POISONING THE FIT SET: END-TO-END ATTACKS ON OPQ-BASED SEMANTIC-ID RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic-ID generative recommenders often rely on an OPQ tokenizer fitted on a subset of catalog items and then used to assign discrete identities to the full catalog. We identify this fit set as an overlooked trust boundary and ask whether injected catalog items can alter the Semantic IDs of legitimate, untouched items and affect downstream recommendation. Clean refitting is exactly reproducible, with Adjusted Rand Index (ARI) 1.000, allowing structural changes to be attributed to fit-set modification rather than tokenizer randomness. Across three OPQ-based recommendation pipelines and three Amazon catalogs, fit-set poisoning consistently causes substantial structural corruption, with mean per-digit ARI falling to 0.14–0.26. The effect also reproduces across ten independently published text encoders and is not confined to idealized embedding-space injection: text-derived poison passed through the same tokenizer-fitting pipeline reaches a comparable corruption regime. However, tokenizer corruption does not translate monotonically into recommendation harm. Across nine recommender–catalog settings trained to convergence, only three show significant downstream changes, and their directions differ across architectures, while substantial structural corruption remains present throughout. Matched benign-growth controls further show that ordinary catalog additions can induce structural drift comparable to malicious insertion, so drift magnitude alone is not a reliable attack signature. Six published anomaly-detection baselines fail to restore the clean tokenizer structure. A fit-aware trusted-reference admission gate strongly separates the non-adaptive attacks from held-out legitimate additions across three catalogs, but an adaptive adjacent-mimicry attack exposes an evasion region. Frozen-rotation warm-started refitting complements screening by constraining admitted updates, maintaining ARI at or above 0.98 across tested single-refit attack severities and geometries, while sequential drip-feed updates reveal gradual erosion under repeated refitting. Code will be provided after acceptance.

1 INTRODUCTION

Generative recommenders increasingly formulate recommendation as sequence generation, representing each item by a short sequence of discrete tokens, or a *Semantic ID*, rather than an atomic item identifier (Rajput et al., 2023). While Semantic IDs can be constructed using different quantization mechanisms, a recent line of generative recommenders, including RPG (Hou et al., 2025) and DiffGRM (Liu et al., 2026), adopts optimized product quantization (OPQ) (Ge et al., 2013a). In these systems, OPQ learns a shared transformation and codebooks from catalog-item representations and subsequently quantizes each item under the fitted structure. This design is consequential: RPG, for example, reports degraded recommendation performance when its OPQ-based IDs are replaced by residual-quantization or random IDs. We focus on this OPQ-based branch because the quantizer is obtained through a separate, data-dependent fitting step whose output is subsequently used to define item identities for recommender training. This creates an implicit trust boundary: if an attacker can introduce manipulated items into the *fit set*, those items can alter the shared quantization geometry and, consequently, the Semantic IDs assigned to legitimate items.

This fit set has received little attention as an attack surface. Prior work has studied poisoning of clustering algorithms (Cinà et al., 2020), manipulation of dense-retrieval corpora (Ben-Tov & Sharif,

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.

2025; Zhong et al., 2023), and attacks on product-quantization boundaries in retrieval (Feng et al., 2020), but these settings do not establish what happens when the tokenizer of a Semantic-ID recommender is poisoned and the resulting IDs are subsequently used to train the recommender itself. This leaves a fundamental end-to-end question unanswered: *Can poisoning the fit set of an OPQ-based Semantic-ID tokenizer materially alter the Semantic IDs assigned to legitimate items, and, if so, do those changes propagate into downstream recommendation behavior?*

We find a clear distinction between the two parts of this question. At the tokenizer level, the vulnerability is substantial and reproducible: fit-set poisoning reorganizes the Semantic-ID structure of legitimate items across catalogs, tokenizer configurations, attack conditions, and text encoders. Yet this structural corruption does not translate monotonically into downstream harm. Under full multi-seed recommender training, larger changes in Semantic-ID structure do not consistently produce larger losses in recommendation quality, and significant downstream effects can differ in direction across attack conditions. The central empirical finding is therefore a separation between *tokenizer vulnerability* and *recommendation harm*: a tokenizer can be strongly corrupted without the magnitude of that corruption reliably predicting its downstream consequence.

Our claim is deliberately scoped to the OPQ fit-set trust boundary rather than Semantic-ID tokenization in general. We evaluate three recommendation pipelines using OPQ-derived IDs: DiffGRM (Liu et al., 2026), RPG (Hou et al., 2025), and an autoregressive recommender following the TIGER decoding paradigm (Rajput et al., 2023) but supplied with the same OPQ IDs under study. This configuration lets us test whether the observed behavior persists under a different downstream generative architecture; it should not be confused with original TIGER, which constructs Semantic IDs using a learned residual-quantization tokenizer. Likewise, jointly optimized tokenizers such as LETTER (Wang et al., 2024) expose a different training and trust surface and fall outside our main claim. Across the OPQ-based configurations and catalogs studied here, however, the structural vulnerability persists, showing that it is not confined to a single recommender implementation.

Finally, we examine whether existing defenses can protect this trust boundary. Six adapted anomaly-detection methods fail to reliably preserve the tokenizer structure. We therefore study two complementary defenses: a fit-aware trusted-reference gate that screens candidate additions before refitting, and frozen-rotation warm-started refitting that constrains admitted updates. The gate strongly separates the non-adaptive attacks from legitimate growth but exhibits adaptive evasion, while constrained refitting substantially limits structural movement after admission.

Our contributions are:

1. We identify and formalize *fit-set poisoning* as an attack surface in OPQ-based Semantic-ID recommendation, and use permutation-invariant measures to quantify structural corruption under tokenizer refitting. We establish that this vulnerability is substantial and reproducible across three OPQ-based recommendation pipelines, three catalogs, multiple attack conditions, and independently published text encoders.
2. We trace fit-set poisoning through full downstream recommender training and uncover a separation between *tokenizer vulnerability* and *recommendation harm*. Despite substantial structural corruption, downstream effects are condition-specific and non-monotonic: greater Semantic-ID corruption does not reliably imply greater degradation in recommendation quality.
3. We systematically evaluate defenses for this trust boundary. Six adapted anomaly-detection methods fail to reliably preserve tokenizer structure; fit-aware admission strongly separates the non-adaptive attacks from legitimate growth but exhibits adaptive evasion, while frozen-rotation refitting constrains admitted updates and maintains high structural stability.

2 Related work

Semantic-ID generative recommendation and tokenization. Generative recommendation represents items as sequences of discrete tokens and formulates recommendation as sequence generation. TIGER (Rajput et al., 2023) established this paradigm using a learned residual-quantization tokenizer to construct Semantic IDs from item representations. Subsequent work has increasingly treated item tokenization as a design problem in its own right. LETTER (Wang et al., 2024), for example, augments residual quantization with collaborative alignment and code-assignment objectives, while DIGER (Fu et al., 2026) makes Semantic-ID learning differentiable so that recommen-

dation objectives can directly influence the tokenizer. Other systems instead construct Semantic IDs through product quantization: RPG (Hou et al., 2025) and DiffGRM (Liu et al., 2026) use OPQ-based tokenization, with DiffGRM introducing parallel semantic encoding to obtain independently quantized SID digits. Recent work has also examined the reliability of Semantic-ID tokenizers themselves, showing that code collisions can materially affect item-level evaluation (Zhang et al., 2026). Collectively, this literature studies how Semantic IDs should be constructed, optimized, and evaluated. We study a complementary question specific to the OPQ-based branch: whether the catalog data used to fit the tokenizer constitutes an adversarial attack surface.

Poisoning representation and retrieval pipelines. Our attack is related to prior work that manipulates the data or geometry underlying learned representations. Cinà et al. (2020) show that injecting adversarial points into a k-means fit can alter the resulting clustering, while Ben-Tov & Sharif (2025); Zhong et al. (2023) manipulate dense-retrieval corpora to promote attacker-chosen content through the retriever's embedding space. Attacks on product quantization have also targeted code-cell boundaries of an already fitted quantizer (Feng et al., 2020), and related work has motivated attacking pre-quantization fitting when discrete quantization obstructs direct end-to-end optimization (Branch et al., 2025). Our setting differs in both attack point and consequence: the adversary poisons the data used to fit an OPQ Semantic-ID tokenizer *before* recommender training, thereby changing the discrete identities assigned to legitimate catalog items. We then follow these changes through full recommender training to determine whether tokenizer-level corruption propagates to downstream recommendation behavior.

Poisoning recommender systems. Recommender-system poisoning has primarily targeted the *interaction side* of the learning pipeline. Classical shilling attacks inject fake users or ratings into collaborative recommenders (Chirita et al., 2005; Mehta et al., 2007; Mehta & Nejdl, 2009), while later work constructs adversarial profiles or behavior sequences against top-N and sequential recommenders (Fang et al., 2020; Zhang et al., 2020; Agate et al., 2026; Zhao et al., 2025; Wang et al., 2025). Recent attacks further target LLM-based or LLM-augmented recommenders through poisoned interactions, prompts, or backdoor triggers (Zhao et al., 2026b; Ning et al., 2025; Li et al., 2025). Our attack surface is orthogonal to these approaches: the adversary does not manipulate users, ratings, or interaction sequences, but instead injects items into the catalog-side fit set used to construct Semantic IDs. Consequently, poisoning occurs upstream of recommender training and can change the representation of legitimate items that the attacker never directly modifies.

Detection and robust quantization. Our defense study draws on established anomaly-detection families, including local-density scoring (Breunig et al., 2000), isolation-based detection (Liu et al., 2008), robust-covariance methods (Rousseeuw & Van Driessen, 1999; Hayase et al., 2021), representation-space spectral detection (Tran et al., 2018), and density-aware detection for poisoned embeddings (Huang et al., 2025). These methods were developed for general outlier or training-data poisoning settings rather than the fit set of a product-quantization tokenizer. Separately, incremental PQ and OPQ methods show that quantizer updates need not require unconstrained refitting from scratch: prior work updates PQ codebooks (Xu et al., 2018) or jointly updates OPQ transformations and codebooks (Liu et al., 2020) as data evolve. Their objective, however, is efficient adaptation to benign streaming data rather than robustness to adversarial fit-set manipulation. Our defense study instead combines trusted-reference admission screening, calibrated against legitimate catalog additions, with constrained OPQ refitting to limit structural movement after an update is admitted.

