SAME RANKINGS, DIFFERENT SEMANTIC IDS: OPTIMIZATION NON-INVARIANCE IN TOKENIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic-ID-based generative recommendation relies on tokenizers such as RQ-VAE to convert continuous item representations into discrete codes. Although scale and distribution are known to affect codebook collapse, their effect on the full recommendation pipeline is harder to isolate when the upstream representation also changes. We study input rescaling that preserves every pairwise recommendation-score difference. In our main study with 512-dimensional SASRec embeddings, the same training procedure collapses to a single base SID on native inputs, whereas rescaling produces broad codebook use and substantially better downstream recommendation. An exact tokenizer reparameterization shows that the same non-collapsed code assignments remain representable at native scale, separating representational capacity from training outcome. At these mapped states, reconstruction loss and its encoder-output gradient scale quadratically, whereas latent quantization terms remain unchanged. This analysis yields a simple reconstruction-loss calibration that keeps the native embeddings fixed. Controlled experiments from matched starting points show that the calibration changes encoder optimization and recovers broad code use. Across three Amazon datasets, calibration improves Hit@10 and NDCG@10 in all nine paired runs, reaching accuracy close to direct input rescaling. These findings show why representation comparisons should account for numerical scale and tokenizer optimization.

1 INTRODUCTION

Semantic-ID-based generative recommendation represents items as discrete token sequences that can be predicted autoregressively. A tokenizer converts continuous item representations into Semantic IDs (SIDs), which connect the representations to a downstream generator (Rajput et al., 2023; Singh et al., 2023; Wang et al., 2024; Zheng et al., 2024). Collaborative embeddings learned by pretrained recommenders already serve as direct tokenizer inputs: ETEGRec uses SASRec item embeddings for tokenization (Liu et al., 2025), and GenSAR quantizes collaborative embeddings to construct recommendation identifiers (Shi et al., 2025). These exported vectors connect recommendation learning to tokenizer training.

An established challenge is codebook collapse: a tokenizer uses only a small fraction of its available codes, limiting the item distinctions captured by the quantization tokens. Prior work has already linked this problem to numerical properties and optimization. In vector quantization, Lancucki et al. (2020) study the relative scale of encoder outputs and codewords and their initialization, while Huh et al. (2023) analyze distribution mismatch, sparse codebook gradients, and commitment-loss optimization. In Semantic-ID systems, GRID whitens inputs to the residual vector-quantized autoencoder (RQ-VAE) to mitigate collapse (Ju et al., 2025); work at Snapchat links low-variance inputs to collapse and studies multi-source fusion (Ju et al., 2026); CARD learns an invertible transformation to balance the quantization space (Wei et al., 2026).

In recommendation pipelines, changing the embedding source can change both the encoded recommendation information and its numerical scale, making their downstream effects difficult to disentangle. Performance differences may reflect how well the tokenizer trains at a given scale, as well as representation quality. We therefore ask: *how does input scale affect tokenizer training and downstream recommendation when upstream recommendation preferences are held fixed?* We isolate

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.

this effect by preserving the pretrained recommender's pairwise score differences, relating tokenizer solutions across scales, and testing a loss adjustment with the original embeddings held fixed.

Our starting observation is that identical upstream preferences can yield sharply different IDs and downstream accuracy. For a pretrained dot-product recommender (the *teacher*), rescaling item embeddings around their mean while inversely rescaling history vectors preserves every pairwise recommendation-score difference, and hence the teacher's rankings. We call two representations *recommendation-equivalent* when their item vectors and corresponding history vectors give identical pairwise score differences for every history. In our main study with 512-dimensional SASRec embeddings (Kang & McAuley, 2018), the original, unrescaled embeddings (*Native*) yield one base SID (the quantization tokens before suffixing) under our RQ-VAE training procedure. Training with the same procedure on rescaled inputs instead uses many entries in each codebook and yields higher downstream accuracy. Figure 1a shows this contrast: unchanged teacher rankings, but different code use, base SIDs, and downstream Hit@10.

To understand this contrast, we first separate representational capacity from the outcome of training (Figure 1b). Reparameterizing the first encoder and last decoder layers preserves latents, assignments, and base SIDs across scales. The native-input tokenizer can therefore produce the same non-collapsed IDs even when training fails to learn them.

The map also reveals a change in the training objective. At mapped states, reconstruction loss and its encoder-output gradient scale quadratically, while latent quantization terms remain unchanged. This relation gives a rule for reweighting reconstruction loss to match a chosen reference scale. We call this reweighting *objective calibration*; it changes only the tokenizer objective and keeps the native embeddings fixed. It matches objective values and latent gradients at mapped states; subsequent optimizer trajectories can still differ. We test it in two complementary experiments. *Mapped continuation* resumes training from the reparameterized checkpoint to test updates and code use (Figure 2). *From-scratch training* compares raw and calibrated objectives from the same native-input K-means initialization (Figure 1c). Calibration improves Hit@10 and NDCG@10 in all nine paired runs across three datasets (Table 1).

Our contributions are:

- **A controlled counterexample:** recommendation-equivalent inputs yield different IDs and downstream accuracy under the same tokenizer training procedure (Section 3).
- **An exact reparameterization and optimization analysis:** the native-input tokenizer can produce the same non-collapsed code assignments; scaling laws for reconstruction and latent quantization yield an analytical reconstruction-loss weighting rule (Section 4).
- **A tokenizer-side intervention with fixed embeddings:** the derived reconstruction weighting changes encoder updates in mapped continuation. From shared native initialization, it restores broad code use and improves Hit@10 and NDCG@10 in all nine paired runs (Section 5).
- **Scope across representations and quantizers:** additional representations, catalog-wide normalization, and fixed-grid quantization show sensitivity beyond the main collapse case, while broader code use does not consistently improve recommendation (Section 6).

2 The Representation–Tokenizer Interface

2.1 One Recommendation Function, Multiple Representations

A frozen dot-product teacher scores item i after a history encoded as h by $a(h, i) = h^\top x_i$, where $x_i \in \mathbb{R}^{d_x}$ and $X = [x_1; \ldots; x_N]$ is the catalog.

Two sets of item vectors and their corresponding history vectors are *recommendation-equivalent* when all pairwise item-score differences agree for every history. They then have identical rankings and softmax probabilities at a fixed temperature (Appendix A.1). For catalog mean μ and $c > 0$, centered scaling gives

$$x_i^{(c)} = \mu + c(x_i - \mu), \qquad h^{(c)} = h/c. \tag{1}$$

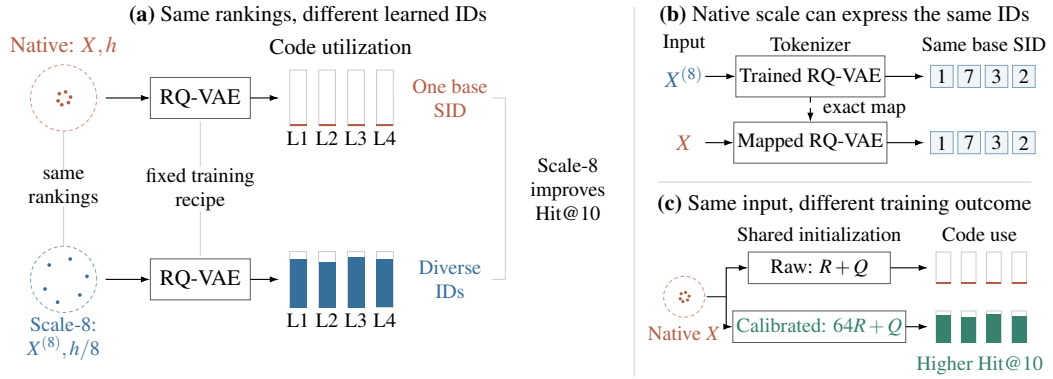

Figure 1: **Same rankings, different tokenization outcomes: the SASRec-512 study.** (a) Scale-8 multiplies centered item embeddings by eight and divides history vectors by eight, preserving pairwise teacher-score differences. The same RQ-VAE procedure yields different IDs and downstream accuracy. (b) An exact map preserves each item's latents and base SID (illustrative tokens shown). At mapped states, reconstruction loss satisfies $R_1 = R_8/64$, while latent quantization loss satisfies $Q_1 = Q_8$; subscripts denote input scale. This motivates the calibration in (c). (c) From the same native K-means initialization, $64R + Q$ restores code use and improves Hit@10 over $R + Q$, with native embeddings fixed. Bars show final code use schematically; clouds are also schematic. Absolute Hit@10 gains over Native raw are 0.0296 for rescaling and 0.0294 for calibration, averaged over three seeds and three datasets (Table 1).

The transformed score satisfies

$$a_c(h,i) = a(h,i) + (c^{-1} - 1)h^\top \mu, \qquad a_c(h,i) - a_c(h,j) = a(h,i) - a(h,j). \tag{2}$$

The item-independent offset establishes recommendation equivalence. The downstream generator consumes item IDs, not the transformed history vectors. Centered scaling preserves item information and centered cosine geometry, while multiplying pairwise distances by c.

Code utilization, base-SID collisions, and downstream retrieval distinguish changes in learned partitions from a mere renaming of code indices.

2.2 RQ-VAE Objective

The encoder produces $z_i = E_\theta(x_i)$. Each of L quantization levels selects a codebook vector (a *codeword*); its index becomes a token. These tokens form the *base SID*, with a suffix distinguishing shared base IDs. The selected codewords sum to q_i for the decoder. We use

$$R = \frac{1}{Bd_x}\sum_{i=1}^{B}\|D_\psi(q_i) - x_i\|_2^2, \qquad Q = \frac{1}{L}\sum_{\ell=1}^{L}(C_\ell + 0.25M_\ell), \qquad \mathcal{L} = f_R R + Q. \tag{3}$$

Here B is minibatch size, d_x is input dimension, and C_ℓ, M_ℓ are codebook and commitment MSEs averaged over items and latent coordinates. Raw training uses $f_R = 1$; calibration changes only f_R.

The implemented residual straight-through estimator (STE) sends reconstruction gradients to the encoder through $\partial_{\mathrm{STE}} q_i/\partial z_i = I$, with no direct path to codebooks. Only the first-level commitment term contributes directly to the encoder gradient; each codebook receives its own C_ℓ gradient. Appendix A.2 specifies the complete forward and backward graph. Training uses greedy assignments at levels 1–3 and Sinkhorn assignment within each minibatch at level 4; the main study evaluates and generates IDs using greedy assignments at every level (Appendix B.3).

Tokenizer diagnostics. MinPPL is the minimum normalized codebook perplexity across levels; NMSE is reconstruction MSE normalized by centered input variance. For code frequencies $p_{\ell k}$ and codebook size K_ℓ, we use

$$\mathrm{MinPPL} = \min_\ell \frac{\exp(-\sum_k p_{\ell k}\log p_{\ell k})}{K_\ell}, \qquad \mathrm{NMSE} = \frac{\sum_i \|x_i - D_\psi(q_i)\|_2^2}{\sum_i \|x_i - \mu\|_2^2}. \tag{4}$$

A one-code level has normalized perplexity $1/K_\ell$; values near one indicate broad, relatively balanced use. Base collision is $1 - U/N$, where U counts unique base SIDs. The suffix-required fraction counts items in shared base-SID groups: all IDs include a suffix, but only these items need it for disambiguation. We evaluate recommendation with Hit@10 and NDCG@10.

3 A CONTROLLED COUNTEREXAMPLE

We use a reference scale $c = 8$, selected using tokenizer-only diagnostics on Beauty and Instruments before downstream evaluation and transfer to Games. *Scale-8* multiplies deviations from the catalog mean by eight; the corresponding history vectors are divided by eight (Equation 1). Our main experiment compares three conditions on the same frozen SASRec-512 catalog: **Native raw** uses the native input with $R + Q$; **Scale-8 raw** applies Equation 1 with $c = 8$ and uses $R + Q$; **Native calibrated** keeps the native input and uses $64R + Q$. We first compare the two raw conditions; Section 4 derives the coefficient 64, and Section 5 examines its effect.

Data and models. We use 5-core Amazon Beauty, Instruments, and Games with chronological leave-two-out splitting and full-catalog evaluation. The SASRec-512 teacher (Kang & McAuley, 2018) is selected through validation and frozen. RQ-VAE uses four 256-entry codebooks, a 32-dimensional latent, and K-means initialization; a TIGER-style generator predicts its IDs autoregressively (Rajput et al., 2023). The tokenizer uses AdamW with learning rate 10^{-3}, weight decay 10^{-4}, and batch size 1,024. Its 20,000 full-catalog epochs correspond to 220,000–340,000 optimizer steps across domains. The generator has four encoder and four decoder layers of width 128 and is randomly initialized. Appendix B gives architecture and optimizer settings; Appendix C.5 gives the motivating SASRec-32 comparison.

One protocol for all three conditions. Three domains and seeds 42–44 give 27 tokenizer–generator runs. Every tokenizer trains for 20,000 full-catalog epochs; we use the final checkpoint and greedy assignments at all four levels, plus a suffix. The two native-input arms share the completed K-means initialization, random state, and minibatch sequence, changing only the reconstruction coefficient. Scale-8 initializes on its transformed input. All conditions use the same tokenizer architecture and Transformer backbone, with matching generator seeds and validation-loss selection. The exported SID catalog determines the generator vocabulary and tied embedding size (Appendix B.3, Table 5); trie-constrained beam search (width 20) and full-catalog evaluation use the same test users and targets, without masking previously interacted items. Comparing Native raw with Scale-8 tests the training procedure as a whole, including initialization on different inputs. Comparing the two native-input conditions isolates the effect of the reconstruction coefficient.

