. Further data-budget and history-encoding experiments appear in Appendices B.6 and D.5.

## 5 Amazon: Comparing SID Diagnostics

We assess GWT on Amazon Beauty, Sports, and Toys, where the conditional item distribution is unknown and mappings differ in code structure. We ask how well these diagnostics predict differences in test NLL across mappings.

### 5.1 Evaluation setting

We compare seven supplied mappings: RecDMON (Medvedev et al., 2026), RQKMEANS (Ju et al., 2025), RQ-GAE (Medvedev et al., 2026), RQVAE (Rajput et al., 2023), LETTER (Wang

et al., 2024), S2GR (Guo et al., 2026), and MMGRec (Liu et al., 2026). These labels identify the supplied constructions; Appendix E.1 documents their provenance and catalog normalization. For each Amazon category (McAuley et al., 2015), SASRec predicts SID targets from atomic item histories. Data splits, architecture, and training settings are fixed across mappings; validation negative log-likelihood (NLL) selects each checkpoint. We evaluate head ranks $r \in \{2, 4, 8\}$ with three training initializations. All mappings assign unique codes to the same retained items, but their code lengths and vocabulary sizes differ, so matching rank does not match parameter count. Appendix E describes the mappings and protocol.

To estimate GWT, one reference SASRec per dataset predicts atomic items from atomic histories. We aggregate its item probabilities along each code tree to obtain the conditional token laws, gains, and spectra. All mappings use the same reference checkpoint and 8,192 validation histories; diagnostics remain fixed across comparison training runs. The diagnostic rank $k$ determines how many spectral directions the tail excludes; the trained head rank $r$ limits the predictor's output capacity.

### 5.2 Diagnostics and trained performance

Table 2 reports signed Spearman correlations with mean test NLL. For each dataset and head rank, we correlate scores across seven mappings after averaging NLL over three initializations, then average the three dataset correlations equally. Catalogue statistics are first averaged across all applicable levels within each mapping. GWT and the unweighted mean tail use the reference's conditional probabilities. (Appendix E.3).

Table 2: Mean within-dataset Spearman correlation with test NLL. The final column averages the signed correlations equally across ranks. Count metrics use the mean across levels.

| Diagnostic / head rank $r$ | 2 | 4 | 8 | Mean |
|---|---|---|---|---|
| GWT ($k = r$) | 0.440 | 0.488 | 0.500 | 0.476 |
| GWT ($k = 16$) | 0.762 | 0.655 | 0.702 | 0.706 |
| Mean tail ($k = 16$) | 0.702 | 0.595 | 0.560 | 0.619 |
| Token utilization | −0.394 | −0.445 | −0.407 | −0.415 |
| Token-count Gini | 0.238 | 0.369 | 0.464 | 0.357 |
| Normalized token-count entropy | −0.429 | −0.476 | −0.452 | −0.452 |
| Effective vocabulary size | −0.333 | −0.440 | −0.512 | −0.429 |
| Decoding-state count | 0.476 | 0.536 | 0.548 | 0.520 |
| Valid-next-token fraction | −0.619 | −0.512 | −0.464 | −0.532 |
| Semantic delta cosine ($L = 3$) | 0.424 | 0.305 | 0.367 | 0.365 |
| Collaborative delta cosine ($L = 2$) | −0.100 | 0.119 | 0.267 | 0.095 |

At $k = 16$, GWT has the largest magnitude of dataset-averaged correlation among the recomputed diagnostics at each head rank, and selects RecDMON in all three datasets. RecDMON has the lowest mean NLL in eight of nine dataset–head-rank settings.Figure 3 shows the comparison at head rank $r = 8$ and Appendix E.4 reports individual scores and NLLs.

## 6 Scope and Limitations of GWT

GWT isolates one aspect of SID learnability: whether predictive variation at each refinement can be represented by a head of specified rank. Its interpretation depends on both the reference law used to estimate gains and spectra and the output factorization induced by the code tree.

On real data, gains and spectra must be estimated from a reference law. Using a common reference makes mappings comparable under that law, but reference error can alter their scores and orderings. GWT is therefore most informative as a diagnostic relative to the chosen reference, rather than as a reference-independent ranking of mappings.

Tree structure also matters. A parent with $B_{i,p}$ children has at most $B_{i,p}-1$ nonzero logit-covariance eigenvalues, so its spectral tail vanishes when $k \geq B_{i,p}-1$. In particular, binary trees have zero tails at diagnostic rank one even when their conditional probabilities are difficult to learn. Long, narrow

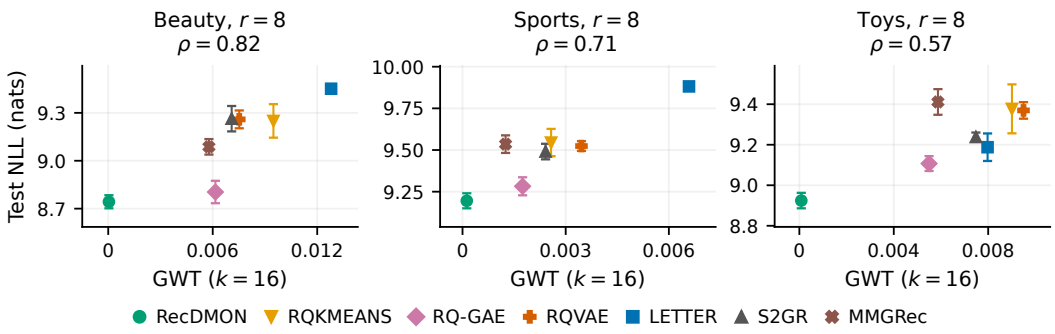


Figure 3: GWT ($k = 16$) and test NLL at head rank $r = 8$. Points and error bars show means and sample standard deviations over three initializations; $\rho$ is Spearman correlation.

trees can therefore have small GWT while requiring many successive predictions. Matching head rank across different structures also leaves parameter counts and computation unmatched.

Finally, spectral approximation is not the only source of prediction error. In the controlled synthetic setting, even the oracle student retains positive trained KL when its spectral tail is zero (Appendix B.2). The probability assumptions determine how tightly the theoretical bounds apply, and an ordering in log-loss does not by itself establish an ordering in retrieval quality. These distinctions define what GWT diagnoses and what must be assessed separately.

## 7 Conclusions and Future Work

An injective SID preserves item identity and total predictive information while changing the conditional prediction problems a model must solve. The filtration view describes how a mapping distributes predictive gain across refinements and how their logit spectra constrain output approximation. Under the assumptions of Theorem 2, gain-weighted spectral tails bound the best attainable excess log-loss at a specified head rank. This motivates GWT as a diagnostic of output factorization. Its levelwise contributions identify where substantial predictive gain coincides with variation that the head cannot represent, giving a concrete criterion for comparing mappings with limited output capacity.

The controlled synthetic experiments show that spectral concentration affects trained error even when predictive gain and numerical rank are matched. They also show differences between mappings with identical code-tree counting statistics. On Amazon, GWT selects the lowest mean-NLL mapping in eight of nine dataset–head-rank settings among four candidates. Together, these results support using GWT to compare the output-factorization demands of SID mappings, while the scope discussed above prevents treating it as a complete measure of learnability.

As discussed in the introduction, a theory of SID learnability should also account for *signal density* and *history representation*. For signal density, the question is when shared tokens let observations of one item improve predictions for others, and when individual refinements receive too little training evidence. For history representation, the question is how the choice of code affects the model's ability to extract predictive information from past interactions. Our supporting experiments show that the preferred head rank changes with the data budget, and that history encodings affect performance despite preserving item identity and matching parameter counts (Appendices B.6 and D.5). In future work, we aim to extend the filtration framework to these questions and connect SID structure to the data and model capacity required for accurate prediction.

### AI use statement

We used OpenAI Codex to assist with implementing and refining experimental code, analyzing results, and drafting and revising the manuscript. The authors take responsibility for the final content, including all mathematical claims, experimental results, and software.

## REFERENCES


Marco Cognetta, Vilém Zouhar, Sangwhan Moon, and Naoaki Okazaki. Two counterexamples to tokenization and the noiseless channel. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 16897–16906, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1469/.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '26, pp. 369–379, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725999. doi: 10.1145/3805712.3809641. URL https://doi.org/10.1145/3805712.3809641.

Octavian Ganea, Sylvain Gelly, Gary Becigneul, and Aliaksei Severyn. Breaking the softmax bottleneck via learnable monotonic pointwise non-linearities. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2073–2082. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ganea19a.html.

Nathan Godey, Éric Villemonte de la Clergerie, and Benoît Sagot. Why do small language models underperform? studying language model saturation via the softmax bottleneck. 2024. URL https://openreview.net/forum?id=MoitXWlXcS.

Zihao Guo, Jian Wang, Ruxin Zhou, Youhua Liu, Jiawei Guo, Jun Zhao, Xiaoxiao Xu, Yongqi Liu, and Kaiqiao Zhan. S2gr: Stepwise semantic-guided reasoning in latent space for generative recommendation, 2026. URL https://doi.org/10.1145/3770855.3817720.

Yacine Jernite, Anna Choromanska, and David Sontag. Simultaneous learning of trees and representations for extreme classification and density estimation. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1665–1674. JMLR.org, 2017. URL https://proceedings.mlr.press/v70/jernite17a.html.

Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. Generative recommendation with semantic ids: A practitioner's handbook. pp. 6420–6425, 2025. doi: 10.1145/3746252.3761612. URL https://doi.org/10.1145/3746252.3761612.

Wang-Cheng Kang and Julian McAuley. Self-Attentive Sequential Recommendation . In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, Los Alamitos, CA, USA, 2018. IEEE Computer Society. doi: 10.1109/ICDM.2018.00035. URL https://doi.ieeecomputersociety.org/10.1109/ICDM.2018.00035.

Han Liu, Yinwei Wei, Xuemeng Song, Weili Guan, Yuan-Fang Li, and Liqiang Nie. Mmgrec: Multimodal generative recommendation with transformer model. *ACM Trans. Multimedia Comput. Commun. Appl.*, August 2026. ISSN 1551-6857. doi: 10.1145/3838594. URL https://doi.org/10.1145/3838594. Just Accepted.

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*, SIGIR '15, pp. 43–52, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336215. doi: 10.1145/2766462.2767755. URL https://doi.org/10.1145/2766462.2767755.

Aleksei Medvedev, Alejandro Ariza-Casabona, Steven Derby, Gonzalo Fiz Pontiveros, Xinyang Shao, and Florian Spiess. Neither black nor white: Balancing semantic and collaborative signals with graph-informed semantic IDs (GrIS). In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management*, 2026. To appear.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In Robert G. Cowell and Zoubin Ghahramani (eds.), *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pp. 246–252. PMLR, 06–08 Jan 2005. URL https://proceedings.mlr.press/r5/morin05a.html.

Nived Rajaraman, Jiantao Jiao, and Kannan Ramchandran. An analysis of tokenization: transformers under markov data. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/724afcaae4ae92a9220a077ffe80088d-Abstract-Conference.html.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Keshavan, Trung Vu, Lukasz Heidt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, 2007.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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*, CIKM '24, pp. 2400–2409, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704369. doi: 10.1145/3627673.3679569. URL https://doi.org/10.1145/3627673.3679569.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkwZSG-CZ.

Vilém Zouhar, Clara Meister, Juan Gastaldi, Li Du, Mrinmaya Sachan, and Ryan Cotterell. Tokenization and the noiseless channel. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5184–5207, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.284. URL https://aclanthology.org/2023.acl-long.284/.

# A Generator Details and Profile Checks

The generator combines a conditional item distribution with a rule for updating the context after each sampled item. We first describe the conditional distribution and its calibration, then the state update and the context samples used in the experiments.

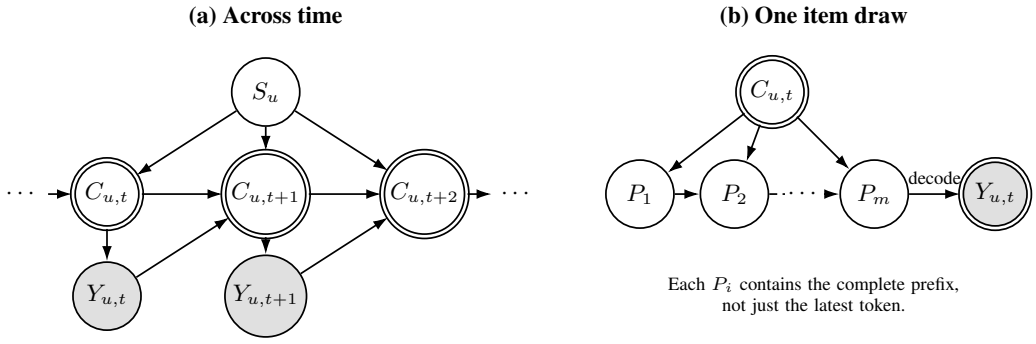


Figure 4: Graphical model of the synthetic generator. (a) The persistent state $S_u$, current context $C_{u,t}$, and sampled item $Y_{u,t}$ determine $C_{u,t+1}$; the user index is fixed. (b) Expansion of one item draw, conditional on the context from (a). Each prefix extension samples a child using the same context and the entire previous prefix; the complete prefix $P_m$ identifies the item. Double borders indicate deterministic nodes given their parents; gray nodes denote items. Dots omit intermediate states or prefixes. Generator parameters are fixed and omitted.

