NEVER LOSE THE THREAD: COLLABORATIVE LATENT REASONING FOR GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

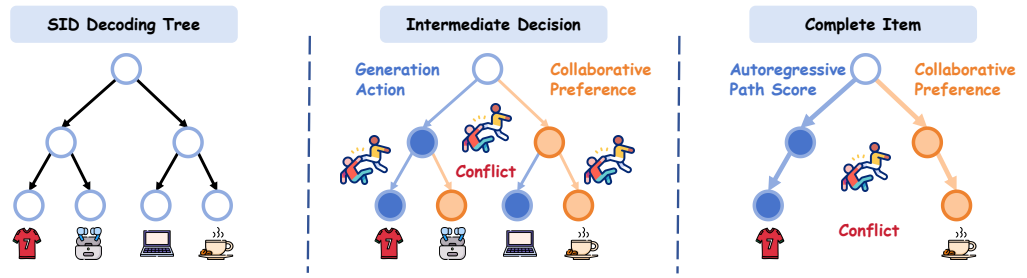
Generative Recommendation (GR) reformulates item recommendation as autoregressive generation over structured item identifiers. Recent studies enhance GR through collaborative alignment and latent reasoning, incorporating behavioral preference and intermediate computation into semantic ID generation. However, these methods overlook collaborative continuity throughout hierarchical generation, i.e., action supervision leaves collaborative preference underdetermined in intermediate states and autoregressive path scoring fails to fully retain collaborative preference for the resulting item ranking. To address the above issues, we first identify the *collaborative continuity gap*, measured by the divergence between level-wise collaborative preference and intermediate-state preference. Subsequently, we propose CoLaGR, a collaborative latent reasoning framework for generative recommendation. CoLaGR derives level-wise collaborative preference from item-level behavioral signals. Furthermore, we develop latent reasoning guided by collaborative preference, where each reasoning state is grounded by the corresponding level-wise collaborative preference. Finally, collaborative leaf calibration is introduced using our constructed collaborative memory to refine complete-item ranking after generation. Extensive experiments demonstrate that CoLaGR consistently outperforms SOTA baselines, with improved collaborative preference modeling at both intermediate-decision and complete-item levels.

1 INTRODUCTION

Generative recommendation predicts a user's next item by producing its semantic ID, a sequence of discrete item codes, from interaction history (Jin et al., 2024; Rajput et al., 2023). By contrast, conventional sequential models such as SASRec and FMLP-Rec score complete candidates, keeping collaborative evidence attached to each item throughout ranking and allowing candidates in a shared semantic branch to receive distinct scores (Kang & McAuley, 2018; Zhou et al., 2022). In generative recommendation, each code selects a branch of the item hierarchy and restricts later choices to items beneath that branch, turning one next-item prediction into successive conditional decisions (Figure 1(a)). Therefore, an early code can be correct for several items without resolving their user-specific ordering, even though the generator must commit to that branch before predicting later codes. How can item-level collaborative evidence guide these successive decisions and remain available to rank complete items after generation?

Early methods combine learned tokenization with autoregressive decoding, making code sequences the units of prediction (Jin et al., 2024; Rajput et al., 2023). LETTER and ETEGRec then add collaborative objectives to identifier learning, placing behavioral structure in the codes before generation (Wang et al., 2024a; Liu et al., 2025). ActionPiece adapts action units to context, whereas RPG explores parallel decoding to change how those units are generated (Hou et al., 2025b;a). TCA4Rec transfers item-level scores into soft next-code targets, bringing collaborative evidence directly to token supervision (Lin et al., 2026). After decoding, MemGen-GR combines generative and conventional scores to rank complete candidates (Ding et al., 2026a). To add intermediate computation during generation, S^2GR interleaves semantically supervised thinking states with code prediction, while Latte places a latent token before the target identifier (Guo et al., 2026c; Hou et al., 2026). These works target different stages of the generative pipeline, from identifier construction and token supervision to intermediate computation and final ranking.

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.



(a) Hierarchical Code Generation **(b) Branch-level Ambiguity** **(c) Leaf-level Mismatch**

Figure 1: Collaborative continuity in semantic ID generation. (a) Codes select branches. (b) Shared actions mask different preferences. (c) Path scores can misalign with item preference.

However, existing methods improve different stages of generative recommendation without ensuring that user-specific collaborative evidence survives code decisions and complete-item ranking. We identify this two-stage failure as the *collaborative continuity gap*. At an intermediate decision, the observed code identifies a branch but cannot specify a user's preferences among its descendant items. Figure 1(b) shows users taking the same action with different collaborative preferences. Therefore, action accuracy cannot reveal whether their decision states preserve this variation. We assess this intermediate component by probing teacher-derived level-wise references in frozen states with a common readout. Figure 1(c) shows that path scores can misrank complete items even with accurate code-level marginals, since those marginals pool preference mass across items sharing a code.

To address the two discontinuities, we propose CoLaGR, a collaborative latent reasoning framework that connects code-level decisions with final item ranking. A frozen sequential recommender supplies item-level preference mass, which we project onto each code vocabulary to form a user-specific reference at every level. Before each code prediction, a collaborative-preference (CP) position creates a decision-aligned hidden state for both the reference and the observed action. Separate readouts supervise these states, so collaborative grounding need not compete with hard code prediction in a single categorical output. After generation, CoLeaf uses collaborative memory to calibrate item scores within the fixed candidate beam, reconnecting path decisions to complete-item preference.

Our contributions are summarized as follows:

- We quantify the *collaborative continuity gap* through recovery of teacher-derived references from hierarchical decision states with a common frozen-state readout, distinguishing intermediate-state underdetermination from complete-item ranking misalignment.
- We introduce CoLaGR, a collaborative latent reasoning framework that separates user-specific preference supervision from hard semantic ID prediction through dedicated readouts before each code decision and calibrates complete items after candidate generation.
- We decompose within-action reference variation into retained and residual information, derive the pointwise shared-output optimum for preference and action, and prove that level-wise marginals cannot determine complete-item ranking, motivating CoLeaf calibration.
- CoLaGR outperforms strong baselines across three domains, with controls isolating intermediate grounding, user correspondence, and complete-item calibration. Code and reproduction resources are available at `https://anonymous.4open.science/r/CoLaGR`.

2 Preliminaries and Problem Definition

Generative Recommendation. Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets. For each user $u \in \mathcal{U}$, the interaction history is $H_u = [i_1^u, \ldots, i_{T_u}^u]$, and the goal is to predict the next item $y_u \in \mathcal{I}$. Following semantic ID generative recommendation (Rajput et al., 2023; Hou et al., 2026), each item i has a code sequence $\mathbf{z}^i = [z_1^i, \ldots, z_L^i]$, with code z_l^i in vocabulary $\mathcal{C}_l$. The conditional probability $P_\theta(\mathbf{z}^{y_u} \mid H_u)$

factorizes over the decision levels:

$$P_\theta(\mathbf{z}^{y_u} \mid H_u) = \prod_{l=1}^{L} P_\theta\left(z_l^{y_u} \mid H_u, \mathbf{z}_{<l}^{y_u}\right), \tag{1}$$

where θ denotes generator parameters, L is the semantic ID length, T_u is the history length, i_t^u is the item at history position t, and $\mathbf{z}_{<l}^{y_u} = [z_1^{y_u}, \ldots, z_{l-1}^{y_u}]$ is the observed code prefix. We denote the target action at level l by $Y_{u,l} = z_l^{y_u}$ and the corresponding intermediate decision state by $h_{u,l}$. Standard next-token training supervises the observed action at each level.

Collaborative continuity gap. Given the same history H_u, a frozen sequential recommender exports a top-M item set $\mathcal{T}_u^M$ with normalized preference mass $p_u(i)$, where $\sum_{i\in\mathcal{T}_u^M} p_u(i) = 1$ and $p_u(i) = 0$ outside $\mathcal{T}_u^M$. We denote the teacher-derived collaborative reference by $q_{u,l}^{\mathrm{col}}$. Each code receives the total preference mass of exported items carrying that code at level l, as formalized in Section 3. This teacher-derived distribution summarizes behavioral preference over the represented items. Examples sharing the same semantic ID action can have different collaborative references, which action supervision alone does not determine.

A common diagnostic readout measures collaborative-reference recoverability from decision states. We define the measured continuity gap $\mathcal{G}_{\mathrm{cont}}$ by:

$$\mathcal{G}_{\mathrm{cont}} = \frac{1}{L}\sum_{l=1}^{L} \mathbb{E}_u\left[D_{\mathrm{KL}}\left(q_{u,l}^{\mathrm{col}} \| r_l(h_{u,l})\right)\right], \tag{2}$$

where D_{KL} denotes Kullback–Leibler divergence, $\mathbb{E}_u$ averages over history–target examples, and r_l is a post-hoc preference readout fitted separately for each model under the same family and fitting protocol. It is distinct from the jointly trained preference head $\pi_{u,l}^{\mathrm{col}}$. For $Q_l = q_{u,l}^{\mathrm{col}}$ and $Z_l = h_{u,l}$, the intrinsic risk is $R_l^\star(Z_l) = \mathbb{E}D_{\mathrm{KL}}(Q_l \| \mathbb{E}[Q_l \mid Z_l])$. The measured risk equals this intrinsic term plus readout approximation and fitting error (Appendix A.2). A smaller measured gap therefore means better recoverability under the specified readout protocol. Autoregressive path scoring can remain misaligned with complete-item preference, motivating post-generation calibration.