3 Problem Setup and Threat Model

System and fit set. Let a catalog contain N legitimate items with text descriptions $\{t_i\}_{i=1}^{N}$. A text encoder $f_\phi : \mathcal{T} \to \mathbb{R}^{d_0}$ produces embeddings $e_i = f_\phi(t_i)$, which are mapped by a dimensionality-reduction function g to $x_i = g(e_i) \in \mathbb{R}^d$. An OPQ tokenizer is fitted on a subset of catalog representations, denoted the *fit set* M. It learns an orthogonal transformation $R \in \mathbb{R}^{d\times d}$ and m sub-codebooks $C_1, \ldots, C_m$, each containing L codewords, by minimizing reconstruction error:

$$(R^\star, C_1^\star, \ldots, C_m^\star) = \arg \min_{R, C_1, \ldots, C_m} \sum_{i \in M} \|Rx_i - \hat{q}_{C_1,\ldots,C_m}(Rx_i)\|_2^2 . \tag{1}$$

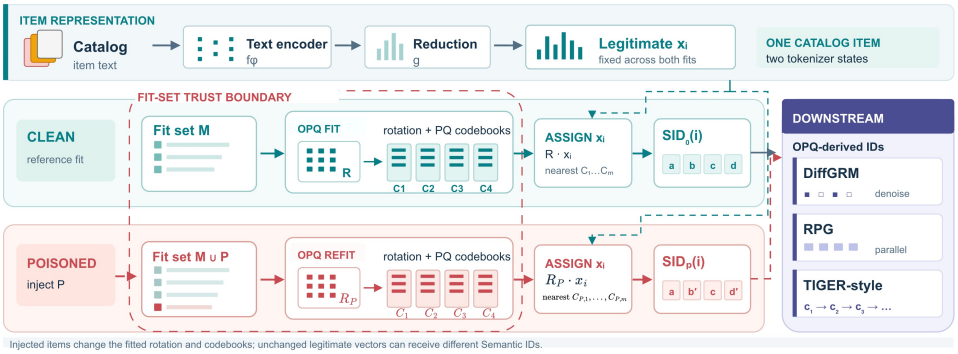

Figure 1: OPQ fit-set poisoning. Injected items P change M to $M \cup P$; refitting can therefore change the Semantic IDs of untouched items. We evaluate the resulting IDs with DiffGRM, RPG, and a TIGER-style recommender.

where $\hat{q}_{C_1,\ldots,C_m}(\cdot)$ denotes product quantization by concatenating the codeword selected independently in each subvector. The fitted tokenizer assigns each catalog item a Semantic ID

$$\mathrm{sid}(i) = \big(c_1(i), \ldots, c_m(i)\big), \qquad c_k(i) = \arg\min_{c \in C_k^\star} \|[R^\star x_i]_k - c\|_2 \,. \tag{2}$$

where $[\cdot]_k$ denotes the k-th subvector. Crucially, the parameters $(R^\star, C_1^\star, \ldots, C_m^\star)$ are estimated from M but subsequently determine the discrete representations assigned to legitimate catalog items and consumed by the downstream recommender. In our evaluated pipelines, M consists of items appearing in the training portion of the interaction data.

Attack surface and objective. We consider an adversary that can introduce a set P of injected items into the data from which the tokenizer is fitted. For an attack budget β, $|P| = \lfloor \beta N \rfloor$, and the tokenizer is refitted on

$$M' = M \cup P. \tag{3}$$

This produces poisoned parameters $(R_P^\star, C_{P,1}^\star, \ldots, C_{P,m}^\star)$ and hence a new assignment $\mathrm{sid}_P(i)$ for each legitimate item. The attack does not modify the representations of legitimate items themselves; instead, it changes the shared quantization structure under which they are discretized. Figure 1 illustrates this fit-set trust boundary: the attacker modifies only the tokenizer fit set, while the resulting change in the shared OPQ quantization structure can alter the Semantic IDs assigned to legitimate catalog items and subsequently affect the downstream recommender. The adversary's objective is therefore to induce structural changes in the Semantic IDs of legitimate items solely through manipulation of the tokenizer fit set. Section 4 instantiates this objective using several injection strategies σ, while the downstream experiments test whether the resulting tokenizer-level corruption propagates to recommendation behavior.

Attacker capability and knowledge. Our primary threat model grants the attacker control only over the injected items P. The attacker has no ability to modify legitimate catalog items or users' interaction histories, no code execution, and no access to the parameters or gradients of the downstream recommender. The embedding-space attacks in Section 4 specify injected representations directly, providing a controlled setting for characterizing the vulnerability of the OPQ fit itself. Because arbitrary embedding vectors need not correspond to realizable catalog content, Appendix B separately evaluates whether such manipulation can be achieved through item text. In that setting, the attacker is additionally granted white-box gradient access to the public text encoder f_ϕ, but remains unable to access or modify the fitted OPQ parameters or downstream recommender.

4 Method

4.1 Measuring tokenizer corruption

Comparing $\mathrm{sid}(\cdot)$ before and after poisoning by counting label disagreements is unsound because k-means/OPQ codebook indices are not identifiable across refits: two equivalent partitions may use entirely different integer labels. A determinism control, in which an identical fit set is refitted

with different random initialization, yields Adjusted Rand Index (ARI) 1.0000 and zero raw-label flips after label alignment. We therefore compare clean and poisoned tokenizers using permutation-invariant partition measures.

We report **Adjusted Rand Index (ARI)** and **Normalized Mutual Information (NMI)** between the partitions of legitimate items induced by the clean and poisoned Semantic IDs. For two partitions U, V of the same n items, with contingency counts n_{uv},

$$\mathrm{ARI}(U,V) = \frac{\sum_{u,v}\binom{n_{uv}}{2} - \left[\sum_u \binom{n_{u\cdot}}{2}\sum_v \binom{n_{\cdot v}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_u \binom{n_{u\cdot}}{2} + \sum_v \binom{n_{\cdot v}}{2}\right] - \left[\sum_u \binom{n_{u\cdot}}{2}\sum_v \binom{n_{\cdot v}}{2}\right]/\binom{n}{2}}. \tag{4}$$

ARI equals 1 for identical partitions up to relabeling and is approximately 0 in expectation for independent random partitions. We report NMI alongside ARI as a complementary, non-chance-corrected measure of shared partition information.

ARI can become unstable in the near-singleton regime encountered in our synthetic control, where the codespace is much larger than the catalog and isolated collisions can dominate its chance-correction term. We therefore additionally report **B-cubed precision/recall/F1** (Bagga & Baldwin, 1998) for this setting. For partitions K (key) and R (response), let K_i and R_i denote the clusters containing item i. B-cubed scores each item as

$$P_i = \frac{|K_i \cap R_i|}{|R_i|}, \qquad \rho_i = \frac{|K_i \cap R_i|}{|K_i|}, \tag{5}$$

and averages P_i and ρ_i over items to obtain precision and recall, whose harmonic mean gives B-cubed F1. We report B-cubed F1 alongside ARI and NMI for the singleton-dominated synthetic control; ARI and NMI are used throughout the real-embedding comparisons.

4.2 Poisoning strategies

We evaluate three poison-placement strategies σ, with poison directions defined in normalized embedding space. **Random** samples injected representations from an isotropic Gaussian, $x_j \sim \mathcal{N}(0, I_d)$, providing an unstructured baseline. **Cluster-targeted** samples Gaussian jitter with scale 0.08 around the centroid of a target cluster of legitimate items. **Leverage** samples Gaussian jitter with scale 0.05 around a point 3.0 units along the catalog's leading principal component, targeting a high-variance, low-density region. For attack budget β, the attacker injects $|P| = \lfloor \beta N \rfloor$ items. We vary both σ and β rather than treating a single poisoning configuration as representative.

4.3 Experimental setup

All experiments reuse each target pipeline's original tokenizer-fitting and recommender-training routines. We evaluate three pipelines using OPQ-derived Semantic IDs: DiffGRM (Liu et al., 2026), RPG (Hou et al., 2025), and an autoregressive encoder–decoder following the TIGER decoding paradigm (Rajput et al., 2023) but supplied with the same OPQ IDs studied here. The latter is therefore a TIGER-style downstream decoder, not the original TIGER tokenizer based on residual quantization. All use `sentence-t5-base` embeddings. DiffGRM and the TIGER-style pipeline use 4×256 OPQ IDs with 256-dimensional PCA and post-PCA L_2 normalization; RPG uses its native configuration, with 32 digits on Beauty and 16 on Toys and Sports, 128-dimensional whitened PCA, and no post-PCA normalization.

We use the Amazon Reviews 2014 5-core datasets (McAuley et al., 2015): Beauty (12,101 items, 22,363 users), Toys and Games (11,924 items, 19,412 users), and Sports and Outdoors (18,357 items, 35,598 users). Structural tokenizer experiments cover all nine recommender–catalog combinations. Downstream clean-versus-poisoned training is also performed for all nine combinations under cluster-targeted poisoning at $\beta = 0.25$, using three paired seeds per cell except DiffGRM/Beauty, which additionally uses the full four-budget sweep with three or five paired seeds depending on budget and attack strategy. Full preprocessing, attack-construction, tokenizer, training, and ablation details are given in Appendix A.

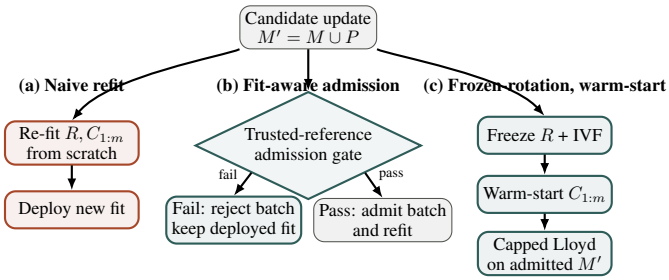

Figure 2: Three ways to process a candidate fit-set update $M' = M \cup P$: (a) naive refitting from scratch; (b) fit-aware admission screening, which scores candidate additions against the trusted catalog before refitting; and (c) frozen-rotation warm-started refitting, which freezes the deployed rotation/IVF centroid, initializes codebooks from the deployed fit, and runs capped Lloyd updates on admitted M'.

4.4 Defense mechanisms

Figure 2 contrasts unconstrained refitting with two complementary mechanisms for the tokenizer fit-set trust boundary: fit-aware admission screening and frozen-rotation warm-started refitting.

4.4.1 Fit-aware trusted-reference admission gate

The matched-growth analysis shows that tokenizer drift itself cannot reliably distinguish malicious insertion from legitimate catalog evolution. We therefore screen candidate additions *before* refitting by comparing them only with the already-trusted catalog. For a candidate batch $B = \{x_i\}_{i=1}^{k}$ and trusted population $\mathcal{T}$, we compute

$$s(B) = \frac{1}{k} \sum_{i=1}^{k} \max_{x_j \in \mathcal{T}} \cos(x_i, x_j).$$

Candidate items cannot serve as neighbors for one another, preventing a coordinated injected batch from providing its own local-density support. We calibrate a two-sided acceptance region using legitimate catalog additions and flag batches whose trusted-neighbor similarity is atypical of benign growth. Appendix E gives the full calibration, operating curves, and adaptive-evasion evaluation.

4.4.2 Frozen rotation with warm-started codebooks

The second defense constrains OPQ refitting directly rather than filtering individual items. Starting from the last trusted deployment, we **freeze the OPQ rotation R and IVF coarse centroid**, **warm-start each PQ codebook C_k from its deployed centroids**, and run a capped number of Lloyd iterations on the admitted candidate fit set M'. Thus, admitted items still contribute to the update, but the tokenizer evolves from the previously deployed quantization geometry instead of being re-derived from scratch.

This differs from incremental OPQ (Liu et al., 2020), which updates both the transformation and codebooks for streaming efficiency: here the rotation is fixed and the iteration cap is used for robustness. Unlike admission screening, this mechanism does not decide whether an update is malicious; it limits how much the tokenizer can change after an update is admitted. We additionally evaluate it under sequential poisoning with warm-started updates across repeated refits.

5 Results

5.1 Structural vulnerability and generalization

Fit-set poisoning consistently reorganizes the Semantic IDs of legitimate, untouched items. Clean refitting is exactly reproducible in every configuration we test (ARI $= 1.000$), whereas poisoning produces substantial structural drift even at the smallest tested budget. Table 1 summarizes this effect under the native OPQ configurations of the three recommender pipelines and across all three

Table 1: Mean per-digit ARI of legitimate items after fit-set poisoning, averaged over three strategies and three repeats, by attack budget. The clean-refit control is 1.000 in every row. Tuple ARI is averaged over all attack cells and is reported only for the 4-digit IDs.