## A.1 Conditional item law and spectral profiles

The generating tree has branching $B$, depth $m$, and $B^m$ leaf items; the context dimension is $d$. For context $c$ and parent prefix $p$ at level $i$, the vector of child logits is

$$\boldsymbol{z}_{i,p}(c) = \boldsymbol{b}_{i,p} + \alpha_i \boldsymbol{P}_{i,p} c, \qquad \boldsymbol{P}_{i,p} \in \mathbb{R}^{B\times d}. \tag{23}$$

The bias vector $\boldsymbol{b}_{i,p}$ is specific to the parent, and $\alpha_i > 0$ is shared across parents at the same level. Each matrix $\boldsymbol{P}_{i,p}$ has rank $R$, where $R \le \min\{d, B-1\}$ is the generator rank and $r$ denotes the student rank. Its columns sum to zero. Softmax gives $q_{i,p}(\cdot \mid c)$. Sampling these conditional laws from root to leaf produces an item and its oracle code, with the same context used at every level (Algorithm 1).

**Constructing the profile matrices.** For each parent independently, draw a Gaussian $d \times R$ matrix and use QR decomposition to obtain $\boldsymbol{H}_{i,p}$ with orthonormal columns. Draw a second Gaussian $B \times R$ matrix, subtract each column's mean, and apply QR decomposition. Scale its orthonormal columns by $\sqrt{B/R}$ to obtain $\boldsymbol{G}_{i,p}$. For positive mode weights $\eta_{i,1}, \dots, \eta_{i,R}$, set

$$\boldsymbol{P}_{i,p} = \boldsymbol{G}_{i,p} \boldsymbol{D}_i \boldsymbol{H}_{i,p}^\top, \qquad \boldsymbol{D}_i = \mathrm{diag}(\eta_{i,1}, \dots, \eta_{i,R}). \tag{24}$$

Its nonzero singular values are $\sqrt{B/R}\,\eta_{i,j}$. The columns of $\boldsymbol{G}_{i,p}$, and hence of $\boldsymbol{P}_{i,p}$, sum to zero.

We use $R = B - 1$ nonzero modes in every profile. Positive exponential mode weights $\eta_{i,j}$ are rescaled so that $\sum_j \eta_{i,j}^2 = R$. Their decay rates are chosen to give the target effective ranks in Table 4, computed from the normalized squared mode weights. The flat profile has $\eta_{i,j} = 1$. This fixes the total squared singular value before calibration of $\alpha_i$. The resulting logit covariance also depends on that scale and on the context distribution at each parent.

Across profiles at a fixed random seed, we reuse the tree, the left and right singular vectors of each matrix, and the contexts used for calibration. Only the singular values and fitted biases and scales change. Each matrix retains rank $R$. The feedback vectors used for sequence generation are also shared across profiles.

**Algorithm 1** Construct the tree and sample an item

**Require:** Branching $B$, depth $m$, context dimension $d$, rank $R$; singular-value profiles; random seed
**Ensure:** A generator that returns an item, its oracle code, and its probability
**Setup: performed once**
Create a balanced $B$-ary tree of depth $m$; assign items to leaves
**for** each level $i$ and parent prefix $p$ **do**
  Construct $\boldsymbol{P}_{i,p}$ with rank $R$ and the prescribed singular values subject to each column summing to zero
**end for**
Set biases $\boldsymbol{b}_{i,p}$ and scales $\alpha_i$ ▷ Calibration: Algorithm 3
**Sampling: repeated for each supplied context** $c$
**procedure** SAMPLEITEM($c$)
  $p \leftarrow ()$; $\pi \leftarrow 1$
  **for** $i = 1, \dots, m$ **do**
    $\boldsymbol{z} \leftarrow \boldsymbol{b}_{i,p} + \alpha_i \boldsymbol{P}_{i,p} c$
    $v \leftarrow \mathrm{softmax}(\boldsymbol{z})$
    Sample child token $a \sim \mathrm{Categorical}(v)$
    $\pi \leftarrow \pi\, v_a$; $p \leftarrow (p, a)$
  **end for**
  $y \leftarrow$ item at leaf $p$
  **return** $(y, p, \pi)$
**end procedure**

**Calibrating biases and level scales.** We calibrate the biases and level scales after constructing $\boldsymbol{P}_{i,p}$. We adjust them so that the profiles have the same target predictive gain at each level and uniform child probabilities after averaging over contexts at each parent. Algorithm 2 constructs the matrices and calls the calibration procedure in Algorithm 3.

Calibration uses a common finite set $\mathcal{C}$ of unit-norm Gaussian directions, disjoint from the student training, validation, and test sets. For each parent $p$, define its context-dependent mass $w_p(c) = \mathbb{P}(P_{i-1} = p \mid C = c)$. Conditioning on that parent weights each calibration context in proportion to $w_p(c)$. For a trial scale $\alpha_i$, we fit the biases so that these weighted child marginals equal $1/B$. We then compute $G_i$ and bisect the scale interval until the gain target is met, refitting the biases at every trial. Levels are fitted from root to leaf, so the parent weights are fixed by the preceding levels. The fitted child probabilities vary with context even though their parent-conditioned averages are uniform.

**Algorithm 2** Construct the calibrated generator

**Require:** Branching $B$, depth $m$, context dimension $d$, mode count $R \le \min\{d, B-1\}$; positive profile vectors $\boldsymbol{\eta}_i$; target gains $G_i^\star$; calibration size $N$; tolerances $\varepsilon$; seed
**Ensure:** Tree and conditional laws $q_{i,p}(\cdot \mid c)$
Create a balanced $B$-ary tree of depth $m$; assign items to leaves
Draw $N$ Gaussian contexts and normalize each to norm 1; collect them in $\mathcal{C}$
**for** $i = 1, \dots, m$ **do**
  Scale $\boldsymbol{\eta}_i$ to $\|\boldsymbol{\eta}_i\|_2^2 = R$; set $\boldsymbol{D}_i \leftarrow \mathrm{diag}(\boldsymbol{\eta}_i)$
  **for** each parent prefix $p$ at level $i$ **do**
    Draw $\boldsymbol{H}_{i,p} \in \mathbb{R}^{d \times R}$ with orthonormal columns
    Draw $\boldsymbol{G}_{i,p} \in \mathbb{R}^{B \times R}$ with orthogonal columns, each summing to zero
    Scale each column of $\boldsymbol{G}_{i,p}$ to norm $\sqrt{B/R}$
    $\boldsymbol{P}_{i,p} \leftarrow \boldsymbol{G}_{i,p} \boldsymbol{D}_i \boldsymbol{H}_{i,p}^\top$
  **end for**
**end for**
$(\{\boldsymbol{b}_{i,p}\}, \{\alpha_i\}) \leftarrow$ CALIBRATE$(\{\boldsymbol{P}_{i,p}\}, \mathcal{C}, \{G_i^\star\}, \varepsilon)$ ▷ Algorithm 3
Define $q_{i,p}(\cdot \mid c) = \mathrm{softmax}(\boldsymbol{b}_{i,p} + \alpha_i \boldsymbol{P}_{i,p} c)$ for each $(i, p)$
**return** tree and conditional laws

**Algorithm 3** Calibrate biases and level scales

**Require:** Fixed tree matrices $\{\boldsymbol{P}_{i,p}\}$; calibration contexts $\mathcal{C}$; target gains $\{G_i^\star\}$; gain and child-marginal tolerances $\varepsilon$

**Ensure:** Biases $\{\boldsymbol{b}_{i,p}\}$ and scales $\{\alpha_i\}$ matching the targets on $\mathcal{C}$

**procedure** CALIBRATE($\{\boldsymbol{P}_{i,p}\}, \mathcal{C}, \{G_i^\star\}, \varepsilon$)
  **for** $i = 1, \dots, m$ **do**
    **for** each parent prefix $p$ at level $i$ **do**
      $\boldsymbol{s}_{i,p}(c) \leftarrow \boldsymbol{P}_{i,p} c$ for every $c \in \mathcal{C}$
      $w_p(c) \leftarrow \mathbb{P}(P_{i-1} = p \mid C = c)$ under the fitted preceding levels ▷ Root weight is 1
    **end for**
    Bracket $G_i^\star$ by doubling the upper scale, refitting biases at each trial
    **repeat**
      $\alpha_i \leftarrow$ midpoint of the current scale interval
      **for** each parent prefix $p$ at level $i$ **do**
        Fit $\boldsymbol{b}_{i,p}$ so that, for every child $a$,
        $\dfrac{\sum_{c \in \mathcal{C}} w_p(c)[\mathrm{softmax}(\boldsymbol{b}_{i,p} + \alpha_i \boldsymbol{s}_{i,p}(c))]_a}{\sum_{c \in \mathcal{C}} w_p(c)} = \dfrac{1}{B}$ within tolerance
        $q_{i,p}(\cdot \mid c) \leftarrow \mathrm{softmax}(\boldsymbol{b}_{i,p} + \alpha_i \boldsymbol{s}_{i,p}(c))$
      **end for**
      Compute $G_i$ on $\mathcal{C}$ using Equation (6)
      If $G_i < G_i^\star$, raise the lower endpoint; otherwise lower the upper endpoint
    **until** gain and child-marginal errors are within tolerance
    Fix this level's biases and scale
  **end for**
  **return** $(\{\boldsymbol{b}_{i,p}\}, \{\alpha_i\})$
**end procedure**

### A.2 State dynamics and numerical settings

We use unit-norm coordinates for feedback and updated contexts. For each tree edge, independently draw a Gaussian vector in $\mathbb{R}^d$ and normalize it to norm 1, obtaining $e_{i,p,a}$. The feedback for item $y$ is the normalized sum of its generating-path edges:

$$e_y = \frac{\sum_{i=1}^m e_{i,p_{i-1}(y),a_i(y)}}{\left\| \sum_{i=1}^m e_{i,p_{i-1}(y),a_i(y)} \right\|_2}. \tag{25}$$

Draw $Z_u \sim \mathcal{N}(0, I_d)$ independently for each user, scale once to obtain the persistent state $S_u = Z_u/\sqrt{d}$, and initialize $C_{u,0} = S_u$. After sampling $Y_{u,t}$ from the current-context item law, update

$$C_{u,t+1} = \frac{\beta S_u + \rho C_{u,t} + \gamma e_{Y_{u,t}}}{\|\beta S_u + \rho C_{u,t} + \gamma e_{Y_{u,t}}\|_2}. \tag{26}$$

The persistent state $S_u$ stays fixed throughout the trajectory. The weights $\beta$, $\rho$, and $\gamma$ determine the contributions of the user state, previous context, and sampled-item feedback; they are nonnegative and sum to one. The update normalizes their weighted sum, so $\|C_{u,t}\|_2 = 1$ for $t \geq 1$. The initial context retains the random length of $S_u$, whose expected squared norm is one. Randomness enters through the initial draws and sampled items; the state update itself is deterministic.

At each step we record the context before drawing the item, the sampled item and its oracle code, and its exact log probability. We discard the burn-in steps and assign complete user trajectories to disjoint training, validation, and test splits.

For parameter seed $s$, the reference-context random seed is $10\,000\,019 + 104\,729s$. Independent $\mathcal{N}(0, I_d)$ draws are normalized to norm 1 and partitioned, in order, into the four sets in Table 3. These sets are shared across profiles at that parameter seed. The trajectory test split instead contains $4{,}096 \times 20 = 81{,}920$ evolving contexts. The fixed-context experiments use the Gaussian directions; the trajectory experiments use the evolving contexts.

Table 3: Generator settings used in the reported synthetic studies. Reference-context counts and recorded-user counts describe different datasets. The larger training sets in the sparse-to-dense experiment are specified separately in Appendix B.5.

| Setting | Value |
|---|---|
| Tree branching and depth | $B = 16$, $m = 3$; 4,096 items |
| Context dimension and nonzero modes | $d = 32$, $R = 15$ at every parent |
| State-update weights | $(\beta, \rho, \gamma) = (0.1, 0.7, 0.2)$ |
| Matching targets | $G_i = 0.30$ nats; each child marginal $1/16$ |
| Fitting tolerances | $10^{-6}$ maximum child-marginal error; $10^{-4}$ nats gain error |
| Reference contexts: fitting / train / validation / test | $4{,}096/2{,}048/512/512$ |
| Recorded users: train / validation / test | $32{,}768/4{,}096/4{,}096$ |
| Burn-in / retained events per user | $5/20$ |
| Source parameter seed | 42 |
| Profile-student parameter seeds | $42, 43, 44$ |
| Source trajectory / user-split seeds | $123/456$ |

### A.3 Realized profile spectra

Effective rank summarizes spectral concentration (Roy & Vetterli, 2007). With $w_{i,j} = \lambda_{i,j}/T_i(0)$ for $T_i(0) > 0$, define

$$r_{\mathrm{eff}}(i) = \exp\left( - \sum_{j : w_{i,j} > 0} w_{i,j} \log w_{i,j} \right), \tag{27}$$

and set it to zero when $T_i(0) = 0$. Here $\lambda_{i,j}$ is the parent-mass-weighted average of the $j$th ordered covariance eigenvalue, as in Equation (11). The corresponding singular-value profile is $(\sqrt{\lambda_{i,j}})_j$.