3 METHOD

3.1 OVERVIEW

As shown in Figure 2, CoLaGR reconstructs level-wise collaborative references from decision-aligned states. A separate action readout retains hard SID supervision, while CoLeaf calibrates fixed-beam ranking with complete-item memory evidence.

3.2 LEVEL-WISE COLLABORATIVE REFERENCE CONSTRUCTION

The decoder predicts individual codes, whereas collaborative evidence concerns complete items and must be projected onto each code vocabulary to preserve user-specific structure.

Level-wise reference projection. We project frozen SASRec teacher preferences onto each code vocabulary (Kang & McAuley, 2018). Summing item masses gives $q_{u,l}^{\mathrm{col}}(c)$:

$$q_{u,l}^{\mathrm{col}}(c) = \sum_{i\in\mathcal{I}} p_u(i)\mathbf{1}[z_l^i = c], \qquad c \in \mathcal{C}_l, \tag{3}$$

where $p_u(i)$ is normalized mass on the exported top-M set $\mathcal{T}_u^M$, z_l^i is item i's level-l code, and $\mathbf{1}[\cdot]$ is the indicator. By construction, $p_u(i) = 0$ outside $\mathcal{T}_u^M$. The teacher and SID mapping remain fixed. References use the available history rather than the held-out target. Each reference is prefix-independent, not a posterior over valid child branches. Conditioning it on the committed subtree would change the target after each action, whereas $h_{u,l}$ carries the changing decision context.

Users taking the same code may assign different collaborative mass to alternatives, so Theorem 1 partitions the reference variation that a decision state can preserve beyond the observed action.

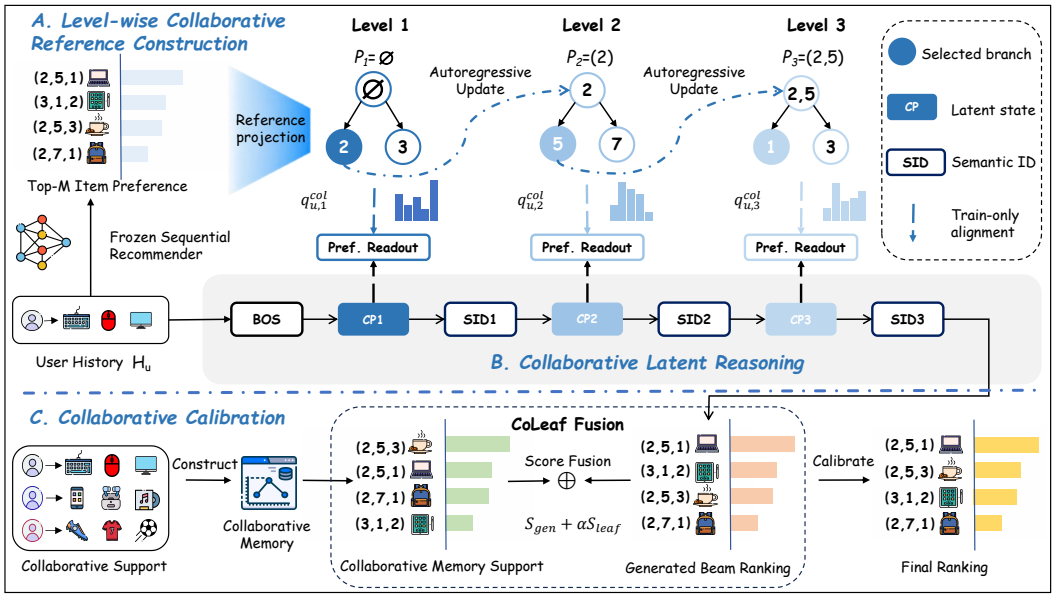


Figure 2: Overview of CoLaGR. Prefix-independent references ground latent states as decoder prefixes evolve; CoLeaf calibrates fixed-beam ranking with collaborative memory.

Theorem 1 (Collaborative continuity decomposition). *At level l, let X_l be the context, Y_l the observed action, $Q_l = q_l^{\mathrm{col}}(X_l)$ the collaborative reference, and $Z_l = f_l(X_l)$ the decision state. Draw an auxiliary code C_l from Q_l conditional on X_l. Define the within-action reference variation as $\mathcal{A}_l = I(C_l; X_l \mid Y_l)$ and the residual variation as $\mathcal{G}_l^\star = I(C_l; X_l \mid Z_l, Y_l)$. The within-action reference variation $\mathcal{A}_l$ can be decomposed into state-retained information and residual variation as:*

$$\mathcal{A}_l = I(C_l; X_l \mid Y_l) = I(C_l; Z_l \mid Y_l) + \mathcal{G}_l^\star, \tag{4}$$

where $I(\cdot;\cdot \mid \cdot)$ denotes conditional mutual information. Since Z_l is determined by X_l, the identity separates state-retained information $I(C_l; Z_l \mid Y_l)$ from residual variation $\mathcal{G}_l^\star$. An action-only state retains none of the within-action variation. The measured gap in Eq. 2 probes Z_l with a finite readout and need not equal this intrinsic residual. Appendix A.2 gives the proof and readout decomposition.

Implication. Reducing residual variation requires preference-grounded states that retain information beyond action identity while predicting hard actions separately.

3.3 Collaborative Latent Reasoning

Collaborative latent reasoning grounds hidden states before SID prediction. Theorem 2 formalizes the trade-off between fitting a hard action and a different soft reference with one categorical output under logarithmic loss and KL divergence (Gneiting & Raftery, 2007).

Theorem 2 (Shared-output preference–action coupling). *At level l, let $\Delta(\mathcal{C}_l)$ denote the probability simplex over the code vocabulary, $D \in \Delta(\mathcal{C}_l)$ the shared categorical output, $y \in \mathcal{C}_l$ the observed code, $q \in \Delta(\mathcal{C}_l)$ the soft collaborative reference, and $\eta > 0$ its weight. Minimizing $\ell(D) = -\log D(y) + \eta D_{\mathrm{KL}}(q \| D)$ gives the unique output $D^\star(c)$:*

$$D^\star(c) = \frac{\mathbf{1}[c = y] + \eta q(c)}{1 + \eta}, \qquad c \in \mathcal{C}_l. \tag{5}$$

Here $D(y)$ is the observed-code probability and $\delta_y(c) = \mathbf{1}[c = y]$ its point-mass target. If $q(y) = 0$, even the optimal shared output assigns the correct code only $1/(1+\eta)$ probability. More generally, smaller $q(y)$ incurs a larger hard-action penalty.

Separate readouts avoid output competition while both losses shape the state. Theorem 2 concerns the shared output's pointwise optimum, not the neural generator's training optimum or accuracy. Appendix A.4 gives the proof and minimum loss.

Implication. Greater reference–action disagreement increases hard-action loss at the shared-output optimum, motivating the conflict-stratified evaluation in Section 4.4.

Dedicated preference and action readouts. Separate preference and action readouts supervise reference reconstruction and SID prediction from the same decision state, so collaborative gradients shape its representation while the action head retains hard-label supervision.

Decision-aligned latent realization. We insert a learned CP token before each code prediction to provide both readouts with a decision-aligned state, yielding the interleaved decoder sequence $\widetilde{\mathbf{z}}^{y_u}$:

$$\widetilde{\mathbf{z}}^{y_u} = [\mathrm{BOS}, \mathrm{CP}_1, z_1^{y_u}, \ldots, \mathrm{CP}_L, z_L^{y_u}], \tag{6}$$

where $\widetilde{\mathbf{z}}^{y_u}$ is the interleaved sequence and CP_l is a level-specific token with a learned embedding. The causal decoder state $h_{u,l} \in \mathbb{R}^{d_h}$ at CP_l, with hidden dimension d_h, uses the encoded history and prefix through that position without accessing current or later target codes. CP positions supply both readouts from hidden states, providing a decision-aligned implementation of collaborative grounding without textual rationale.

Level-wise collaborative grounding. Each CP state is grounded with the KL loss $\mathcal{L}_{\mathrm{col}}$:

$$\mathcal{L}_{\mathrm{col}} = \mathbb{E}_{(H_u, y_u)\sim\mathcal{D}} \left[\frac{1}{L} \sum_{l=1}^{L} D_{\mathrm{KL}}(q_{u,l}^{\mathrm{col}} \| \pi_{u,l}^{\mathrm{col}}) \right], \tag{7}$$

where $\pi_{u,l}^{\mathrm{col}} = \mathrm{softmax}(W_l^{\mathrm{col}} h_{u,l} + b_l^{\mathrm{col}})$ is the preference readout over $\mathcal{C}_l$, $W_l^{\mathrm{col}} \in \mathbb{R}^{|\mathcal{C}_l| \times d_h}$ and $b_l^{\mathrm{col}} \in \mathbb{R}^{|\mathcal{C}_l|}$ parameterize its head, and $\mathcal{D}$ is the training set. Unlike the post-hoc diagnostic readout r_l, this head is optimized jointly with the generator. The collaborative target shapes training without entering the model input or any stage of decoding at inference time.