Appendix C.2 reports the earlier rescaling study used to select the reference, including results excluding the pilot seed on Beauty and Instruments, under its alternative checkpoint-selection and ID-generation rules.

The teacher's score differences and rankings are preserved. Native and Scale-8 give identical top-10 results when tied scores are taken into account, and identical teacher Hit@10/NDCG@10. The maximum discrepancy in pairwise score differences is 5.25×10^{-7}, consistent with Equation 2 in finite precision.

The learned IDs change. All nine Native raw tokenizers map the entire catalog to one base SID, with one active code at each level and NMSE near one. Scale-8 uses all four levels broadly, with mean MinPPL .8709–.8784 and NMSE .2139–.3697 (Table 1). The inputs implement the same upstream preferences, but training retains very different catalog distinctions.

The difference reaches the final recommender. Scale-8 improves Hit@10 and NDCG@10 over Native raw in all nine paired runs (Table 1; paired results in Appendix C.1). Under Native raw, the suffix supplies all item discrimination; under Scale-8, the mean fraction requiring the suffix for disambiguation is 0.60–1.52% (Table 7).

A ranking-preserving change thus alters both the learned IDs and downstream accuracy. Can a tokenizer at native scale produce the same diverse base IDs?

Table 1: Main three-condition experiment: the same training and ID-generation rules connect tokenizer behavior to downstream recommendation. Native raw and Scale-8 raw use $R + Q$; Native calibrated uses $64R+Q$. All 27 tokenizers use the checkpoint at epoch 20,000 and greedy assignments at all levels, followed by a suffix. Values are mean $\pm$ sample standard deviation over seeds 42–44; Hit and NDCG are on a 0–1 scale. Complete level collapse gives MinPPL $1/256 \approx .0039$.

		Tokenizer		Recommendation	
Dataset	Condition	MinPPL ↑	NMSE ↓	Hit@10 ↑	NDCG@10 ↑
Beauty	Native raw	.0039 ± .0000	1.0007 ± .0001	.04515 ± .00251	.02302 ± .00150
	Scale-8 raw	.8709 ± .0904	.3697 ± .0797	.06629 ± .00212	.03512 ± .00107
	Native calibrated	.8019 ± .0980	.5288 ± .0373	.06663 ± .00079	.03584 ± .00058
Instruments	Native raw	.0039 ± .0000	1.0006 ± .0000	.08984 ± .00112	.06865 ± .00124
	Scale-8 raw	.8784 ± .0066	.2139 ± .0057	.11228 ± .00341	.08508 ± .00312
	Native calibrated	.8431 ± .0125	.3718 ± .0265	.11300 ± .00104	.08650 ± .00147
Games	Native raw	.0039 ± .0000	1.0007 ± .0000	.07669 ± .00514	.05004 ± .00341
	Scale-8 raw	.8726 ± .0083	.2876 ± .0083	.12182 ± .00124	.08170 ± .00080
	Native calibrated	.8195 ± .0092	.4357 ± .0102	.12020 ± .00024	.08135 ± .00035

4 The Same Code Assignments, a Different Objective

4.1 Native Inputs Can Represent the Same IDs

Consider a tokenizer trained at source scale c_s, with affine first encoder and last decoder operations $u = W_e x + b_e$ and $\widehat{x} = W_d v + b_d$. Let $\alpha = c_s/c_t$ and $\gamma = c_t/c_s$ for target scale c_t.

Proposition 1 (Exact tokenizer reparameterization)**.** *Keep every interior parameter and codebook fixed and change these two layers by*

$$\begin{aligned} W'_e &= \alpha W_e, \quad b'_e = b_e + (1-\alpha) W_e \mu, \\ W'_d &= \gamma W_d, \quad b'_d = \gamma b_d + (1-\gamma)\mu. \end{aligned} \tag{5}$$

On corresponding source and target inputs, the mapped tokenizer has identical encoder preactivations, latents, residual assignments, and base SIDs in exact arithmetic. Reconstruction residual vectors scale by γ, preserving centered NMSE.

The same architecture can therefore produce the same code assignments at native scale. Substituting $x^{(c_s)} = \alpha x^{(c_t)} + (1-\alpha)\mu$ gives the encoder identity; the decoder satisfies $\widehat{x}_t = \mu + \gamma(\widehat{x}_s - \mu)$ (full proof in Appendix A.3). Across three domains and both mapping directions, float64 preserves all assignments with maximum latent error 8.88×10^{-16}. In float32, two of 318,248 level assignments change at near ties, while aggregate utilization and collision remain unchanged (Appendix D).

The map also applies to the mixed assignment rule used during training. Corresponding minibatches have identical latent distance matrices, so both greedy selection and the level-4 Sinkhorn procedure receive the same inputs.

Capacity alone does not explain the failure. Native-input controls remain collapsed when codebook size varies from 64 to 512, or when the latent width increases from 32 to 64 or 128 (Appendix D.2). The exact map gives stronger evidence: the original architecture can already represent the non-collapsed IDs.

4.2 Calibrating Objective Values and Latent Gradients

Moving a solution to a smaller input scale shrinks reconstruction residuals but preserves latent quantization residuals. At the mapped states,

$$R_t = \gamma^2 R_s, \qquad Q_t = Q_s. \tag{6}$$

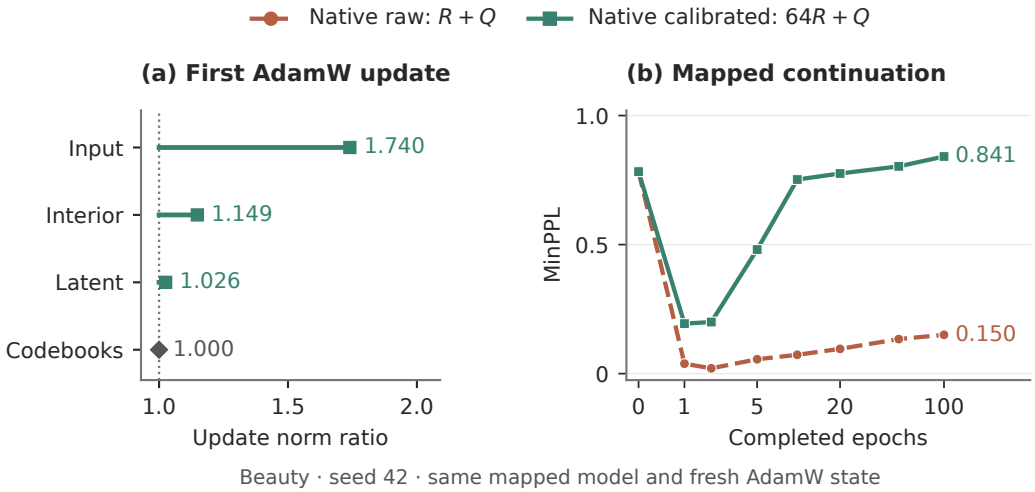

Figure 2: **Encoder updates change before codebook updates.** Both arms start from the same mapped tokenizer and fresh AdamW state (Beauty, seed 42). (a) Measured first-step update norm, calibrated divided by raw, for encoder groups and all codebooks. The dotted line marks a ratio of one; each codebook's first update is identical across arms. (b) Full-catalog greedy MinPPL at observed epochs. Horizontal position is $\log_{10}(\text{epoch} + 1)$; ticks show completed epochs. Lines connect observations without smoothing.

The same code assignments are evaluated with a different relative weight on reconstruction, which also changes the gradient reaching the encoder. Let $g^{(a)}_{R,z}$ and $g^{(a)}_{Q,z}$ be the reconstruction and quantization gradients at the continuous latent z, using the residual STE described in Section 2 (full graph in Appendix A.2). On corresponding minibatches at mapped states,

$$g^{(t)}_{R,z} = \gamma^2 g^{(s)}_{R,z}, \qquad g^{(t)}_{Q,z} = g^{(s)}_{Q,z}. \tag{7}$$

The residual and decoder output weights each supply one factor of γ; interior backward paths are unchanged. Matching the source balance at these states gives

$$f_t = \gamma^{-2} f_s = (c_s/c_t)^2 f_s. \tag{8}$$

Then $f_t g^{(t)}_{R,z} + g^{(t)}_{Q,z} = f_s g^{(s)}_{R,z} + g^{(s)}_{Q,z}$: calibration restores the source gradient balance as well as its objective value (Corollary 1, Appendix A.4). This is a local equality at mapped states; subsequent parameter updates also depend on the parameterization of the first encoder and last decoder layers, and on AdamW's moment estimates (Kingma & Ba, 2015; Loshchilov & Hutter, 2019).

For the pilot-selected reference $c_s = 8$ with $f_s = 1$, matching its balance at native scale $c_t = 1$ gives $f_t = 64$. This coefficient is conditional on the selected reference scale. The three-arm comparison in Table 1 tests its downstream effect; the controlled continuations in Section 5 trace how the reweighting changes encoder updates and code use.

5 How Objective Balance Changes Training

We test reconstruction weighting at two starting points: a reparameterized trained tokenizer and a shared native-input K-means initialization. Each native-input pair shares model weights, random state, and minibatch sequence, changing only the reconstruction weight. The first experiment measures updates and code use (Figure 2); the second measures recommendation (Figure 1c).

5.1 Continuing from the Same Tokenizer

We reparameterize the earlier study's non-collapsed Scale-8 checkpoints for native input (seed 42; Appendix C.2). These checkpoints were selected by minimum base-SID collision, measured using

greedy assignments. Two copies continue with $R + Q$ or $64R + Q$. Both reset AdamW's moment estimates, so inherited optimizer history cannot explain their differences.

Encoder updates change before codebook updates. On the first Beauty minibatch (seed 42), calibrated-to-raw AdamW update-norm ratios are 1.740, 1.149, and 1.026 for the input, interior, and latent-projection groups (Figure 2a). Every codebook's first update remains identical.

To explain this difference, write an encoder group's gradient as $f_R g_R + g_Q$. Raising f_R from 1 to 64 changes $\|f_R g_R\|/\|g_Q\|$ from 0.058–0.218 to 3.73–13.96 (Table 16). The cosine similarities between the reconstruction and quantization gradients are -0.332 and -0.086 at the input and interior groups, so reweighting also changes the total gradient's direction (Appendix D.4). Reconstruction has no direct codebook gradient path under the implemented STE; codebook gradients initially agree, then later updates separate as the encoder changes residuals and assignments (Appendix A.2).

Code use develops differently from the same start. After one epoch, raw Native drops from MinPPL 0.783 to 0.039, while calibrated Native retains 0.194. By epoch 100 the values are 0.150 and 0.841 (Figure 2b). A separate 2,000-epoch continuation on all three domains at seed 42 reaches mean MinPPL across datasets of 0.1942 for raw Native, 0.8921 for calibrated Native, and 0.8818 for the Scale-8 source reference (Appendix D.5). Every calibrated codebook uses all 256 entries; the least-used raw levels use 47–98. At their endpoints, raw base collision is 1.14–2.23%, so this reduced utilization is distinct from the complete from-scratch collapse below.

5.2 From Shared Initialization to Downstream Recommendation

Calibration recovers useful codes at native scale. With calibration, only 1.00–1.86% of items then share a base SID, compared with 100% under the raw objective (Appendix C.1). The Beauty seed-42 pair starts at MinPPL .9046, then both arms reach one code per level after the first epoch. Calibrated MinPPL subsequently rises to .2407 at epoch 100 and .6933 at epoch 20,000, while raw Native remains at $1/256$ (Table 19). Thus calibration enables recovery after the initial loss of code use, rather than simply preserving a favorable K-means partition. Figure 5 shows the recovery trajectories across all nine pairs.

Calibration improves Hit@10 and NDCG@10 in all nine paired runs. Mean Hit@10 gains are $.02148 \pm .00178$, $.02316 \pm .00190$, and $.04350 \pm .00535$ across the three domains (mean $\pm$ sample standard deviation of paired seed-wise differences). Averaging dataset means equally gives gains of .02938 and .02066, respectively; Appendix C.1 reports paired differences and their variability.

Utilization and recommendation measure different outcomes. Calibrated minus Scale-8 mean Hit@10 is $+.00034$, $+.00072$, and $-.00163$ across the three domains. Scale-8 achieves lower NMSE and higher MinPPL throughout, but does not consistently achieve higher recommendation accuracy. The generator backbone is shared across conditions, but the SID catalog determines its vocabulary and tied input/output embeddings. Under complete base-code collapse, all items share the same four-token prefix, leaving the suffix to distinguish items. With broad base codes, item distinctions enter earlier in the autoregressive sequence. The exported vocabulary shrinks from 42,543–49,360 tokens for raw Native to 33,152–33,229 for calibrated Native, so the calibrated generators have fewer total parameters despite their higher accuracy (Table 5). The downstream effect includes this change in prediction structure and embedding size; the experiment does not separately identify the contribution of semantic grouping.

Recovery extends to several reconstruction weights. A separate Beauty seed-42 experiment tests $f_R \in \{16, 32, 128, 256\}$ from a shared K-means initialization and minibatch sequence. Every level uses all 256 codes at the final checkpoint, with MinPPL .8408–.8857 (Appendix E.2). This tests tokenizer recovery; downstream evaluation uses 64. Because 64 used a separate initialization, these measurements do not rank it against neighboring weights.