Table 4 reports the realized effective ranks on the 81,920-context trajectory test cohort for generator seed 42. Every oracle level has numerical rank 15, while the effective ranks reflect the intended spectral profiles. Directions are shared across profiles, and each profile generates its own trajectory cohort. The atomic logit law has numerical rank 287 for each profile.

Table 4: Realized oracle effective ranks on held-out trajectory contexts. Parentwise ordered spectra are averaged with their parent-mass weights.

| Profile | Target | Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| Low constant | $(2, 2, 2)$ | 1.969 | 1.989 | 2.000 |
| Increasing | $(2, 4, 8)$ | 1.973 | 3.968 | 7.945 |
| Reversed | $(8, 4, 2)$ | 7.777 | 3.949 | 1.997 |
| Medium constant | $(8, 8, 8)$ | 7.786 | 7.892 | 7.942 |
| Flat full | $(15, 15, 15)$ | 14.665 | 14.801 | 14.791 |

### A.4 Mapping construction and structural diagnostics

All SID candidates are bijections onto the same balanced code tree. The *oracle* uses the generator hierarchy. Twenty *random* controls independently permute complete balanced codes across items. Two data-derived controls use only the first $N \in \{512, 2048, 8192, 32768\}$ training users. The *popularity* control orders items by training count and assigns lexicographic codes. For the *predictive* control, let $N_y$ be the number of occurrences of item $y$ in the selected training prefix and let $\bar{C}$ be the mean of all contexts in that prefix. It computes

$$\widehat{\mu}_y = \frac{\sum_{t : Y_t = y} C_t + 10\,\bar{C}}{N_y + 10}, \tag{28}$$

so the ten global-mean pseudo-observations shrink rare-item means toward $\bar{C}$. Within each current node, it centers the $\widehat{\mu}_y$ vectors, projects them onto their first right-singular vector, orders the projections, and divides them into 16 equal blocks before recursing. The singular-vector sign is fixed

by making its first nonzero coordinate positive. Popularity and predictive controls use independent seeded random priorities to break ties, avoiding an item-index or oracle-lexicographic tie rule. Random mapping seeds are 789–808; the popularity and predictive tie-priority seeds are 10,007 and 20,011. The predictive control uses the generator contexts $C_t$ recorded with the selected training occurrences.

**Spectral diagnostics.** We enumerate the generator's complete 4,096-item law in float64 at every test context and estimate $G_i$ and $K_{i,p}$ by averaging their exact conditional-law quantities over the finite test-context cohort. Throughout the experiments, "exact-law test" denotes this integration over the exact next-item law followed by an empirical average over held-out contexts. Within each parent, we threshold eigenvalues at $B_{i,p}\epsilon_{\mathrm{mach}}\lambda_{i,p,1}$ in float64, where $B_{i,p}$ is the number of children. We then average the ordered eigenvalues with parent-mass weights. Numerical rank counts nonzero averaged modes; tail fractions and effective rank use that same averaged profile. Each comparison between candidate mappings uses one fixed generator profile and the same evaluation contexts.

For every profile and candidate mapping, we report the predictive-gain sequence, the parentwise spectral energies, their rank-dependent tails, and the corresponding normalized tails $\tau_i(r)$. The atomic case has one parent and is evaluated using its exact hierarchical logit representation.

**Exposure diagnostics.** Parent exposure counts examples with a given target prefix; edge exposure counts how often that prefix is followed by a particular token; token exposure pools those occurrences across parents at the same level. Held-out missing mass is the probability assigned to outcomes absent from training, averaged over the exact test item distributions. Each parent-specific student head receives the examples reaching that parent. Exposure reports contain parent, edge, and level-specific token counts, support, and held-out missing mass at every training budget.

**Item and prefix coverage.** At 512 training users (10,240 transitions), every level-specific token is observed, but only 3,692 of 4,096 items are observed. Unseen items carry $8.58\%$ of held-out mass. Final-level edges correspond bijectively to items, so this leaf-level coverage is identical across mappings. Intermediate coverage is not: all 256 final-refinement parents are observed under the oracle and each random mapping, compared with 231 under popularity and 232 under predictive mapping. Entirely unseen parents carry $8.50\%$ and $8.18\%$ of held-out mass for the latter two controls. The mapping therefore changes where missing supervision appears even when item coverage is fixed. At 32,768 users every item is observed, so the full-budget history-encoding gaps in Appendix D.5 cannot be explained simply by unseen items. Here the data-derived mappings are rebuilt at each construction budget; the recorded-data student experiment instead freezes the mappings constructed from 32,768 source users.

**Information allocation and spectral tails.** For the increasing profile, all mappings preserve total predictive gain $0.8917$ nats on the trajectory test cohort, but expose different refinement laws (Table 5). The oracle distributes gain across all three levels; random mappings place approximately $88\%$ at the final level. The oracle also has the smallest gain-weighted rank-4 tail, followed by predictive, popularity, and random mappings. The mappings therefore allocate the same total information to refinements with different rank-dependent tails.

Table 5: Predictive gains and weighted tails on the increasing-profile trajectory test cohort (81,920 contexts). Data-derived mappings use 32,768 training users; random entries average 20 mappings. Gains and their weighted tail are in nats.

| Mapping | $G_1$ | $G_2$ | $G_3$ | $\sum_i G_i\tau_i(4)$ |
|---|---|---|---|---|
| Oracle | 0.3152 | 0.2901 | 0.2865 | 0.0936 |
| Predictive | 0.2385 | 0.1559 | 0.4973 | 0.2019 |
| Popularity | 0.0713 | 0.0912 | 0.7292 | 0.2528 |
| Random | 0.0074 | 0.1018 | 0.7825 | 0.2753 |

# B SYNTHETIC STUDENT EXPERIMENTS

The fixed-context experiments vary output rank and spectral profile at controlled label budgets per parent. The recorded-data experiments instead train on nested sets of user trajectories. Both use the linear student in Equation (22) and evaluate complete-item KL by summing the parent-mass-weighted levelwise risks.

## B.1 FIXED-CONTEXT STUDENT PROTOCOL AND VARIABILITY

The student class is defined in Equation (22). For the oracle mapping, the true logits and their spectral projections are affine in $C$, satisfying the realizability condition in Equation (19). Remapped conditional laws need not remain affine.

**Experimental grid.** The profile comparison uses three generator seeds (42–44), all five profiles, all three levels, ranks $r \in \{0, 1, 2, 4, 8, 15\}$, budgets of 5, 20, 80, 320, and 1,280 labels per parent, and three initialization seeds (1001–1003), for 4,050 level fits. The mapping comparison uses the increasing-profile source with generator seed 42, the oracle, popularity, and predictive mappings, and random mapping seeds 789–793. It uses the same ranks and initialization seeds at 1,280 labels per parent, for 432 level fits. The data-derived mappings are constructed from 32,768 trajectory-training users.

**Sampling and training.** Context cohorts contain 2,048/512/512 normalized Gaussian vectors for training/validation/test. Within each parent, training contexts are sampled with probability proportional to its exact mass, followed by a conditional child sample. Budgets are nested. Each fit uses AdamW with learning rate $0.01$, weight decay $10^{-4}$, batch size 1,024, gradient clipping at 1, and 30 epochs. At 1,280 labels per parent, this gives 60, 600, and 9,600 optimizer updates at the three successive levels. Test KL is weighted by evaluation parent mass and summed over levels.

**Evaluation and aggregation.** We compute the spectral diagnostics on the same evaluation cohorts as the students and compare raw KL with $T_i(r)$ and $G_i\tau_i(r)$. Deterministic mappings are summarized by mean and standard deviation over initializations. For random mappings, initialization means are computed within each mapping before summarizing across mappings.

The flat-minus-low-constant contrast pairs generator seed, level, and initialization. Its 2,000 hierarchical-bootstrap replicates resample generator seeds and then paired level–initialization differences within each sampled seed.

## B.2 COMPLETE MAPPING CAPACITY COMPARISON

In Figure 2 and Table 6, the random mappings form one family. We average their normalized GWT values and, within each student initialization, their complete-item KLs. The plotted random error bar is the sample standard deviation of these initialization means. It describes training variation after averaging the fixed random constructions. The individual-mapping correlations below retain each random construction separately and therefore use eight points rather than the four family means in the main text.

Table 6: Correlation between GWT and mean trained KL across four mapping families: oracle, predictive, popularity, and averaged random. At ranks 0 and 15, GWT is constant and correlations are undefined.

| Head rank | Pearson | Spearman |
|---|---|---|
| 1 | 0.518 | 0.400 |
| 2 | 0.867 | 0.400 |
| 4 | 0.997 | 1.000 |
| 8 | 1.000 | 1.000 |

At rank 4, the eight individual mappings have Spearman correlation $0.738$ between GWT and mean KL; at rank 8 it is $0.810$. These correlations use individual random mappings, whereas Table 7 averages them for display.

Table 7: Normalized GWT and trained complete-item KL on the fixed mapping-comparison task. KL averages three initializations; random entries then average five mappings. GWT is computed separately for each mapping before averaging. All entries use the same test-context cohort, with total gain 0.8998 nats. Table 8 gives initialization SDs.

| | Oracle | | Predictive | | Popularity | | Random | |
|---|---|---|---|---|---|---|---|---|
| Rank | GWT | KL | GWT | KL | GWT | KL | GWT | KL |
| 0 | 1.000 | 0.923 | 1.000 | 0.924 | 1.000 | 0.923 | 1.000 | 0.923 |
| 1 | 0.478 | 0.741 | 0.508 | 0.767 | 0.595 | 0.773 | 0.630 | 0.759 |
| 2 | 0.269 | 0.629 | 0.388 | 0.722 | 0.461 | 0.727 | 0.489 | 0.712 |
| 4 | 0.107 | 0.463 | 0.237 | 0.596 | 0.294 | 0.674 | 0.315 | 0.679 |
| 8 | 0.022 | 0.518 | 0.085 | 0.620 | 0.114 | 0.672 | 0.124 | 0.686 |
| 15 | 0.000 | 0.619 | 0.000 | 0.702 | 0.000 | 0.726 | 0.000 | 0.743 |

Table 8: Students: complete-item test KL at 1,280 labels per parent on the increasing profile, generator seed 42. Entries are initialization means ± standard deviations (three initializations). Random means average five mapping means; their displayed SD averages the within-mapping initialization SDs. Between-mapping SDs range from 0.0008 to 0.0076 across ranks.

| Rank | Oracle | Predictive | Popularity | Random |
|---|---|---|---|---|
| 0 | $0.9227 \pm 0.0002$ | $0.9236 \pm 0.0010$ | $0.9234 \pm 0.0002$ | $0.9233 \pm 0.0008$ |
| 1 | $0.7407 \pm 0.0460$ | $0.7674 \pm 0.0336$ | $0.7726 \pm 0.0034$ | $0.7589 \pm 0.0039$ |
| 2 | $0.6289 \pm 0.0148$ | $0.7222 \pm 0.0445$ | $0.7270 \pm 0.0155$ | $0.7121 \pm 0.0035$ |
| 4 | $0.4631 \pm 0.0019$ | $0.5957 \pm 0.0167$ | $0.6739 \pm 0.0223$ | $0.6788 \pm 0.0050$ |
| 8 | $0.5176 \pm 0.0033$ | $0.6199 \pm 0.0031$ | $0.6716 \pm 0.0045$ | $0.6860 \pm 0.0059$ |
| 15 | $0.6190 \pm 0.0170$ | $0.7019 \pm 0.0117$ | $0.7260 \pm 0.0088$ | $0.7433 \pm 0.0074$ |

**Effect of head rank.** At ranks 1 and 2, random mappings have larger GWT than the data-derived controls but lower mean KL; the eight-mapping Spearman correlations are $-0.048$ and $0.000$. Increasing rank always reduces GWT, but need not improve a trained model: oracle KL rises from $0.463$ at rank 4 to $0.619$ at rank 15, where its tail is zero. At rank 15 the oracle student class contains the true conditional laws, so its positive trained KL comes from finite-data learning and optimization. For remapped laws, nonlinearity in $C$ can additionally prevent a linear student from realizing the projected logits.

### B.3 Levelwise weighted-tail comparison

The main diagnostic sums $G_i\tau_i(r)$ and divides by total gain. Across 45 seed–profile–level means, the weighted contribution and level KL have Pearson correlation $0.849$; within levels 1, 2, and 3 the values are $0.769$, $0.996$, and $0.999$. Figure 5 examines its unnormalized contributions separately, keeping the prediction level visible. Each point uses the same test contexts for the contribution and level KL.