Joint training with hard-action supervision. Since the same CP state supports reference reconstruction and next-code prediction, we combine collaborative grounding with hard-label supervision through its action readout. The generation loss $\mathcal{L}_{\mathrm{gen}}$ and joint objective $\mathcal{L}$ take the form:

$$\mathcal{L}_{\mathrm{gen}} = -\mathbb{E}_{\mathcal{D}} \left[\frac{1}{L} \sum_{l=1}^{L} \log \pi_{u,l}^{\mathrm{act}}(Y_{u,l}) \right], \qquad \mathcal{L} = \mathcal{L}_{\mathrm{gen}} + \lambda_{\mathrm{pref}} \mathcal{L}_{\mathrm{col}}, \tag{8}$$

where $\mathbb{E}_{\mathcal{D}}$ averages over $(H_u, y_u) \sim \mathcal{D}$, $\pi_{u,l}^{\mathrm{act}}$ is the action distribution, and $\lambda_{\mathrm{pref}} \geq 0$ is the collaborative weight. For the Direct baseline, we set its collaborative weight $\eta = \lambda_{\mathrm{pref}}$ under the same level averaging to match the relative loss scale. The generation and grounding losses use separate output heads and backpropagate through the shared decoder.

The KL grounding objective bounds utility discrepancies between the reference and its preference readout. The following lemma formalizes this relationship.

Lemma 1 (Preference-utility control). *For any bounded code utility $V_l : \mathcal{C}_l \to [0, 1]$, the average utility discrepancy $\frac{1}{L}\sum_l \mathbb{E}e_l$ obeys:*

$$\frac{1}{L} \sum_{l=1}^{L} \mathbb{E}e_l \leq \sqrt{\frac{1}{2L} \sum_{l=1}^{L} \epsilon_l}, \tag{9}$$

where $e_l = |\mathbb{E}_{q_{u,l}^{\mathrm{col}}} V_l - \mathbb{E}_{\pi_{u,l}^{\mathrm{col}}} V_l|$ is the utility discrepancy and $\epsilon_l = \mathbb{E}D_{\mathrm{KL}}(q_{u,l}^{\mathrm{col}} \| \pi_{u,l}^{\mathrm{col}})$ is the expected grounding loss at level l. Thus reducing the grounding loss bounds reference–readout discrepancies for every bounded code utility. Appendix A.5 gives the proof and extends the result with the context shift induced by autoregressively generated decision prefixes.

Teacher-free generation. At inference, the fixed CP pattern interleaves with constrained semantic ID generation under the valid-prefix trie. Beam scores accumulate action log-probabilities. Neither the teacher nor the level-wise collaborative reference is queried, and the preference heads are unnecessary for decoding. Collaborative memory is used only after candidate generation.

3.4 COLLABORATIVE LEAF CALIBRATION

Because level-wise marginals do not determine complete-item order, CoLeaf combines generation confidence with collaborative memory on the fixed beam. Proposition 1 formalizes the ranking ambiguity that motivates this calibration.

Proposition 1 (Marginal non-identifiability). *The projection from complete-item distributions to their collection of level-wise marginals is not injective. For semantic IDs* $(0,0),(0,1),(1,0),(1,1)$*, the item distributions* $\xi = (0.4, 0.1, 0.1, 0.4)$ *and* $\xi' = (0.1, 0.4, 0.4, 0.1)$ *have identical level-wise marginals but reverse the order of the first two items.* Thus level-wise marginals alone do not identify complete-item rankings; the ambiguity arises from marginalization. Appendix A.6 provides details.

We associate each generated item with an item-level collaborative utility $S_{\text{leaf}}(u, j)$. Our default implementation uses recency-weighted transition statistics from training interactions. Its construction is detailed in Appendix B. The teacher supervises intermediate states, whereas the memory scores complete items after candidate generation.

Generation-regularized calibration. To incorporate memory evidence while retaining generation confidence, we regularize candidate preference toward the generator's distribution. Let $\mathcal{B}_u$ be the beam and $g_u(j) \propto \exp(S_{\text{gen}}(u, j))$ its normalized generation distribution. CoLeaf obtains ρ_u^α by solving the following problem:

$$\rho_u^\alpha = \arg \max_{\rho \in \Delta(\mathcal{B}_u)} \{\alpha \mathbb{E}_{j\sim\rho}[S_{\text{leaf}}(u, j)] - D_{\text{KL}}(\rho \| g_u)\}, \tag{10}$$

where $\Delta(\mathcal{B}_u)$ is the candidate probability simplex, $S_{\text{leaf}}(u, j)$ is the collaborative item utility, and $\alpha \geq 0$ is selected on validation data. The variational identity in Appendix A.7 gives the unique solution $\rho_u^\alpha(j) \propto g_u(j) \exp(\alpha S_{\text{leaf}}(u, j))$. Candidates are ordered by the final score $S_{\text{final}}(u, j)$:

$$S_{\text{final}}(u, j) = S_{\text{gen}}(u, j) + \alpha S_{\text{leaf}}(u, j), \qquad j \in \mathcal{B}_u, \tag{11}$$

where S_{gen} is the accumulated generation score and S_{final} is the calibrated score. The closed-form solution requires no additional optimization. CoLeaf neither updates the generator nor affects intermediate-state reference reconstruction. Ranking improvements depend on the quality of candidate-specific collaborative memory evidence.

Corollary 1 (Candidate-support invariance). *For a fixed candidate set* $\mathcal{B}_u$*, finite generation and memory scores, and finite* α*, the calibrated distribution satisfies* $\text{supp}(\rho_u^\alpha) = \mathcal{B}_u$*. Calibration can change candidate order but cannot add or remove an item.* Full-beam Recall is therefore invariant, while smaller-cutoff ranking remains sensitive to the correspondence between candidates and memory evidence. Appendix A.7 provides the variational proof and pairwise crossover condition.

4 EXPERIMENTS

Our experiments evaluate recommendation quality (RQ1), components (RQ2), intermediate preference (RQ3), complete-item calibration (RQ4), and weight sensitivity (RQ5).

4.1 EXPERIMENTAL SETUP

Datasets and protocol. We use Industrial and Scientific, Musical Instruments, and Video Games from Amazon Reviews 2023 under the Latte leave-one-out protocol (Hou et al., 2026). Locally adapted generative baselines and CoLaGR share the item universe, data splits, semantic ID assets, and evaluator. Appendix C reports the full dataset statistics and experimental settings.

Metrics. We report NDCG and Recall at 5 and 10. Recall@50 measures target coverage in the generated beam, separating candidate generation from subsequent ranking.

Baselines. We compare CoLaGR with sequential recommenders SASRec, FMLP-Rec, HSTU, FDSA, and S^3-Rec, and generative methods TIGER, LETTER, ActionPiece, TCA4Rec, MemGen-GR, Latte, and S^2GR (Kang & McAuley, 2018; Zhou et al., 2022; Zhai et al., 2024; Zhang et al., 2019; Zhou et al., 2020; Rajput et al., 2023; Wang et al., 2024a; Hou et al., 2025b; Lin et al., 2026; Ding et al., 2026a; Hou et al., 2026; Guo et al., 2026c). TCA4Rec, S^2GR, and MemGen-GR use local adaptations detailed in Appendix B.

Table 1: Full-system performance across three datasets. N and R denote NDCG and Recall. Bold and underline mark the best and second-best results, respectively.

Method	Industrial and Scientific				Musical Instruments				Video Games			
	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10
SASRec	0.0150	0.0199	0.0259	0.0412	0.0213	0.0274	0.0333	0.0523	0.0331	0.0438	0.0535	0.0847
FMLP-Rec	0.0155	0.0204	0.0269	0.0422	0.0218	0.0282	0.0339	0.0536	0.0338	0.0444	0.0528	0.0857
HSTU	0.0147	0.0198	0.0271	0.0429	0.0191	0.0271	0.0343	0.0577	0.0334	0.0442	0.0578	0.0903
FDSA	0.0169	0.0213	0.0262	0.0421	0.0230	0.0293	0.0347	0.0545	0.0361	0.0448	0.0544	0.0852
S^3-Rec	0.0171	0.0219	0.0263	0.0418	0.0199	0.0257	0.0317	0.0496	0.0315	0.0406	0.0485	0.0769
TIGER	0.0175	0.0226	0.0264	0.0422	0.0244	0.0306	0.0370	0.0564	0.0366	0.0467	0.0559	0.0868
LETTER	0.0182	0.0232	0.0279	0.0435	0.0246	0.0313	0.0372	0.0580	0.0372	0.0473	0.0563	0.0877
ActionPiece	0.0182	0.0236	0.0284	0.0452	0.0243	0.0318	0.0383	0.0615	0.0382	0.0490	0.0591	0.0927
TCA4Rec	0.0168	0.0216	0.0259	0.0408	0.0246	0.0315	0.0373	0.0584	0.0366	0.0471	0.0565	0.0894
S^2GR	0.0176	0.0228	0.0270	0.0434	0.0237	0.0300	0.0359	0.0556	0.0369	0.0477	0.0564	0.0901
MemGen-GR	<u>0.0198</u>	0.0247	0.0302	0.0455	<u>0.0262</u>	0.0329	0.0393	0.0600	<u>0.0415</u>	<u>0.0520</u>	<u>0.0621</u>	0.0949
Latte	0.0196	<u>0.0249</u>	<u>0.0304</u>	<u>0.0470</u>	0.0261	<u>0.0331</u>	<u>0.0401</u>	<u>0.0618</u>	0.0406	0.0515	0.0618	<u>0.0958</u>
CoLaGR	**0.0257**	**0.0309**	**0.0374**	**0.0536**	**0.0324**	**0.0395**	**0.0472**	**0.0694**	**0.0510**	**0.0618**	**0.0740**	**0.1073**
Δ vs. best baseline	+29.8%	+24.1%	+23.0%	+14.0%	+23.7%	+19.3%	+17.7%	+12.3%	+23.0%	+18.8%	+19.2%	+12.1%

Table 2: Component results. CoLaGR-G is CoLaGR without CoLeaf. N/R denote NDCG/Recall.