5.3 Code Use and Endpoint Loss

Code utilization does not follow the ordering of endpoint scalar losses. Reevaluating the two continuation endpoints under $64R + Q$ gives a lower loss for the raw arm on all three domains,

Table 2: Downstream accuracy at target RMS .008 and .064 (seed 42; checkpoints selected by minimum base-SID collision). Δ is RMS .064 minus .008; all 45 native/intermediate/endpoint results appear in Appendix F.2.

Representation	Dataset	H@10 (.008)	H@10 (.064)	ΔH@10	ΔN@10
BERT-768	Beauty	0.0369	0.0519	+0.0150	+0.0100
BERT-768	Instruments	0.0972	0.1040	+0.0068	+0.0034
BERT-768	Games	0.0932	0.1026	+0.0095	+0.0041
SASRec-32	Beauty	0.0466	0.0608	+0.0142	+0.0066
SASRec-32	Instruments	0.1090	0.1141	+0.0051	+0.0037
SASRec-32	Games	0.1199	0.1181	-0.0018	-0.0012
MoRec-512	Beauty	0.0685	0.0672	-0.0013	-0.0005
MoRec-512	Instruments	0.1208	0.1229	+0.0021	+0.0007
MoRec-512	Games	0.1241	0.1247	+0.0006	+0.0002

despite its narrower code use (Appendix D.3). For example, the Beauty 2,000-epoch raw endpoint has loss 0.006353 under $64R + Q$ and MinPPL 0.1741; the calibrated endpoint has the higher loss 0.01331 under that same objective and MinPPL 0.9255. With stop-gradient and STE, a scalar loss does not fully describe the implemented updates, and assignments change discretely.

6 Scope and Controls

We examine how numerical sensitivity depends on the representation, the normalization convention, and the quantizer.

6.1 Scale Sensitivity across Representations

An exploratory scale sweep uses BERT-768, SASRec-32, and MoRec-512 catalogs (Devlin et al., 2019; Kang & McAuley, 2018; Yuan et al., 2023) on all three domains. We set the root mean square (RMS) of the centered embedding coordinates, $r = (\sum_i \|x_i - \mu\|^2/(Nd_x))^{1/2}$, to $\{0.008, 0.016, 0.032, 0.064\}$, with native inputs as a fifth condition. All tokenizers and generators use seed 42, giving 45 downstream evaluations. For SASRec and MoRec, inverse scaling of the history vector preserves pairwise score differences; BERT supplies only item content features. At final checkpoints, raising RMS from .008 to .064 improves MinPPL and NMSE in all nine representation–domain pairs. Equal RMS still leaves different utilization and reconstruction quality (Appendix F.1).

The downstream sweep selects tokenizers by minimum base-SID collision (Table 4). As shown in Table 2, between RMS .008 and .064, Hit@10 increases in seven pairs and decreases for SASRec-32 on Games ($-.0018$) and MoRec on Beauty ($-.0013$). The full sweep also shows non-monotonicity at intermediate and native scales (Appendix F.2). Numerical sensitivity extends beyond SASRec-512, while larger RMS and broader code use do not consistently predict higher recommendation accuracy.

6.2 Catalog-wide Input Normalization

Subtract the catalog mean and divide every item vector by the same number, $\rho(X) = (N^{-1}\sum_i \|x_i - \mu\|^2)^{1/2}$, to obtain $G_i(X) = (x_i - \mu)/\rho(X)$. This makes the mean squared item norm one, without forcing each item's norm to be one. Since $G(X^{(c)}) = G(X)$, all positive centered rescalings give the same normalized training input. The resulting per-coordinate RMS is $1/\sqrt{d_x}$, or approximately 0.0442 for $d_x = 512$. The reconstruction loss still satisfies $R = \mathrm{NMSE}/d_x$ over the full catalog. Thus fixing input scale does not fix the relative weights of reconstruction and latent quantization: loss reduction and weighting still matter for training.

A three-seed study with a common implementation finds collapse on Beauty at seeds 43/44 and Games at seed 44; Instruments remains non-collapsed throughout (Appendix F.3). Mean Hit@10 for Beauty, Instruments, and Games is .04144, .11303, and .11030, respectively. This convention removes

scale ambiguity while leaving seed-dependent training outcomes. The study selects checkpoints by minimum base-SID collision and reassigns codes within groups of items sharing a base SID before export, so it does not establish a matched comparison with calibration.

6.3 Fixed-Grid Quantization

A four-stage Residual-FSQ control replaces learned codebooks and commitment loss with fixed scalar grids (Mentzer et al., 2024). In this tokenizer-level comparison at seed 42, native SASRec-512 remains non-collapsed, with final MinPPL .126–.213 across the three domains. Rescaling still changes utilization, reconstruction, and neighborhood preservation (Appendix F.4). Thus numerical sensitivity persists in a quantizer whose training system avoids the native-input collapse observed with the studied RQ-VAE recipe.

7 Related Work

Semantic IDs. Discrete identifiers support retrieval (Tay et al., 2022; Sun et al., 2023) and recommendation (Rajput et al., 2023; Singh et al., 2023). VQ-Rec quantizes text features for transfer (Hou et al., 2023); LC-Rec, LETTER, and TokenRec incorporate collaborative signals (Zheng et al., 2024; Wang et al., 2024; Qu et al., 2024). Later work couples tokenization to generation or recommendation geometry (Liu et al., 2025; Fu et al., 2026; Liang et al., 2026; Wang et al., 2026; Zheng et al., 2026). Input preprocessing and distribution-aware tokenization also address collapse (Ju et al., 2025; 2026; Wei et al., 2026), as discussed in Section 1. We preserve teacher-score differences and test derived loss weighting with native embeddings fixed.

Scale and collapse in vector quantization. VQ-VAE learns nearest-neighbor codebooks with straight-through gradients (van den Oord et al., 2017); RQ-VAE quantizes successive residuals (Lee et al., 2022). Beyond scale, initialization, and distribution matching (Lancucki et al., 2020; Huh et al., 2023), remedies include stochastic quantization (Takida et al., 2022), normalized low-dimensional codebooks (Yu et al., 2022), coupled codeword optimization in SimVQ (Zhu et al., 2025), and geometry-aware updates in GRIT-VQ, which includes recommendation benchmarks (You et al., 2026). FSQ replaces learned codebooks with fixed scalar grids (Mentzer et al., 2024).

Function-preserving reparameterization. Equivalent parameterizations can optimize differently. Path-SGD addresses rescaling symmetries (Neyshabur et al., 2015); parameter scaling also changes apparent sharpness (Dinh et al., 2017), while natural gradients seek coordinate-invariant updates (Amari, 1998). Here an unchanged ranking function supplies different inputs across a module boundary to a separately trained tokenizer.

8 Discussion and Conclusion

RQ-VAE Semantic-ID training can be sensitive to input scale even when every upstream recommendation-score difference is preserved. Our exact reparameterization shows that the native-input architecture can represent the same non-collapsed IDs, so capacity alone does not explain the observed failure to learn them. At mapped states, scale changes the balance between reconstruction and latent quantization losses and gradients. The resulting reconstruction-loss calibration keeps the native embeddings fixed, changes encoder updates, and restores broad code use. In the main SASRec-512 study, it improves Hit@10 and NDCG@10 in all nine paired runs across three datasets.

Representation comparisons should therefore control for numerical scale and tokenizer optimization before attributing downstream differences to representation quality. Reporting input normalization, loss weights, and checkpoint and ID-generation rules makes these comparisons easier to interpret and reproduce. Broader code use does not consistently improve recommendation across representations, and our calibration coefficient depends on a pilot-selected reference scale. Choosing the reference scale or loss weight automatically, and testing transfer to other tokenizer training procedures, remain open directions.

Reproducibility Statement

Appendix B specifies data processing, model architectures, optimization, checkpoint selection, and evaluation. The appendices provide proofs, paired results, and diagnostic measurements supporting the main findings. We will release source snapshots, configurations, and a machine-readable result registry connecting reported cells to their checkpoints, SID catalogs, and evaluation outputs (Appendix B.6).

AI Use Statement

AI tools were used to assist with code development and debugging, data and result checking, experiment bookkeeping, literature and source inspection, manuscript review and language polishing, and feedback on experimental methodology and interpretation. All research decisions, theoretical claims, experimental results, and scientific conclusions were reviewed and verified by the authors. The authors reviewed all AI-assisted work and take full responsibility for the accuracy and integrity of the manuscript.

References

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1019–1028, 2017.

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

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

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 14096–14113, 2023.

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. *arXiv preprint arXiv:2507.22224*, 2025.

Clark Mingxuan Ju, Tong Zhao, Leonardo Neves, Liam Collins, Bhuvesh Kumar, Jiwen Ren, Lili Zhang, Wenfeng Zhuo, Vincent Zhang, Xiao Bai, Jinchao Li, Karthik Iyer, Zihao Fan, Yilun Xu, Yiwen Chen, Peicheng Yu, Manish Malik, and Neil Shah. Semantic IDs for recommender systems at Snapchat: Use cases, technical challenges, and design choices. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2026. doi: 10.1145/3805712.3808405. URL https://arxiv.org/abs/2604.03949.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. *arXiv preprint arXiv:1808.09781*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Adrian Lancucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J. G. A. Dolfing, Sameer Khurana, Tanel Alumae, and Antoine Laurent. Robust training of vector quantized bottleneck models. In *International Joint Conference on Neural Networks*, pp. 1–7, 2020.

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

Yu Liang, Zhongjin Zhang, Yuxuan Zhu, Kerui Zhang, Zhiluohan Guo, Wenhang Zhou, Zonqi Yang, Kangle Wu, Yabo Ni, Anxiang Zeng, Cong Fu, Jianxin Wang, and Jiazhi Xia. Rethinking generative recommender tokenizer: Recsys-native encoding and semantic quantization beyond llms. *arXiv preprint arXiv:2602.02338*, 2026.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025. doi: 10.1145/3726302.3729989. URL https://arxiv.org/abs/2409.05546.

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

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-SGD: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, 2015.

Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li. Tokenrec: Learning to tokenize id for llm-based generative recommendation. *arXiv preprint arXiv:2406.10450*, 2024.

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

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Enyun Yu. GenSAR: Unifying balanced search and recommendation with generative retrieval. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, 2025. doi: 10.1145/3705328.3748071.

Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, Ed H. Chi, and Xinyang Yi. Better generalization with semantic ids: A case study in ranking for recommendations. *arXiv preprint arXiv:2306.08121*, 2023.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Learning to tokenize for generative retrieval. In *Advances in Neural Information Processing Systems*, 2023.

Yuhta Takida, Takashi Shibuya, Wei-Hsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshimitsu Kumakura, and Yuki Mitsufuji. SQ-VAE: Variational bayes on discrete representation with self-annealed stochastic quantization. *arXiv preprint arXiv:2205.07547*, 2022.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, and Donald Metzler. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*, 2022.

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

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

Xuesi Wang, Junjie Wang, Ziliang Wang, Weijie Bian, and Guanxing Zhang. Decoupled residual quantization for robust semantic ids in recommendation. *arXiv preprint arXiv:2606.01844*, 2026.

Yibiao Wei, Jie Zou, Pengfei Zhang, Xiao Ao, Weikang Guo, Zeyu Ma, and Yang Yang. CARD: Non-uniform quantization of visual semantic unit for generative recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1959–1970, 2026. doi: 10.1145/3805712.3809538.

Haochen You, Heng Zhang, Hongyang He, Yuqi Li, and Baojing Liu. Generalized radius and integrated codebook transforms for differentiable vector quantization. In *Conference on Parsimony and Learning*, volume 328 of *Proceedings of Machine Learning Research*, pp. 1125–1160, 2026.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. Where to go next for recommender systems? id- vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835*, 2023.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering*, pp. 1435–1448, 2024. doi: 10.1109/ICDE60146.2024.00118.

Ziyu Zheng, Zhengshun Du, Yaming Yang, Bin Tong, Guan Wang, Meng Yan, Ziyu Guan, and Wei Zhao. Topogr: Revealing and preserving latent structure of semantic id in generative recommendation. *arXiv preprint arXiv:2607.25216*, 2026.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22968–22977, 2025.

GUIDE TO THE SUPPLEMENTARY EVIDENCE

The main three-condition experiment uses checkpoints at epoch 20,000 and generates IDs with greedy assignments at all levels. Earlier rescaling and scope studies retain their own protocols; Table 4 distinguishes them. Collision is $1 - U/N$ for U unique base SIDs; suffix-required fraction counts items in shared base-SID groups that need the suffix for disambiguation. Tables state whether these quantities are fractions or percentages.

Question	Direct evidence
Do the maps preserve scores and IDs?	Proofs: Appendices A.1 and A.3; finite-precision map: Table 14.
Do the main gains hold in each seed?	Appendix C.1: Figure 4, paired summary and individual contrasts in Tables 6 and 8.
Which training and export rules are used?	Table 4, Appendix B.3; STE paths: Figure 3.
How does a mapped solution evolve?	First updates: Table 16; three-domain continuation: Table 17; endpoint losses: Table 15.
Do shared starts recover code use?	Appendix D.6: Figure 5 and Table 19.
What changes outside the main setting?	RMS response: Figure 6 and Table 25; normalization and FSQ: Tables 26 and 27.