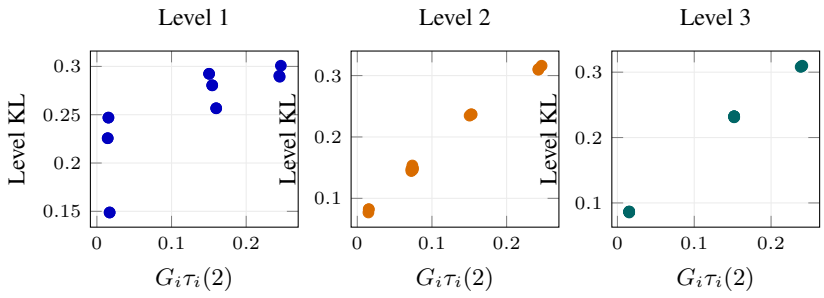


Figure 5: At head rank 2 and 1,280 labels per parent, level contributions $G_i\tau_i(2)$ versus trained level KL. Each panel contains five profiles and three generator seeds, with KL averaged over three initializations. Both axes are in nats.

**Weighted and unweighted tails.** Table 9 compares GWT with the arithmetic mean $\frac{1}{3}\sum_{i=1}^{3}\tau_i(r)$ on the eight fixed-task mappings. Both use the same test contexts and trained KL values. GWT has the higher rank correlation at ranks 1, 2, and 8; the unweighted mean has the higher correlation at rank 4. The mapping comparison varies both gain allocation and spectra, whereas the profile comparison holds the level gains approximately equal.

Table 9: Spearman correlation with trained complete-item KL across the eight synthetic mappings. Higher correlation means closer agreement with the observed ordering.

| Head rank | GWT | Unweighted mean tail |
|---|---|---|
| 1 | $-0.048$ | $-0.357$ |
| 2 | $0.000$ | $-0.190$ |
| 4 | $0.738$ | $0.857$ |
| 8 | $0.810$ | $0.571$ |

### B.4 Spectral profiles and trained performance

This experiment studies how spectral concentration affects the error achieved by training when the student class can realize the generator's spectral projections. For each profile, we train a separate student using the generator's original item codes. Head rank and training budget are held fixed when comparing profiles.

**Spectral profiles.** We vary the singular values of the generator matrices $\boldsymbol{P}_{i,p}$ while retaining their directions and numerical rank. The five profiles in Table 10 control how concentrated these singular values are at each level. The target effective ranks are computed from the normalized squared singular values. Biases and level scales are calibrated to common marginal and gain targets, as described in Appendix A.

Table 10: Generator profiles. Targets are effective ranks at successive levels; all profiles have the same numerical rank. Measured logit covariance spectra are reported in Appendix A.

| Profile | Target effective ranks | Spectral concentration |
|---|---|---|
| Low constant | $(2, 2, 2)$ | Strong at every level |
| Medium constant | $(8, 8, 8)$ | Moderate at every level |
| Increasing | $(2, 4, 8)$ | Decreases with depth |
| Reversed | $(8, 4, 2)$ | Increases with depth |
| Flat | $(15, 15, 15)$ | Equal singular values at every level |

For the profile comparison at head rank 2 and 1,280 labels per parent, the mean GWT is $0.0516$ for the low-constant profile and $0.8179$ for the flat profile; their mean KLs are $0.374$ and $0.915$ nats. Across the 15 generator-seed–profile means, Pearson correlation is $0.973$ and Spearman correlation is $0.836$. The paired flat-minus-low-constant KL difference is $0.180$ nats per refinement, with a hierarchical-bootstrap $95\%$ interval $[0.146, 0.209]$; 26 of 27 paired differences are positive. Appendix B.3 reports the levelwise association.

Figure 1 shows that concentrated profiles generally have smaller GWT and lower trained KL than diffuse profiles, despite matched gain and numerical rank. Table 11 reports the correlations at each head rank.

At rank 2, the increasing and reversed profiles have similar mean GWT ($0.269$ and $0.273$), with mean KLs $0.589$ and $0.511$. At full generator rank, the oracle student class contains the true conditional laws. Its positive trained KL in Table 12 then measures the remaining learning cost.

**Complete capacity results.** Table 12 retains all six head ranks in the three-seed profile comparison at the largest label budget. Its entries sum the three refinement KLs before averaging across generator seeds and initializations; they are complete-item risks rather than the per-refinement paired contrast reported in Appendix B.1.

Table 11: Correlation between GWT and mean trained KL across the 15 profile–generator combinations. Means average student initializations. Correlations are undefined when GWT is constant, at ranks 0 and 15.

| Head rank | Pearson | Spearman |
|---|---|---|
| 1 | 0.958 | 0.868 |
| 2 | 0.973 | 0.836 |
| 4 | 0.989 | 0.907 |
| 8 | 0.973 | 0.914 |

Table 12: Mean complete-item test KL at 1,280 labels per parent, averaged over three generator seeds and three initializations. Each column uses the indicated rank at every parent.

| Profile | $r = 0$ | $r = 1$ | $r = 2$ | $r = 4$ | $r = 8$ | $r = 15$ |
|---|---|---|---|---|---|---|
| Low constant | 0.9175 | 0.5317 | 0.3738 | 0.3486 | 0.4594 | 0.5628 |
| Increasing | 0.9179 | 0.7411 | 0.5886 | 0.4600 | 0.5120 | 0.6125 |
| Reversed | 0.9147 | 0.6302 | 0.5108 | 0.4738 | 0.5314 | 0.5943 |
| Medium constant | 0.9155 | 0.8225 | 0.7442 | 0.6365 | 0.6056 | 0.6458 |
| Flat full | 0.9131 | 0.9201 | 0.9145 | 0.8947 | 0.8333 | 0.7118 |

### B.5 Recorded-data training protocol

**Data, mappings, and evaluation.** We reuse the increasing-profile generator with parameter seed 42, but generate three independent largest training datasets with trajectory-sampling seeds 2001–2003. Each retains 32,768 training users after user-level splitting (seed 456), with five burn-in events and 20 recorded transitions per user. One fixed shuffle per dataset (seed 2601) defines nested budgets of 32, 128, 512, 2,048, 8,192, and 32,768 users. Every SID and atomic condition receives the same recorded context–item examples within a data seed and budget. The data-derived mappings are frozen artifacts constructed from the original source's 32,768 training users. Those construction examples are additional to the reported student training budgets. Random seed 789 defines the fixed random mapping.

Validation and test use 128 and 512 users from the original source's disjoint held-out splits (trajectory-sampling seed 123), shared across all fits. The test cohort therefore contains 10,240 recorded contexts. Test KL integrates over the exact next-item law at each context, uses evaluation parent-mass weights, and sums the three SID refinement risks. Both the training and evaluation contexts in this experiment come from the state dynamics in Equation (26).

**Training and capacity.** We use the student parameterization in Equation (22). Atomic prediction uses one 4,096-class head. Table 13 reports the actual trainable parameter counts, including biases and both factors. Atomic ranks immediately below and above each SID parameter budget are both retained.

All fits use initialization seed 1001, add-one frequency-bias initialization, AdamW with learning rate $0.01$, zero weight decay, batch size 1,024, gradient clipping at 1, and 30 epochs. We report the final checkpoint, without validation-based checkpoint selection. The six budgets give 30, 90, 300, 1,200, 4,800, and 19,200 optimizer updates per fitted SID level or atomic model. The SID students use ranks 2 and 4; atomic ranks 6 and 7 bracket the rank-2 SID parameter count, and ranks 12 and 13 bracket the rank-4 count. There are $4 \times 2 \times 3 \times 6 \times 3 = 432$ SID level fits and $4 \times 6 \times 3 = 72$ atomic fits, forming 216 complete predictors. Means and sample SDs are over the three training-data seeds, with one fixed generator and one initialization.

### B.6 Learning curves from recorded trajectories

A larger head can reduce approximation error, but its extra parameters must be estimated from the available data. We ask whether SID advantages at sparse budgets persist as the sample grows, using the same student class throughout.

Table 13: Parameter brackets for the recorded-data student comparison. SID counts sum all three levels; atomic counts are for one complete-item head. The final column is the atomic-minus-SID relative difference.

| SID rank | SID parameters | Atomic rank | Atomic parameters | Difference |
|---|---|---|---|---|
| 2 | 30,576 | 6 | 28,864 | −5.60% |
| 2 | 30,576 | 7 | 32,992 | +7.90% |
| 4 | 56,784 | 12 | 53,632 | −5.55% |
| 4 | 56,784 | 13 | 57,760 | +1.72% |

Table 14: Complete recorded-data learning curves: mean test KL over three training-data seeds. Columns give training item transitions, not users. Corresponding data-seed SDs are shown in Figure 6.

| Mapping | Rank | 640 | 2,560 | 10,240 | 40,960 | 163,840 | 655,360 |
|---|---|---|---|---|---|---|---|
| Oracle | 2 | 2.595 | 2.391 | 2.629 | 1.215 | 0.409 | 0.323 |
| Predictive | 2 | 2.630 | 2.466 | 2.722 | 1.308 | 0.511 | 0.426 |
| Popularity | 2 | 2.606 | 2.557 | 2.849 | 1.425 | 0.615 | 0.536 |
| Random 789 | 2 | 2.609 | 2.590 | 2.920 | 1.416 | 0.632 | 0.555 |
| Atomic | 6 | 1.822 | 2.961 | 2.429 | 0.866 | 0.534 | 0.474 |
| Atomic | 7 | 1.979 | 3.363 | 2.780 | 0.895 | 0.523 | 0.457 |
| Oracle | 4 | 3.013 | 2.944 | 4.013 | 2.076 | 0.350 | 0.203 |
| Predictive | 4 | 2.973 | 2.967 | 4.107 | 2.208 | 0.491 | 0.344 |
| Popularity | 4 | 3.063 | 3.109 | 4.112 | 2.361 | 0.582 | 0.434 |
| Random 789 | 4 | 3.125 | 3.164 | 4.205 | 2.314 | 0.592 | 0.448 |
| Atomic | 12 | 2.914 | 5.751 | 4.413 | 1.076 | 0.478 | 0.374 |
| Atomic | 13 | 2.998 | 6.176 | 4.821 | 1.112 | 0.472 | 0.359 |

The protocol in Appendix B.5 holds the generator, mappings, and evaluation contexts fixed while increasing the number of training users. Students receive the recorded contexts directly, so their model input is $X = C$.

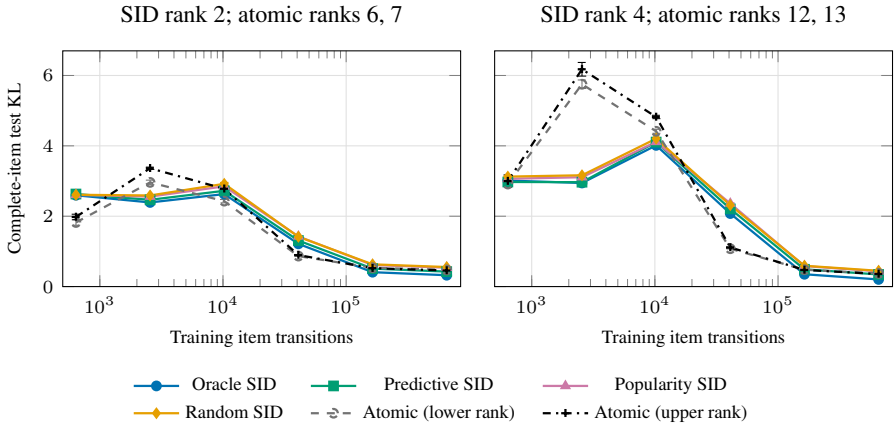


Figure 6: Recorded-data student learning curves. Each panel compares a fixed SID rank with two atomic ranks whose parameter counts bracket the SID total (Table 13). Solid curves are SID mappings; dashed and dash-dotted curves are atomic models. Points are mean complete-item test KL over three independent training-data seeds; bars show their sample SD. The large sparse-data SID advantage in the higher-capacity panel narrows at larger budgets, with an intermediate reversal and a mapping-dependent dense-data ordering.

**Mapping comparisons across data budgets.** Figure 6 shows the largest advantage in the higher-capacity comparison. At 2,560 training transitions, all four rank-4 SID schemes have mean KL

between 2.944 and 3.164, versus 5.751 and 6.176 for atomic ranks 12 and 13. At 10,240 transitions, every SID scheme still outperforms both atomic models: 4.013–4.205 versus 4.413–4.821. Both comparisons hold separately in all three data seeds. At 655,360 transitions, the differences are much smaller and no longer favor SIDs generally: oracle and predictive KLs are 0.203 and 0.344, below the atomic values 0.374 and 0.359, while popularity and random KLs are 0.434 and 0.448. Thus the sparse-data benefit is shared across the SID mappings, whereas the dense-data ordering depends on the mapping. The lower-capacity panel also shows a shared SID advantage at 2,560 transitions, but a smaller one. The trend is not monotone: there is no shared SID advantage at the smallest budget, and the higher-capacity ordering reverses at 40,960 transitions before separating again.