	Industrial				Musical				Video Games			
Variant	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10
Base	0.0172	0.0221	0.0269	0.0421	0.0251	0.0319	0.0382	0.0593	0.0387	0.0495	0.0588	0.0925
Latent-only	0.0183	0.0236	0.0284	0.0450	0.0254	0.0324	0.0388	0.0608	0.0392	0.0502	0.0594	0.0938
Direct	0.0187	0.0239	0.0285	0.0446	0.0261	0.0333	0.0393	0.0616	0.0404	0.0510	0.0616	0.0946
CoLaGR-G	0.0216	0.0272	0.0330	0.0503	0.0281	0.0353	0.0425	0.0648	0.0414	0.0527	0.0629	0.0980
CoLaGR	**0.0257**	**0.0309**	**0.0374**	**0.0536**	**0.0324**	**0.0395**	**0.0472**	**0.0694**	**0.0510**	**0.0618**	**0.0740**	**0.1073**

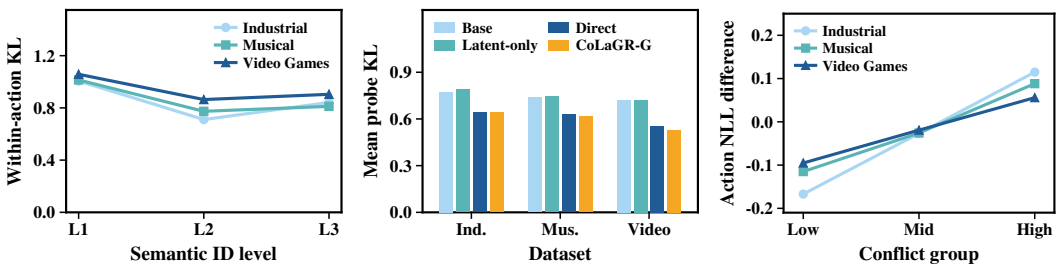
Implementation. Controlled CoLaGR variants share semantic IDs and prefix-trie masking; collaborative supervision uses fixed teacher references. We select checkpoints, supervision weights, and calibration weights using validation NDCG@10. Appendix B gives configurations, and Appendix C reports repeated runs and sensitivity analyses.

4.2 Overall Recommendation Performance (RQ1)

Table 1 compares the complete system with sequential and generative baselines. CoLaGR leads on all twelve metrics across Industrial, Musical, and Video Games, with NDCG@10 improvements of 18.8%–24.1% over the strongest listed baselines. Its gains in NDCG are larger than those in Recall at the same cutoff, suggesting that improved ranking within the generated candidate set contributes alongside target coverage. These are full-system gains combining generator-side collaborative grounding with candidate-preserving calibration. Table 2 reports the generator-side contribution before CoLeaf, while Section 4.5 isolates calibration on a fixed beam. Repeated runs show consistent full-system performance; Appendices C.5 and C.13 provide multi-seed results and calibrated controls across the domains and ranking cutoffs.

4.3 Component Analysis (RQ2)

Table 2 separates latent computation, collaborative grounding, and complete-item calibration. Latent-only makes a small improvement over Base, while Direct adds a soft collaborative target at the action output. CoLaGR-G improves on both through separate hard-action and reference readouts of the same decision state. Native-State Grounding applies this supervision to existing pre-action states and retains comparable gains across domains (Appendix C.6). Thus the grounding principle extends beyond dedicated CP positions, which CoLaGR uses as a uniform decision-aligned interface. CoLeaf then improves all reported metrics, consistent with the benefit of revisiting preference at the complete-item level. Finally, breaking user–reference correspondence weakens the generator (Appendix C.7); the benefit depends on matching the collaborative reference to the user, not simply adding an auxiliary loss.



(a) Action ambiguity **(b) Reference recoverability** **(c) Action conflict**

Figure 3: Intermediate-state diagnostics. (a) Within-action ambiguity. (b) Probe KL. (c) Direct minus CoLaGR-G action NLL, with positive values favoring CoLaGR-G.

4.4 Collaborative Preference at Intermediate Decisions (RQ3)

Figure 3(a) shows that examples sharing the same observed code can have different collaborative references, the within-action variation identified by Theorem 1 that action labels alone cannot resolve. We therefore ask whether the decision state retains collaborative information beyond action identity. Probes on frozen observed-prefix states recover the collaborative reference more accurately from CoLaGR-G than from Base across all three domains (Figure 3(b)). Within-action shuffling degrades recovery, and action-conditioned probes further support reference recoverability beyond action identity (Appendix C.11). The effect on action prediction is conflict-dependent. Grouping decisions by the preference–action disagreement in Theorem 2, Figure 3(c) shows that Direct has lower NLL at low and medium conflict, whereas CoLaGR-G has lower NLL at high conflict. The NLL advantage is concentrated at L1–L2 (Appendix C.12), showing that the effect varies across decision levels.

4.5 Collaborative Preference in Complete-Item Ranking (RQ4)

Complete-item calibration. Figure 4 fixes the CoLaGR-G beam to isolate the ranking stage. CoLeaf improves NDCG@10 by 12.0%–17.1% across the three domains when its memory evidence is matched to the candidate item. Candidate membership and Recall@50 remain unchanged, so calibration promotes relevant items already in the beam rather than retrieving new ones. A stricter control permutes memory scores among those same candidates while retaining the generation scores and the set of memory scores. Performance then falls below the uncalibrated generator, showing that the gain depends on which item receives each collaborative signal. This complements the intermediate-state result: preference retained during code generation does not fully determine the order of complete items. CoLeaf supplies item-level evidence for that final distinction. Calibration also benefits other local generators, with the native-state grounding control slightly ahead on Industrial after calibration (Appendix C.13).

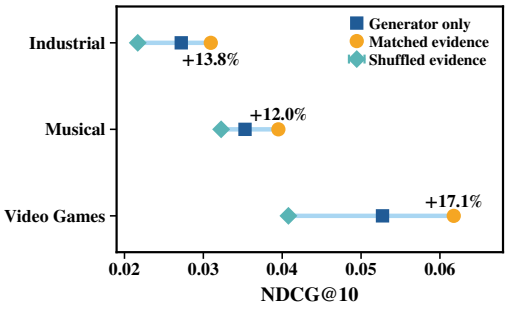


Figure 4: Fixed-candidate calibration under matched and shuffled evidence.

4.6 Hyperparameter Sensitivity (RQ5)

Figure 5 examines the two weights that connect collaborative evidence to the generator and the final ranking: the state-supervision weight λ_{pref} and CoLeaf coefficient α. Validation favors a positive calibration weight in every domain, consistent with the fixed-candidate experiment. Neither stronger state supervision nor stronger calibration improves performance indefinitely; the useful range balances the collaborative signal with hard semantic ID prediction and path scores. We therefore select both weights on validation NDCG@10 rather than treating maximum weight as stronger collaboration. The figure reports Recall@10 at the same settings, while Appendix C.2 provides the full validation NDCG@10 sweeps and shows how each domain responds around its selected weights.

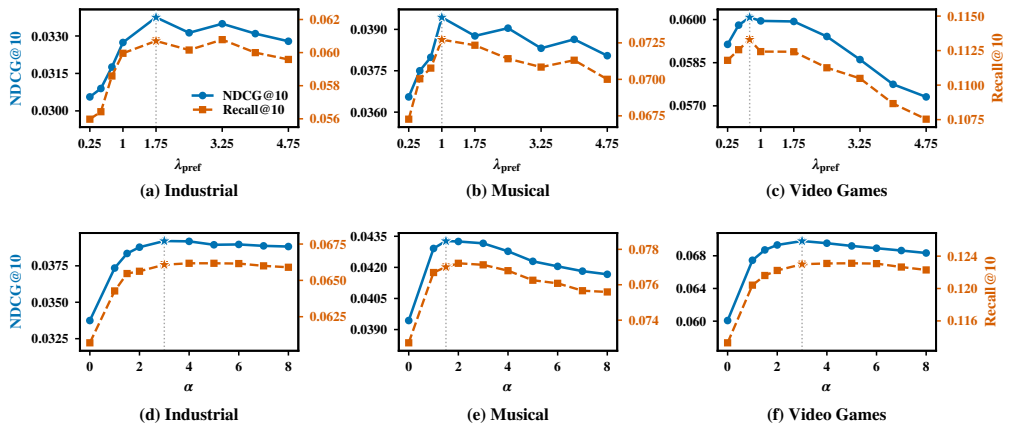


Figure 5: Validation sensitivity to supervision (top) and calibration (bottom). Blue and orange report NDCG@10 and Recall@10, respectively. Stars mark validation-selected weights.