A PROOFS AND MATHEMATICAL DETAILS

A.1 RECOMMENDATION-SCORE EQUIVALENCE

Definition 1 (Recommendation-equivalent exported representations)**.** Consider two sets of item vectors and corresponding history vectors over the same catalog and history set $\mathcal{H}$, with scores $a(h_u, i)$ and $a'(h'_u, i)$ for history u. They are recommendation-equivalent if

$$a'(h'_u, i) - a'(h'_u, j) = a(h_u, i) - a(h_u, j) \quad \text{for every } u \in \mathcal{H} \text{ and items } i, j. \tag{9}$$

For $x_i^{(c)} = \mu + c(x_i - \mu)$ and $h^{(c)} = h/c$,

$$(h^{(c)})^\top x_i^{(c)} = \frac{1}{c} h^\top \mu + h^\top (x_i - \mu) \tag{10}$$

$$= h^\top x_i + \left(\frac{1}{c} - 1\right) h^\top \mu. \tag{11}$$

The second term does not depend on i. Subtracting the scores of items i and j proves Equation 2. Consequently, rankings and softmax probabilities over the catalog are identical in exact arithmetic. The construction operates on exported item and context vectors; the internal SASRec history encoder is unchanged.

A.2 RESIDUAL QUANTIZATION BACKWARD GRAPH

For $z_i = E_\theta(x_i)$, assignment rule A_ℓ selects codeword $e^\ell_{k_i^\ell}$ from codebook E^ℓ. With sg denoting stop-gradient, the reported implementation uses

$$\begin{aligned} r_i^0 &= z_i, \qquad k_i^\ell = A_\ell(r_i^{\ell-1}, E^\ell), \\ \widetilde{e}_i^\ell &= r_i^{\ell-1} + \mathrm{sg}(e^\ell_{k_i^\ell} - r_i^{\ell-1}), \\ r_i^\ell &= r_i^{\ell-1} - \widetilde{e}_i^\ell, \qquad q_i = \sum_{\ell=1}^{L} \widetilde{e}_i^\ell. \end{aligned} \tag{12}$$

For a minibatch of B items, write $\langle \|v_i\|^2 \rangle_m = (Bm)^{-1} \sum_i \|v_i\|_2^2$. With input dimension d_x and latent dimension d_z, the component losses are

$$\begin{aligned} R &= \langle \|D_\psi(q_i) - x_i\|^2 \rangle_{d_x}, \\ C_\ell &= \langle \|\,\mathrm{sg}(r_i^{\ell-1}) - e^\ell_{k_i^\ell}\|^2 \rangle_{d_z}, \quad M_\ell = \langle \|r_i^{\ell-1} - \mathrm{sg}(e^\ell_{k_i^\ell})\|^2 \rangle_{d_z}. \end{aligned} \tag{13}$$

Here C_ℓ is the codebook loss and M_ℓ the commitment loss; $Q = L^{-1}\sum_\ell (C_\ell + \beta_{\text{com}} M_\ell)$ with $\beta_{\text{com}} = 0.25$.

Each selected codeword is wrapped in a level-wise STE before being subtracted from the residual and added to the quantized output. Its forward value is $e^\ell_{k^\ell_i}$; hard assignment indices are not differentiated. The supplied Jacobians are

$$\frac{\partial_{\text{STE}} \widetilde{e}^\ell_i}{\partial r^{\ell-1}_i} = I, \qquad \frac{\partial_{\text{STE}} r^\ell_i}{\partial r^{\ell-1}_i} = I - I = 0. \tag{14}$$

Consequently, only the first level has an STE path to z_i:

$$\frac{\partial_{\text{STE}} q_i}{\partial z_i} = I, \qquad \frac{\partial M_\ell}{\partial z_i} = 0 \quad (\ell \geq 2), \qquad g_{Q,z_i} = \frac{2\beta_{\text{com}}}{LBd_z}(z_i - e^1_{k^1_i}). \tag{15}$$

Thus reconstruction and the first-level commitment term supply the direct encoder gradients. Later residuals still change in the forward pass as the encoder and earlier codebooks change during training.

The STE blocks the path from reconstruction to every codebook. Each C_ℓ directly updates the selected entries of its own codebook:

$$\frac{\partial Q}{\partial e^\ell_k} = \frac{2}{LBd_z} \sum_{i:k^\ell_i = k} (e^\ell_k - r^{\ell-1}_i). \tag{16}$$

The C_ℓ and M_ℓ losses have equal numerical values, but their stop-gradient operators and the residual STE give them different update paths. These rules apply to both greedy and Sinkhorn hard assignments. Figure 3 separates forward residual dependence from the resulting direct gradient paths.

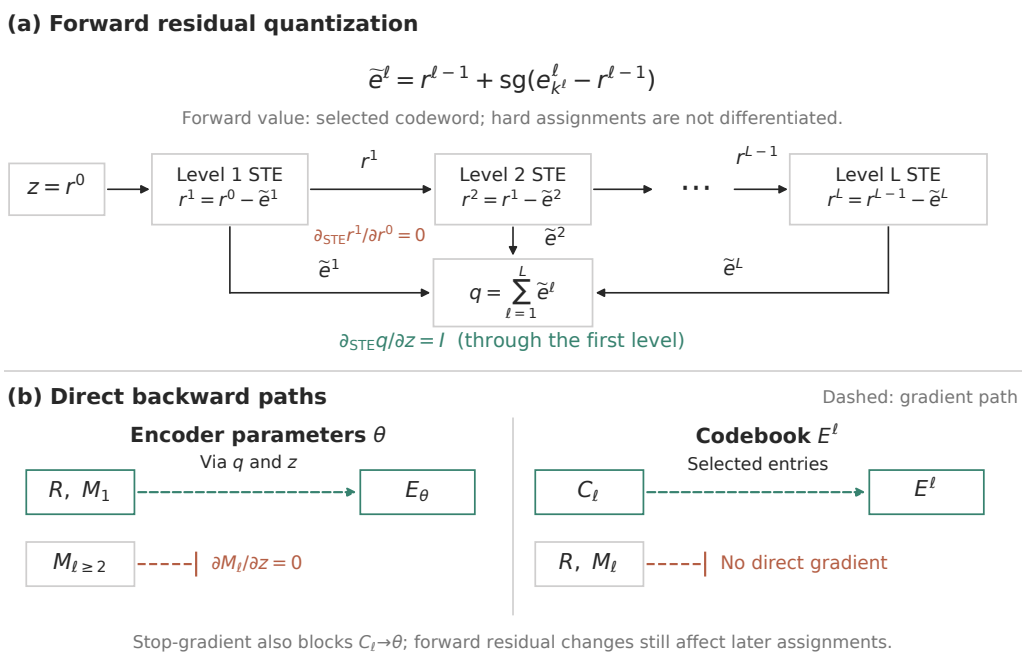

Figure 3: **Forward residual quantization and its implemented backward paths.** The level-wise straight-through estimator (STE) gives $\partial \widetilde{e}^\ell / \partial r^{\ell-1} = I$, so the residual update has Jacobian $I - I = 0$. Reconstruction and the first-level commitment loss therefore have direct gradient paths to the encoder; later commitment terms do not. Every codebook loss C_ℓ updates its selected codewords, while reconstruction has no direct codebook gradient. The total quantized output retains $\partial q / \partial z = I$. These backward rules leave the forward residual dependence intact.

A.3 Proof of Proposition 1

Write the corresponding source and target inputs as x_s and x_t. They satisfy

$$x_s = \mu + \alpha(x_t - \mu) = \alpha x_t + (1-\alpha)\mu. \tag{17}$$

Substituting the reparameterized first encoder layer gives

$$\begin{aligned} W_e' x_t + b_e' &= \alpha W_e x_t + b_e + (1-\alpha) W_e \mu && (18)\\ &= W_e\big(\alpha x_t + (1-\alpha)\mu\big) + b_e && (19)\\ &= W_e x_s + b_e. && (20) \end{aligned}$$

The first preactivation is identical. Since every subsequent encoder parameter, codebook, and quantization operation is unchanged, all encoder latents, residuals, assignment outputs, and quantized latents are identical. Finite-precision effects near assignment ties are reported in Appendix D.

Let $\widehat{x}_s = W_d v + b_d$ be the source reconstruction for the shared hidden decoder state v. The mapped output is

$$\begin{aligned} \widehat{x}_t &= W_d' v + b_d' && (21)\\ &= \gamma(W_d v + b_d) + (1-\gamma)\mu && (22)\\ &= \mu + \gamma(\widehat{x}_s - \mu). && (23) \end{aligned}$$

Because $x_t = \mu + \gamma(x_s - \mu)$, the centered reconstruction residual obeys $x_t - \widehat{x}_t = \gamma(x_s - \widehat{x}_s)$. Both reconstruction SSE and centered input variance scale by γ^2, proving equal NMSE.

A.4 Objective and Latent-Gradient Calibration

The parameter mapping leaves all latent and quantized vectors unchanged, hence it leaves the codebook and commitment losses unchanged. The preceding residual identity gives $R_t = \gamma^2 R_s$. Equating the weighted reconstruction terms, $f_t R_t = f_s R_s$, yields $f_t = \gamma^{-2} f_s = (c_s/c_t)^2 f_s$.

Corollary 1 (Latent-gradient calibration)**.** *Consider the mapped models of Proposition 1, evaluated on corresponding minibatches with the same assignments and backward rules. Let $g_{R,z}^{(a)}$ and $g_{Q,z}^{(a)}$ be the reconstruction and quantization gradients with respect to the continuous encoder output z, using STE and stop-gradient, for $a \in \{s,t\}$. Then*

$$g_{R,z}^{(t)} = \gamma^2 g_{R,z}^{(s)}, \qquad g_{Q,z}^{(t)} = g_{Q,z}^{(s)}. \tag{24}$$

Consequently, $f_t = \gamma^{-2} f_s$ gives

$$f_t g_{R,z}^{(t)} + g_{Q,z}^{(t)} = f_s g_{R,z}^{(s)} + g_{Q,z}^{(s)}. \tag{25}$$

Proof. Write the decoder as $D_a(q) = W_{d,a} F(q) + b_{d,a}$, where F contains all shared interior decoder layers. For item i, let $v_i = F(q_i)$, $J_i = \partial v_i / \partial q_i$, and $S_i = \partial_{\mathrm{STE}} q_i / \partial z_i$, the Jacobian supplied by the straight-through estimator. For the implemented residual graph, $S_i = I$ (Appendix A.2). The mapped models have identical z_i, q_i, interior activations, and assignment outputs, so they share J_i and S_i.

Let $e_{a,i} = \widehat{x}_{a,i} - x_{a,i}$. Under the coordinate-mean reduction in Equation 3, the reconstruction branch gives

$$g_{R,z_i}^{(a)} = \frac{2}{B d_x} S_i^\top J_i^\top W_{d,a}^\top e_{a,i}. \tag{26}$$

The output-layer reparameterization gives $W_{d,t} = \gamma W_{d,s}$ and $e_{t,i} = \gamma e_{s,i}$, hence

$$g_{R,z_i}^{(t)} = \frac{2}{B d_x} S_i^\top J_i^\top (\gamma W_{d,s})^\top (\gamma e_{s,i}) = \gamma^2 g_{R,z_i}^{(s)}. \tag{27}$$

For Q, the latents and selected first-level codewords in Equation 15 are identical, as are the loss coefficients, giving $g_{Q,z_i}^{(t)} = g_{Q,z_i}^{(s)}$. Applying this to every item and substituting $f_t = \gamma^{-2} f_s$ proves Equation 25. □

Both greedy and Sinkhorn-based hard assignments preserve these relations: on corresponding minibatches, the distance matrices and selected indices are identical.

The corollary matches the local backward signal at corresponding model states. Subsequent parameter updates also depend on the parameterization of the first encoder and last decoder layers and on AdamW's moment normalization, so matching latent gradients alone does not yield identical training trajectories (Kingma & Ba, 2015; Loshchilov & Hutter, 2019). The rule converts a chosen source coefficient to the target scale. In the experiments, the pilot-selected $c_s = 8$ and $f_s = 1$ determine $f_t = 64$ at $c_t = 1$; choosing a successful reference scale remains an empirical step.

A.5 Equivalence-Class Interpretation

The family of representations generated by positive centered rescaling in Equation 1 describes the same recommendation preferences, while presenting different numerical inputs to the tokenizer. This function-preserving reparameterization is analogous to weight-rescaling symmetries in neural networks (Neyshabur et al., 2015; Dinh et al., 2017). Here the scale changes at the interface between two separately trained modules, rather than between layers of one model.

B Data and Implementation

B.1 Data Processing

We use review timestamps to order each user history. Iterative 5-core filtering is applied before chronological splitting; catalog items without the text needed by content encoders are removed consistently from all conditions. The last interaction is the test target and the penultimate interaction is the validation target. Training prefixes contain neither target. Evaluation ranks against the complete processed catalog rather than sampled negatives.

B.2 Teacher Selection

SASRec-32 and SASRec-512 use the same data loader, sequence construction, negative sampler, evaluator, seed 42, and six-configuration search budget. A configuration is selected by validation metrics within 300 epochs, retrained for at most 1,000 epochs with patience 15, and evaluated once on test.

For the score-equivalence comparison, we export the already computed history contexts and apply the inverse context transform from Equation 1. The internal SASRec parameters and history encoder remain frozen.