Catalog	Tokenizer	1%	5%	10%	25%	Tuple ARI
Beauty	DiffGRM / TIGER-style (4×256)	0.156	0.172	0.150	0.155	0.446
Beauty	RPG-native (32×256)	0.260	0.257	0.255	0.247	—
Toys	DiffGRM / TIGER-style (4×256)	0.161	0.157	0.167	0.160	0.456
Toys	RPG-native (16×256)	0.159	0.158	0.156	0.152	—
Sports	DiffGRM / TIGER-style (4×256)	0.190	0.178	0.170	0.170	0.435
Sports	RPG-native (16×256)	0.143	0.145	0.143	0.138	—

catalogs. Because DiffGRM and the TIGER-style pipeline share the same 4×256 OPQ configuration, they share the corresponding tokenizer-level result. RPG instead uses its native 32×256 configuration on Beauty and 16×256 on Toys and Sports. For comparability across these different SID lengths, we report mean per-digit ARI and additionally report full-tuple ARI where it remains informative.

The vulnerability is therefore not specific to one catalog, SID length, or OPQ preprocessing configuration: every poisoned cell remains far below the 1.000 clean-refit ceiling, with mean per-digit ARI never exceeding 0.260. The same conclusion holds when the upstream representation model is varied. Across ten independently published text encoders spanning 2019–2026, clean-refit ARI remains exactly 1.000, while poisoned ARI ranges from 0.415 to 0.625. Thus, changing the encoder changes the magnitude of the reorganization but does not remove the fit-set vulnerability.

Finally, the effect is not restricted to idealized embedding-space injection. We construct poison through writable item text and pass the resulting representations through the same unmodified tokenizer-fitting pipeline. At batch scale, naive text mimicry ($\beta = 0.25$) yields ARI 0.4487, while a gradient-optimized adversarial suffix at only 0.33% budget yields ARI 0.4357, both within the corruption regime produced by idealized embedding-space attacks. Detailed strategy-by-budget results, synthetic controls, localization analysis, the complete ten-encoder evaluation, and the text-realizability experiments are reported in Appendix B.

5.2 Structural corruption does not predict downstream harm

The structural vulnerability in Section 5.1 is consistent across tokenizers and catalogs, but its downstream consequence is not. Table 2 compares clean and poisoned training across all nine recommender–catalog combinations at the common `cluster_targeted`, $\beta = 0.25$ condition. Only three cells reach significance, and they disagree in both sign and architecture: DiffGRM is harmed on Sports (-11.5%), the TIGER-style decoder is harmed on Beauty (-4.0%) but improves on Toys ($+5.2\%$), and RPG is not significantly changed on any catalog. Thus, the structural corruption established in Table 1 does not imply a consistent direction or magnitude of recommendation-quality change.

The same separation appears when attack geometry and budget are varied within DiffGRM on Beauty. Table 3 shows that `random` is indistinguishable from clean at every tested budget, whereas `cluster_targeted` is significant only at 5% and `leverage` changes from a small positive effect at 1% to significant degradation at larger budgets. In particular, increasing the poisoning budget does not produce a monotonic increase in downstream harm, despite the strong tokenizer corruption already present throughout this range.

Together, these experiments reveal a separation between *tokenizer vulnerability* and *recommendation harm*. Fit-set poisoning reliably changes the discrete structure assigned to legitimate items, but neither the existence nor severity of that structural change determines whether downstream quality improves, degrades, or remains statistically unchanged. Appendix C provides the full per-metric analysis, including the 10% budget condition (Appendix C.3) and additional Toys results (Appendix C.4). We separately document the targeting correction (Appendix C.2) and retain the earlier

Table 2: Paired clean-vs.-poisoned NDCG@10 under `cluster_targeted` poisoning at $\beta = 0.25$. Means are over the seeds specified in Table 5; p is from a paired t-test. Bold marks $p < 0.05$.

Recommender	Catalog	Clean	Poisoned	Change (p)
DiffGRM	Beauty	0.0278	0.0285	$+2.6\%$ ($p = .078$)
DiffGRM	Toys	0.0324	0.0329	$+1.5\%$ ($p = .426$)
DiffGRM	Sports	0.0141	0.0125	$\mathbf{-11.5\%}$ ($\mathbf{p = .028}$)
TIGER-style	Beauty	0.0402	0.0386	$\mathbf{-4.0\%}$ ($\mathbf{p = .020}$)
TIGER-style	Toys	0.0409	0.0430	$\mathbf{+5.2\%}$ ($\mathbf{p = .018}$)
TIGER-style	Sports	0.0253	0.0250	-1.1% ($p = .312$)
RPG	Beauty	0.0416	0.0424	$+1.9\%$ ($p = .210$)
RPG	Toys	0.0453	0.0456	$+0.7\%$ ($p = .739$)
RPG	Sports	0.0242	0.0249	$+3.2\%$ ($p = .182$)

Table 3: DiffGRM/Beauty NDCG@5 change relative to clean across attack budgets. At 1% and 5%, all conditions use three paired seeds. At 10%, `cluster_targeted` uses five seeds while `leverage` and `random` use three; at 25%, all conditions use five. Bold marks $p < 0.05$.

Budget	`cluster_targeted`	`leverage`	`random`
1%	-1.4% ($p = .69$)	$+2.4\%$ ($p = .26$)†	-1.5% ($p = .55$)
5%	$\mathbf{-7.8\%}$ ($\mathbf{p = .036}$)	$\mathbf{-12.0\%}$ ($\mathbf{p = .023}$)	-4.7% ($p = .28$)
10%	$+0.5\%$ ($p = .85$)	$\mathbf{-14.4\%}$ ($\mathbf{p = .024}$)	-0.6% ($p = .74$)
25%	$+2.2\%$ ($p = .15$)	$\mathbf{-6.3\%}$ ($\mathbf{p = .009}$)	-1.4% ($p = .48$)

†At 1%, `leverage` is non-significant on NDCG@5 but significantly improves the weighted score ($+4.4\%$, $p = .014$).

collision-count diagnostic for transparency (Appendix C.5); it is not used to support the corrected downstream conclusions.

5.3 Benign growth exposes the underlying trust-boundary sensitivity

The preceding results raise a stronger question: is this instability specific to adversarially chosen items, or is the OPQ fit itself highly sensitive to ordinary catalog growth? We hold out 700 legitimate catalog items and compare adding k of them back with adding the same number of attacker-chosen items. Structurally, the two are nearly indistinguishable. At $k = 6$, benign insertion yields tuple ARI 0.433 versus 0.438–0.453 for the three attack strategies; at $k = 121$, the corresponding values are 0.404 versus 0.384–0.397. This behavior persists across all tested k and on both Beauty and Toys.

The downstream comparison leads to the same caution. At $k = 121$, benign growth reduces NDCG@10 by 10.9% ($p = .125$), within the 8.8–17.7% degradation range of the malicious conditions; only cluster-targeted and leverage reach significance. Thus, structural drift alone is not a malicious-input signature. More broadly, these results expose a sensitive fit-set trust boundary in which legitimate catalog evolution can produce changes comparable to adversarial insertion. Full matched controls are reported in Appendix D.

5.4 Defense evaluation

We finally test whether the fit-set vulnerability can be mitigated. Table 4 first compares six anomaly-detection baselines with frozen-rotation warm-started refitting on Beauty under `cluster_targeted` poisoning at $\beta = 0.25$. None of the six filtering baselines restores the clean tokenizer structure: resulting ARI ranges from 0.273 to 0.457, compared with the undefended 0.432 and the clean-refit ceiling of 1.000. Even methods with relatively strong poison-detection accuracy do not necessarily preserve the fitted tokenizer, indicating that item-level detection and tokenizer stability are distinct objectives.

The fit-aware admission gate addresses a complementary failure mode by screening candidate additions before refitting. Across Beauty, Toys and Games, and Sports and Outdoors, the original

Table 4: Defense comparison on Beauty under `cluster_targeted` poisoning at $\beta = 0.25$. Filtering baselines refit OPQ after filtering; frozen rotation constrains the refit. Clean-refit ARI is 1.000.

Method	ARI	Method	ARI
Undefended	0.432	LOF	0.457
Centroid-distance trim	0.449	Isolation Forest	0.455
Spectral Signatures	0.378	MCD robust covariance	0.273
DAO	0.431	Frozen-rotation warm-start	$\mathbf{0.987}$

random, `cluster_targeted`, and leverage attacks are strongly separated from held-out legitimate growth, with ROC AUC at least 0.965. However, an adaptive adjacent-mimicry attack exposes a genuine evasion region: at jitter 0.05, AUC falls to 0.623–0.830 across the three catalogs, with only 24–48% TPR at held-out FPR $\leq 5\%$. Admission screening is therefore useful but not sufficient by itself.

Frozen-rotation refitting provides the complementary stabilization mechanism. Across the tested poisoning budgets it maintains ARI 0.998, 0.995, 0.990, and 0.987 at 1%, 5%, 10%, and 25%, respectively, and remains between 0.980 and 0.987 across the three attack geometries. More importantly, when applied to batches that evade the admission gate, full-tuple ARI remains approximately 0.97–0.99 across all three catalogs, comparable to matched legitimate growth. Thus, the two mechanisms address different failure modes: the admission gate rejects updates that are distinguishable from ordinary growth, while constrained refitting limits structural movement for updates that are admitted.

The stabilization is not absolute. When the full 25% `cluster_targeted` budget is distributed over ten sequential refits, ARI falls from the single-shot 0.987 to 0.945 at the one-iteration operating point, while matched legitimate growth reaches 0.971. Full detection statistics, operating curves, adaptive-evasion results, downstream checks, severity and strategy sweeps, and cross-dataset evaluation are reported in Appendix E.

6 Discussion and limitations

Our results separate structural vulnerability from downstream harm: OPQ refitting is highly sensitive to fit-set composition, but the magnitude of Semantic-ID corruption does not reliably predict recommendation effects. Matched benign growth shows that comparable drift can arise without malicious insertion, explaining why item-level filtering is insufficient. Admission screening and frozen-rotation refitting address complementary failure modes, but adaptive evasion and drip-feed updates show that neither provides absolute protection.

Our conclusions are limited to OPQ-based Semantic-ID tokenizers; learned residual tokenizers such as the original TIGER require separate evaluation. The realistic text attack and detailed defense sweeps are narrower than the structural and downstream study, and several downstream cells use three seeds while the baseline and frozen-rotation severity/strategy sweeps use single trials. Frozen-rotation also damps legitimate catalog evolution and weakens under repeated refits, while the admission gate has an adaptive evasion region. Stronger adaptive attacks, deployment-specific calibration, and the long-term balance between tokenizer stability and adaptation remain open.

7 Conclusion

We identify the tokenizer fit set as an overlooked trust boundary in OPQ-based Semantic-ID recommendation. Fit-set poisoning reliably changes the Semantic IDs of legitimate items across catalogs, recommender configurations, attack conditions, and text encoders, but the resulting structural corruption does not reliably predict the direction or magnitude of downstream recommendation effects. Matched benign growth further shows that this sensitivity is a broader property of OPQ refitting rather than a uniquely malicious signature. Existing item-level anomaly detectors fail to restore the tokenizer structure, whereas directly constraining the refitting mechanism maintains ARI ≥ 0.98 across the tested single-refit attack severities and geometries. These results motivate treating tokenizer fitting itself as part of the security boundary of generative recommendation systems.

AI USE STATEMENT

Generative AI tools were used only for language editing and improving the clarity and readability of the manuscript. They were not used to generate the research idea, formulate hypotheses, design the methodology or experiments, implement the experimental pipeline, analyze data, or interpret the results. All scientific content, experimental decisions, claims, and conclusions were developed and verified by the authors. The authors reviewed all AI-assisted language edits and take full responsibility for the final manuscript.