5 Related Work

Collaborative-Aware Generative Recommendation. Sequential models use recurrent, attention-based, or filtered history representations (Hidasi et al., 2016; Kang & McAuley, 2018; Sun et al., 2019; Zhou et al., 2022); generative methods produce semantic item codes (Jin et al., 2024; Rajput et al., 2023). Recent work explores collaborative tokenization and end-to-end identifiers (Wang et al., 2024a; Liu et al., 2025), context-dependent action units (Hou et al., 2025b), parallel decoding (Hou et al., 2025a), and multimodal quantization (Wei et al., 2026). Behavioral evidence also enters via LLM adaptation, dual streams, sparse-dense representations, and soft token targets (Zheng et al., 2024; Zhang et al., 2025; Wang et al., 2024b; Yang et al., 2025; Lin et al., 2026). AlphaRec studies language representations (Sheng et al., 2025). SpecGR drafts and verifies candidates (Ding et al., 2026b), while MemGen-GR combines generative and conventional recommenders (Ding et al., 2026a). Live-streaming work adds real-time multimodal feedback (Qu et al., 2026; Guo et al., 2026a) and dynamic room-level collaboration (Guo et al., 2026b). CoLaGR recovers collaborative references from intermediate states and calibrates candidates with collaborative memory.

Reasoning-Enhanced Recommendation. Reasoning-enhanced recommendation introduces intermediate computation before item prediction. LatentR^3 optimizes latent reasoning through reinforcement learning, while SIREN internalizes long-term interest sketches and reasoning in an LLM recommender (Zhang et al., 2026; Ding et al., 2026c). R^{2}ec combines textual reasoning and recommendation with dual heads, and ThinkRec distills synthesized reasoning traces (You et al., 2025; Yu et al., 2026). ReaRec expands inference-time computation through iterative latent refinement, whereas ManCAR grounds refinement in a collaborative item-space prior with adaptive stopping (Tang et al., 2026; Yang et al., 2026). Within semantic ID generation, S^{2}GR interleaves semantically supervised thinking states with code prediction, and Latte inserts a discrete latent token before the complete target identifier (Guo et al., 2026c; Hou et al., 2026). CoLaGR reconstructs a user-specific collaborative reference through a readout separated from hard semantic ID prediction, linking intermediate-state recoverability, preference–action coupling, and complete-item calibration.

6 Conclusion and Future Work

CoLaGR frames collaborative continuity as a design principle for hierarchical generative recommendation. At intermediate decisions, level-wise references and separate readouts improve recovery of user-specific collaborative references beyond action identity while retaining hard semantic ID supervision. At the complete-item level, CoLeaf adds item-level evidence to refine candidate rankings beyond level-wise collaborative supervision. The resulting framework links representation, prediction, and ranking through complementary collaborative interfaces. Across three domains, gains over strong baselines, multi-seed results, and correspondence controls consistently support this account. Future work will extend the framework to evolving user histories and online recommendation environments.

Reproducibility Statement

Appendix B documents collaborative assets, implementation settings, and validation-selected hyperparameters. Appendix C reports multi-seed results, sensitivity analyses, and diagnostic controls.

AI Use Statement

Generative AI tools were used to assist with language polishing, including grammar, phrasing, clarity, and organization, and to support figure presentation and technical discussion. The authors independently verified all theoretical claims, derivations, citations, experimental results, and final manuscript content and retain full responsibility for the work.

References

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, second edition, 2006.

Yijie Ding, Zitian Guo, Jiacheng Li, Letian Peng, Shuai Shao, Wei Shao, Xiaoqiang Luo, Luke Simon, Jingbo Shang, Julian McAuley, and Yupeng Hou. On the Memorization and Generalization of Generative Recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026a.

Yijie Ding, Jiacheng Li, Julian McAuley, and Yupeng Hou. Inductive Generative Recommendation via Retrieval-based Speculation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14675–14683, 2026b.

Zhihao Ding, Jinming Li, Shuai Mu, and Jieming Shi. Token-Efficient Long-Term Interest Sketching and Internalized Reasoning for LLM-based Recommendation. In *International Conference on Learning Representations*, 2026c.

Tilmann Gneiting and Adrian E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Ke Guo, Changle Qu, Jiayaqi Cheng, Xiao Zhang, Shijun Wang, Xiaoyu Zhang, Xueliang Wang, Le Zhang, Lantao Hu, and Jun Xu. KuaiLive-M3: A multi-modal, multi-domain, and multi-feedback dataset for live streaming recommendation. *arXiv preprint arXiv:2607.24862*, 2026a.

Ke Guo, Changle Qu, Xiao Zhang, Liqin Zhao, Shijun Wang, Yanan Niu, and Jun Xu. Room matters: Dynamic room-level collaboration information modeling for live streaming recommendation. In *Proceedings of the ACM Web Conference 2026*, pp. 6045–6056, 2026b.

Zihao Guo, Jian Wang, Ruxin Zhou, Youhua Liu, Jiawei Guo, Jun Zhao, Xiaoxiao Xu, Yongqi Liu, and Kaiqiao Zhan. S^2GR: Stepwise Semantic-Guided Reasoning in Latent Space for Generative Recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 1498–1507, 2026c.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based Recommendations with Recurrent Neural Networks. In *International Conference on Learning Representations*, 2016.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. Generating Long Semantic IDs in Parallel for Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 956–966, 2025a.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian McAuley, and Derek Zhiyuan Cheng. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24004–24024, 2025b.

Yupeng Hou, Haven Kim, Clark Mingxuan Ju, Eduardo Escoto, Neil Shah, and Julian McAuley. Expressiveness Limits of Autoregressive Semantic ID Generation in Generative Recommendation. *arXiv preprint arXiv:2605.06331*, 2026.

Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, Suhang Wang, Jiawei Han, and Xianfeng Tang. Language Models as Semantic Indexers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 22244–22259, 2024.

Wang-Cheng Kang and Julian McAuley. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining*, pp. 197–206, 2018.

Fake Lin, Binbin Hu, Zhi Zheng, Xi Zhu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, and Tong Xu. Token-level Collaborative Alignment for LLM-based Generative Recommendation. In *Proceedings of the ACM Web Conference 2026*, pp. 6909–6919, 2026.

Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative Recommender with End-to-End Learnable Item Tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 729–739, 2025.

Changle Qu, Sunhao Dai, Ke Guo, Xiao Zhang, Liqin Zhao, Shijun Wang, Yanan Niu, Lantao Hu, Han Li, and Jun Xu. KuaiLive: A real-time interactive dataset for live streaming recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2026.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10299–10315, 2023.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. Language Representations Can Be What Recommenders Need: Findings and Potentials. In *International Conference on Learning Representations*, 2025.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019.

Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Jian Wu, and Yuning Jiang. Think Before Recommend: Unleashing the Latent Reasoning Power for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 38(9):6226–6239, 2026.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable Item Tokenization for Generative Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024a.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and Zhenhua Dong. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3245–3254, 2024b.

Yibiao Wei, Jie Zou, Pengfei Zhang, Xiao Ao, Weikang Guo, Zeyu Ma, and Yang Yang. CARD: Non-Uniform Quantization of Visual Semantic Unit for Generative Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1959–1970, 2026.

Kun Yang, Yuxuan Zhu, Yazhe Chen, Siyao Zheng, Bangyang Hong, Kangle Wu, Yabo Ni, Anxiang Zeng, Cong Fu, and Hui Li. ManCAR: Manifold-Constrained Latent Reasoning with Adaptive Test-Time Computation for Sequential Recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 5998–6009, 2026.

Yuhao Yang, Zhi Ji, Zhaopeng Li, Yi Li, Zhonglin Mo, Yue Ding, Kai Chen, Zijian Zhang, Jie Li, Shuanglong Li, and Lin Liu. Sparse Meets Dense: Unified Generative Recommendations with Cascaded Sparse-Dense Representations. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Runyang You, Yongqi Li, Xinyu Lin, Xin Zhang, Wenjie Wang, Wenjie Li, and Liqiang Nie. R^2ec: Towards Large Recommender Models with Reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Qihang Yu, Kairui Fu, Zheqi Lv, Shengyu Zhang, Xinhui Wu, Chen Lin, Feng Wei, Bo Zheng, and Fei Wu. ThinkRec: Thinking-based Recommendation via LLM. In *Proceedings of the ACM Web Conference 2026*, 2026.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 58484–58509, 2024.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 4320–4326, 2019.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. CoLLM: Integrating Collaborative Embeddings Into Large Language Models for Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2329–2340, 2025.

Yang Zhang, Wenxin Xu, Xiaoyan Zhao, Wenjie Wang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Reinforced Latent Reasoning for LLM-based Recommendation. In *International Conference on Learning Representations*, 2026.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. In *2024 IEEE 40th International Conference on Data Engineering*, pp. 1435–1448, 2024.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S^3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 1893–1902, 2020.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-Enhanced MLP Is All You Need for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022*, pp. 2388–2399, 2022.

A Theoretical Analysis of Collaborative Guidance

A.1 Distributional Setup

Fix a semantic ID level l and a distribution over history–prefix examples. Let X denote the available context, Y the observed next code, $Q = q^{\mathrm{col}}_{u,l}(X)$ the collaborative reference over the full vocabulary $\mathcal{C}_l$, and $Z = h_{u,l}(X)$ the hidden state used for evaluation. These are local random-variable versions of the notation in Sections 2–3.3. Q is a random probability vector, not a discrete class label. All logarithms are natural and $0 \log 0 = 0$. A readout is required to be positive wherever its reference has positive mass. The reference is a fixed teacher estimate. None of the results below assumes that it equals the population next-item distribution.