B.3 RQ-VAE Implementation

All central conditions use four residual levels, 256 entries per codebook, and a 32-dimensional quantization latent. The MLP encoder widths are $[2048, 1024, 512, 256, 128, 64]$ followed by the latent projection; the decoder is symmetric. The first encoder and last decoder layers retain the input dimensionality. Codebooks are initialized by constrained K-means. Hidden layers use ReLU, zero dropout, and no batch normalization; affine weights use Xavier-normal initialization and biases start at zero. Constrained K-means uses 10 iterations, 10 restarts, and the full catalog (the initialization cap is 20,000 items). For N inputs and K clusters, its minimum cluster size is $\max(1, \min(\lfloor N/(2K) \rfloor, 50))$; the maximum is the largest of four times that minimum, the minimum plus one, and $\lceil N/K \rceil$. AdamW uses constant learning rate 10^{-3}, weight decay 10^{-4}, and batch size 1,024 for 20,000 epochs. Each epoch visits the full catalog, retaining the last partial minibatch. The final runs therefore use 240,000, 220,000, and 340,000 optimizer steps on Beauty, Instruments, and Games, respectively.

The assignment path is fixed across all central conditions. In a training minibatch, levels 1–3 use Euclidean nearest-code assignment because their Sinkhorn coefficients are zero. At level 4, the $B \times 256$ squared-distance matrix is centered and scaled as

$$\widetilde{\Delta} = \frac{\Delta - (\Delta_{\max} + \Delta_{\min})/2}{\Delta_{\max} - (\Delta_{\max} + \Delta_{\min})/2 + 10^{-5}}, \tag{28}$$

Table 3: Dataset statistics after iterative 5-core filtering and removal of items without text. The penultimate and final interactions form validation and test targets, respectively.

Dataset	Users	Items	Interactions
Beauty	22,332	12,086	198,215
Instruments	27,390	10,439	230,224
Games	54,955	17,256	494,934

Table 4: The main three-arm study uses the same training and ID-generation rules for both rescaling and calibration. The earlier rescaling study uses different checkpoint-selection and ID-generation rules. All downstream IDs have four base tokens and one suffix.

Study	Tokenizer state	Assignment / export
Main three-arm study	Final checkpoint, epoch 20,000; native arms share K-means start	Greedy diagnostics and export, suffix
Earlier rescaling study	Minimum greedy base collision	Greedy, duplicate-group Sinkhorn, suffix
Shared-state dynamics	Same mapped start; fixed horizons	Greedy diagnostics; training Sinkhorn at L4
Target-RMS diagnostics	Final checkpoint	Greedy at all levels
Target-RMS downstream	Minimum greedy base collision	Greedy, duplicate-group Sinkhorn, suffix
Normalization study	Minimum greedy base collision; final checkpoint also reported	Duplicate-group reassignment, suffix; exported-base diagnostics

then passed to 50 Sinkhorn iterations with $\epsilon = 0.003$; the largest entry in each transport row determines the code. K-means initialization passes residuals through the same level-specific assignment rules. Training minimizes Equation 3: reconstruction, codebook, and commitment MSEs use PyTorch's `reduction=mean`; the commitment coefficient is 0.25; the four level losses are averaged; and the aggregate VQ term has unit outer weight. The reconstruction coefficient is one in the baseline recipe and takes the stated value in calibration experiments. All optional auxiliary losses are zero. Equation 12 and Appendix A.2 specify the residual STE used in these runs, including the commitment gradient path from level 1 to the encoder.

Main three-arm study. On each domain and seed, the native raw and calibrated arms start from the same completed K-means initialization, random state, and minibatch sequence. Only f_R changes from 1 to 64. Scale-8 starts from K-means on its transformed input and uses $f_R = 1$. All arms retain training-time Sinkhorn at level 4; their final checkpoints are evaluated and exported greedily at all four levels, followed by a deterministic group-local suffix, with index zero for singleton groups. Tokenizer and generator seeds are paired at 42–44. The completed Beauty seed-42 native pair also supplies the trajectory in Appendix D.6.

Earlier checkpoint and export protocol. Every 20 epochs, the earlier rescaling study performs a full-catalog pass with greedy nearest-code assignment at all four levels and saves the first checkpoint attaining each new minimum in unsuffixed base-SID collision. This selection uses no interaction labels. Its export starts with greedy codes, then reassigns each duplicate group with Sinkhorn at level 4 for at most 20 rounds before adding the suffix. Export collision is measured after reassignment and before suffixing; checkpoint-selection collision uses the initial greedy assignments. The scale sweeps use the selection rules listed in Table 4; checkpoints selected by minimum loss supply additional diagnostics.

B.4 SID Export and Generator

The four residual assignments form the base SID. All items have four base tokens plus one suffix token; the suffix distinguishes items sharing a base SID. Before suffixing, base collision is $1 - U/N$

Table 5: Generator vocabulary and total parameter ranges across the three domains and seeds 42–44 in the final calibration comparison. All conditions use the same Transformer backbone and tied embeddings.

Condition	Vocabulary size V	Total parameters
Native, $R + Q$	42,543–49,360	9,905,152–10,777,728
Native, $64R + Q$	33,152–33,229	8,703,104–8,712,960
Scale-8, $R + Q$	33,151–33,199	8,702,976–8,709,120

for U unique SIDs; suffix-required fraction counts items in shared base-SID groups that need the suffix for disambiguation. Singleton groups still receive suffix zero. The generator is a randomly initialized T5-style network with four encoder and four decoder layers, model width 128, feed-forward width 1,024, six heads, key/value width 64, ReLU, and dropout 0.1. T5-small supplies the text-tokenizer assets, not pretrained generator weights.

Vocabulary and parameterization. Each distinct base or suffix token in a condition's exported catalog is added as one atomic token. The resulting vocabulary has size V and is used by the shared input embedding and tied output projection. Training cross-entropy spans this full vocabulary; the catalog trie constrains decoding. The Transformer backbone is fixed across conditions, while the catalog-dependent vocabulary gives a total of $4{,}459{,}648 + 128V$ parameters. Table 5 gives the ranges in the three-arm comparison. Raw Native's single base SID requires a distinct suffix for each catalog item, producing a larger vocabulary.

Training and retrieval. Training uses a maximum of 200 epochs, learning rate 5×10^{-4}, batch size 256, gradient accumulation two, and early-stopping patience 20, AdamW weight decay .01, a .01 warmup ratio, and cosine scheduling. The saved generator is selected by validation loss. Beam width is 20. A prefix trie restricts decoding to catalog-valid identifiers, after which predictions are mapped to items and deduplicated. In the three-arm study, all generators train in BF16 with effective batch size 512 and a maximum history of 20 items. The validation-loss checkpoint is selected independently for each run; decoding returns 20 catalog-valid candidates with at most 10 generated tokens. Previously interacted items are not masked. Every condition is evaluated on the same ordered test users and targets: 22,332 on Beauty, 27,390 on Instruments, and 54,955 on Games.

B.5 Randomness and Reporting

Tokenizer confirmation uses seeds 42–46 on Beauty and Instruments. The main three-arm and earlier rescaling studies pair tokenizer/generator seeds 42–44 on all three domains. Mapping, continuation, matched-RMS, and Residual-FSQ controls use seed 42. Native-input objective comparisons share the initial model, random state, and minibatch sequence. We report sample standard deviations and paired differences; Appendix C.2 identifies pilot reuse.

B.6 Source and Result Resources

The companion result registry links each reported cell to its configuration, input representation, checkpoint, SID catalog, and evaluation output through file hashes. Source snapshots and table-generation scripts accompany the registry. Full-catalog diagnostics using greedy assignments and loss components evaluated with the training assignment rule are recorded separately. These resources provide the per-run implementation details underlying the protocols in Table 4.

C Complete Results and Protocol Comparisons

The final three-arm experiment in Table 1 supplies the main rescaling and calibration comparisons. This appendix also reports the earlier rescaling study, which selected the reference scale and tested an alternative checkpoint and export protocol. The studies are reported separately using the protocols in Table 4.

C.1 Complete Results for the Three Main Conditions

All three conditions in the main experiment use tokenizer checkpoints at epoch 20,000 and greedy assignments followed by a suffix. Table 6 reports paired calibration gains without repeating the condition means in Table 1; Tables 7 and 8 give complete tokenizer diagnostics and the downstream contrasts for each seed. Figure 4 visualizes the direction and magnitude of the nine paired effects for both interventions.

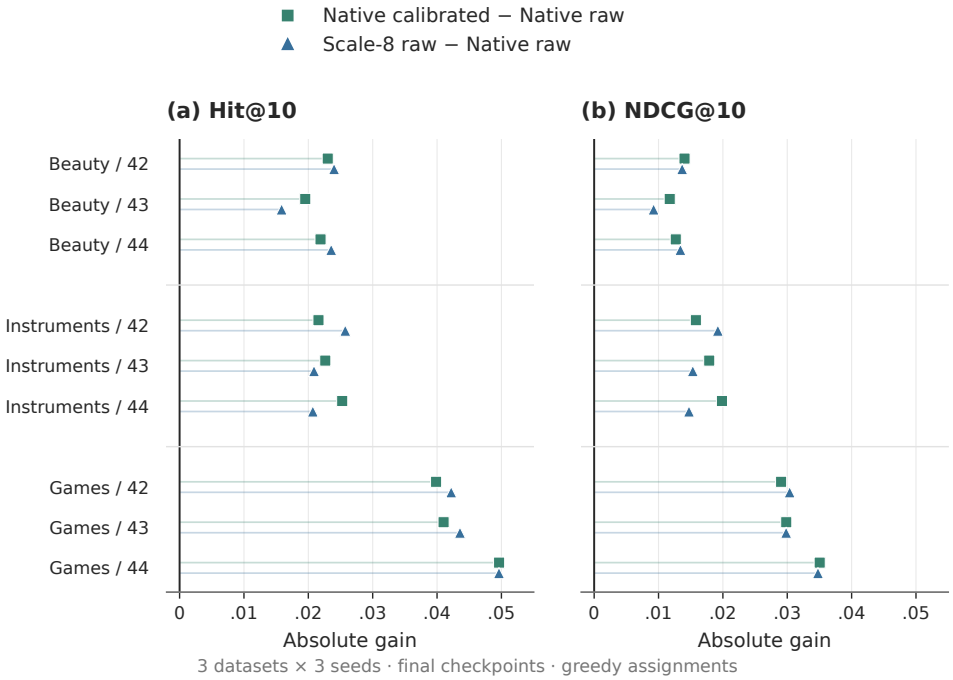

Figure 4: **Both interventions improve recommendation in every paired run.** Each row identifies a dataset and paired tokenizer/generator seed. Squares show Native calibrated minus Native raw; triangles show Scale-8 raw minus Native raw under the main protocol with final checkpoints and greedy assignments. The vertical black line marks zero gain. Every Hit@10 and NDCG@10 contrast is positive; individual seed effects carry no error bars. Thin stems connect zero to each observed effect and do not represent uncertainty. Points are slightly separated vertically for visibility. Table 8 gives the exact contrasts, including calibrated minus Scale-8.

Table 6: Paired native-scale calibration gains in the main study using final checkpoints and greedy assignments. Entries are calibrated minus raw Native, mean ± sample standard deviation of the three seed-wise differences. Hit and NDCG use the 0–1 scale. Each of the nine pairs improves both metrics; condition means are in Table 1.

Dataset	ΔHit@10	ΔNDCG@10
Beauty	$+0.02148 \pm 0.00178$	$+0.01282 \pm 0.00116$
Instruments	$+0.02316 \pm 0.00190$	$+0.01785 \pm 0.00203$
Games	$+0.04350 \pm 0.00535$	$+0.03131 \pm 0.00325$

All nine native raw tokenizers have one active code at every level and one unique base SID. Eight calibrated runs use all 256 entries in every codebook; Beauty seed 42 uses 212 at level 1 and 256 at the remaining levels. The calibrated suffix vocabulary ranges from 34 to 105, compared with one suffix per catalog item (10,439–17,256) for Native raw.

Calibration improves Hit and NDCG at cutoffs 5 and 10 in all nine pairs. Its paired Hit@10 improvement ranges from .01952 to .04964. Mean calibrated minus Scale-8 Hit@10 differences are $+.00034 \pm .00290$, $+.00072 \pm .00445$, and $-.00163 \pm .00143$ on Beauty, Instruments, and Games. The corresponding NDCG@10 differences are $+.00072 \pm .00164$, $+.00142 \pm .00438$, and $-.00035 \pm .00086$. Thus native calibration and input rescaling reach close average accuracy through different learned code assignments and reconstruction errors.

Table 7: The same final-checkpoint tokenizers used in Table 1. Full-catalog greedy diagnostics, mean $\pm$ sample standard deviation over three seeds. Collision precedes suffixing; suffix-required fraction counts items sharing a base SID and thus needing the suffix for disambiguation.

Dataset	Condition	MinPPL	NMSE	Collision (%)	Suffix (%)
Beauty	Native raw	0.0039 ± 0.0000	1.0007 ± 0.0001	99.99 ± 0.00	100.00 ± 0.00
	Native calibrated	0.8019 ± 0.0980	0.5288 ± 0.0373	1.44 ± 0.12	1.86 ± 0.14
	Scale-8 raw	0.8709 ± 0.0904	0.3697 ± 0.0797	0.91 ± 0.02	1.14 ± 0.07
Instruments	Native raw	0.0039 ± 0.0000	1.0006 ± 0.0000	99.99 ± 0.00	100.00 ± 0.00
	Native calibrated	0.8431 ± 0.0125	0.3718 ± 0.0265	0.80 ± 0.15	1.00 ± 0.24
	Scale-8 raw	0.8784 ± 0.0066	0.2139 ± 0.0057	0.53 ± 0.03	0.60 ± 0.06
Games	Native raw	0.0039 ± 0.0000	1.0007 ± 0.0000	99.99 ± 0.00	100.00 ± 0.00
	Native calibrated	0.8195 ± 0.0092	0.4357 ± 0.0102	1.33 ± 0.04	1.79 ± 0.06
	Scale-8 raw	0.8726 ± 0.0083	0.2876 ± 0.0083	1.17 ± 0.06	1.52 ± 0.10

Table 8: All nine paired differences in the three-condition study at final checkpoints. C denotes Native calibrated, R Native raw, and S Scale-8 raw. Differences use the 0–1 metric scale. All four recommendation metrics (Hit and NDCG at 5 and 10) improve in every C–R pair.