ETHICS STATEMENT

We use publicly available Amazon 5-core datasets for Beauty, Toys and Games, and Sports and Outdoors, and collect no new personal data. All poisoning experiments were conducted offline against local research pipelines and were never deployed against a production recommender, marketplace, real account, or user. This work studies a potential security weakness in tokenizer fitting with the aim of understanding and improving the robustness of Semantic-ID recommendation systems. We do not release a ready-to-deploy attack against any production system.

REPRODUCIBILITY STATEMENT

All experiment scripts reuse the target pipeline's own tokenizer-fitting, embedding, and training routines unmodified. Appendix A reports the tokenizer and poisoning configurations, evaluated text encoders, baseline defenses, frozen-rotation implementation details, model architecture, and training settings. The rotation matrices, IVF coarse centroids, and PQ codebooks used by frozen-rotation refitting are extracted directly from a trained `faiss` index through its public API and verified against `faiss`'s own encoding before use (Section 4.4.2), rather than reimplemented from an algorithmic description. We use the SNAP Amazon 5-core Beauty, Toys_and_Games, and Sports_and_Outdoors datasets (`https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/`). Cross-catalog structural and defense experiments reuse the corresponding pipeline, encoder, preprocessing, and experimental protocol across the three catalogs.

REFERENCES

Vincenzo Agate, V. P. Barreca, Giuseppe Lo Re, Marco Morana, and A. Virga. Black-box poisoning attacks on sequential recommender systems via cross-domain profiles. In *Research Challenges in Information Science (RCIS 2026)*, 2026. doi: 10.1007/978-3-032-26836-5_28. Poisoned sequences crafted by transferring real user sequences from a source domain.

Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC) Workshop on Linguistics Coreference*, volume 1, pp. 563–566, 1998. Introduces the B-cubed precision/recall/F1 metric for comparing partitions dominated by singleton clusters, the regime in which coreference mention sets and (in this paper) near-collision-free semantic-ID catalogs both live. Applied here, verified on this paper's own real degenerate trials, to fix a specific numerical instability in Adjusted Rand Index (Section 4.1).

Matan Ben-Tov and Mahmood Sharif. GASLITEing the retrieval: Exploring vulnerabilities in dense embedding-based search. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2025. arXiv:2412.20953. Gradient corpus poisoning of dense retrievers; basis for the HotFlip-style adversarial-suffix attack used here.

Alexander Branch, Omead Pooladzandi, Radin Khosraviani, Sunay Gajanan Bhat, Jeffrey Jiang, and Gregory Pottie. Purevq-gan: Defending data poisoning attacks through vector-quantized bottlenecks. *arXiv preprint arXiv:2509.25792*, 2025. Confirms discrete VQ/PQ bottlenecks block gradient end-to-end attacks, motivating the pre-quantization / fit-set attack surface used throughout.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.

Paul-Alexandru Chirita, Wolfgang Nejdl, and Cristian Zamfir. Preventing shilling attacks in online recommender systems. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management (WIDM)*, pp. 67–74, 2005. doi: 10.1145/1097047.1097061. Introduces RDMA (Rating Deviation from Mean Agreement) and related profile-level shilling-detection statistics for explicit-rating collaborative filtering. Adapted here to implicit sequential data (no ratings exist in this pipeline) as an item-rarity profile score, an adaptation stated explicitly, not a literal port.

Antonio Emanuele Cinà, Alessandro Torcinovich, and Marcello Pelillo. A black-box adversarial attack for poisoning clustering. *arXiv preprint arXiv:2009.05474*, 2020. General k-means fit-set poisoning; basis for the core hypothesis applied here to OPQ for the first time to our knowledge.

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: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. Influence function based data poisoning attacks to top-n recommender systems. *arXiv preprint arXiv:2002.08025*, 2020. WWW 2020. Classical fake-profile shilling; traditional recommenders, not semantic-ID.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 878–891, 2022.

Yan Feng, Bin Chen, Tao Dai, and Shu-Tao Xia. Adversarial attack on deep product quantization network for image retrieval. In *Proceedings of AAAI 2020*, 2020. arXiv:2002.11374. PQ cell-boundary adversarial attacks, image retrieval.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. SIGIR '26, pp. 369–379, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725999. doi: 10.1145/3805712.3809641. URL https://doi.org/10.1145/3805712.3809641.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6894–6910, 2021.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013a.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(4):744–755, 2013b. Origin of the OPQ rotation-plus-codebook fit this paper's tokenizer and second defense both build on; a CVPR 2013 conference version precedes this journal version.

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. SPECTRE: Defending against backdoor attacks using robust statistics. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 4129–4139, 2021.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. Generating long semantic IDs in parallel for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025. RPG: OPQ-based semantic IDs of up to 64 unordered tokens per item.

Hanxun Huang, Sarah Erfani, Yige Li, Xingjun Ma, and James Bailey. Detecting backdoor samples in contrastive language image pretraining. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2502.01385. Dimensionality-Aware Outlier (DAO) detection, a

domain-agnostic local-density-ratio statistic weighted by local intrinsic dimensionality; applied here at the paper's own default $k = 16$.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Runze Li, Di Jin, Xiaobao Wang, Dongxiao He, Bingdao Feng, and Zhen Wang. Single-node trigger backdoor attacks in graph-based recommendation systems. *arXiv preprint arXiv:2506.08401*, 2025. One injected node sufficient to expose target items to $\geq$50% of targeted users in 99% of cases.

Chong Liu, Defu Lian, Min Nie, and Hu Xia. Online optimized product quantization. In *Proceedings of the 20th IEEE International Conference on Data Mining (ICDM)*, pp. 362–371, 2020. Incrementally updates both the OPQ rotation and PQ codebooks for streaming data, motivated by index-maintenance efficiency rather than trust. Section 4.4.2 freezes the rotation entirely rather than updating it online, and motivates capped-iteration codebook updates by poisoning resistance, not throughput.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)*, 2008.

Zhao Liu, Yichen Zhu, Yiqing Yang, Xiao Lv, Guoping Tang, Rui Huang, Qiang Luo, Ruiming Tang, Kun Gai, and Guorui Zhou. DiffGRM: Diffusion-based generative recommendation model. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*, pp. 5853–5864, 2026.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.

Bhaskar Mehta and Wolfgang Nejdl. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction*, 19:65–97, 2009. doi: 10.1007/s11257-008-9050-4. Extends Mehta et al. (2007)'s robust/clustering-based shilling detection; cited alongside it as the basis for this paper's PCA/clustering-analog baseline.

Bhaskar Mehta, Thomas Hofmann, and Wolfgang Nejdl. Robust collaborative filtering. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys)*, pp. 49–56, 2007. doi: 10.1145/1297231.1297240. Correlation/covariance-clustering detection of tightly-correlated attack profiles in a user-item rating matrix. Adapted here to implicit data via pairwise Jaccard item-overlap among profiles, in place of rating correlation.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1864–1874, 2022a.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9844–9855, 2022b.

Liangbo Ning, Wenqi Fan, and Qing Li. Exploring backdoor attack and defense for llm-empowered recommendations. *arXiv preprint arXiv:2504.11182*, 2025. Proposes BadRec, trigger-phrase backdoors against LLM-based RecSys, and a defense (P-Scanner).

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. TIGER: the semantic-ID generative-retrieval architecture family this pipeline (DiffGRM-style) belongs to.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1102–1121, 2023.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, 2024. LETTER: a gradient-trained, learnable residual-quantization tokenizer for generative recommendation, jointly optimized with the recommender rather than fit once and frozen; representative of the gradient-trained tokenizer family this paper's OPQ-based scope excludes.

Zihao Wang, Tianhao Mao, Xiaofeng Wang, Di Tang, and Xiaozhong Liu. Indirectad: Practical data poisoning attacks against recommender systems for item promotion. *arXiv preprint arXiv:2511.05845*, 2025. Black-box promotion via a trigger item and induced co-occurrence, evading shilling detectors this pipeline has none of.

Donna Xu, Ivor W. Tsang, and Ying Zhang. Online product quantization. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 30(11):2185–2198, 2018. arXiv:1711.10775. Incremental codebook updates for plain PQ (no learned rotation) on streaming data, budget-constrained to update only part of the codebook per batch; motivates Liu et al. (2020)'s later extension to OPQ's rotation.

Hengtong Zhang, Yaliang Li, Bolin Ding, and Jing Gao. Practical data poisoning attack against next-item recommendation. In *Proceedings of The Web Conference 2020 (WWW)*, pp. 2458–2464, 2020. doi: 10.1145/3366423.3379992. LOKI: gradient-based generation of malicious user-behavior sequences to promote a target item in next-item/sequential recommendation, on conventional (non-semantic-ID) sequential recommenders. Closest published sequence-poisoning prior art to this paper's threat model; this paper differs by targeting a semantic-ID generative recommender and by using unoptimized classical bandwagon/random fillers rather than gradient-optimized sequences, to isolate whether the architecture is vulnerable to the simplest possible version of the attack before asking whether an optimized one would do better.

Qian Zhang, Lech Szymanski, Haibo Zhang, and Jeremiah D Deng. How reliable are semantic-id tokenizer comparisons in generative recommendation? *arXiv preprint arXiv:2605.25330*, 2026.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Qian Chen, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. Kalm-embedding-v2: Superior training techniques and data inspire a versatile embedding model. In *International Conference on Learning Representations (ICLR)*, 2026a. arXiv:2506.20923. Bidirectional-attention, mean-pooled embedding model built on Qwen2-0.5B.

Yuchuan Zhao, Tong Chen, Junliang Yu, Kai Zheng, Lizhen Cui, and Hongzhi Yin. Diversity-aware dual-promotion poisoning attack on sequential recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1634–1644, 2025. doi: 10.1145/3726302.3729955. Optimizes poisoning sequences

for target-item promotion on sequential recommenders while explicitly preserving normal recommendation relevance and improving stealthiness against detection — the same collateral-quality-preservation property this paper observes empirically as a byproduct rather than an optimization objective (Table 15). From the same research group as Zhao et al. (2026b), which extends this line with adversarial item-text modification.

Yuchuan Zhao, Tong Chen, Junliang Yu, Zongwei Wang, Lizhen Cui, and Hongzhi Yin. Prompt-unknown promotion attacks against llm-based sequential recommender systems. *arXiv preprint arXiv:2604.23640*, 2026b. Accepted SIGIR 2026. Combines adversarial item-text revision with fake interactions against LLM-based sequential recommenders.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of EMNLP 2023*, 2023. Surrogate-transferable corpus poisoning against dense text retrieval, not recommendation.

A Experimental Setup and Full Configuration

All three downstream pipelines use `sentence-t5-base` representations. DiffGRM (Liu et al., 2026) uses `OPQ4,IVF1,PQ4x8`, corresponding to four Semantic-ID digits with 256 codes per digit, after PCA to 256 dimensions and post-PCA L_2 normalization.

Our TIGER-style pipeline follows the autoregressive encoder–decoder and decoding paradigm of TIGER (Rajput et al., 2023), but replaces TIGER's original residual-quantization tokenizer with the same OPQ-derived Semantic IDs used by DiffGRM. Thus, DiffGRM and the TIGER-style pipeline share the same `OPQ4,IVF1,PQ4x8` tokenizer and therefore the same tokenizer-level structural results; they differ in the downstream recommendation architecture.