The action ambiguity $\mathcal{A}_l$ quantifies reference variation among examples sharing an observed code:

$$\mathcal{A}_l = \mathbb{E} D_{\mathrm{KL}} \left(Q \| \mathbb{E}[Q \mid Y] \right), \tag{12}$$

where $\mathcal{A}_l$ is the action-conditioned reference variation, $Y = Y_{u,l}$ is the random observed semantic ID code, and $\mathbb{E}[Q \mid Y]$ is its conditional mean reference. It is positive when users sharing an observed action retain different collaborative references.

A.2 Proof of Recoverability Decomposition

Corollary A.1 (Action-conditioned reference ambiguity). For reference $Q = q^{\mathrm{col}}_{u,l}$ and action $Y = Y_{u,l}$, consider a state $Z = \psi(Y)$ that retains only action information. Its optimal reconstruction risk $R^\star(Z)$ obeys the lower bound:

$$R^\star(Z) \geq R^\star(Y) = \mathbb{E} D_{\mathrm{KL}}(Q \| \mathbb{E}[Q \mid Y]) = \mathcal{A}_l, \tag{13}$$

where $R^\star(Z) = \inf_r \mathbb{E} D_{\mathrm{KL}}(Q \| r(Z))$. The bound is positive whenever Q varies within an action class of positive probability.

Lemma A.1 (Recoverability decomposition). Let Q be a random collaborative reference, Z a state, and $\overline{Q}_Z = \mathbb{E}[Q \mid Z]$ its conditional mean. For a measurable readout r with compatible support and finite risk, the expected divergence $\mathbb{E} D_{\mathrm{KL}}(Q \| r(Z))$ admits the exact decomposition:

$$\mathbb{E} D_{\mathrm{KL}}(Q \| r(Z)) = \mathbb{E} D_{\mathrm{KL}}(Q \| \overline{Q}_Z) + \mathbb{E} D_{\mathrm{KL}}(\overline{Q}_Z \| r(Z)). \tag{14}$$

This conditional-mean identity follows from the KL/Bregman decomposition and the strict propriety of logarithmic loss (Banerjee et al., 2005; Cover & Thomas, 2006; Gneiting & Raftery, 2007).

Proof. Write $\overline{Q}_Z = \mathbb{E}[Q \mid Z]$. We expand the expected KL risk $\mathbb{E} D_{\mathrm{KL}}(Q \| r(Z))$ by adding and subtracting $\overline{Q}_Z$ inside the logarithm:

$$\begin{aligned} \mathbb{E} D_{\mathrm{KL}}(Q \| r(Z)) &= \mathbb{E} \sum_c Q(c) \log \frac{Q(c)}{\overline{Q}_Z(c)} + \mathbb{E} \sum_c Q(c) \log \frac{\overline{Q}_Z(c)}{r(Z)(c)} \\ &= \mathbb{E} D_{\mathrm{KL}}(Q \| \overline{Q}_Z) + \mathbb{E} D_{\mathrm{KL}}(\overline{Q}_Z \| r(Z)), \end{aligned} \tag{15}$$

where c ranges over $\mathcal{C}_l$, $Q(c)$ is the reference mass, and $r(Z)(c)$ is the readout probability. The second term vanishes at the conditional mean, giving the minimum risk $R^\star(Z)$:

$$R^\star(Z) := \inf_r \mathbb{E} D_{\mathrm{KL}}(Q \| r(Z)) = \mathbb{E} D_{\mathrm{KL}}(Q \| \overline{Q}_Z), \tag{16}$$

where $R^\star(Z)$ is the minimum expected reconstruction risk and the infimum ranges over all measurable readouts with compatible support and finite risk. If a readout observes only Y, its optimum is $\overline{Q}_Y = \mathbb{E}[Q \mid Y]$, whose risk is $\mathcal{A}_l$. If $Z = \psi(Y)$ for an action-encoding function ψ, the recoverability risk is no smaller than $\mathcal{A}_l$. Thus, correct action prediction does not imply collaborative sufficiency. The claim does not say that every action-trained representation retains only the action.

Proof of Theorem 1. Sample C independently conditional on X, so that $C \perp Y \mid X$ and $P(C = c \mid X, Y) = Q(c)$. Since $Z = f(X)$, its conditional law is $P(C = c \mid Z, Y) = \mathbb{E}[Q(c) \mid Z, Y]$. Ambiguity $\mathcal{A}_l$ and residual risk $\mathcal{G}_l^\star$ can be expressed through the entropy identities:

$$\mathcal{A}_l = H(C \mid Y) - \mathbb{E}H(Q) = I(C; X \mid Y), \tag{17}$$

$$\mathcal{G}_l^\star = H(C \mid Z, Y) - \mathbb{E}H(Q) = I(C; X \mid Z, Y). \tag{18}$$

For finite-risk $r_l(Z_l, Y_l)$, let $\overline{Q}_{Z,Y} = \mathbb{E}[Q_l \mid Z_l, Y_l]$ be the conditional mean reference. The KL risk $\mathbb{E}D_{\mathrm{KL}}(Q_l \| r_l(Z_l, Y_l))$ decomposes:

$$\mathbb{E}D_{\mathrm{KL}}(Q_l \| r_l(Z_l, Y_l)) = \mathcal{G}_l^\star + \mathbb{E}D_{\mathrm{KL}}(\overline{Q}_{Z,Y} \| r_l(Z_l, Y_l)). \tag{19}$$

Subtracting the entropy identities gives Eq. 4, equivalently the conditional mutual-information chain rule for a deterministic state (Cover & Thomas, 2006). The preceding application of Lemma A.1 proves Eq. 19. The unconditional operational risk in Eq. 2 instead decomposes around $\mathbb{E}[Q \mid Z]$. Both intrinsic risks allow unrestricted measurable readouts. An injective state can have zero intrinsic risk even when a finite probe cannot reconstruct its reference. Finite-probe ordering may therefore differ from intrinsic-risk ordering.

A.3 Within-Action Shuffle Control

Let $\widetilde{Q} \mid Y \sim P(Q \mid Y)$ with $\widetilde{Q} \perp Z \mid Y$, and draw $\widetilde{C} \mid \widetilde{Q}, Z, Y \sim \widetilde{Q}$. For the real pairing, draw $C \mid X \sim Q(X)$. Both ideal readouts observe (Z, Y). The shuffled reference preserves the conditional distribution given Y, while $\widetilde{C} \perp Z \mid Y$. Their optimal risks $R^\star_{\mathrm{real}}$ and $R^\star_{\mathrm{shuffle}}$ therefore satisfy:

$$\begin{aligned} R^\star_{\mathrm{real}} &= \mathrm{H}(C \mid Z, Y) - \mathbb{E}\mathrm{H}(Q), \\ R^\star_{\mathrm{shuffle}} &= \mathrm{H}(\widetilde{C} \mid Y) - \mathbb{E}\mathrm{H}(\widetilde{Q}) = \mathrm{H}(C \mid Y) - \mathbb{E}\mathrm{H}(Q), \end{aligned} \tag{20}$$

where H is Shannon entropy, C and $\widetilde{C}$ are the real and shuffled auxiliary codes, and $\perp$ denotes conditional independence here. The equality uses $(\widetilde{Q}, Y) \overset{d}{=} (Q, Y)$ and $(\widetilde{C}, Y) \overset{d}{=} (C, Y)$, where $\overset{d}{=}$ denotes equality in distribution. Their difference $R^\star_{\mathrm{shuffle}} - R^\star_{\mathrm{real}}$ gives the shuffle penalty:

$$\boxed{R^\star_{\mathrm{shuffle}} - R^\star_{\mathrm{real}} = \mathrm{I}(C; Z \mid Y)}, \tag{21}$$

where $\mathrm{I}(C; Z \mid Y)$ is the conditional mutual information between code C and state Z given the observed action. A finite linear probe with JS (Lin, 1991) or KL loss is not literally conditional mutual information. The practical shuffle experiment should therefore be interpreted as a controlled recoverability contrast within each action class.