		Hit@10			NDCG@10		
Dataset	Seed	C–R	S–R	C–S	C–R	S–R	C–S
Beauty	42	+0.02302	+0.02400	-0.00099	+0.01404	+0.01367	+0.00037
	43	+0.01952	+0.01585	+0.00367	+0.01174	+0.00924	+0.00250
	44	+0.02190	+0.02355	-0.00166	+0.01268	+0.01340	-0.00072
Instruments	42	+0.02158	+0.02574	-0.00416	+0.01580	+0.01922	-0.00341
	43	+0.02264	+0.02088	+0.00175	+0.01788	+0.01533	+0.00255
	44	+0.02526	+0.02070	+0.00456	+0.01985	+0.01473	+0.00512
Games	42	+0.03985	+0.04220	-0.00235	+0.02905	+0.03037	-0.00133
	43	+0.04102	+0.04356	-0.00255	+0.02984	+0.02983	+0.00002
	44	+0.04964	+0.04962	+0.00002	+0.03504	+0.03477	+0.00027

C.2 Earlier Rescaling Study: Scale Selection

The earlier study selects the checkpoint with minimum full-catalog greedy base-SID collision, then applies duplicate-group Sinkhorn reassignment before suffixing. Its tokenizer confirmation and downstream results below use this protocol; the main three-arm experiment uses final checkpoints and greedy export.

Scale selection used tokenizer-only results on Beauty and Instruments at seed 42. Candidates were $c \in \{2, 4, 8, 16\}$ and a domain-specific scale matching SASRec-32 RMS. For each candidate, we added its MinPPL increase and input-space NMSE reduction within each domain, then selected the candidate with the largest minimum of these sums across Beauty and Instruments. Scores for Scale-8, Scale-4, RMS matching, and Scale-16 were 1.0665, .9599, .9467, and .5665; Scale-2 remained collapsed. Scale-8 was fixed before downstream testing and transfer to Games.

The earlier study reuses the Beauty and Instruments pilot tokenizers at seed 42. Among the four additional tokenizer seeds 43–46, every pair improves both MinPPL and NMSE. Mean MinPPL increase/NMSE decrease is .6121/.3208 on Beauty and .6517/.5491 on Instruments. Excluding seed 42 from downstream results gives ΔHit@10 $= .02982 \pm .00241$ on Beauty and $.02158 \pm .00496$ on Instruments; corresponding ΔNDCG@10 is $.01686 \pm .00222$ and $.01295 \pm .00355$. All retained pairs improve. Games is the transfer domain, with no role in scale selection.

C.3 Tokenizer Confirmation under the Earlier Protocol

Table 9: Tokenizer confirmation at checkpoints selected by minimum base-SID collision. MinPPL, NMSE, and collision show mean $\pm$ sample standard deviation; $\text{nPPL}_{1\text{–}4}$ show per-level means. Beauty and Instruments use five tokenizer seeds; Games uses three. Greedy collision is a fraction in $[0, 1]$, $1 - U/N$, where U is the number of unique unsuffixed base SIDs. For $K = 256$, one-code collapse gives $\text{nPPL} = 1/K = 0.00390625$, rounded to 0.0039 in the table.

Dataset	Input	MinPPL ↑	NMSE ↓	nPPL_1	nPPL_2	nPPL_3	nPPL_4	Collision ↓
Beauty	Native	.0039±.0000	1.0006±.0001	.0039	.0039	.0039	.6001	.9816±.0035
	Scale-8	**.6494±.1790**	**.6726±.0318**	.6779	.7701	.7417	.9239	**.0041±.0002**
Instruments	Native	.0039±.0000	1.0007±.0000	.0039	.0039	.0039	.2771	.9857±.0060
	Scale-8	**.6388±.0728**	**.4618±.0537**	.8020	.6388	.6718	.9373	**.0033±.0002**
Games	Native	.0039±.0000	1.0006±.0001	.0039	.0039	.0039	.4185	.9882±.0026
	Scale-8	**.7564±.0208**	**.5398±.0375**	.8427	.7896	.7811	.9410	**.0041±.0003**

At checkpoints selected by minimum base-SID collision, native inputs collapse the first three levels while retaining partial use at level 4. Scale-8 improves code use and reconstruction across all confirmation seeds, as summarized in Table 9 and detailed below.

Table 10: Scale-8 improves code use and reconstruction in all five seeds on each selection domain. Checkpoints minimize base-SID collision; Native MinPPL is .003906 in every cell. Positive differences denote MinPPL increase and NMSE decrease.

Dataset	Seed	Scale-8 MinPPL	ΔMinPPL	Native NMSE	Scale-8 NMSE	ΔNMSE
Beauty	42	.7829	.7790	1.0005	.6440	.3565
	43	.7558	.7519	1.0005	.6724	.3282
	44	.3512	.3473	1.0006	.6642	.3364
	45	.7431	.7392	1.0007	.6563	.3444
	46	.6140	.6101	1.0005	.7264	.2741
Instruments	42	.5719	.5680	1.0007	.5022	.4985
	43	.6343	.6304	1.0007	.4724	.5282
	44	.5992	.5953	1.0007	.4983	.5024
	45	.6276	.6237	1.0008	.4660	.5348
	46	.7612	.7573	1.0007	.3699	.6309

C.4 Downstream Effects under the Earlier Protocol

Table 11: Earlier rescaling study: checkpoints minimize base-SID collision; codes are then reassigned within groups sharing a base SID before suffixing. Item-level test results over paired tokenizer/generator seeds 42–44 (mean ± sample standard deviation). Δ is the within-seed Scale-8 minus Native difference; all nine pairs improve on both metrics.

Dataset	Metric	Native	Scale-8	Paired Δ
Beauty	Hit@10	.03432±.00104	.06500±.00174	**+.03069±.00227**
	NDCG@10	.01810±.00128	.03484±.00080	**+.01674±.00159**
Instruments	Hit@10	.09399±.00374	.11583±.00104	**+.02184±.00354**
	NDCG@10	.07494±.00325	.08815±.00135	**+.01321±.00255**
Games	Hit@10	.09275±.00241	.12378±.00167	**+.03103±.00315**
	NDCG@10	.06402±.00118	.08323±.00092	**+.01921±.00136**
Macro	Hit@10			**+.02785**
	NDCG@10			**+.01639**

Table 12: Paired downstream effects of rescaling. Entries are Scale-8 minus Native for tokenizer/generator seeds 42–44; tokenizers selected by minimum base-SID collision.

Dataset	Seed	ΔH@10	ΔN@10
Beauty	42	.032420	.016480
	43	.031524	.018436
	44	.028121	.015290
Instruments	42	.022380	.013726
	43	.025082	.015462
	44	.018072	.010441
Games	42	.030570	.019346
	43	.034374	.020498
	44	.028132	.017790

All 36 paired differences across the four reported metrics are positive. The mean differences over the nine dataset–seed pairs are .027853 for Hit@10, .016386 for NDCG@10, .018544 for Hit@5, and .013413 for NDCG@5.

C.5 The Motivating Teacher Comparison

Table 13: A stronger upstream teacher can yield a worse directly quantized SID recommender. Each dimension uses the same validation-only search budget, chronological split, and seed 42. This motivating dimension-control block is independent of the main Native/Scale-8 comparison. Bold marks the better dimension within each dataset and evaluation stage.

Dataset	Teacher	Teacher Hit@10 ↑	Direct SID Hit@10 ↑	Direct SID NDCG@10 ↑	MinPPL ↑	NMSE ↓
Beauty	SASRec-32	0.04903	**0.05933**	**0.03170**	**0.73820**	**0.02677**
	SASRec-512	**0.07227**	0.03139	0.01564	0.00391	1.00048
Instruments	SASRec-32	0.08934	**0.11062**	**0.08651**	**0.22969**	**0.29131**
	SASRec-512	**0.11161**	0.09843	0.07904	0.00391	1.00073
Games	SASRec-32	0.09915	**0.12048**	**0.08171**	**0.91775**	**0.03913**
	SASRec-512	**0.11557**	0.09047	0.06205	0.00391	1.00073

The same-budget dimension study is independent of the paired Scale-8 comparison. Increasing SASRec dimension improves teacher Hit@10 by 16.6–47.4% but reduces directly quantized SID Hit@10 by 11.0–47.1%. The 512-dimensional catalogs have per-coordinate centered RMS .01456–.01683, below their 32-dimensional counterparts. The within-teacher rescaling study holds recommendation preferences fixed to isolate the numerical interface from this cross-model comparison.

D Representability and Training Mechanisms

Reparameterizing the input and output layers preserves the code assignments; reweighting reconstruction changes how training develops from that state. We connect these two results through loss components, encoder updates, and code-use trajectories.

D.1 Preserving Code Assignments across Scales

Table 14 shows the exact reparameterization on the earlier checkpoints selected by minimum base-SID collision in all three domains and both directions. These checkpoints also supply the mapped starts for the continuations. Moving Scale-8 to Native places a tokenizer using many codes at native scale; the reverse mapping preserves the collapsed code assignments. Latents and assignments are unchanged in float64, while MSE follows the predicted factor of 64 or $1/64$.

Table 14: Reparameterization preserves code assignments in both directions. Seed 42; latent error uses float64, while SID agreement and paired MinPPL/NMSE use float32. Scales 8 and 1 denote Scale-8 and Native. Paired diagnostic cells list source above target; MSE ratios are target/source.

Dataset	Scale	Latent error	SID agreement	MinPPL	NMSE	MSE ratio
Beauty	$8 \to 1$	6.66e−16	1.000000	.782919 .782919	.643979 .643979	.015625
Beauty	$1 \to 8$	5.20e−18	1.000000	.003906 .003906	1.000490 1.000490	64.0000
Instruments	$8 \to 1$	8.88e−16	.999904	.571879 .571879	.502210 .502209	.015625
Instruments	$1 \to 8$	4.34e−18	1.000000	.003906 .003906	1.000726 1.000726	64.0000
Games	$8 \to 1$	6.66e−16	1.000000	.740338 .740338	.566318 .566318	.015625
Games	$1 \to 8$	3.90e−18	.999942	.003906 .003906	1.000724 1.000724	64.0000

Finite precision. Two of 318,248 float32 level assignments differ, both at near-tied fourth-level distances with margins below 10^{-8}. Aggregate code utilization and collision are unchanged; the maximum NMSE difference is 6.82×10^{-7}. In float64, all assignments match and the maximum latent error is 8.88×10^{-16}.

D.2 Increasing Nominal Capacity

Varying codebook size over $K \in \{64, 128, 256, 512\}$ leaves native SASRec-512 collapsed on every domain under the same training recipe. The first three levels each use one code, MinPPL equals $1/K$, and NMSE is 1.00049–1.00086 across the 12 cells. At $K = 256$, widening the latent from 32 to 64 or 128 also leaves all six added cells at MinPPL $1/256$ and NMSE 1.00048–1.00080. These capacity changes do not repair the observed training failure; the exact map establishes that the original architecture can already represent a tokenizer using many codes.

D.3 Loss Components at Mapped States

The raw and calibrated Native continuations share the same input and mapped starting model. Their reconstruction-to-quantization ratio is .0976 on Beauty, .0609 on Instruments, and .0926 on Games

under $R + Q$. Multiplication by 64 matches the respective Scale-8 ratios of 6.2452, 3.9001, and 5.9288, as predicted by Equation 8.

Table 15: Reweighting reconstruction reverses the loss ordering of the initial collapsed and mapped tokenizers, but lower endpoint loss does not imply broader code use. Frozen checkpoints on Native input (seed 42). Evaluation uses catalog-order batches of 1,024 with the training assignment rule: greedy at levels 1–3 and Sinkhorn at level 4. R, C, and M are the coordinate-mean reconstruction, level-mean codebook, and level-mean commitment MSEs. The values of C and M coincide at each frozen state, so they share a column; their gradient paths differ. The last columns evaluate every checkpoint under the same two objectives, $\mathcal{L}_1 = R + C + 0.25M$ and $\mathcal{L}_{64} = 64R + C + 0.25M$.

Dataset	Checkpoint state	R	$C = M$	$\mathcal{L}_1$	$\mathcal{L}_{64}$
Beauty	Collapsed from scratch	2.835e−4	7.638e−9	2.835e−4	1.814e−2
	Exact mapped	1.825e−4	1.499e−3	2.056e−3	1.356e−2
	Raw continuation	9.721e−5	1.055e−4	2.291e−4	6.353e−3
	Calibrated continuation	9.376e−5	5.846e−3	7.402e−3	1.331e−2
Instruments	Collapsed from scratch	2.487e−4	6.254e−9	2.487e−4	1.592e−2
	Exact mapped	1.250e−4	1.641e−3	2.177e−3	1.005e−2
	Raw continuation	7.787e−5	6.380e−5	1.576e−4	5.064e−3
	Calibrated continuation	7.261e−5	3.263e−3	4.151e−3	8.726e−3
Games	Collapsed from scratch	2.120e−4	5.321e−9	2.121e−4	1.357e−2
	Exact mapped	1.201e−4	1.039e−3	1.418e−3	8.982e−3
	Raw continuation	6.620e−5	9.673e−5	1.871e−4	4.358e−3
	Calibrated continuation	6.645e−5	4.838e−3	6.114e−3	1.030e−2

Table 15 compares frozen models under common evaluation objectives. Reweighting reverses the ordering of the initial collapsed and mapped solutions on every domain. At the continuation endpoints, however, the raw arm still has the lower loss under $64R + Q$ despite its narrower code use. These evaluations distinguish objective values from the code assignments produced by training.