**Capacity across data budgets.** For every SID mapping, rank 4 has higher mean KL than rank 2 through 40,960 transitions, but lower KL at 163,840 and 655,360 transitions. The intermediate-budget SID error peaks are concentrated at the final refinement. For example, at 10,240 transitions it contributes 3.727 of the oracle rank-4 model's total 4.013 KL. All parent prefixes are already observed by 2,560 transitions; observing a parent does not imply having enough examples to estimate its predictor. The crossover between ranks 2 and 4 shows how the useful output capacity changes with the training budget.

**Exposure and the intermediate-budget peaks.** Every level-specific token is observed at the smallest budget of 640 transitions. Yet unseen items account for 84.83% of held-out mass, averaged over data seeds; final SID edges have exactly the same missing mass because each identifies one item. At 2,560 transitions all parent prefixes are observed under every fixed mapping. The final refinement nevertheless has only ten training examples per parent on average at this budget. Each parent-specific head is trained on the examples reaching that parent.

At 10,240 transitions, the final-refinement contribution to oracle rank-4 test KL is 3.727, compared with 0.015 and 0.271 at the first two levels. Its final-level training NLL is 0.413, versus sampled test NLL 6.181, showing substantial overfitting. Atomic rank 13 has a pronounced error peak at the smaller budget of 2,560 transitions, with training NLL 3.141 and sampled test NLL 13.585. These training–test gaps accompany the intermediate-budget peaks in the learning curves.

# C Proofs

We first give the mutual-information identities used to define gains and remainders. The spectral bounds then follow from two ingredients: optimal affine approximation of centered logits and curvature bounds for the softmax loss.

## C.1 Gain as a mean child divergence

The same gain can be expressed using the context distributions associated with the children. Define $\nu_{i,p,a} = \mathbb{P}(C \in \cdot \mid P_{i-1} = p, A_i = a)$ and $\bar{\nu}_{i,p} = \mathbb{P}(C \in \cdot \mid P_{i-1} = p)$.

**Identity.** For every refinement,

$$G_i = \sum_p \rho_i(p) \sum_{a \in \mathcal{A}_{i,p}} \bar{q}_{i,p}(a) \mathrm{KL}(\nu_{i,p,a} \,\|\, \bar{\nu}_{i,p}) \,. \tag{29}$$

Each child's context distribution is compared with the mixture over all children of its parent, using their conditional occurrence probabilities as weights.

**Proof.** The two standard conditional-mutual-information identities are

$$I(A_i; C \mid P_{i-1}) = \mathbb{E}_{P_{i-1},C}[\mathrm{KL}(\mathbb{P}(A_i \in \cdot \mid C, P_{i-1}) \,\|\, \mathbb{P}(A_i \in \cdot \mid P_{i-1}))] \,, \tag{30}$$

$$= \mathbb{E}_{P_{i-1},A_i}[\mathrm{KL}(\mathbb{P}(C \in \cdot \mid A_i, P_{i-1}) \,\|\, \mathbb{P}(C \in \cdot \mid P_{i-1}))] \,. \tag{31}$$

Substituting the definitions of $q_{i,p}$ and $\bar{q}_{i,p}$ into the first identity gives Equation (6). Conditioning the second identity on $P_{i-1} = p$ and summing over $A_i = a$ gives Equation (29). □

### C.2 Proof of Proposition 1

The mutual-information chain rule and $P_i = (P_{i-1}, A_i)$ give

$$I(P_m; C) = \sum_{i=1}^{m} I(A_i; C \mid P_{i-1}) = \sum_{i=1}^{m} G_i. \tag{32}$$

Because the complete code is injective, $P_m$ and $Y$ determine one another, so $I(P_m; C) = I(Y; C)$. □

### C.3 Spectral approximation bounds

**Inputs and the loss decomposition.** Work with the joint law of $(C, X, Y)$ and assume

$$\mathbb{P}(Y \in \cdot \mid C, X) = \mathbb{P}(Y \in \cdot \mid C). \tag{33}$$

This holds for $X = C$, and follows from Equation (1) when $X$ is a function of history. It identifies the conditional KL below with excess log-loss on observed targets.

For $f \in \mathcal{M}_r$, write $\widehat{q}_{i,p} = \mathrm{softmax}(\widehat{\boldsymbol{\ell}}_{i,p})$ on the actual children. The item law is

$$\widehat{q}_f(y \mid x) = \prod_{i=1}^{m} \widehat{q}_{i,p_{i-1}(y)}(a_i(y) \mid x). \tag{34}$$

Define the levelwise risk by

$$\mathcal{R}_i(f) = \mathbb{E}_{C,X,P_{i-1}}\left[\mathrm{KL}\big(q_{i,P_{i-1}}(\cdot \mid C) \,\|\, \widehat{q}_{i,P_{i-1}}(\cdot \mid X)\big)\right]. \tag{35}$$

By Equation (33), the conditional distribution of $A_i$ given $(C, X, P_{i-1})$ is $q_{i,P_{i-1}}(\cdot \mid C)$. Hence this KL is the expected difference between the model's token log-loss and the true context-conditioned token log-loss. Summing over levels yields $\sum_i \mathcal{R}_i(f) = \mathbb{E}[-\log \widehat{q}_f(Y \mid X)] - \mathrm{H}(Y \mid C)$, which is $\mathcal{R}_\phi(f)$ in Equation (16).

**Probability and support conditions.** Fix an integer $r \geq 0$. Assume throughout that $q_{i,p}(a \mid c) \geq \epsilon > 0$ on the support of $(C, P_{i-1})$ for each actual child $a \in \mathcal{A}_{i,p}$. This bound makes the centered logits bounded and their covariances finite, including for continuous context distributions.

All softmax distributions are defined on the actual child set $\mathcal{A}_{i,p}$. Each parent's centered logits and projections therefore lie in $\mathbf{1}^\perp \subseteq \mathbb{R}^{B_{i,p}}$. Children outside this set have probability zero and are excluded from the logit vector. Eigenvalue lists are zero-padded as in Equation (11).

**Theorem 2(a): lower bound.** Let the level-$i$ logits of $f$ have the rank-$r$ form in Equation (14). If $\widehat{q}_{i,p}(a \mid x) \geq \epsilon$ for every actual child on the joint support of $(C, X, P_{i-1})$, then

$$\mathcal{R}_i(f) \geq \frac{\epsilon}{2} T_i(r) \geq 2\epsilon G_i \tau_i(r). \tag{36}$$

Consequently, for any complete-predictor class $\mathcal{M}_r$ with these rank restrictions, Equation (18) holds if its infimum is attained by a predictor with all fitted child probabilities at least $\epsilon$. Attainment can be replaced by a minimizing sequence satisfying that same uniform bound. The feature functions may be nonlinear.

**Theorem 2(b): upper bound.** For each parent define the optimal rank-$r$ spectral projection

$$\boldsymbol{\ell}_{i,p}^{(r)}(c) = \boldsymbol{\mu}_{i,p} + \Pi_{E_{i,p,r}} \boldsymbol{u}_{i,p}(c). \tag{37}$$

Here $E_{i,p,r}$ has dimension $\min\{r, B_{i,p} - 1\}$, so the projection equals the true logits when the available rank covers the entire centered child space. At a tied eigenvalue cutoff, realizability may use any one choice of the corresponding optimal projections. Suppose there is a single predictor $f^{(r)} \in \mathcal{M}_r$ such that, at every parent and level,

$$\widehat{\boldsymbol{\ell}}_{i,p}^{\,f^{(r)}}(X) = \boldsymbol{\ell}_{i,p}^{(r)}(C) \quad \text{almost surely conditional on } P_{i-1} = p. \tag{38}$$

Then its levelwise risks satisfy

$$\mathcal{R}_i(f^{(r)}) \leq \frac{1}{4} T_i(r) \leq \frac{G_i \tau_i(r)}{2\epsilon}. \tag{39}$$

Summing and taking the class infimum gives Equation (19). The same predictor must realize the projections at every parent and level, respecting the model's parameter-sharing constraints.

**Spectral geometry.** Fix a level $i$. We take $\boldsymbol{\beta}_{i,p}$ and the columns of $V_{i,p}$ to be centered across children, as in Equation (14). For any predictor of this form, let $E_p$ be the column space of its matrix $V_{i,p}$ and let $\Pi_{E_p}$ be the orthogonal projector onto that space. Its dimension is at most $\min\{r, B_{i,p} - 1\}$; no orthonormality or full column rank of $V_{i,p}$ is assumed. Pointwise in $(c, x)$,

$$\|\boldsymbol{\ell}_{i,p}(c) - \widehat{\boldsymbol{\ell}}_{i,p}(x)\|_2^2 \geq \|(I - \Pi_{E_p})(\boldsymbol{\ell}_{i,p}(c) - \boldsymbol{\beta}_{i,p})\|_2^2. \tag{40}$$

The conditional mean of $\boldsymbol{u}_{i,p}(C)$ is zero, so

$$\begin{aligned}\mathbb{E}\big[\|(I - \Pi_{E_p})(\boldsymbol{\ell}_{i,p}(C) - \boldsymbol{\beta}_{i,p})\|_2^2 \mid P_{i-1} = p\big] \\ = \|(I - \Pi_{E_p})(\boldsymbol{\mu}_{i,p} - \boldsymbol{\beta}_{i,p})\|_2^2 + \operatorname{tr}((I - \Pi_{E_p})K_{i,p}) \\ \geq \sum_{j>r} \lambda_{i,p,j}.\end{aligned} \tag{41}$$

The last step is the Ky Fan principle within this parent's centered child space. It also holds when $r \geq B_{i,p} - 1$, since the padded tail is zero. Averaging these scalar residuals over parents yields

$$\mathbb{E}\|\boldsymbol{\ell}_{i,P_{i-1}}(C) - \widehat{\boldsymbol{\ell}}_{i,P_{i-1}}(X)\|_2^2 \geq T_i(r). \tag{42}$$

The projected logits in Equation (37) attain equality in squared-logit error when they can be realized from $X$.

**KL curvature and gain.** For centered logits $z, \widehat{z} \in \mathbf{1}^\perp \subseteq \mathbb{R}^{B_{i,p}}$, conditional KL is the Bregman divergence of log-sum-exp. Its Hessian at probability vector $s$ is $\operatorname{diag}(s) - ss^\top$. For any $v \in \mathbf{1}^\perp$,

$$\epsilon\|v\|_2^2 \leq v^\top(\operatorname{diag}(s) - ss^\top)v \leq \frac{1}{2}\|v\|_2^2 \tag{43}$$

whenever every coordinate of $s$ is at least $\epsilon$; the upper inequality holds for every probability vector $s$. Integrating along the logit segment contributes a factor $1/2$. Thus KL is at most one quarter of squared-logit distance globally, and at least $\epsilon/2$ times that distance when both endpoint laws are $\epsilon$-interior. The latter condition suffices because the normalized geometric interpolation between the endpoint laws remains $\epsilon$-interior. Explicitly, each unnormalized coordinate $q_a^{1-t}\widehat{q}_a^{\,t}$ is at least $\epsilon$, and their sum is at most one by Hölder's inequality. Normalizing cannot decrease a coordinate. For a one-child parent the centered space is $\{0\}$ and all these divergences and residuals are zero.

The best context-independent law at parent $p$ is $\bar{q}_{i,p}$; its KL risk, averaged over parents, is $G_i$. For the lower bound, compare the true logits with the centered logits of $\bar{q}_{i,p}$. Both laws have probabilities at least $\epsilon$, and the mean $\boldsymbol{\mu}_{i,p}$ minimizes squared error among constant logit vectors. For the upper bound, use $\operatorname{softmax}(\boldsymbol{\mu}_{i,p})$ as a constant candidate and apply the global upper curvature bound. Averaging over parents gives

$$\frac{\epsilon}{2}T_i(0) \leq G_i \leq \frac{1}{4}T_i(0). \tag{44}$$

**Proof of Theorem 2(a).** Apply the lower KL curvature bound to the level-$i$ predictions of $f$ and use Equation (42). This gives $\mathcal{R}_i(f) \geq \epsilon T_i(r)/2$. The inequality $G_i \leq T_i(0)/4$ and the identity $T_i(r) = T_i(0)\tau_i(r)$ give Equation (36). Sum over levels and apply the result to the class minimizer, or to the stated minimizing sequence, to obtain Equation (18). □

**Proof of Theorem 2(b).** The realizable predictor $f^{(r)}$ has squared-logit error exactly $T_i(r)$ at each level. The global upper KL curvature bound gives $\mathcal{R}_i(f^{(r)}) \leq T_i(r)/4$. Using $G_i \geq \epsilon T_i(0)/2$ proves Equation (39). This one predictor belongs to $\mathcal{M}_r$, so its summed risk upper-bounds the class infimum. Zero-variation levels have $G_i = T_i(r) = 0$ and obey the same conclusions under the stated convention for $\tau_i$. □

**Input information and the reference risk.** Equation (33) implies

$$\mathcal{R}_\phi(f) = \big(\mathbb{E}[-\log \widehat{q}_f(Y \mid X)] - \mathrm{H}(Y \mid X)\big) + I(Y; C \mid X). \tag{45}$$

The first term is excess log-loss relative to the Bayes predictor given the available input; the second is information supplied by $C$ beyond that input. For $X = C$ it vanishes. For an injective history encoding $X = \psi(H)$, it equals $I(Y; C \mid H)$ and vanishes when $C = C(H)$ is sufficient. The SASRec analysis below uses the history-conditioned law and measures excess loss relative to $\mathrm{H}(Y \mid H)$.