A.4 Proof of Preference–Action Target Coupling

For a fixed level, let $y = Y_{u,l}$ be the observed code and $q = q^{\mathrm{col}}_{u,l}$ its collaborative reference. A categorical output a incurs the shared-output loss $\ell_{\mathrm{direct}}(a)$:

$$\ell_{\mathrm{direct}}(a) = -\log a_y + \eta D_{\mathrm{KL}}(q \| a), \tag{22}$$

where $a_y = a(y)$ is the observed-code probability and $\eta > 0$ is the effective preference weight. Expanding $\ell_{\mathrm{direct}}(a)$ and regrouping its logarithms gives:

$$\begin{aligned} \ell_{\mathrm{direct}}(a) &= -\sum_c [\delta_y(c) + \eta q(c)] \log a_c - \eta \mathrm{H}(q) \\ &= (1 + \eta)\,[\mathrm{H}(m_\eta) + D_{\mathrm{KL}}(m_\eta \| a)] - \eta \mathrm{H}(q), \end{aligned} \tag{23}$$

where $m_\eta = (\delta_y + \eta q)/(1 + \eta)$ is the mixture target and $\delta_y(c) = \mathbf{1}[c = y]$. Minimization gives the optimizer $a^\star$ and its hard-action penalty $-\log a^\star_y$:

$$a^\star = \frac{\delta_y + \eta q}{1 + \eta}, \qquad -\log a^\star_y = \log \frac{1 + \eta}{1 + \eta q_y}. \tag{24}$$

Set $\omega = 1/(1 + \eta)$. The minimum shared-output loss $\Gamma_\eta(q, y)$ takes the form:

$$\Gamma_\eta(q, y) = (1 + \eta)\,\mathrm{JS}_{1/(1+\eta)}(\delta_y, q), \qquad \Gamma_\eta(q, y) = 0 \iff q = \delta_y, \tag{25}$$

where $\mathrm{JS}_\omega(P, Q) = \omega D_{\mathrm{KL}}(P \| \omega P + (1 - \omega)Q) + (1 - \omega) D_{\mathrm{KL}}(Q \| \omega P + (1 - \omega)Q)$ is weighted Jensen–Shannon divergence. Equations 24–25 prove the shared-output result.

With separate action and preference distributions a and b, the loss $\ell_{\text{sep}}(a, b) = -\log a_y + \eta D_{\text{KL}}(q\|b)$ reaches zero in the closure at $(a, b) = (\delta_y, q)$. For the shared output, the optimal hard-label loss $\ell_y^\star$ and its derivative with respect to disagreement κ obey:

$$\ell_y^\star = \log \frac{1+\eta}{1+\eta q_y}, \qquad \frac{\mathrm{d}\ell_y^\star}{\mathrm{d}\kappa} = \frac{\eta e^{-\kappa}}{1+\eta e^{-\kappa}} > 0, \quad \kappa = -\log q_y, \tag{26}$$

where $\ell_y^\star$ is the hard-action loss at the shared-output optimum and $\kappa = -\log q_y$ measures reference–action disagreement. The derivative assumes $q_y > 0$. The boundary case follows by continuity. This is a pointwise loss identity, not a claim about the exact optimum of a neural network. Direct alignment is not universally inferior, and η is the effective relative loss weight after reduction. In the Direct implementation, both losses are averaged over levels, and we set $\eta = \lambda_{\text{pref}}$ to match the relative loss scale in Eq. 8.

Corollary A.2 (Conflict-dependent action penalty). For reference–action disagreement $\kappa = -\log q_y \geq \tau \geq 0$, the optimal hard-action loss $-\log a_y^\star$ obeys:

$$-\log a_y^\star \geq \log \frac{1+\eta}{1+\eta e^{-\tau}}. \tag{27}$$

The bound follows from the monotonicity in Eq. 26.

Population-risk interpretation. Let $p_x(c) = \Pr(Y = c \mid X = x)$ be the action posterior and q_x the reference. Minimizing expected action loss plus $\eta D_{\text{KL}}(q_x\|a_x)$ over categorical outputs yields the optimum $a_x^\star$ and its excess population log loss:

$$a_x^\star = \frac{p_x + \eta q_x}{1+\eta}, \qquad \mathbb{E}_{Y|x}[-\log a_x^\star(Y)] - H(p_x) = D_{\text{KL}}(p_x\|a_x^\star). \tag{28}$$

The excess population log loss vanishes when $q_x = p_x$, including soft references. Thus shared-output coupling changes the population optimum when the collaborative reference differs from the action posterior. Dedicated readouts allow these two distributions to be represented separately. The empirical comparisons measure the resulting behavior under finite data and shared neural parameters.

A.5 State Alignment and Generated-Prefix Extension

Proof of Lemma 1. Under supervised context distribution ν_l, let $\epsilon_l = \mathbb{E}_{\nu_l} D_{\text{KL}}(q_{u,l}^{\text{col}}\|\pi_{u,l}^{\text{col}}) < \infty$. For bounded utility $V_l : \mathcal{C}_l \to [0, 1]$, define $e_l = |\mathbb{E}_{q_{u,l}^{\text{col}}} V_l - \mathbb{E}_{\pi_{u,l}^{\text{col}}} V_l|$. For $V_l(c) \in [0, 1]$, Pinsker's inequality implies the pointwise utility control:

$$\left|\mathbb{E}_{q_{u,l}^{\text{col}}} V_l - \mathbb{E}_{\pi_{u,l}^{\text{col}}} V_l\right| \leq \text{TV}(q_{u,l}^{\text{col}}, \pi_{u,l}^{\text{col}}) \leq \sqrt{\frac{1}{2} D_{\text{KL}}(q_{u,l}^{\text{col}}\|\pi_{u,l}^{\text{col}})}, \tag{29}$$

where $\text{TV}(P, Q) = \frac{1}{2}\sum_c |P(c) - Q(c)|$ is total variation distance. Taking expectations and applying Cauchy–Schwarz gives Eq. 9. Let μ_l and ν_l denote generated- and supervised-prefix distributions, with shift $\Delta_l^{\text{ctx}} = \text{TV}(\mu_l, \nu_l)$. Under generated prefixes, the average discrepancy $L^{-1}\sum_l \mathbb{E}_{\mu_l} e_l$ obeys:

$$\frac{1}{L}\sum_{l=1}^{L} \mathbb{E}_{\mu_l} e_l \leq \sqrt{\frac{1}{2L}\sum_{l=1}^{L} \epsilon_l} + \frac{1}{L}\sum_{l=1}^{L} \Delta_l^{\text{ctx}}, \tag{30}$$

where e_l and ϵ_l are the bounded utility discrepancy and expected alignment KL defined in Lemma 1. Pinsker's inequality and Jensen's inequality give $\mathbb{E}_{\nu_l} e_l \leq \sqrt{\epsilon_l/2}$. Since $0 \leq e_l \leq 1$, changing the context distribution adds at most $\text{TV}(\mu_l, \nu_l)$. Averaging over levels gives the displayed bound. The two terms separate supervised state alignment from generated-prefix distribution shift.

A.6 Why Marginal Alignment Does Not Determine Item Ranking

Proof of Proposition 1. To exhibit the information omitted by level-wise marginals, consider IDs $(0, 0), (0, 1), (1, 0), (1, 1)$ and two item distributions ξ and ξ' in that order:

$$\xi = (0.4, 0.1, 0.1, 0.4), \qquad \xi' = (0.1, 0.4, 0.4, 0.1), \tag{31}$$

where ξ and ξ' are illustrative item distributions in the listed semantic ID order, rather than additional model predictions. Both have identical level-wise marginals $(0.5, 0.5)$ at every level, but they reverse

the ordering of $(0, 0)$ and $(0, 1)$. Level-wise marginal recoverability therefore does not determine complete-item ranking, motivating collaborative leaf calibration.

A.7 Variational Properties of Collaborative Leaf Calibration

For fixed candidates, let g_u denote the generation distribution on $\mathcal{B}_u$ and $\mathcal{Z}_u(\alpha)$ its normalizer. The calibrated distribution $\rho_u^\alpha(j)$ takes the normalized exponential form:

$$\rho_u^\alpha(j) = \frac{g_u(j)\exp(\alpha S_{\text{leaf}}(u, j))}{\mathcal{Z}_u(\alpha)}, \tag{32}$$

where $\mathcal{Z}_u(\alpha) > 0$. For any $\rho \in \Delta(\mathcal{B}_u)$, the regularized utility satisfies the identity:

$$\alpha\mathbb{E}_{j\sim\rho}[S_{\text{leaf}}(u, j)] - D_{\text{KL}}(\rho\|g_u) = \log \mathcal{Z}_u(\alpha) - D_{\text{KL}}(\rho\|\rho_u^\alpha), \tag{33}$$

where both KL terms use the fixed candidate support. Nonnegativity of KL establishes ρ_u^α as the unique maximizer in Eq. 10, ordered by $S_{\text{gen}} + \alpha S_{\text{leaf}}$. Differentiating its expected memory score with respect to α yields:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathbb{E}_{j\sim\rho_u^\alpha}[S_{\text{leaf}}(u, j)] = \text{Var}_{j\sim\rho_u^\alpha}[S_{\text{leaf}}(u, j)] \geq 0, \tag{34}$$

where $\text{Var}_{j\sim\rho_u^\alpha}$ is variance under the calibrated distribution. An initially lower-scoring item j overtakes item i precisely when the weighted memory advantage satisfies:

$$\alpha\left(S_{\text{leaf}}(u, j) - S_{\text{leaf}}(u, i)\right) > S_{\text{gen}}(u, i) - S_{\text{gen}}(u, j), \tag{35}$$

where $i, j \in \mathcal{B}_u$ are two candidate items and the strict inequality describes a rank reversal, with equality giving a tie. Calibration preserves the candidate set and full-beam Recall, but it does not by itself imply that NDCG improves.

Proof of Corollary 1. For a fixed set of distinct candidates $\mathcal{B}_u$, finite generation and memory scores, and finite α, $g_u(j) > 0$ and $\exp(\alpha S_{\text{leaf}}(u, j)) > 0$ for every $j \in \mathcal{B}_u$. Thus $\text{supp}(\rho_u^\alpha) = \mathcal{B}_u$. A permutation of this set preserves target membership and Recall at any cutoff covering the complete set, in particular Recall@50 when $|\mathcal{B}_u| \leq 50$. Empirical evaluation determines the resulting smaller-cutoff ranking quality.