RPG (Hou et al., 2025) uses `OPQ32,IVF1,PQ32x8` on Beauty and `OPQ16,IVF1,PQ16x8` on Toys and Games and Sports and Outdoors. Its representations are reduced to 128 dimensions using whitened PCA without post-PCA L_2 normalization. Poison directions are constructed in the normalized space used to define the attack strategies and then rescaled to the catalog's mean embedding norm before RPG tokenizer fitting.

Tokenizer-level structural experiments cover all nine recommender–catalog combinations. The common downstream comparison trains all nine cells under cluster-targeted poisoning at $\beta = 0.25$. DiffGRM on Beauty additionally includes the complete four-budget sweep. Seed counts for this sweep vary by budget and, at 10%, by attack strategy, as summarized below.

Table 5: Downstream clean-versus-poisoned training coverage. For DiffGRM/Beauty, the 1% and 5% conditions use three paired seeds. At 10%, cluster-targeted uses five seeds while leverage and random use three; at 25%, all three strategies use five seeds.

Recommender	Beauty	Toys and Games	Sports and Outdoors
DiffGRM	3/5 seeds, full sweep	three seeds, one condition	three seeds, one condition
TIGER-style	three seeds, one condition	three seeds, one condition	three seeds, one condition
RPG	three seeds, one condition	three seeds, one condition	three seeds, one condition

For the nine-cell cross-pipeline comparison, each downstream recommender uses its pipeline-specific training routine. Architecture and training hyperparameters are listed below for all three pipelines: DiffGRM, the TIGER-style pipeline, and RPG. Within each pipeline, clean and poisoned runs use identical settings and differ only in the Semantic-ID file.

Shared DiffGRM/TIGER-style OPQ tokenization

The shared OPQ tokenizer contains 4 codebooks of size 256 each (vocabulary size $= 3 + 4 \times 256 = 1027$). The primary text encoder is Ni et al. (2022a) (768-dimensional raw representations), followed by PCA to 256 dimensions with whitening, post-PCA L_2 normalization, and OPQ rotation.

POISONING STRATEGIES

`random` samples Gaussian noise followed by normalization. `cluster_targeted` applies Gaussian jitter with scale 0.08 around a target-cluster centroid followed by normalization. `leverage` applies jitter with scale 0.05 around a point 3.0 units along the catalog's leading principal component, followed by normalization. Attack budgets are $\beta \in \{0.01, 0.05, 0.10, 0.25\}$, corresponding to $1\%, 5\%, 10\%$, and 25% of catalog size. Structural experiments use three tokenizer refits per (strategy, budget) cell unless otherwise noted.

ENCODERS EVALUATED IN THE GENERALIZATION STUDY

Encoder	Venue	Raw dim	Pooling
Zhao et al. (2026a)	ICLR 2026	896	native
Su et al. (2023)	ACL Findings 2023	768	native
Ni et al. (2022b)	EMNLP 2022	768	native
Ni et al. (2022a)	ACL Findings 2022	768	native
Feng et al. (2022)	ACL 2022	768	native
Izacard et al. (2022)	TMLR 2022	768	mean, uniform
Gao et al. (2021)	EMNLP 2021	768	mean, uniform
Song et al. (2020) (via Reimers & Gurevych, 2019)	NeurIPS 2020 / EMNLP 2019	768	native
Wang et al. (2020) (via Reimers & Gurevych, 2019)	NeurIPS 2020 / EMNLP 2019	384	native
Devlin et al. (2019)	NAACL 2019	768	mean, uniform

"Native" pooling uses each checkpoint's published sentence-embedding pooling head. "Mean, uniform" denotes the shared mean-pooling recipe applied identically to the three checkpoints released without a dedicated sentence-embedding pooling head.

BASELINE DEFENSES

LOF (Breunig et al., 2000): local outlier factor with $k = 20$ neighbors, excluding the 25% most negative-scoring samples within the candidate fit set.

Centroid-distance trim: exclude the 25% of candidate-fit-set items farthest in Euclidean distance from the candidate-set mean.

Isolation Forest (Liu et al., 2008): scikit-learn default hyperparameters, excluding the 25% lowest `score_samples` values.

Spectral Signatures (Tran et al., 2018): score each candidate-fit-set point by its squared projection onto the top right singular vector of the centered representation matrix and exclude the top-scoring 25%. The original method scores examples within classifier classes; here it is adapted to the class-free pooled fit set because the tokenizer fit set has no analogous label structure.

MCD robust covariance (Rousseeuw & Van Driessen, 1999): Minimum Covariance Determinant on a 30-component PCA projection of the candidate fit set using the scikit-learn default support fraction, excluding the 25% with largest robust Mahalanobis distance. This is the robust-covariance primitive used by the QUE-score component of SPECTRE (Hayase et al., 2021).

DAO (Huang et al., 2025): Dimensionality-Aware Outlier scoring with the paper's default $k = 16$ neighbors, using local intrinsic dimensionality estimated with the Levina–Bickel MLE estimator, and excluding the 25% highest-scoring samples.

All six baselines use a 25% exclusion budget, matching the poisoning budget in the principal defense comparison, with no detection threshold tuned to this dataset.

FROZEN-ROTATION WARM-START DEFENSE

The rotation R and IVF coarse centroid are frozen at their values from a real `faiss OPQ4,IVF1,PQ4x8` index trained once on the trusted-only fit set (Ge et al., 2013b). The extracted components were verified against `faiss`'s own encoding before use, with $200/200$ sampled item codes matching exactly.

Each of the four PQ codebooks, containing 256 centroids over 64-dimensional subvectors, is warm-started using scikit-learn `KMeans`, with `init` set to the centroids from the preceding trusted deployment, `n_init=1`, and `algorithm='lloyd'`.

For the single-shot defense sweeps (Tables 16–18), `max_iter=1` and one refit are used. An iteration-budget sweep over `max_iter`$\in \{0, 1, 2, 3, 5, 10, 25\}$ identifies `max_iter=1` as the operating point used in the single-shot experiments.

For the multi-cycle drip-feed stress test (Table 19), the candidate additions are distributed across ten sequential refit cycles. We evaluate `max_iter`$\in \{1, 2, 5\}$ per cycle. The complete $\beta = 0.25$ poisoning budget, or the matched 5% organic-growth budget, is split evenly across cycles using `numpy.array_split`. Each cycle warm-starts its codebooks from the immediately preceding cycle's output centroids rather than resetting to the original trusted centroids.

DiffGRM model architecture

DiffGRM uses one encoder layer and four decoder layers, with four attention heads, embedding dimension 256, inner dimension 1024, and a shared decoder output embedding. Dropout is 0.1 and label smoothing is 0.2. Guided masking uses least-confidence selection with maximum softmax probability as the confidence metric. The model contains 5,602,048 trainable parameters.

DiffGRM training

DiffGRM uses batch size 1024, learning rate 0.01 with a cosine schedule and 10,000-step warmup, gradient clipping at 1.0, mixed precision, and a leave-one-out split without a sliding window.

The pilot configuration uses seed 42, at most 15 epochs, evaluation beginning at epoch 10, and patience 5. The standard three-seed runs use seeds $\{1, 2, 3\}$, at most 60 epochs, evaluation beginning at epoch 20, and patience 15.

The five-seed strengthening experiment (Table 12) uses seeds $\{1, \ldots, 5\}$, at most 100 epochs, evaluation beginning at epoch 20, and patience 15. For the Beauty budget sweep, the 1% and 5% conditions use three paired seeds. At 10%, cluster-targeted uses five seeds while leverage and random use three; at 25%, all three attack strategies use five paired seeds.

Inference uses beam size 128 with confidence-guided decoding on the test split.

TIGER-style model architecture

Our TIGER-style pipeline is an autoregressive encoder–decoder that generates the four Semantic-ID digits of the next item one digit at a time, conditioned on the user's item history. It uses one encoder layer and four decoder layers, four attention heads, embedding dimension 256, inner dimension 1024, a shared decoder output embedding, LayerNorm ($\epsilon = 10^{-5}$), dropout 0.1 for embedding, attention, and residual connections, and label smoothing 0.2. The item history is truncated to the 50 most recent items. It consumes the same `OPQ4,IVF1,PQ4x8` Semantic IDs as DiffGRM (vocabulary size 1027), so the TIGER-style clean and poisoned runs use the same tokenizations as the corresponding DiffGRM runs. The model contains 5,602,304 trainable parameters and uses the same architecture on all three catalogs.

TIGER-style training

Training uses AdamW with learning rate 0.01 and weight decay 0, a cosine schedule with 10,000-step warmup, gradient clipping at 1.0, mixed precision, batch size 1024, and a leave-one-out split without a sliding window. Runs use at most 100 epochs, with evaluation beginning at epoch 20 and early-stopping patience 15. Checkpoints are selected by the validation score $0.8\,\mathrm{NDCG@10} + 0.2\,\mathrm{Recall@10}$; validation uses beam size 32. The reported three-seed comparison uses seeds $\{1, 2, 3\}$. Test-time inference uses beam search with beam size 128 and returns the top 10 items.

Table 6: Structural change under poisoning on real Beauty embeddings across 36 trials. ARI denotes full-tuple ARI; collision counts are totals across the 36 poisoned trials.

Condition	Tuple ARI	New collisions	Broken collisions
Determinism control (no poisoning)	1.0000	0 (exact)	0 (exact)
Poisoned, all 36 trials	0.4474 ± 0.0326	54 total	12 total

RPG MODEL ARCHITECTURE

RPG (Hou et al., 2025) uses the official model and training implementation, with our precomputed clean or poisoned Semantic-ID files supplied in place of RPG's tokenizer output. Its backbone is a GPT-2-style decoder with two layers, four attention heads, embedding dimension 448, inner dimension 1024, and the `gelu_new` activation. Embedding and attention dropout are 0.5, residual dropout is 0, and LayerNorm uses $\epsilon = 10^{-12}$. The item history is truncated to 50 items. Each item is represented by 32 Semantic-ID tokens on Beauty and 16 on Toys and Games and Sports and Outdoors, with 256 codes per codebook. The model contains $13{,}581{,}824$ trainable parameters on Beauty and $8{,}528{,}384$ on Toys and Games and Sports and Outdoors.

RPG TRAINING

Hyperparameters follow RPG's per-dataset settings. Beauty uses learning rate 0.01, temperature 0.03, 32 codebooks, 20 beams, 200 graph edges, and 3 propagation steps. Toys and Games uses learning rate 0.003, temperature 0.03, 16 codebooks, 200 beams, 20 graph edges, and 3 propagation steps. Sports and Outdoors uses learning rate 0.003, temperature 0.03, 16 codebooks, 100 beams, 30 graph edges, and 5 propagation steps. All other settings follow RPG's defaults: AdamW with weight decay 0, a cosine schedule with $10{,}000$-step warmup, gradient clipping at 1.0, training batch size 256, evaluation batch size 32, at most 150 epochs with validation every epoch, early-stopping patience 20, checkpoint selection by validation NDCG@10, and a leave-one-out split. The reported three-seed comparison uses seeds $\{1, 2, 3\}$. Test-time decoding uses RPG's graph-based decoding with the per-dataset settings above and returns the top 10 items.

B EXTENDED STRUCTURAL-VULNERABILITY ANALYSES

This section provides additional experiments supporting the structural-vulnerability results in Section 5.1, including synthetic controls, detailed Beauty results, localization, text-realizable poisoning, cross-encoder generalization, and additional cross-catalog analysis. Unless otherwise stated, ARI values in this appendix refer to the full Semantic-ID tuple induced by the shared four-digit DiffGRM/TIGER-style OPQ tokenizer. The main paper additionally reports mean per-digit ARI to enable comparison with RPG's longer Semantic IDs.