# D History-Conditioned Output Analysis

## D.1 History-conditioned laws and gains

The generating context $C$ may be hidden from a model that observes only $H$. By context sufficiency, the probability law available to that model is

$$\mathbb{P}(Y = y \mid H) = \mathbb{E}[\mathbb{P}(Y = y \mid C) \mid H]. \tag{46}$$

This averages the next-item probabilities over the contexts compatible with the observed history. Use the same target prefixes and put tildes on the quantities defined from $(H, Y)$:

$$\widetilde{q}_{i,p}(a \mid h) = \mathbb{P}(A_i = a \mid H = h, P_{i-1} = p), \tag{47}$$

$$\widetilde{G}_i = I(A_i; H \mid P_{i-1}), \qquad \widetilde{D}_i = I(Y; H \mid P_i). \tag{48}$$

In particular,

$$\widetilde{G}_i = \mathbb{E}_{H,P_{i-1}}\left[\mathrm{KL}\big(\widetilde{q}_{i,P_{i-1}}(\cdot \mid H) \,\|\, \bar{q}_{i,P_{i-1}}\big)\right], \qquad \sum_i \widetilde{G}_i = I(Y; H). \tag{49}$$

The conditional chain rule applied to $(A_i, C, H)$ given $P_{i-1}$ gives

$$\begin{aligned} G_i - \widetilde{G}_i &= I(A_i; C \mid H, P_{i-1}) - I(A_i; H \mid C, P_{i-1}) \\ &= I(A_i; C \mid H, P_{i-1}) \geq 0. \end{aligned} \tag{50}$$

The final equality follows from context sufficiency: conditioning additionally on a target prefix preserves the conditional independence of the target and history given $C$. The same argument gives

$$D_i - \widetilde{D}_i = I(Y; C \mid H, P_i), \qquad \sum_i (G_i - \widetilde{G}_i) = I(Y; C \mid H). \tag{51}$$

The history-based gains have the same remainder identity, $\widetilde{D}_i = \sum_{j=i+1}^{m} \widetilde{G}_j$. Their mean-KL representations follow from the conditional-mutual-information identities under $(H, Y)$. The gains and logit spectra for history-based prediction are both computed under the joint law of $(H, Y)$.

## D.2 Spectral bounds for prediction from history

Define $\widetilde{\boldsymbol{\ell}}_{i,p}(h)$ by Equation (8), replacing $q_{i,p}$ with $\widetilde{q}_{i,p}$ on the same actual child set $\mathcal{A}_{i,p}$. Let

$$\widetilde{\boldsymbol{\mu}}_{i,p} = \mathbb{E}[\widetilde{\boldsymbol{\ell}}_{i,p}(H) \mid P_{i-1} = p], \tag{52}$$

$$\widetilde{K}_{i,p} = \mathbb{E}\Big[(\widetilde{\boldsymbol{\ell}}_{i,p}(H) - \widetilde{\boldsymbol{\mu}}_{i,p})(\widetilde{\boldsymbol{\ell}}_{i,p}(H) - \widetilde{\boldsymbol{\mu}}_{i,p})^\top \mid P_{i-1} = p\Big]. \tag{53}$$

Order the eigenvalues on $\mathbf{1}^\perp$ decreasingly as $\widetilde{\lambda}_{i,p,j}$ and zero-pad to length $B_i - 1$. Define

$$\widetilde{T}_i(r) = \sum_p \rho_i(p) \sum_{j>r} \widetilde{\lambda}_{i,p,j}, \qquad \widetilde{\tau}_i(r) = \frac{\widetilde{T}_i(r)}{\widetilde{T}_i(0)}, \tag{54}$$

with $\widetilde{\tau}_i(r) = 0$ if $\widetilde{T}_i(0) = 0$. Ranks above a parent's centered child dimension and one-child parents are handled by the same zero-tail conventions as in the context analysis.

For history-based predictors with the parentwise rank restriction of Equation (14), define

$$\widetilde{\mathrm{App}}_r(\mathcal{M}) = \inf_{f \in \mathcal{M}_r} \mathbb{E}_H[\mathrm{KL}(\mathbb{P}(Y \in \cdot \mid H) \,\|\, \widehat{q}_f(\cdot \mid H))]. \tag{55}$$

The notation $\widehat{q}_f(\cdot \mid H)$ includes the fixed history encoding and the model's feature computation. Applying Theorem 2 to $(H, Y)$ gives, when both sets of assumptions hold,

$$2\epsilon \sum_i \widetilde{G}_i \widetilde{\tau}_i(r) \;\leq\; \widetilde{\mathrm{App}}_r(\mathcal{M}) \;\leq\; \frac{1}{2\epsilon} \sum_i \widetilde{G}_i \widetilde{\tau}_i(r). \tag{56}$$

For the lower bound, the true child probabilities and an attaining predictor's child probabilities must be at least $\epsilon$ on their support; an attaining predictor can again be replaced by a uniformly bounded

minimizing sequence. For the upper bound, the true probabilities must be at least $\epsilon$ and one member of the class must realize all the mean-plus-projection logits simultaneously from its available history input.

To verify the extension, repeat the proof of Appendix C.3 with $H$ replacing $C$. The parentwise squared-error residual is $\widetilde{T}_i(r)$ and the gain comparison is $\epsilon\widetilde{T}_i(0)/2 \le \widetilde{G}_i \le \widetilde{T}_i(0)/4$. These give the lower bound for rank-restricted logits and the upper bound for the jointly realizable projections. The reference entropy is $\mathrm{H}(Y \mid H)$. The normalized diagnostic uses $\sum_i \widetilde{G}_i\widetilde{\tau}_i(r)/I(Y;H)$ when $I(Y;H) > 0$.

### D.3 Rank-restricted SASRec

Keep the history encoded as atomic items and vary the target mapping. Each model learns its own embeddings, history features, and output head. Suppose its next-token logits have a bias and a linear output matrix of rank at most $r$, applied to features of the history and target prefix. For any fixed parent, restricting the output rows to its actual children and centering the logits cannot increase that rank. The model therefore satisfies Equation (14), regardless of the nonlinear computation of its features.

The history-conditioned lower bound applies under Theorem 2(a)'s probability conditions. The upper bound requires one model to realize all the history-conditioned projections, respecting any constraints from sharing an output head across parents.

### D.4 Estimating the history-conditioned quantities

Our Amazon diagnostic uses the fitted item law $\widehat{p}_{\mathrm{ref}}(y \mid h)$ of an atomic-history, atomic-target SASRec reference with no imposed low-rank output factorization. Let $\mu$ be the uniform empirical distribution over the selected evaluation histories. Together they define

$$\widehat{\mathbb{P}}_{\mathrm{ref}}(H = h, Y = y) = \mu(h)\widehat{p}_{\mathrm{ref}}(y \mid h). \tag{57}$$

Every candidate mapping is applied to this same joint law. Parent masses, child marginals, conditional means, and spectra are all computed under it. The resulting gains obey the conservation identity for this fitted reference law. The comparison models' NLL is evaluated on observed held-out targets.

### D.5 History-encoding comparison

This synthetic comparison changes the history code while fixing atomic targets. Let $\psi$ denote the history item mapping, extended by concatenating complete codewords:

$$\psi(H) = \psi(Y_1) \,\|\, \cdots \,\|\, \psi(Y_T). \tag{58}$$

Its population complete-item risk for a restricted sequence-model class is

$$R_{\mathcal{M}}(\psi) = \inf_{f \in \mathcal{M}} \mathbb{E}_{H,Y}[-\log \widehat{q}_f(Y \mid \psi(H))]. \tag{59}$$

An injective history encoding preserves the information in $H$ and hence the unrestricted Bayes loss, but need not preserve this class optimum. Fitted models can incur additional estimation and optimization error. The history-conditioned atomic gain and spectrum are unchanged by this encoding. Performance differences measure how the model learns from the alternative representations of the same history.

**Protocol.** We hold the atomic target fixed and train SASRec on the increasing-profile trajectories using atomic, oracle, predictive, popularity, or random SID histories. Complete three-token codewords replace each observed item, and only item boundaries predict the next atomic item. All conditions use the same training transitions, a 4,098-row input embedding table, an independent untied 4,096-class output head, two Transformer layers, and embedding width 64. The SID models retain unused input rows to match parameter counts: every condition has 628,608 trainable parameters, uses the same corpus of 655,360 training transitions, and receives 3,840 optimizer updates.

Training uses 30 epochs and restores the checkpoint with the best atomic validation NLL. Three initialization seeds are paired across the four deterministic histories and five random mappings, giving 27 fits. Evaluation predicts the final held-out item for each of 4,096 test users.

For a fitted predictor $f$, exact-law KL satisfies

$$\mathbb{E}_{H,C}[\mathrm{KL}(\mathbb{P}(Y \in \cdot \mid C) \,\|\, \widehat{q}_f(\cdot \mid \psi(H)))] = \mathbb{E}_{H,C,Y}[-\log \widehat{q}_f(Y \mid \psi(H))] - \mathrm{H}(Y \mid C). \quad (60)$$

The entropy term is common across history mappings. KL differences therefore measure differences in expected log-loss. Normalized Recall@10 is the exact item mass captured by the model's top ten divided by that of the oracle top ten; normalized NDCG@10 applies logarithmic rank discounting before taking the same ratio. Aggregation follows the initialization-first, mapping-second convention of the student comparison.

**Results.** Atomic and oracle-SID histories have similar mean KL, $0.6716$ and $0.6743$ (Table 15). Their paired difference changes sign across initializations: the oracle-minus-atomic mean is $0.0027$ nats, with paired standard deviation $0.0097$. Predictive, popularity, and random histories increase mean KL by $0.0330$, $0.1384$, and $0.2190$ nats relative to atomic histories. Each is worse than atomic in every paired initialization and, for random histories, every tested mapping. Oracle-normalized Recall@10 is also substantially lower for popularity and random histories. The history encoding can therefore impede learning despite preserving item identity, even with equal parameter counts and complete training-item coverage.

Table 15: History-encoding comparison with fixed atomic outputs: all conditions have 628,608 parameters and 3,840 training updates. KL reports initialization mean ± SD; random aggregation follows Table 8 and has between-mapping KL SD $0.00008$. Differences use the atomic baseline. Recall is oracle-normalized and reports means.

| History | Test KL ↓ | KL difference | Recall@10 ↑ |
|---|---|---|---|
| Atomic | $0.6716 \pm 0.0035$ | — | 0.2046 |
| Oracle SID | $0.6743 \pm 0.0122$ | $+0.0027$ | 0.2084 |
| Predictive | $0.7046 \pm 0.0062$ | $+0.0330$ | 0.1940 |
| Popularity | $0.8099 \pm 0.0010$ | $+0.1384$ | 0.1469 |
| Random | $0.8906 \pm 0.0002$ | $+0.2190$ | 0.1103 |

# E Amazon Protocol and Complete Results

## E.1 Data preparation, mappings, and training

We apply iterative 5-core filtering separately to Beauty, Sports, and Toys. The threshold counts interaction records, including repeated user–item pairs, and is applied before splitting. Timestamp ties preserve input order. The last two interactions per user form validation and test examples. Each training prefix of length at least two supplies a history–target pair, with history truncated to the most recent 20 items. The catalog contains items from training prefixes. Validation and test examples with unknown targets or unknown history items are excluded.

Table 16: Retained Amazon catalogs and history–target pairs.

| Dataset | Items | Train | Validation | Test |
|---|---|---|---|---|
| Beauty | 12068 | 131413 | 22312 | 22176 |
| Sports | 18284 | 189543 | 35480 | 35166 |
| Toys | 11876 | 109361 | 19312 | 19048 |

**Mapping provenance.** The seven supplied mappings are RecDMON, RQKMEANS, RQ-GAE, RQVAE, LETTER, S2GR, and MMGRec. These mappings were provided as fixed item-to-code artifacts. We retain their supplied labels and describe the associated papers' identifier constructions below. Our experiments compare the provided mappings with a common SASRec predictor, rather than the source papers' complete recommendation systems.

**RecDMON.** The RecDMoN construction from GrIS (Medvedev et al., 2026) recursively partitions an item interaction graph using differentiable clustering with a modularity objective. Each item's sequence of cluster assignments forms its code.

**RQKMEANS.** Applies residual $k$-means quantization to item content embeddings. Successive codebooks approximate the residual left by earlier stages; the selected centroid indices form the code.

**RQ-GAE.** The RQ-GAE construction from GrIS (Medvedev et al., 2026) combines item content embeddings with information propagated through an interaction graph, then learns a residual-quantization autoencoder with embedding- and graph-reconstruction objectives. Its sequence of quantizer indices forms the code.