D.4 First-Step Gradients and Updates

Table 16: Calibration changes the relative reconstruction and quantization gradient norms and the resulting AdamW updates from the same tokenizer. Beauty, first minibatch. $\rho = \|f_R g_R\|/\|g_Q\|$; cosine is between the weighted reconstruction and quantization gradients. The update column is the factor-64/factor-1 update-norm ratio.

Encoder group	ρ_1	ρ_{64}	Cosine	Update ratio
Input	.0583	3.7293	-.3323	1.7399
Interior	.1380	8.8300	-.0863	1.1485
Latent	.2182	13.9617	.1452	1.0255

At the shared Beauty starting state, raising the reconstruction coefficient changes encoder gradient composition immediately. Reconstruction has no direct gradient path to codebooks, whose first gradients and AdamW updates therefore match across arms. Subsequent residuals, assignments, and codebook updates diverge. Figure 2 shows the resulting 100-epoch trajectory; Table 16 resolves the initial encoder measurements by parameter group.

D.5 THREE-DOMAIN CONTINUATION

Table 17: Final state after 2,000 full-catalog continuation epochs (seed 42). Native-raw and Native-calibrated start from the same mapped model and differ only by the analytically derived reconstruction factor. "Used" lists active codes at levels 1–4, out of 256. Collision is a fraction in $[0, 1]$. Mean MinPPL across datasets is .8818 for Scale-8, .1942 for Native raw, and .8921 for Native calibrated.

Dataset	Continuation	MinPPL ↑	NMSE ↓	Collision ↓	Used L1/L2/L3/L4
Beauty	Scale-8 raw	.9182	.3460	.0090	256/256/256/256
	Native raw	.1741	.3409	.0147	50/141/190/256
	Native calibrated	**.9255**	**.3300**	**.0079**	256/256/256/256
Instruments	Scale-8 raw	**.8739**	.2996	.0053	256/256/256/256
	Native raw	.2526	.3133	.0114	98/93/135/255
	Native calibrated	.8670	**.2918**	**.0048**	256/256/256/256
Games	Scale-8 raw	.8535	.3117	.0112	256/256/256/256
	Native raw	.1558	**.3104**	.0223	47/162/145/256
	Native calibrated	**.8838**	.3122	**.0097**	256/256/256/256

Table 18: The code-use trajectories separate within the first epoch on every domain. MinPPL after one and 20 full-catalog epochs, with 11–17 minibatch updates per epoch.

Dataset	Epoch	Scale-8 raw	Native raw	Native calibrated
Beauty	1	.4749	.0386	.1938
	20	.7947	.0960	.7756
Instruments	1	.5103	.1154	.4099
	20	.5743	.1337	.5773
Games	1	.4088	.0523	.2015
	20	.7847	.0999	.7865

All conditions briefly lose utilization after restarting with fresh optimizers. After one epoch, raw Native collision reaches 74.1%, 22.4%, and 56.0% on Beauty, Instruments, and Games, respectively, versus at most 10.5% for calibrated Native. At epoch 2,000, calibration uses all 256 codes at every level. Table 17 gives reconstruction and collision at the same endpoint.

D.6 TRAINING FROM SHARED K-MEANS INITIALIZATION

Figure 5 tracks all nine shared-initialization pairs from the three-arm study across 11 saved checkpoints. The raw arms finish with complete collapse, whereas every calibrated arm recovers code use. Table 19 provides selected measurements for the Beauty seed-42 pair.

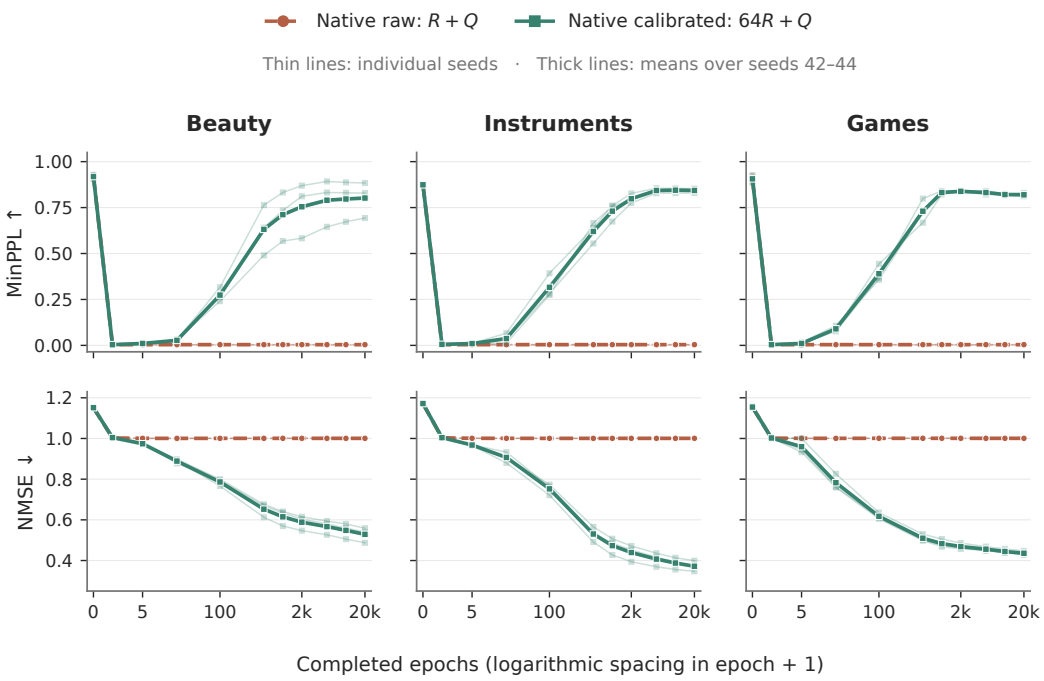

Figure 5: **Recovery from shared native-input initialization across all nine pairs.** Raw and calibrated tokenizers share the completed K-means initialization, random state, and minibatch sequence within each domain–seed pair. Thin lines show seeds 42–44; thick lines show their means at the 11 saved checkpoints. The horizontal position is $\log_{10}(\text{epoch} + 1)$, with ticks labeled by completed epochs; lines connect observations without smoothing. Every raw endpoint uses one code per level (MinPPL $1/256$); all nine calibrated runs recover code use. These are from-scratch trajectories, complementary to the mapped continuation in Figure 2.

Table 19: Only the calibrated arm recovers after the initial loss of code use. Beauty seed 42, shared K-means initialization; full-catalog greedy diagnostics.

	MinPPL		NMSE	
Epoch	$R + Q$	$64R + Q$	$R + Q$	$64R + Q$
0	.9046	.9046	1.1513	1.1513
1	.0039	.0039	1.0024	1.0038
20	.0039	.0327	1.0005	.8775
100	.0039	.2407	1.0006	.7929
1,000	.0039	.5680	1.0007	.6341
5,000	.0039	.6454	1.0006	.5799
20,000	.0039	.6933	1.0007	.5403

In the Beauty seed-42 pair, both arms reach MinPPL $1/256$ after one epoch; only the calibrated arm subsequently recovers. Its 11,901 unique base SIDs leave 1.95% of items requiring the suffix for disambiguation, versus 100% under the raw objective.

E Objective and Protocol Sensitivity

These controls ask how the findings depend on reconstruction weighting, checkpoint selection, and the exported ID structure.

Table 20: Normalizing reconstruction scale removes the MinPPL gap on Beauty and reduces it on Instruments. Metric values over seeds 42–44 (mean ± sample standard deviation). Each dataset has two input scales and two reconstruction objectives. Differences are Scale-8 minus Native for MinPPL and Native minus Scale-8 for NMSE, so a positive value favors Scale-8.

Dataset	Recon. term	MinPPL			NMSE		
		Native	Scale-8	Δ	Native	Scale-8	Δ
Beauty	R	.0039±.0000	.6300±.2418	+.6261	1.0005±.0000	.6602±.0146	+.3404
	$c^{-2}R$	.0039±.0000	.0039±.0000	+.0000	1.0005±.0000	1.0000±.0000	+.0005
Instruments	R	.0039±.0000	.6018±.0313	+.5979	1.0007±.0000	.4910±.0162	+.5097
	$c^{-2}R$	.0039±.0000	.2022±.0592	+.1983	1.0007±.0000	.2134±.0157	+.7873

E.1 Reversing the Reconstruction-Scale Change

The factorial experiment crosses input scale $c \in \{1, 8\}$ with raw or c^{-2}-normalized reconstruction MSE, leaving the latent term unchanged. Models initialize on each input distribution. Normalization removes the MinPPL difference on Beauty and reduces it on Instruments. On Instruments, it further improves Scale-8 NMSE despite reducing code use. Thus changing objective balance affects both domains, with different reconstruction and utilization responses.

E.2 Sensitivity to Reconstruction Weight

Table 21: Broad code use is attained at several reconstruction weights. Beauty seed 42, shared K-means initialization and minibatch sequence, final checkpoints. Every level uses all 256 codes in each row.

f_R	MinPPL ↑	NMSE ↓	Collision $[0, 1]$ ↓
16	.8408	.4711	.0103
32	.8757	.4778	.0120
128	.8639	.5008	.0137
256	.8857	.4678	.0142

Reconstruction weights $16, 32, 128, 256$ share a Beauty seed-42 K-means initialization and minibatch sequence, separate from the weight-64 comparison. All recover broad utilization, with non-monotone endpoint reconstruction and collision. The recovery therefore occurs at several weights. This experiment does not include 64 and does not establish its performance relative to those weights at a common initialization.

E.3 Checkpoint Dependence

In the seed-42 scale sweep, SASRec-512 collapses at $c \leq 2$ and recovers at $c \geq 4$ on Beauty and Instruments at the final checkpoint. Within the recovered region, final MinPPL has a broad plateau while MinPPL at checkpoints selected by minimum base-SID collision is non-monotone. The transition is present under both checkpoint rules; the degree of utilization depends on the selected state.

E.4 What Changes in the Exported IDs

Table 22 summarizes all 18 downstream SID catalogs from the earlier rescaling comparison. Both conditions uniquely identify every item using five tokens. The contrast lies in base-code sharing and suffix vocabulary: collapse transfers much of item identification to the suffix.

Table 22: Diagnostics of the final IDs in the earlier Scale-8 comparison, using checkpoints selected by minimum base-SID collision. Ranges span seeds 42–44; length counts semantic tokens, excluding decoder special tokens. Collision and suffix-required values are fractions in $[0, 1]$; the latter counts items requiring the suffix to distinguish a shared four-token base.

Dataset	Input	Length	Export coll.	Suffix req.	Suffix vocab.	Max group
Beauty	Native	5	0.9788–0.9791	0.9998–1.0000	144–159	144–159
Beauty	Scale-8	5	0.0000–0.0001	0.0000–0.0002	1–2	1–2
Instruments	Native	5	0.9764–0.9785	0.9963–0.9983	217–286	217–286
Instruments	Scale-8	5	0.0000–0.0002	0.0000–0.0004	1–2	1–2
Games	Native	5	0.9853–0.9875	0.9978–0.9995	209–551	209–551
Games	Scale-8	5	0.0000–0.0003	0.0000–0.0005	1–3	1–3

F Generalization across Representations and Quantizers

F.1 Scale Sensitivity across Representations

Table 23: Utilization changes with scale in every representation, with different maximizing scales. Checkpoints selected by minimum base-SID collision (tokenizer seed 42). Ranges are minimum–maximum MinPPL over $c \in \{0.5, 1, 2, 4, 8, 16, 32\}$; "best c" maximizes MinPPL. Collision range (fraction in $[0, 1]$) is maximum minus minimum base-code collision. BERT is a content control, not a recommendation teacher.

	Beauty			Instruments		
Representation	MinPPL range	Best c	Collision range	MinPPL range	Best c	Collision range
BERT-768	.1175–.7849	16	.0142	.0286–.3604	2	.0139
SASRec-32	.1371–.9276	4	.0022	.1600–.4853	1	.0007
SASRec-512	.0039–.7829	8	.9910	.0039–.8503	4	.9958
MoRec-512	.3439–.9193	1	.0066	.2482–.5269	2	.0046

The rescaling sweep varies $c \in \{0.5, 1, 2, 4, 8, 16, 32\}$ at seed 42. Utilization changes in every representation, while its maximizing scale depends on both representation and domain. BERT supplies content features; SASRec and MoRec provide recommendation representations.

Table 24: Low absolute scale reduces utilization across representation families. Target-RMS sweep at epoch 20,000 (tokenizer seed 42). Entries are MinPPL; † marks an exact level collapse with normalized perplexity $1/256$. Native RMS is the source per-coordinate centered RMS. BERT is a content control rather than a recommendation teacher.