B.1 SYNTHETIC SANITY CHECK AND DETAILED BEAUTY RESULTS

On a synthetic 15-cluster catalog, poisoning the fit set at every tested budget ($\beta \in \{0.01, 0.05, 0.10, 0.25\}$) and under every strategy produces zero sampled-pair collision changes, $\mathrm{NMI} \geq 0.999994$, and B^3-F1 ≥ 0.9999 across all 36 trials. Full-tuple ARI averages 0.947, but two trials enter the near-singleton ARI regime discussed in Section 4.1 and return exactly 0.000000 despite NMI and B-cubed confirming near-perfect agreement. With a code space of 256^4 possible tuples for only $12{,}000$ well-separated items, this synthetic setting is therefore unrealistically easy. Repeating the same experiment on the real SentenceT5-encoded, PCA-reduced Beauty catalog produces substantial structural reorganization instead.

Every poisoned trial exhibits nonzero collision churn. Even at $\beta = 0.01$, corresponding to only 121 injected items against $12{,}101$ legitimate Beauty items, the tokenizer substantially reorganizes the assignments of untouched catalog items. Because the clean refit is deterministic, this drift cannot be attributed to refitting noise.

Table 7: Mean full-tuple ARI by poisoning strategy and budget on real Beauty embeddings.

Budget	random	cluster_targeted	leverage
1%	0.4601	0.4770	0.4508
5%	0.4829	0.4417	0.4298
10%	0.4781	0.4672	0.4147
25%	0.4217	0.4452	0.3998

Table 8: Cosine similarity to the target centroid achieved by text-based and idealized attack constructions.

Attacker	Cosine sim.	Notes
Real cluster members	0.090	Natural catalog items
Naive copycat text	0.242	Two descriptions concatenated
Gradient-optimized suffix	0.469	40 iterations, still rising
Idealized raw-vector injection	0.575	Embedding-space reference

B.2 Localization analysis

We test whether cluster-targeted poisoning preferentially corrupts the region toward which poison is placed. We divide the catalog into 15 k-means clusters and compare the targeted cluster's ARI with that of the remaining catalog over $n = 30$ independent trials. The target-minus- other-clusters gap has mean -0.007 and is not significantly different from zero ($t = -0.190$, $p = 0.851$); only 14 of the 30 trials have a positive gap. Target-cluster ARI ranges from 0.113 to 0.714, within the range produced for arbitrary clusters by untargeted random and leverage poisoning.

The weak localization is consistent with the geometry of the processed catalog. Real cluster members have mean cosine similarity only 0.090 to their own centroid, barely above the approximately 0.06 expected from noise on a 256-dimensional hypersphere. Retesting with Amazon category metadata gives the same qualitative conclusion. The "Body" category (939 items) has member-to-centroid similarity 0.139 ± 0.090, yet category-targeted poisoning produces no significant localization effect relative to the comparison condition ($t = -0.881$, $p = 0.399$). Thus, although poisoning strongly reorganizes the tokenizer globally, we find no evidence that this construction lets the attacker reliably select which catalog region is damaged.

B.3 Text-realizable poisoning

The embedding-space attacks used in the main structural experiments are idealized because a realistic attacker controls item text rather than an arbitrary embedding vector. We therefore test two text-based attackers against the public text encoder while targeting the largest real cluster ($n = 2{,}198$).

The gradient-optimized suffix reaches approximately 82% of the idealized attack's target similarity ($0.469/0.575$), while even naive copycat text substantially exceeds the natural baseline. The optimized suffix uses valid vocabulary but is visibly nonsensical, reflecting the usual potency-versus-fluency trade-off.

More importantly, we evaluate text-derived poison through the actual tokenizer rather than treating cosine similarity as a proxy. We generate a full naive-mimicry batch ($n = 3{,}025$, $\beta = 0.25$) and a HotFlip batch ($n = 40$, $\beta = 0.0033$), encode them with the same public encoder, and pass the resulting representations through the unmodified fit-set-poisoning pipeline.

Both text-derived constructions therefore reach the same broad structural-corruption regime as idealized embedding-space poisoning despite their very different attack budgets. The naive-mimicry artifact was additionally carried through five-seed downstream training, yielding mean NDCG@5 of 0.02508 and weighted score of 0.03170, compared with 0.02452 and 0.03149 for clean training. Its downstream effect is therefore approximately neutral despite substantial tokenizer-level corruption.

Table 9: Text-derived poison passed through the real tokenizer-fitting pipeline at batch scale. ARI denotes full-tuple ARI.

Attacker	n poison	Budget	Tuple ARI
Naive text mimicry	3,025	25%	0.4487
HotFlip optimized	40	0.33%	0.4357
Idealized raw-vector reference	—	—	0.40–0.48

Table 10: Full-tuple structural corruption across ten text encoders (σ =`cluster_targeted`, $\beta = 0.25$; mean $\pm$ std over three repeats).

Encoder	Venue	Dim	Determ. ARI	Baseline coll.	Poisoned ARI
Zhao et al. (2026a)	ICLR 2026	896	1.0000	0.020%	0.460 ± 0.029
Su et al. (2023)	ACL Findings 2023	768	1.0000	0.016%	0.428 ± 0.013
Ni et al. (2022b)	EMNLP 2022	768	1.0000	0.018%	0.504 ± 0.030
Ni et al. (2022a)	ACL Findings 2022	768	1.0000	0.002%	0.415 ± 0.011
Feng et al. (2022)	ACL 2022	768	1.0000	0.016%	0.577 ± 0.021
Izacard et al. (2022)	TMLR 2022	768	1.0000	0.012%	0.615 ± 0.040
Gao et al. (2021)	EMNLP 2021	768	1.0000	0.008%	0.580 ± 0.018
Song et al. (2020)	NeurIPS 2020	768	1.0000	0.016%	0.513 ± 0.018
Wang et al. (2020)	NeurIPS 2020	384	1.0000	0.020%	0.536 ± 0.008
Devlin et al. (2019)	NAACL 2019	768	1.0000	0.008%	0.625 ± 0.029

B.4 Generalization across text encoders

We repeat cluster-targeted poisoning at $\beta = 0.25$ across ten independently published text encoders while holding the catalog, poisoning protocol, and four-digit OPQ tokenizer fixed.

All ten encoders retain exact clean-refit reproducibility but exhibit substantial poisoning-induced drift. Poisoned full-tuple ARI ranges from 0.415 to 0.625. We observe no monotonic relationship with publication recency, representation dimensionality, or sentence-embedding specialization. The supported conclusion is therefore that the vulnerability persists across the tested encoders, not that its severity can be predicted from these properties.

B.5 Detailed second-catalog results

We additionally repeat the complete strategy-by-budget grid on Toys and Games (11,924 items) using the same Sentence-T5 representations and the shared four-digit DiffGRM/TIGER-style OPQ configuration used in the corresponding Beauty experiment.

All twelve poisoned cells remain substantially below the deterministic clean ceiling. The aggregate poisoned mean full-tuple ARI is 0.453 ± 0.048, close to Beauty's corresponding 0.445 ± 0.041, providing an independent catalog-level replication of the structural effect.

C Extended Downstream Evaluation

This section provides the detailed downstream results supporting Section 5.2, including the full DiffGRM/Beauty per-metric evaluation, provenance of the targeting correction, detailed budget sensitivity, an additional Toys and Games result, and a historical collision-count diagnostic associated with the superseded targeting artifact.

C.1 Detailed DiffGRM results on Beauty

We first examine all five recommendation metrics for DiffGRM on Beauty at $\beta = 0.25$. The `cluster_targeted` condition reported here uses genuine k-means cluster targeting. The `leverage` and `random` constructions do not depend on the target-cluster assignment.

Table 11: Mean full-tuple ARI by strategy and budget on Toys and Games (three repeats per cell); clean-refit ARI is 1.0000.

Budget	random	cluster_targeted	leverage
1%	0.4748	0.4912	0.4907
5%	0.4686	0.4992	0.4813
10%	0.4482	0.4605	0.4189
25%	0.4439	0.3981	0.3565

Table 12: DiffGRM/Beauty test metrics at $\beta = 0.25$, mean $\pm$ std over five paired seeds. Reported p-values are from paired t-tests against the corresponding clean runs.

Metric	Clean	`cluster_targeted`	`leverage`	`random`
Recall@5	0.0350 ± 0.0007	$0.0354 \pm 0.0011^{p=.56}$	$0.0335 \pm 0.0005^{p=.017}$	$0.0347 \pm 0.0008^{p=.56}$
NDCG@5	0.0245 ± 0.0005	$0.0251 \pm 0.0007^{p=.15}$	$0.0230 \pm 0.0005^{p=.009}$	$0.0242 \pm 0.0006^{p=.48}$
Recall@10	0.0450 ± 0.0015	$0.0459 \pm 0.0012^{p=.16}$	$0.0434 \pm 0.0024^{p=.23}$	$0.0445 \pm 0.0019^{p=.68}$
NDCG@10	0.0278 ± 0.0005	$0.0285 \pm 0.0006^{p=.08}$	$0.0262 \pm 0.0011^{p=.046}$	$0.0274 \pm 0.0010^{p=.51}$
Weighted score	0.0312 ± 0.0007	$0.0320 \pm 0.0007^{p=.09}$	$0.0296 \pm 0.0013^{p=.075}$	$0.0308 \pm 0.0011^{p=.55}$

With correct targeting, `cluster_targeted` has positive point estimates on all five metrics but is significant on none ($p = .08$–$.56$). `leverage` significantly degrades Recall@5, NDCG@5, and NDCG@10, whereas `random` is statistically indistinguishable from clean on all five metrics. Thus, even at a fixed budget and encoder, comparable tokenizer-level corruption can translate into different downstream outcomes.

C.2 Targeting correction and provenance

An earlier version of the downstream artifact generator passed `victim_assign=np.zeros(m)` and `target_cluster=0` to the shared poisoning routine. In that artifact family, these arguments selected the whole-catalog mean rather than a genuine sub-cluster centroid. The resulting target vector had norm approximately 0.007–0.008, roughly 150–170$\times$ smaller than the typical norm of the Gaussian perturbation subsequently added. After normalization, the resulting construction therefore behaved similarly to random placement rather than genuine cluster targeting.

We rebuilt the affected `cluster_targeted` artifacts using a genuine target centroid obtained from `KMeans(n_clusters=15, n_init=3, random_state=0)` on a 4,000-item sample, matching the targeting procedure used in the structural experiments. We then reran the affected 10% and 25% conditions. The `random` and `leverage` conditions are unaffected because their constructions do not depend on the target-cluster assignment. All main-paper results use the corrected artifacts wherever cluster targeting is required.

The correction changes the interpretation of the `cluster_targeted` condition. At $\beta = 0.25$, the corrected condition is non-significant on all five metrics (Table 12). We therefore use only the corrected artifacts for the paper's downstream conclusions and do not rely on the superseded targeting result.

C.3 Detailed budget sensitivity

The main paper summarizes NDCG@5 across all four attack budgets. For completeness, Table 13 reports all five metrics at $\beta = 0.10$. The corrected `cluster_targeted` condition uses five paired seeds. The `leverage` and `random` conditions, which were unaffected by the targeting correction, use three paired seeds.

These results reinforce the non-monotonic pattern summarized in Table 3. At $\beta = 0.10$, `cluster_targeted` is statistically indistinguishable from clean on every metric. `leverage` significantly degrades NDCG@5 but not the other four metrics, while `random` remains statistically indistinguishable from clean throughout.