**RQVAE.** Learns a residual-quantization autoencoder to reconstruct item content embeddings (Rajput et al., 2023). It encodes each item into a latent vector and approximates that vector through successive codebooks, whose selected indices form the code.

**LETTER.** Wang et al. (2024) learn residual-quantized item codes with semantic reconstruction, alignment to pretrained collaborative embeddings, and codebook-diversity regularization. The successive quantizer indices form the item identifier.

**S2GR.** The CoBa RQ-VAE tokenizer in $S^2$GR (Guo et al., 2026) propagates item features over a co-occurrence graph, then applies residual quantization with codebook uniformity and adaptive load balancing. The resulting codeword sequence is its SID. This tokenizer is distinct from the paper's subsequent latent-reasoning decoder, which is outside this mapping comparison.

**MMGRec.** Liu et al. (2026) construct Rec-IDs by quantizing graph representations that combine item features and interactions, with a popularity-rank suffix to distinguish collisions. Their Graph RQ-VAE combines multimodal and collaborative information; the Rec-ID consists of the selected semantic codewords followed by that suffix.

RecDMON, RQKMEANS, RQVAE, LETTER, S2GR, and MMGRec use five-token codes. RQ-GAE uses four-token codes in Beauty and Sports and five-token codes in Toys. For each dataset, all seven mappings use the same atomic reference checkpoint and the same 8,192 validation histories, selected without replacement using seed 2001. Diagnostics weight these histories equally. Each parent's spectrum uses its actual children and is weighted by its reference probability of occurrence. The diagnostic remains fixed across comparison training seeds 1001, 1002, and 1003.

Training uses batch size 64, learning rate $10^{-3}$, no weight decay, warmup fraction $0.05$, gradient clipping at 1, and at most 30 epochs. The Transformer has two layers, two attention heads, embedding width 64, feed-forward width 256, fixed sinusoidal positions, embedding and attention dropout $0.5$, and residual dropout zero. The output head is untied from the input embeddings. Validation item NLL selects the checkpoint; training stops after five epochs without improvement.

For SID targets, the output is a shared affine rank-$r$ head. At each prefix, logits outside its actual child set are masked before normalization. The rank restriction continues to hold after restricting and centering those logits, as explained in Appendix D. Target NLL sums exact conditional negative log-probabilities. Retrieval uses beam width 50 with no sampled negatives; previously observed items remain eligible. Different code vocabularies allocate different numbers of parameters, even at the same head rank.

### E.2 Diagnostic construction and rank sensitivity

The reference has no imposed low-rank output-head factorization. For each history $h$, parent $p$, and child $a$, its induced token law is

$$\widehat{q}^{\,\mathrm{ref}}_{i,p}(a \mid h) = \frac{\sum_{y:\, p_i(y)=(p,a)} \widehat{p}_{\mathrm{ref}}(y \mid h)}{\sum_{y:\, p_{i-1}(y)=p} \widehat{p}_{\mathrm{ref}}(y \mid h)}. \tag{61}$$

These conditional laws determine the gains and parentwise spectra under the reference joint law in Equation (57).

We evaluate diagnostic ranks $k \in \{1, 2, 4, 8, 16, 32\}$ against every trained head rank $r \in \{2, 4, 8\}$, computing each tail from the same saved spectra. At $k = 16$, the arithmetic mean of the nine within-setting Spearman correlations is $0.706$, compared with $0.567$ at $k = 8$ and $0.476$ when $k = r$. The

value $k = 16$ maximizes this average within the sweep. At $k = 16$, correlations with individual-initialization NLL also range from $0.321$ to $0.964$.

GWT at $k = 16$ agrees with 147 of the 189 pairwise mean-NLL comparisons: 21 mapping pairs in each of nine dataset–head-rank settings. On Beauty at head rank 2, RQ-GAE's mean NLL is $9.025$ versus $9.194$ for RecDMON, and RQ-GAE has lower NLL in all three initializations. All tested diagnostic ranks prefer RecDMON. The unweighted comparator is $m^{-1} \sum_i \tau_i(k)$; its mean correlation at $k = 16$ is $0.619$. With seven candidates it no longer induces the same orderings as GWT. On Toys, GWT orders the three added mappings as MMGRec, S2GR, LETTER, whereas mean NLL at head ranks 4 and 8 orders them as LETTER, S2GR, MMGRec. Selecting the best candidate thus does not imply correctly ordering all mappings.

### E.3 Count and coherence metrics correlations

For a mapping of depth $m$, each count statistic is evaluated at every applicable level and then averaged arithmetically across levels before correlating mappings. Token usage and prefix-sharing metrics use lengths $1, \ldots, m$, including the final collision-resolving digit. Decoding metrics use the $m$ prediction steps, whose parent prefixes have lengths $0, \ldots, m-1$; the first step includes one empty-prefix state. Levels receive equal weight even when their numbers of states differ.

Token counts give every catalog item weight one. At level $i$, the allocated vocabulary size $K_i$ is the manifest size (one plus the largest retained token value), including inactive token indices. Observed-token count is the number of positive counts. Token-count Gini uses all $K_i$ counts, including zeros, with the population convention. If $f_{ia}$ is the fraction of items using token $a$, token entropy is $H_i = -\sum_a f_{ia} \log f_{ia}$, normalized by $\log K_i$ (zero when $K_i = 1$). Effective vocabulary size is $\exp(H_i)$, calculated before averaging levels; maximum token share is $\max_a f_{ia}$.

Prefix metrics count observed groups of items sharing a prefix. We record the number of groups, their mean, median, maximum, and population Gini, the fraction of items in nonsingleton groups, and the fraction of distinct item pairs sharing a group. Group-size summaries weight observed groups equally. Decoding-state count is the number of observed parent prefixes. Valid-next-token summaries count distinct children per parent, with parents weighted equally; the valid-token fraction divides the mean child count by $K_i$ at that prediction step.

Frequency–prefix-size correlation is the within-level Pearson correlation across items between log training-interaction count and log prefix-group size. Each original training interaction is counted once, without duplicating events across overlapping training histories or using held-out events. Levels with constant group size have undefined correlation and are excluded from this statistic's level mean: all 21 final levels and the first level of Toys RQVAE. Other count metrics include these levels.

Each recomputed correlation uses seven mapping scores and seven mean NLLs within a dataset and head rank; each NLL averages seeds 1001–1003. The reported value is the arithmetic mean of the Beauty, Sports, and Toys correlations, not a correlation pooled across datasets. Thus there are 21 mapping–NLL pairs in three separate panels for each head rank. Table 17 gives all recomputed count metrics and the retained coherence entries.

For the semantic and collaborative metrics, we use *Qwen3-Embedding-0.6B* text embeddings and item representations from a pretrained SASRec model, respectively. At prefix length $l$, we sample up to 1,000 intra-prefix item pairs and an equal number of inter-prefix pairs formed by taking an anchor item from an intra pair and another item from a different prefix group. Let $c_{\text{intra}}$ and $c_{\text{inter}}$ denote the mean cosine similarities of the intra- and inter-prefix pairs in a given embedding space. Semantic intra-prefix cosine is $c_{\text{intra}}$ measured on semantic vectors. Delta cosine is $c_{\text{intra}} - c_{\text{inter}}$, reported separately for semantic and collaborative representations. Semantic separation ratio is the corresponding ratio of cosine distances, $(1 - c_{\text{intra}})/(1 - c_{\text{inter}})$, on semantic embeddings. For each metric, we report the prefix length $l$ that yields the highest correlation with the target metric. Note that these metrics become uninformative or undefined at deeper prefix lengths as prefix groups collapse into singletons, which prevents the formation of intra-prefix pairs and restricts sampling to lingering collision clusters.

Table 17: Complete diagnostic comparison: signed correlations averaged across datasets at each head rank. Count statistics are averaged across levels before computing correlations.

| Diagnostic / head rank $r$ | 2 | 4 | 8 |
|---|---|---|---|
| GWT ($k = r$) | 0.440 | 0.488 | 0.500 |
| GWT ($k = 16$) | 0.762 | 0.655 | 0.702 |
| Mean tail ($k = r$) | 0.321 | 0.226 | 0.381 |
| Mean tail ($k = 16$) | 0.702 | 0.595 | 0.560 |
| Observed-token count | −0.429 | −0.464 | −0.440 |
| Token-count Gini | 0.238 | 0.369 | 0.464 |
| Normalized token-count entropy | −0.429 | −0.476 | −0.452 |
| Effective vocabulary size | −0.333 | −0.440 | −0.512 |
| Maximum token share | 0.286 | 0.321 | 0.345 |
| Distinct-prefix count | 0.381 | 0.464 | 0.464 |
| Mean prefix-group size | −0.048 | −0.238 | −0.345 |
| Median prefix-group size | 0.107 | −0.095 | −0.214 |
| Maximum prefix-group size | 0.000 | −0.262 | −0.310 |
| Prefix-group Gini | 0.345 | 0.202 | 0.202 |
| Shared-prefix item fraction | −0.345 | −0.417 | −0.369 |
| Pair prefix-sharing rate | 0.060 | −0.214 | −0.298 |
| Decoding-state count | 0.476 | 0.536 | 0.548 |
| Mean valid-next-token count | 0.000 | 0.202 | 0.298 |
| Median valid-next-token count | −0.012 | 0.190 | 0.286 |
| Maximum valid-next-token count | 0.000 | 0.083 | 0.131 |
| Valid-next-token fraction | −0.619 | −0.512 | −0.464 |
| Frequency–prefix-size correlation | −0.310 | −0.131 | −0.095 |
| Semantic intra-prefix cosine ($L = 3$)$^\dagger$ | 0.443 | 0.305 | 0.367 |
| Semantic intra-prefix cosine distance ($L = 3$)$^\dagger$ | −0.443 | −0.305 | −0.367 |
| Semantic inter-prefix cosine ($L = 2$)$^\dagger$ | 0.314 | 0.352 | 0.333 |
| Semantic inter-prefix cosine distance ($L = 2$)$^\dagger$ | −0.314 | −0.352 | −0.333 |
| Semantic delta cosine ($L = 3$)$^\dagger$ | 0.424 | 0.305 | 0.367 |
| Semantic separation ratio ($L = 3$)$^\dagger$ | −0.443 | −0.305 | −0.367 |
| Collaborative delta cosine ($L = 2$)$^\dagger$ | −0.100 | 0.119 | 0.267 |

### E.4 RESULTS ACROSS TRAINING SEEDS

Table 18 gives a compact comparison of mapping scores and mean NLLs across trained head ranks. Figure 7 plots the same comparison, with error bars showing variation across training initializations.

Table 18: GWT at diagnostic rank $k = 16$ and mean test NLL at trained head ranks $r = 2, 4, 8$. GWT is dimensionless; NLL is in nats, averaged over three initializations. The choice of $k$ follows the sweep in Appendix E.2. Table 19 reports variability and Recall@10.

| Dataset | Mapping | GWT ($k = 16$) | NLL ($r = 2$) | NLL ($r = 4$) | NLL ($r = 8$) |
|---|---|---|---|---|---|
| Beauty | RecDMON | 0.00004 | 9.194 | 8.865 | 8.743 |
| | RQKMEANS | 0.00949 | 9.457 | 9.391 | 9.249 |
| | RQ-GAE | 0.00616 | 9.025 | 8.934 | 8.804 |
| | RQVAE | 0.00752 | 9.669 | 9.409 | 9.259 |
| | LETTER | 0.01280 | 9.568 | 9.529 | 9.451 |
| | S2GR | 0.00709 | 9.549 | 9.467 | 9.263 |
| | MMGRec | 0.00578 | 9.320 | 9.188 | 9.087 |
| Sports | RecDMON | 0.00012 | 9.428 | 9.302 | 9.195 |
| | RQKMEANS | 0.00258 | 9.910 | 9.770 | 9.545 |
| | RQ-GAE | 0.00175 | 9.582 | 9.372 | 9.282 |
| | RQVAE | 0.00346 | 9.913 | 9.660 | 9.524 |
| | LETTER | 0.00659 | 9.941 | 9.910 | 9.882 |
| | S2GR | 0.00241 | 9.940 | 9.783 | 9.491 |
| | MMGRec | 0.00125 | 9.757 | 9.707 | 9.535 |
| Toys | RecDMON | 0.00008 | 9.260 | 9.189 | 8.924 |
| | RQKMEANS | 0.00901 | 9.622 | 9.557 | 9.377 |
| | RQ-GAE | 0.00550 | 9.282 | 9.307 | 9.107 |
| | RQVAE | 0.00950 | 9.624 | 9.430 | 9.369 |
| | LETTER | 0.00797 | 9.487 | 9.417 | 9.188 |
| | S2GR | 0.00748 | 9.647 | 9.455 | 9.238 |
| | MMGRec | 0.00587 | 9.602 | 9.566 | 9.411 |