Dataset	Representation	Native RMS	Native	$r = .008$	$r = .016$	$r = .032$	$r = .064$
Beauty	BERT-768	.1496	.7504	.0039†	.0039†	.0972	.3070
	SASRec-32	.0629	.7795	.0039†	.2878	.6617	.7188
	MoRec-512	.0997	.9093	.1340	.4475	.4411	.6605
Instruments	BERT-768	.1480	.6939	.0039†	.0039†	.0383	.3308
	SASRec-32	.6516	.4926	.0862	.2166	.3124	.8278
	MoRec-512	.1014	.9179	.1364	.4288	.6788	.8879
Games	BERT-768	.1541	.8656	.0039†	.0072	.0701	.4465
	SASRec-32	.0520	.8689	.1522	.2375	.8736	.8621
	MoRec-512	.1004	.8798	.1644	.4499	.8147	.8661

The target-RMS sweep sets per-coordinate centered RMS to $\{.008, .016, .032, .064\}$. Across all nine representation–domain pairs, raising RMS between the endpoints improves final MinPPL, NMSE, and reconstructed-neighborhood Recall@10. The MinPPL endpoint direction is the same for final checkpoints and those selected by minimum base-SID collision or minimum loss. Equal per-coordinate RMS still leaves dimensions, covariance spectra, and neighborhood structures different.

F.2 Complete Downstream Scale Sweep

Table 25 reports all 45 cells underlying Table 2: three representations, three domains, and five inputs, with tokenizer and generator seed 42. Figure 6 makes the intermediate responses visible alongside each Native baseline. The .008-to-.064 contrast improves Hit@10 and NDCG@10 in seven pairs and reduces both in two, with further non-monotonicity at intermediate and native scales.

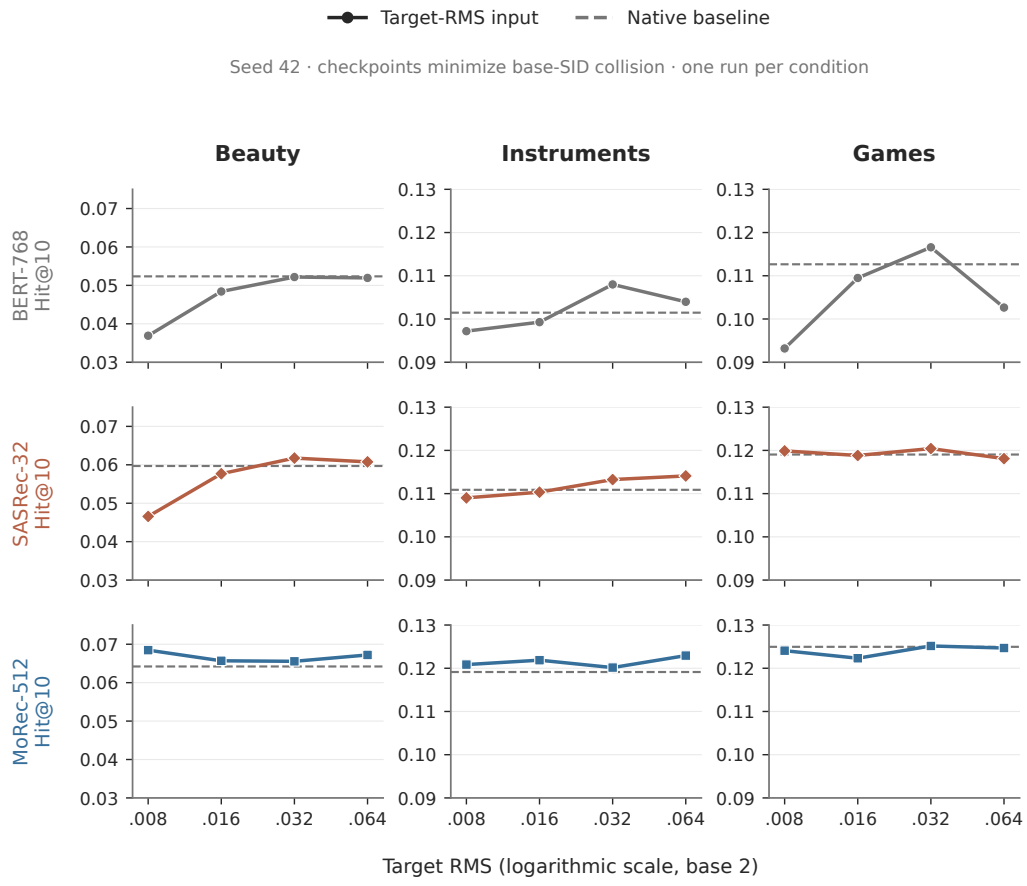

Figure 6: **Downstream responses to scale depend on representation and domain.** Each panel shows the four target-RMS conditions; the dashed horizontal line is that panel's Native-input result. RMS is logarithmically spaced, and vertical limits are shared within each dataset column. All 45 results use tokenizer and generator seed 42 and checkpoints selected by minimum base-SID collision. Before appending a suffix, codes are reassigned within groups of items sharing a base SID. These are individual runs, without multi-seed uncertainty bands. Increasing RMS from .008 to .064 improves Hit@10 in seven of nine panels; the intermediate points reveal further non-monotonicity. Table 25 retains all numerical results.

Table 25: The complete scale sweep shows representation- and domain-dependent downstream responses. All 45 cells use tokenizer and generator seed 42 and tokenizers selected by minimum base-SID collision. Collision and suffix-required fraction use the actual exported SID catalogs and are fractions in $[0, 1]$; each full SID has five tokens. Hit and NDCG also use the 0–1 scale.

Dataset	RMS	Hit@10	NDCG@10	Export coll.	Suffix req.
BERT-768					
Beauty	Native	0.05235	0.02780	0.0007	0.0014
	.008	0.03690	0.01905	0.9830	0.9960
	.016	0.04841	0.02513	0.0015	0.0027
	.032	0.05217	0.02871	0.0024	0.0040
	.064	0.05194	0.02902	0.0021	0.0040
Instruments	Native	0.10146	0.07815	0.0026	0.0049
	.008	0.09719	0.07678	0.9798	0.9960
	.016	0.09927	0.07622	0.0031	0.0057
	.032	0.10800	0.08232	0.0026	0.0048
	.064	0.10398	0.08017	0.0026	0.0049
Games	Native	0.11266	0.07676	0.0060	0.0114
	.008	0.09319	0.06389	0.9734	0.9977
	.016	0.10949	0.07307	0.0065	0.0121
	.032	0.11659	0.07931	0.0061	0.0115
	.064	0.10265	0.06795	0.0059	0.0112
SASRec-32					
Beauty	Native	0.05969	0.03279	0.0001	0.0002
	.008	0.04657	0.02536	0.9990	0.9997
	.016	0.05768	0.03104	0.0000	0.0000
	.032	0.06175	0.03341	0.0000	0.0000
	.064	0.06076	0.03201	0.0002	0.0003
Instruments	Native	0.11088	0.08582	0.0000	0.0000
	.008	0.10902	0.08474	0.0000	0.0000
	.016	0.11033	0.08370	0.0000	0.0000
	.032	0.11325	0.08821	0.0000	0.0000
	.064	0.11409	0.08842	0.0001	0.0002
Games	Native	0.11904	0.08064	0.0001	0.0002
	.008	0.11988	0.08108	0.0001	0.0001
	.016	0.11882	0.07971	0.0003	0.0005
	.032	0.12043	0.08173	0.0001	0.0001
	.064	0.11810	0.07989	0.0004	0.0008
MoRec-512					
Beauty	Native	0.06421	0.03384	0.0025	0.0049
	.008	0.06847	0.03620	0.0026	0.0050
	.016	0.06569	0.03542	0.0025	0.0049
	.032	0.06556	0.03487	0.0025	0.0049
	.064	0.06721	0.03574	0.0025	0.0049
Instruments	Native	0.11913	0.09187	0.0046	0.0084
	.008	0.12085	0.09264	0.0046	0.0084
	.016	0.12187	0.09292	0.0046	0.0084
	.032	0.12015	0.09297	0.0046	0.0084
	.064	0.12293	0.09331	0.0046	0.0084
Games	Native	0.12496	0.08361	0.0291	0.0524
	.008	0.12407	0.08308	0.0291	0.0524
	.016	0.12232	0.08139	0.0291	0.0524
	.032	0.12516	0.08367	0.0291	0.0524
	.064	0.12468	0.08333	0.0291	0.0524

F.3 CATALOG-WIDE INPUT NORMALIZATION

For a non-degenerate catalog, define

$$\rho(X) = \left(N^{-1} \sum_i \|x_i - \mu\|_2^2 \right)^{1/2}, \qquad g_i = \frac{x_i - \mu}{\rho(X)}. \tag{29}$$

The normalized catalog $G(X) = [g_1; \ldots; g_N]$ satisfies

$$G(X^{(c)}) = G(X), \qquad (\rho(X)h)^\top g_i = h^\top x_i - h^\top \mu. \tag{30}$$

It removes the positive centered scale while preserving preferences, anisotropy, and relative geometry. Reconstruction maps back as $\widehat{x}_i = \mu + \rho(X)\widehat{g}_i$, preserving centered NMSE. Normalizing the mean squared item norm to one gives full-catalog $R = \mathrm{NMSE}/d_x$, so input dimension still affects the relative reconstruction and quantization losses.

Table 26: Catalog-wide normalization to mean squared item norm one, with a shared tokenizer implementation, seeds 42–44. Selected checkpoints minimize base-SID collision under greedy assignments; codes are then reassigned within shared base-SID groups before suffixing. MinPPL and NMSE use these exported base assignments. Final MinPPL uses the same export procedure at epoch 20,000. The last three rows give downstream mean ± sample standard deviation.

Dataset	Seed	MinPPL		Selected	Export	H@10	N@10
		Selected	Final	NMSE	coll. $[0, 1]$		
Beauty	42	0.35440	0.47821	0.48905	0.00017	0.06381	0.03526
Beauty	43	0.00391	0.00391	1.00005	0.99669	0.02929	0.01453
Beauty	44	0.00391	0.00391	1.00006	0.97882	0.03121	0.01607
Instruments	42	0.58810	0.82008	0.48354	0.00029	0.11406	0.08760
Instruments	43	0.77798	0.83073	0.33347	0.00000	0.11081	0.08566
Instruments	44	0.52741	0.59863	0.34501	0.00057	0.11424	0.08734
Games	42	0.78527	0.82311	0.40349	0.00035	0.11872	0.08071
Games	43	0.74665	0.84024	0.42991	0.00046	0.12099	0.08152
Games	44	0.00391	0.00391	1.00022	0.98656	0.09120	0.06339
Beauty mean ± SD						0.04144±0.01940	0.02195±0.01155
Instruments mean ± SD						0.11303±0.00193	0.08687±0.00106
Games mean ± SD						0.11030±0.01658	0.07521±0.01024

This study trains all nine domain–seed cells with the same tokenizer implementation and configuration, using one normalized input catalog per domain. Tokenizer and generator seeds vary jointly over 42–44; the teacher remains frozen. Tokenizers train for 20,000 epochs. The primary checkpoint minimizes full-catalog greedy base collision, and export applies duplicate-group Sinkhorn reassignment before suffixing. Table 26 reports utilization and reconstruction from these exported base assignments, rather than a separate all-greedy diagnostic pass. The final-checkpoint column uses the same export procedure; downstream generators use only the selected checkpoint.

Beauty seeds 43/44 and Games seed 44 collapse in the first three levels at the selected checkpoint and in all four at the final checkpoint. Instruments remains non-collapsed at both views for all three seeds. Hit@10 is $.04144 \pm .01940$, $.11303 \pm .00193$, and $.11030 \pm .01658$ on Beauty, Instruments, and Games. The study documents seed-dependent outcomes for this normalization convention, whose per-coordinate RMS is $1/\sqrt{512} \approx 0.0442$ for SASRec-512.

Equation 30 removes the positive scale ambiguity exactly. Training on this normalized input can still collapse under the specified recipe. The checkpoint and export rules differ from the main study using final checkpoints and greedy assignments, so these results do not provide a matched comparison with objective calibration. The result registry records the source code hashes and all nine completed test cells.

F.4 ALTERNATIVE QUANTIZER: RESIDUAL-FSQ

Table 27: Residual-FSQ avoids native-input collapse but remains scale-sensitive. Seed 42; results use the final checkpoint after 20,000 completed epochs (zero-based index 19,999); levels 2–4 use all 256 codes in every row. Base collision is a fraction in $[0, 1]$. Normalized inputs use the catalog-wide scaling in Appendix F.3. No downstream generator is trained for this control.

Dataset	Input	MinPPL ↑	L1 codes	NMSE ↓	Base coll. ↓	kNN R@10 ↑
Beauty	Native	.1353	131	.2578	.0041	.2430
	Scale-8	.2756	219	.2837	.0031	.2742
	Normalized	.2009	176	.2496	.0026	.2625
Instruments	Native	.1264	156	.1703	.0041	.3094
	Scale-8	.1655	224	.1792	.0028	.3680
	Normalized	.1800	230	.1494	.0033	.3844
Games	Native	.2133	209	.1749	.0036	.3836
	Scale-8	.2969	246	.2455	.0026	.3594
	Normalized	.2195	207	.2128	.0025	.3633

The control uses ResidualFSQ from vector-quantize-pytorch 1.31.1, with scalar levels $[4, 4, 4, 4]$, four residual stages, and quantization dropout disabled. Each stage emits one of 256 indices. Native remains non-collapsed, while changing scale alters utilization, reconstruction, and neighborhood preservation.