Table 13: DiffGRM/Beauty test metrics at $\beta = 0.10$. Percent changes and paired p-values are computed relative to each condition's matched clean reference. The corrected `cluster_targeted` condition uses five seeds; `leverage` and `random` use three.

Metric	`cluster_targeted`	`leverage`	`random`
Recall@5	0.0357, $+2.0\%^{p=.58}$	0.0317, $-10.7\%^{p=.12}$	0.0346, $-2.5\%^{p=.12}$
NDCG@5	0.0247, $+0.5\%^{p=.85}$	0.0212, $-14.4\%^{p=.024}$	0.0246, $-0.6\%^{p=.74}$
Recall@10	0.0473, $+5.1\%^{p=.14}$	0.0423, $-6.2\%^{p=.47}$	0.0450, $-0.2\%^{p=.94}$
NDCG@10	0.0285, $+2.5\%^{p=.40}$	0.0247, $-11.5\%^{p=.078}$	0.0280, $+0.4\%^{p=.81}$
Weighted score	0.0322, $+3.3\%^{p=.29}$	0.0282, $-10.2\%^{p=.15}$	0.0314, $-0.1\%^{p=.89}$

Table 14: DiffGRM downstream training on Toys and Games under `cluster_targeted` poisoning at $\beta = 0.25$; values are three-seed means.

Condition	NDCG@5	Weighted score	Δ NDCG@5
Clean	0.03184	0.03422	—
Poisoned	0.03246	0.03464	$+2.0\%$

C.4 Additional DiffGRM/Toys result

We additionally train DiffGRM on Toys and Games under corrected `cluster_targeted` poisoning at $\beta = 0.25$. This experiment uses genuine k-means targeting rather than the superseded whole-catalog-mean construction.

Because this experiment uses only three seeds, the $+2.0\%$ NDCG@5 point estimate should not be interpreted as evidence of a reliable improvement. It is consistent with the broader finding that downstream effects are condition-specific even when tokenizer corruption is substantial.

C.5 Superseded collision-count diagnostic

For transparency, we retain one diagnostic performed on the superseded artifact described in Appendix C.2. That artifact produced $1{,}473$ colliding legitimate-item pairs, compared with $1{,}447$ under the corresponding clean fit. To test whether its downstream behavior could be explained solely by this increase, we constructed a non-poisoned control from the clean fit and directly reassigned 26 item codewords to reproduce the same collision-count increase.

Under otherwise identical five-seed training, the collision-matched control achieves NDCG@5 0.0242 ± 0.0010, statistically indistinguishable from clean (0.0245 ± 0.0005; paired t-test, $p = .465$). Because this diagnostic concerns the superseded targeting artifact, it is not used as evidence for any corrected downstream conclusion. It establishes only that collision count alone did not explain the behavior of that earlier artifact.

D Matched Benign-versus-Malicious Catalog Growth

The strong structural drift under poisoning raises an important question: is the observed instability specific to attacker-chosen items, or can comparable changes arise when legitimate items are added to the tokenizer fit set? We test this directly using DiffGRM's four-digit OPQ tokenizer.

On Beauty, we hold out a pool of 700 legitimate catalog items from the tokenizer fit set and use them as a controlled proxy for benign catalog growth. We compare: (A) a base fit with the held-out pool excluded; (B) the base fit plus k legitimate items sampled from the held-out pool; and (C) the base fit plus k attacker-chosen items generated using the `random`, `cluster_targeted`, or `leverage` constructions. All conditions are evaluated on the same base items and use the same addition size k, allowing benign and malicious fit-set changes to be compared under matched perturbation size.

Structural comparison. Benign and malicious additions produce similar levels of structural drift. At $k = 6$, adding legitimate held-out items yields full-tuple ARI 0.433, compared with 0.438–0.453

Table 15: Matched benign-versus-malicious control on DiffGRM/Beauty. Results are paired against the same held-out-pool clean baseline (NDCG@10 $= 0.0296$, Recall@10 $= 0.0484$), with $n = 3$ paired seeds. Bold marks $p < 0.05$ for NDCG@10.

k	Condition	NDCG@10	Recall@10	p
6	benign	0.0280 (-5.3%)	0.0465 (-3.9%)	.173
6	`random`	0.0299 ($+1.0\%$)	0.0489 ($+1.0\%$)	.555
6	`cluster_targeted`	0.0308 ($+4.2\%$)	0.0516 ($+6.5\%$)	.356
6	`leverage`	0.0286 (-3.2%)	0.0486 ($+0.4\%$)	.599
121	benign	0.0264 (-10.9%)	0.0429 (-11.4%)	.125
121	`random`	0.0254 (-14.1%)	0.0422 (-12.7%)	.082
121	`cluster_targeted`	0.0270 (-8.8%)	0.0449 (-7.3%)	**.019**
121	`leverage`	0.0244 (-17.7%)	0.0412 (-15.0%)	**.005**

across the three attack strategies. At $k = 121$, benign addition yields full-tuple ARI 0.404, compared with 0.384–0.397 for the attacks.

The same qualitative behavior holds over the complete tested sequence

$$k \in \{6, 12, 24, 60, 121, 242, 605\},$$

and on both Beauty and Toys and Games. Across these matched comparisons, structural-drift magnitude alone does not cleanly separate legitimate fit-set additions from attacker-chosen additions.

Downstream comparison. We additionally train DiffGRM on Beauty for $k \in \{6, 121\}$ using three paired seeds per condition. Table 15 reports NDCG@10 and Recall@10 relative to the same held-out-pool clean baseline.

At $k = 6$, none of the four addition conditions differs significantly from clean. At $k = 121$, all four NDCG@10 point estimates are below the clean baseline. Benign addition reduces NDCG@10 by 10.9% ($p = .125$), while the three malicious conditions range from -8.8% to -17.7%. The `cluster_targeted` and `leverage` conditions reach significance ($p = .019$ and $p = .005$, respectively), whereas benign growth and `random` insertion do not. Thus, the statistical evidence is stronger for two malicious conditions, but the magnitude of the benign point estimate lies within the range produced by the tested attacks.

These matched controls refine the interpretation of the fit-set threat surface. Attacker-chosen additions can substantially reorganize the Semantic IDs assigned to legitimate items, but comparable structural movement can also arise when legitimate items are added at the same scale. Structural drift is therefore evidence that the tokenizer fit is sensitive to fit-set composition, but is not by itself a reliable signature of malicious activity.

This distinction motivates the defense analysis in Section 4.4: an effective defense should not rely solely on the magnitude of tokenizer drift to distinguish benign from malicious updates, and should also consider mechanisms that directly limit how much the tokenizer can change when the fit set evolves.

E Extended Defense Evaluation

This section provides the complete defense evaluation supporting Section 4.4. We first evaluate item-filtering baselines and the frozen-rotation warm-start mechanism under changes in attack severity and geometry. We then evaluate a fit-aware admission gate calibrated against legitimate catalog additions, construct an adaptive attack against that gate, and test whether frozen-rotation refitting contains the updates that pass admission screening. Unless otherwise stated, structural ARI in this section denotes full-tuple ARI on legitimate items.

E.1 Baseline-defense comparison

We benchmark six anomaly- or poisoning-detection baselines, including published detectors and a simple centroid-distance control, against cluster-targeted poisoning at $\beta = 0.25$. Each baseline

Table 16: Baseline prevention methods on Beauty under `cluster_targeted` poisoning at $\beta = 0.25$. Detection metrics measure identification of injected items; resulting ARI measures preservation of the legitimate tokenizer structure.

Method	Precision	Recall	F1	Resulting ARI
Undefended	—	—	—	0.432
LOF (Breunig et al., 2000)	50.3%	62.7%	55.8%	0.457
Centroid-distance trim	0.0%	0.0%	0.0%	0.449
Isolation Forest (Liu et al., 2008)	15.8%	19.7%	17.6%	0.455
Spectral Signatures (Tran et al., 2018)	**72.0%**	**89.8%**	**79.9%**	0.378
MCD robust covariance (Rousseeuw & Van Driessen, 1999; Hayase et al., 2021)	31.0%	38.7%	34.4%	0.273
DAO (Huang et al., 2025)	55.3%	69.0%	61.4%	0.431
Frozen-rotation warm-start	—	—	—	**0.987**

Table 17: Resulting full-tuple ARI by defense and attack severity on Beauty. The clean-refit ceiling is 1.000 at every budget.

Method	$\beta = .01$	$\beta = .05$	$\beta = .10$	$\beta = .25$
Undefended	0.467	0.509	0.461	0.446
LOF	0.445	0.455	0.430	0.473
Centroid-distance trim	0.446	0.459	0.477	0.415
Isolation Forest	0.402	0.459	0.455	0.429
Spectral Signatures	0.356	0.374	0.460	0.453
MCD robust covariance	0.315	0.313	0.265	0.259
DAO	0.482	0.455	0.441	0.461
Frozen-rotation warm-start	**0.998**	**0.995**	**0.990**	**0.987**

filters the candidate fit set before OPQ is fitted. We therefore report both its ability to identify injected items and the structural ARI obtained after fitting the tokenizer on the filtered set. Precision, recall, and F1 are computed against known poison labels for evaluation only.

None of the six filtering baselines approaches the clean-refit ceiling of 1.000. Moreover, poison-detection accuracy is not sufficient for preserving the tokenizer. Spectral Signatures obtains the highest F1 (79.9%) but produces ARI 0.378, below the undefended 0.432. DAO obtains F1 61.4% while leaving ARI almost unchanged at 0.431. MCD illustrates the same mismatch in the opposite direction: filtering based on its robust-distance score produces ARI only 0.273.

Frozen-rotation warm-start is not an item-level classifier. It retains all candidate items and directly constrains the refitting process, producing a deployable tokenizer with ARI 0.987.

E.2 Severity sensitivity

We next vary the poisoning budget over the same range used in the structural evaluation, $\beta \in \{0.01, 0.05, 0.10, 0.25\}$, while holding the attack strategy fixed to `cluster_targeted`. This tests whether the baseline failures are specific to the 25% condition.

The baseline failure persists throughout the severity sweep: none approaches the clean ceiling at any tested budget. In contrast, frozen-rotation warm-start remains at ARI ≥ 0.987 throughout, including 0.998 at the smallest budget and 0.987 at the largest.

E.3 Strategy sensitivity

We also vary poisoning geometry at fixed $\beta = 0.25$, repeating the comparison under cluster-targeted, leverage, and random insertion.

The filtering baselines remain far from the clean ceiling under all three attack geometries. Frozen-rotation warm-start avoids explicit per-item classification and remains stable across the three strategies, with ARI between 0.980 and 0.987.

Table 18: Resulting full-tuple ARI by defense and poisoning strategy on Beauty at $\beta = 0.25$. The clean-refit ceiling is 1.000.

Method	`cluster_targeted`	`leverage`	`random`
Undefended	0.451	0.369	0.430
LOF	0.449	0.427	0.483
Centroid-distance trim	0.423	0.380	0.420
Isolation Forest	0.439	0.437	0.468
Spectral Signatures	0.365	0.406	0.355
MCD robust covariance	0.287	0.269	0.236
DAO	0.450	0.395	0.436
Frozen-rotation warm-start	**0.987**	**0.980**	**0.987**

Table 19: Full-tuple ARI after ten sequential refit cycles under drip-fed poison and legitimate catalog additions, as a function of the per-cycle Lloyd-iteration cap.