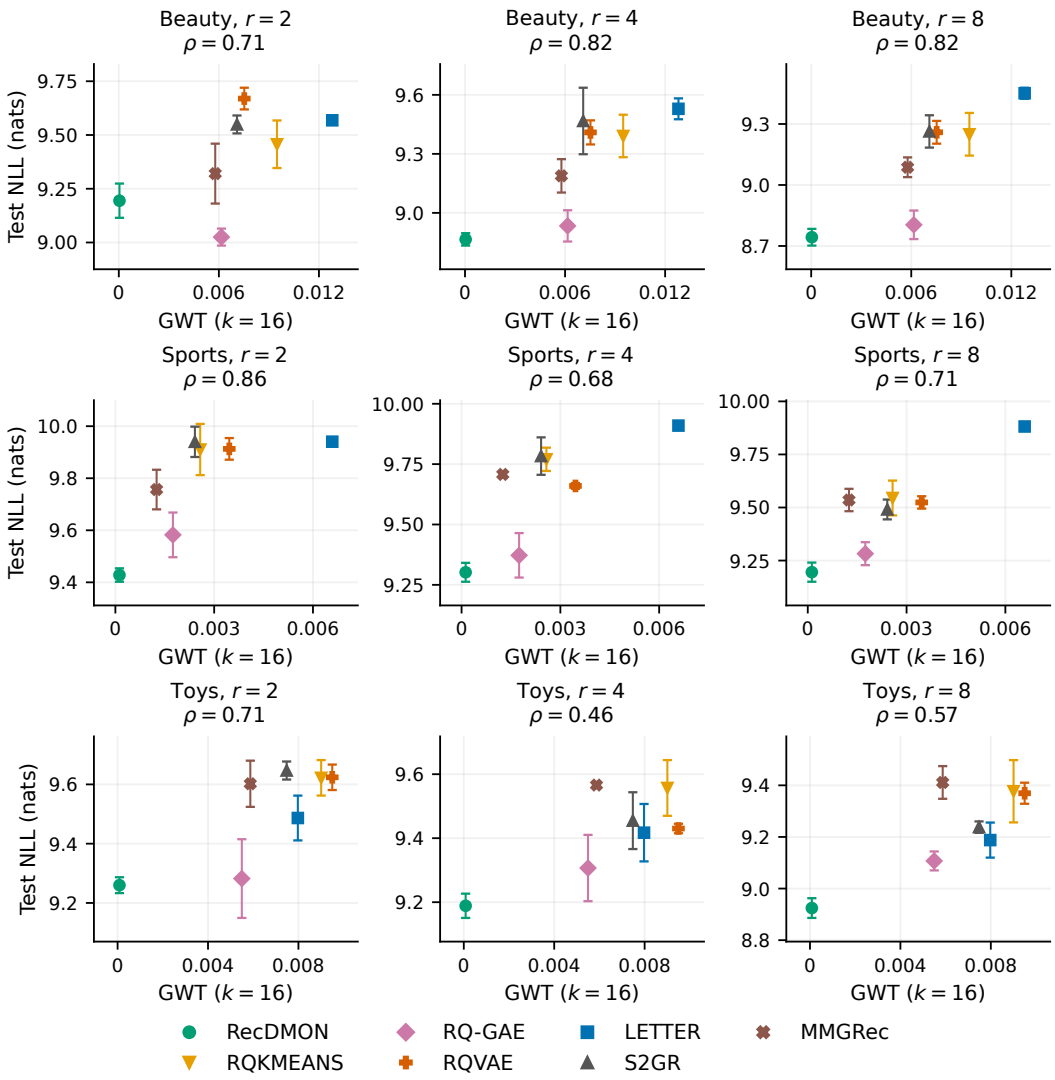


Figure 7: GWT and trained prediction error for the seven provided Amazon mappings. Rows correspond to datasets and columns to trained head ranks $r = 2, 4, 8$. GWT uses the fixed diagnostic rank $k = 16$ in every panel, as in Table 18. Points and error bars show mean test NLL and sample standard deviation across three training initializations. Panel annotations give Spearman correlations across the seven mapping means.

Table 19 summarizes all 189 completed fits as 63 dataset–mapping–rank combinations. NLL and Recall@10 entries are means and sample standard deviations over three initialization seeds. GWT uses diagnostic cutoff $k = 16$, independently of trained head rank $r$, and is fixed across training seeds. Main-text Spearman correlations compare the seven mappings' GWT values with their mean NLL within each dataset and head rank, then average the three dataset correlations.

Table 19: Seven provided Amazon mappings: GWT at $k = 16$ and test performance. NLL is in nats; Recall@10 is a percentage. Entries show mean $\pm$ sample standard deviation over three seeds.

| Dataset | Mapping | $r$ | GWT ($k = 16$) | Test NLL | Recall@10 (%) |
|---|---|---|---|---|---|
| Beauty | RecDMON | 2 | 0.00004 | $9.194 \pm 0.080$ | $2.67 \pm 0.48$ |
| Beauty | RQKMEANS | 2 | 0.00949 | $9.457 \pm 0.111$ | $1.39 \pm 0.32$ |
| Beauty | RQ-GAE | 2 | 0.00616 | $9.025 \pm 0.039$ | $2.68 \pm 0.17$ |
| Beauty | RQVAE | 2 | 0.00752 | $9.669 \pm 0.050$ | $1.32 \pm 0.49$ |
| Beauty | LETTER | 2 | 0.01280 | $9.568 \pm 0.017$ | $1.16 \pm 0.11$ |
| Beauty | S2GR | 2 | 0.00709 | $9.549 \pm 0.042$ | $1.76 \pm 0.30$ |
| Beauty | MMGRec | 2 | 0.00578 | $9.320 \pm 0.139$ | $1.06 \pm 0.22$ |
| Beauty | RecDMON | 4 | 0.00004 | $8.865 \pm 0.031$ | $3.72 \pm 0.09$ |
| Beauty | RQKMEANS | 4 | 0.00949 | $9.391 \pm 0.108$ | $2.34 \pm 0.15$ |
| Beauty | RQ-GAE | 4 | 0.00616 | $8.934 \pm 0.080$ | $3.51 \pm 0.08$ |
| Beauty | RQVAE | 4 | 0.00752 | $9.409 \pm 0.061$ | $2.22 \pm 0.19$ |
| Beauty | LETTER | 4 | 0.01280 | $9.529 \pm 0.053$ | $1.38 \pm 0.20$ |
| Beauty | S2GR | 4 | 0.00709 | $9.467 \pm 0.169$ | $2.13 \pm 0.10$ |
| Beauty | MMGRec | 4 | 0.00578 | $9.188 \pm 0.085$ | $1.82 \pm 0.16$ |
| Beauty | RecDMON | 8 | 0.00004 | $8.743 \pm 0.041$ | $3.81 \pm 0.21$ |
| Beauty | RQKMEANS | 8 | 0.00949 | $9.249 \pm 0.105$ | $2.86 \pm 0.04$ |
| Beauty | RQ-GAE | 8 | 0.00616 | $8.804 \pm 0.070$ | $3.78 \pm 0.49$ |
| Beauty | RQVAE | 8 | 0.00752 | $9.259 \pm 0.056$ | $2.87 \pm 0.09$ |
| Beauty | LETTER | 8 | 0.01280 | $9.451 \pm 0.027$ | $2.18 \pm 0.15$ |
| Beauty | S2GR | 8 | 0.00709 | $9.263 \pm 0.080$ | $2.83 \pm 0.16$ |
| Beauty | MMGRec | 8 | 0.00578 | $9.087 \pm 0.049$ | $2.04 \pm 0.34$ |
| Sports | RecDMON | 2 | 0.00012 | $9.428 \pm 0.026$ | $1.77 \pm 0.04$ |
| Sports | RQKMEANS | 2 | 0.00258 | $9.910 \pm 0.098$ | $0.93 \pm 0.05$ |
| Sports | RQ-GAE | 2 | 0.00175 | $9.582 \pm 0.086$ | $1.76 \pm 0.02$ |
| Sports | RQVAE | 2 | 0.00346 | $9.913 \pm 0.042$ | $1.01 \pm 0.06$ |
| Sports | LETTER | 2 | 0.00659 | $9.941 \pm 0.009$ | $0.55 \pm 0.07$ |
| Sports | S2GR | 2 | 0.00241 | $9.940 \pm 0.058$ | $0.91 \pm 0.10$ |
| Sports | MMGRec | 2 | 0.00125 | $9.757 \pm 0.076$ | $0.64 \pm 0.08$ |
| Sports | RecDMON | 4 | 0.00012 | $9.302 \pm 0.039$ | $2.23 \pm 0.03$ |
| Sports | RQKMEANS | 4 | 0.00258 | $9.770 \pm 0.048$ | $1.50 \pm 0.16$ |
| Sports | RQ-GAE | 4 | 0.00175 | $9.372 \pm 0.092$ | $2.12 \pm 0.13$ |
| Sports | RQVAE | 4 | 0.00346 | $9.660 \pm 0.012$ | $1.70 \pm 0.03$ |
| Sports | LETTER | 4 | 0.00659 | $9.910 \pm 0.007$ | $0.76 \pm 0.05$ |
| Sports | S2GR | 4 | 0.00241 | $9.783 \pm 0.078$ | $1.35 \pm 0.07$ |
| Sports | MMGRec | 4 | 0.00125 | $9.707 \pm 0.014$ | $0.95 \pm 0.04$ |
| Sports | RecDMON | 8 | 0.00012 | $9.195 \pm 0.045$ | $2.29 \pm 0.11$ |
| Sports | RQKMEANS | 8 | 0.00258 | $9.545 \pm 0.082$ | $1.80 \pm 0.23$ |
| Sports | RQ-GAE | 8 | 0.00175 | $9.282 \pm 0.054$ | $2.32 \pm 0.06$ |
| Sports | RQVAE | 8 | 0.00346 | $9.524 \pm 0.029$ | $1.95 \pm 0.22$ |
| Sports | LETTER | 8 | 0.00659 | $9.882 \pm 0.012$ | $1.22 \pm 0.05$ |
| Sports | S2GR | 8 | 0.00241 | $9.491 \pm 0.047$ | $1.67 \pm 0.16$ |
| Sports | MMGRec | 8 | 0.00125 | $9.535 \pm 0.053$ | $1.53 \pm 0.16$ |

Continued on next page

Table 19 continued

| Dataset | Mapping | $r$ | GWT ($k = 16$) | Test NLL | Recall@10 (%) |
|---|---|---|---|---|---|
| Toys | RecDMON | 2 | 0.00008 | $9.260 \pm 0.027$ | $1.17 \pm 0.18$ |
| Toys | RQKMEANS | 2 | 0.00901 | $9.622 \pm 0.060$ | $0.76 \pm 0.05$ |
| Toys | RQ-GAE | 2 | 0.00550 | $9.282 \pm 0.133$ | $0.83 \pm 0.26$ |
| Toys | RQVAE | 2 | 0.00950 | $9.624 \pm 0.043$ | $0.36 \pm 0.16$ |
| Toys | LETTER | 2 | 0.00797 | $9.487 \pm 0.076$ | $0.76 \pm 0.15$ |
| Toys | S2GR | 2 | 0.00748 | $9.647 \pm 0.030$ | $0.69 \pm 0.06$ |
| Toys | MMGRec | 2 | 0.00587 | $9.602 \pm 0.078$ | $0.51 \pm 0.08$ |
| Toys | RecDMON | 4 | 0.00008 | $9.189 \pm 0.038$ | $2.59 \pm 0.32$ |
| Toys | RQKMEANS | 4 | 0.00901 | $9.557 \pm 0.087$ | $1.24 \pm 0.21$ |
| Toys | RQ-GAE | 4 | 0.00550 | $9.307 \pm 0.104$ | $1.01 \pm 0.22$ |
| Toys | RQVAE | 4 | 0.00950 | $9.430 \pm 0.015$ | $0.59 \pm 0.11$ |
| Toys | LETTER | 4 | 0.00797 | $9.417 \pm 0.090$ | $1.22 \pm 0.20$ |
| Toys | S2GR | 4 | 0.00748 | $9.455 \pm 0.089$ | $1.17 \pm 0.16$ |
| Toys | MMGRec | 4 | 0.00587 | $9.566 \pm 0.002$ | $0.42 \pm 0.24$ |
| Toys | RecDMON | 8 | 0.00008 | $8.924 \pm 0.038$ | $2.74 \pm 0.21$ |
| Toys | RQKMEANS | 8 | 0.00901 | $9.377 \pm 0.121$ | $1.88 \pm 0.38$ |
| Toys | RQ-GAE | 8 | 0.00550 | $9.107 \pm 0.037$ | $1.58 \pm 0.14$ |
| Toys | RQVAE | 8 | 0.00950 | $9.369 \pm 0.041$ | $1.18 \pm 0.13$ |
| Toys | LETTER | 8 | 0.00797 | $9.188 \pm 0.068$ | $1.95 \pm 0.16$ |
| Toys | S2GR | 8 | 0.00748 | $9.238 \pm 0.022$ | $2.10 \pm 0.17$ |
| Toys | MMGRec | 8 | 0.00587 | $9.411 \pm 0.063$ | $1.00 \pm 0.16$ |