Per-cycle `max_iter`	1	2	5
Drip-fed poison	**0.945**	0.927	0.915
Legitimate additions	0.971	0.963	0.957

E.4 Sequential drip-feed stress test

The preceding experiments evaluate a single tokenizer refit. We therefore test a patient attacker that distributes the complete $\beta = 0.25$ cluster-targeted poisoning budget across ten sequential refit cycles. Once admitted, items remain in the candidate set, and each cycle warm-starts its codebooks from the preceding deployed fit. For comparison, we also simulate cumulative legitimate additions totaling 5% across the same number of cycles.

Drip-feeding partially erodes the defense. At `max_iter`$= 1$, final ARI falls from the single-shot 0.987 to 0.945 after the complete poisoning budget has accumulated. Increasing the per-cycle iteration allowance further reduces ARI to 0.927 and 0.915. Legitimate additions also induce cumulative movement, reaching 0.971, 0.963, and 0.957 under the same iteration caps.

Thus, frozen-rotation warm-start stabilizes rather than authenticates fit-set changes. It damps both malicious and legitimate composition changes, and its protection weakens as additional optimization is permitted across repeated updates. We use `max_iter`$= 1$ as the operating point in the single-shot experiments.

E.5 Fit-aware trusted-reference admission gate

The matched-growth experiment in Appendix D shows that tokenizer drift itself does not reliably distinguish malicious insertion from legitimate fit-set evolution. We therefore evaluate an admission statistic that scores candidate items before refitting against the already-trusted catalog rather than against the possibly contaminated candidate batch.

For a candidate batch $B = \{x_i\}_{i=1}^{k}$ and trusted population $\mathcal{T}$, we compute

$$s(B) = \frac{1}{k} \sum_{i=1}^{k} \max_{x_j \in \mathcal{T}} \cos(x_i, x_j).$$

Candidate items are never permitted to serve as neighbors for one another. This distinction is important for coordinated attacks: multiple injected points placed near one another can support each other under same-population density scoring, whereas they provide no mutual cover under trusted-reference scoring.

We calibrate the statistic using fresh legitimate-item additions, separate from the held-out draws used for evaluation. Calibration is performed independently at each tested batch size, including $k = 40$ for the HotFlip construction. For operating-curve evaluation, each batch is assigned the

Table 20: Fit-aware admission performance across three catalogs. Each cell reports ROC AUC followed by TPR at held-out FPR $\leq 1\%/5\%/10\%$.

Attack	Beauty	Toys and Games	Sports and Outdoors
`random`	1.000; 100/100/100	1.000; 100/100/100	1.000; 100/100/100
`cluster_targeted`	1.000; 100/100/100	0.998; 95/100/100	1.000; 100/100/100
`leverage`	1.000; 100/100/100	0.965; 57/86/86	1.000; 100/100/100
Adjacent mimicry, 0.01	0.928; 81/81/86	0.896; 71/81/81	0.791; 43/57/62
Adjacent mimicry, 0.05	0.623; 24/24/29	0.791; 33/38/43	0.830; 48/48/57
Adjacent mimicry, 0.15	0.999; 100/100/100	0.998; 90/100/100	1.000; 100/100/100
Adjacent mimicry, 0.30	1.000; 100/100/100	1.000; 100/100/100	1.000; 100/100/100

Table 21: Adaptive adjacent-mimicry attack on Beauty. TPR is for the trusted-reference gate at the calibrated fixed operating point, pooling the seven tested batch sizes.

Jitter scale	Mean $s(B)$	TPR
0.01	0.552	81.0%
0.05	0.437	28.6%
0.15	0.266	100%
0.30	0.218	100%

standardized two-sided score

$$z(B) = \frac{|s(B) - \mu_k|}{\sigma_k},$$

where μ_k and σ_k are estimated from the corresponding calibration draws. Held-out legitimate batches are sampled independently from the calibration draws and are used only to estimate false-positive rates and operating curves.

The original idealized attacks are well separated from held-out legitimate additions across all three catalogs, with ROC AUC at least 0.965. The weakest idealized case is leverage on Toys and Games, where TPR is 86% at held-out FPR $\leq 5\%$.

We also evaluate the two text-controlled attacks from Appendix B.3. On Beauty, the same calibrated trusted-reference statistic flags both HotFlip ($k = 40$, $s(B) = 0.268$) and naive text mimicry ($k = 3{,}025$, $s(B) = 0.363$). The HotFlip batch has a directly calibrated $k = 40$ reference. The 3,025-item mimicry batch exceeds the held-out calibration range and is therefore compared with the $k = 605$ threshold as the nearest available reference; we treat this case explicitly as an extrapolation rather than native calibration.

E.6 Adaptive adjacent-mimicry evasion

The trusted-reference statistic suggests a natural adaptive attack: place each injected item near a different genuine trusted item rather than around a shared derived centroid. For each candidate, we select a distinct trusted anchor, add isotropic Gaussian jitter, and renormalize the resulting representation. This construction differs from `cluster_targeted`, which places an entire attack batch around one derived centroid.

The attack exposes a genuine evasion region. At jitter 0.05, the candidate-to-trusted similarity falls inside the range occupied by legitimate additions and detection drops to 28.6% at the fixed operating point. The stricter held-out operating-curve evaluation in Table 20 reaches only 24%, 38%, and 48% TPR at FPR $\leq 5\%$ on Beauty, Toys and Games, and Sports and Outdoors, respectively. Near-duplicate candidates at jitter 0.01 become anomalously similar to the trusted catalog, while looser perturbations at 0.15 and 0.30 become anomalously dissimilar. The evasion region therefore lies between these two extremes.

We next pass the adaptive batches through a real OPQ refit rather than evaluating admission alone. Across the 84 Beauty refits covering all tested batch sizes and jitter settings, mean per-digit ARI is approximately 0.16. Crucially, matched legitimate additions produce the same level of structural movement. Relative to benign growth at the same k, adjacent mimicry changes mean per-digit ARI by only -0.0007 on average, with a range of -0.019 to $+0.014$. The gate-evading subset

Table 22: Downstream comparison for gate-admitted adjacent mimicry on DiffGRM/Beauty. Results are paired against the held-out-pool clean baseline (NDCG@10 $= 0.0296$, Recall@10 $= 0.0484$), with $n = 3$ seeds.

k	Addition	NDCG@10	Recall@10	p
6	legitimate	0.0280 (-5.3%)	0.0465 (-3.9%)	.173
6	adjacent mimicry	0.0287 (-3.0%)	0.0481 (-0.6%)	.213
121	legitimate	0.0264 (-10.9%)	0.0429 (-11.4%)	.125
121	adjacent mimicry	0.0267 (-9.8%)	0.0433 (-10.6%)	.061

Table 23: Full-tuple ARI under frozen-rotation refitting for legitimate and gate-evading adjacent-mimicry batches. Values are mean [min, max] over the evaluated batches.

Catalog	Lloyd iterations	Legitimate	Gate-evading mimicry
Beauty	1	0.992 [0.986, 0.995]	0.993 [0.988, 0.995]
Beauty	3	0.986 [0.975, 0.992]	0.988 [0.970, 0.992]
Beauty	10	0.975 [0.955, 0.984]	0.978 [0.954, 0.983]
Toys and Games	1	0.996 [0.988, 0.999]	0.998 [0.997, 0.999]
Toys and Games	3	0.992 [0.980, 0.996]	0.996 [0.993, 0.996]
Toys and Games	10	0.983 [0.962, 0.991]	0.990 [0.987, 0.992]
Sports and Outdoors	1	0.996 [0.993, 0.998]	0.997 [0.997, 0.998]
Sports and Outdoors	3	0.991 [0.984, 0.994]	0.993 [0.991, 0.994]
Sports and Outdoors	10	0.983 [0.968, 0.988]	0.988 [0.985, 0.990]

exhibits the same average difference. Thus, the adaptive construction successfully evades admission screening, but in these matched tests it does not produce additional structural damage beyond that induced by legitimate fit-set growth.

We also test the downstream effect of two gate-admitted jitter-0.05 batches using DiffGRM on Beauty.

At $k = 6$, adjacent mimicry changes NDCG@10 by -3.0%, compared with -5.3% for matched legitimate addition. At $k = 121$, the corresponding changes are -9.8% and -10.9%. Direct paired comparisons between mimicry and legitimate addition differ by $+2.4\%$ ($p = .121$) and $+1.2\%$ ($p = .607$), respectively. Accordingly, in these tested conditions, evading the gate does not produce additional aggregate recommendation degradation beyond matched legitimate growth. This adaptive construction has no target item, so this experiment evaluates aggregate recommendation quality rather than target-item promotion.

E.7 Combining admission screening with constrained refitting

We finally ask whether the batches that evade fit-aware admission can be handled by either the ARI-drift tripwire or frozen-rotation warm-started refitting.

ARI-drift tripwire. The tripwire provides little additional discrimination. On Beauty, gate-evading adjacent-mimicry batches have full-tuple ARI 0.362–0.449 against the trusted fit (mean 0.412, $n = 19$), while matched legitimate additions occupy 0.375–0.470 (mean 0.421, $n = 21$). At the original 0.999 tripwire threshold, both malicious and legitimate updates are blocked. Across thresholds, separation is close to chance (AUC 0.586); at an operating point that rejects only 5–10% of legitimate updates, the tripwire detects only 11% of the gate-evading batches. Structural drift alone therefore does not close the adaptive-evasion gap.

Frozen-rotation warm-start. We instead apply frozen-rotation refitting to the batches that pass the admission gate. The rotation and IVF centroid remain fixed, and the PQ codebooks are warm-started from the trusted deployment. We compare legitimate additions with adjacent-mimicry batches admitted by the fit-aware gate.

Table 24: Filtering-baseline and frozen-rotation comparison on Toys and Games under cluster-targeted poisoning at $\beta = 0.25$.

Method	Precision	Recall	F1	Resulting ARI
Undefended	—	—	—	0.414
LOF (Breunig et al., 2000)	72.2%	90.0%	80.1%	0.457
Centroid-distance trim	0.8%	1.0%	0.9%	0.376
Isolation Forest (Liu et al., 2008)	17.6%	22.0%	19.6%	0.468
Spectral Signatures (Tran et al., 2018)	43.9%	54.7%	48.8%	0.428
MCD robust covariance (Rousseeuw & Van Driessen, 1999)	33.9%	42.2%	37.6%	0.432
DAO (Huang et al., 2025)	61.5%	76.6%	68.2%	0.422
Frozen-rotation warm-start	—	—	—	**0.979**

Under frozen rotation, gate-evading batches alter the IDs of legitimate items no more than matched legitimate additions. Across all three catalogs and iteration settings, full-tuple ARI remains approximately 0.97–0.99, rather than falling to the roughly 0.41 regime observed for from-scratch Beauty refits. The two mechanisms therefore address different failure modes: trusted-reference admission rejects the attack constructions that are geometrically distinguishable from legitimate growth, while constrained refitting limits structural movement for the batches that remain admissible.

E.8 Cross-dataset filtering-baseline validation

Finally, we repeat the original filtering-baseline comparison on Toys and Games under cluster-targeted poisoning at $\beta = 0.25$. As in the Beauty evaluation, extracted codes were verified against FAISS before the comparison, with all $200/200$ sampled codes matching exactly.

The cross-dataset experiment reproduces the Beauty pattern. All six filtering baselines remain far below the clean-refit ceiling, with resulting ARI between 0.376 and 0.468, whereas frozen-rotation warm-start reaches 0.979. Detection quality also varies substantially without a corresponding recovery in tokenizer structure. For example, LOF's F1 rises from 55.8% on Beauty to 80.1% on Toys and Games, while its resulting ARI remains 0.457 in both evaluations.

Together with the three-catalog admission results in Table 20 and the three-catalog constrained-refit results in Table 23, these experiments show that the defense conclusions are not restricted to Beauty.