B Experimental Details

B.1 Datasets

We evaluate Industrial and Scientific, Musical Instruments, and Video Games from Amazon Reviews 2023 under the Latte preprocessing protocol (Hou et al., 2026).

Table B.1 gives the published preprocessing reference. These domains span 50,985–94,762 users and average sequence lengths of 8.10–8.91, covering varied catalog and interaction scales.

Table B.1: Amazon Reviews 2023 statistics under the Latte preprocessing protocol (Hou et al., 2026). These published counts provide the protocol reference for the CoLaGR preprocessing pipeline.

Dataset	Users	Items	Interactions	Avg. length
Industrial and Scientific	50,985	25,848	412,947	8.10
Musical Instruments	57,439	24,587	511,836	8.91
Video Games	94,762	25,612	814,586	8.60

B.2 Baselines

We compare with five sequential and seven generative recommenders in Table 1. Each method is summarized below.

SASRec (Kang & McAuley, 2018): Uses causal self-attention to model dependencies in a user's interaction history. It scores candidate items using the resulting sequence representation.

FMLP-Rec (Zhou et al., 2022): Applies learnable filters in the frequency domain to process interaction sequences. Filtering and feed-forward layers transform historical interactions into representations for predicting the next item.

HSTU (Zhai et al., 2024): Uses hierarchical sequential transduction units to model user action sequences. The stacked units aggregate historical interactions into representations for recommendation.

FDSA (Zhang et al., 2019): Models item identities and item features through separate self-attention streams. Combining these representations captures sequential patterns in both user interactions and associated item attributes.

S^3-Rec (Zhou et al., 2020): Learns sequential representations with mutual-information-based self-supervised objectives. The objectives connect attributes, items, and sequence contexts to support next-item prediction.

TIGER (Rajput et al., 2023): Encodes items as semantic IDs through residual vector quantization. An autoregressive generator predicts the next item's code sequence from the user's interaction history.

LETTER (Wang et al., 2024a): Learns an item tokenizer that combines semantic representation, collaborative alignment, and code diversity. The resulting identifiers provide the discrete item representation for generative recommendation.

ActionPiece (Hou et al., 2025b): Constructs context-dependent tokens for user actions. Its tokenization incorporates interaction context into the discrete units used by the recommendation generator.

TCA4Rec (Lin et al., 2026): Converts item-level collaborative scores into token-level soft targets. Aligning next-token predictions with these targets introduces collaborative information into the generation objective.

MemGen-GR (Ding et al., 2026a): Combines generative and conventional recommendation scores to exploit their complementary memorization and generalization behavior. Score fusion brings both sources of evidence into the final item ranking.

Latte (Hou et al., 2026): Generates a latent reasoning token before the target semantic ID. Conditioning subsequent code generation on this token introduces intermediate computation and diversifies generation paths.

S^2GR (Guo et al., 2026c): Interleaves continuous thinking states with semantic ID prediction. Stepwise semantic supervision guides each state before the generator predicts the corresponding item code.

B.3 Implementation Details

Comparison setup. We follow Latte's comparison protocol (Hou et al., 2026). The local TCA4Rec, S^2GR, and MemGen-GR adaptations use the shared splits and evaluator. TCA4Rec transfers its objective to the T5/SID backbone instead of its original LLM/LoRA architecture; S^2GR uses fixed semantic IDs instead of training its own tokenizer. Table B.3 gives the shared and adaptation-specific settings.

Teacher preference export. The frozen sequential recommender is SASRec with two self-attention blocks, one head, hidden size 64, maximum sequence length 128, and dropout 0.1. It is trained for 200 epochs with Adam ($\mathrm{lr} = 10^{-3}$, $\beta = (0.9, 0.98)$) and binary cross-entropy. The final-epoch checkpoint is frozen for export. The exporter scores the item vocabulary and retains $M = 200$ complete-item candidates for each available history. Their logits are converted to preference masses with temperature 2.0. Training references supply generator supervision. Diagnostic readouts use training references for fitting, validation KL for selection, and test references for evaluation. CoLaGR constructs $q^{\mathrm{col}}_{u,l}$ by aggregating these candidates according to their level-l semantic ID codes. The exporter removes unmapped items and renormalizes retained mass. An export with no positive valid mass falls back to a uniform distribution over each level vocabulary. Held-out labels serve as evaluation metadata, not reference-construction inputs. Generator inference does not query the teacher.

The teacher audit in Table B.2 evaluates this frozen export checkpoint; the SASRec row in Table 1 reports the comparison result under the baseline protocol.

Table B.2 records the teacher evidence used by the released references. All exported items map to semantic IDs and no example invokes uniform fallback. The nonzero zero-target rates show that the

Table B.2: Frozen-teacher and collaborative-reference audit on the test split. Zero-target reports the percentage of examples with $q^{\mathrm{col}}_{u,l}(Y_{u,l}) = 0$.

Dataset	Frozen teacher		Collaborative reference				
	N@10	R@10	Unmapped	Uniform	Zero L1	Zero L2	Zero L3
Industrial	0.0110	0.0213	0.0%	0.0%	28.0%	20.8%	28.2%
Musical	0.0110	0.0225	0.0%	0.0%	24.4%	20.9%	26.9%
Video Games	0.0226	0.0462	0.0%	0.0%	20.6%	18.0%	21.4%

soft collaborative reference can disagree with the observed hard action, which motivates keeping reference reconstruction and action prediction in separate readouts.

Semantic ID assets and trie constraints. The semantic ID tokenizer, item-to-code mapping, level vocabularies, and valid-prefix trie are fixed across locally controlled variants. Training uses cross-entropy on the full level vocabulary, averaged over the batch and then over levels. The collaborative KL loss uses the same batch/level averaging. Each preference head has a learned bias. Distinct CP token IDs occupy decoder positions $2l - 1$ under zero-based indexing, and these states predict the following semantic ID codes. During decoding, a prefix-trie mask removes codes that cannot extend the current prefix to a valid item. The level-wise reference aggregates the complete represented top-M set independently of the generated prefix.

Collaborative memory. Collaborative leaf calibration scores complete items using transitions in the training data. For each sequence $s = (i^s_1, \ldots, i^s_{|s|})$, the memory entry $\mathcal{M}(i, j)$ accumulates distance-weighted transitions according to:

$$\mathcal{M}(i,j) = \sum_s \sum_{a<b} \frac{\mathbf{1}[i^s_a = i,\, i^s_b = j,\, 1 \le b - a \le w]}{b - a}, \tag{36}$$

where s ranges over training sequences, $a, b \in \{1, \ldots, |s|\}$ are positions with $a < b$, and $w = 5$ is the window. The 200 largest outgoing entries are retained per source item. Let $r_{u,t}$ be history age, with zero for the newest retained item, and $\gamma_{u,t} = 1/(1 + r_{u,t})$. The item utility $v_u(j)$ and normalized leaf score $S_{\mathrm{leaf}}(u, j)$ are computed by:

$$\begin{aligned} v_u(j) &= \mathbf{1}[j \notin \mathcal{I}(H_u)] \sum_{t \in \mathcal{J}_u} \gamma_{u,t} \mathcal{M}(i^u_t, j), \\ S_{\mathrm{leaf}}(u,j) &= \begin{cases} v_u(j) / \max_{k \in \mathcal{K}_u} v_u(k), & j \in \mathcal{K}_u,\ \max_{k \in \mathcal{K}_u} v_u(k) > 0, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{37}$$

where $\mathcal{I}(H_u)$ contains history items, $\mathcal{J}_u$ contains retained history positions, and $\mathcal{K}_u$ is the retained top-50 memory support. A generated item outside this support receives zero leaf score, and memory-only items are not added to the beam.

Computational cost. Let B be the beam width and L the semantic ID depth. CoLaGR inserts $O(L)$ CP positions. Collaborative leaf calibration computes and fuses one memory score per generated item in $O(B)$ time. Sorting costs $O(B \log B)$, excluding the configured transition lookup.

Table B.3 summarizes the settings shared by the local comparisons. The semantic ID mapping and candidate evaluator remain fixed, while validation data determine the dataset-specific supervision and calibration strengths. Preference guidance is applied during generator training. Memory scoring is applied after candidate generation.

These stages address different points in the generation process. The frozen teacher supplies a distribution over alternatives at each semantic ID level, allowing users with the same observed action to receive different preference targets. The separate readout grounds the intermediate state in these targets while retaining the hard action objective. At inference, the generator carries this learned preference information into decoding without querying the teacher. The memory then supplies evidence about complete items, whose relative order can remain unresolved by code-level preferences.

Table B.3: Shared local settings. Validation NDCG@10 selects each dataset-specific weight.

Field	Value
Backbone	T5 (Raffel et al., 2020): 4 encoder and 4 decoder layers, hidden size 128, feed-forward size 1024
Attention / dropout	6 heads, key/value dimension 64, dropout 0.1
Frozen teacher	SASRec: 2 blocks, 1 head, width 64, max length 128; 200 epochs
Teacher optimization	Adam, learning rate 0.001; $\beta = (0.9, 0.98)$, binary cross-entropy
Teacher export	Top-M with $M = 200$, temperature 2.0, final-epoch frozen checkpoint
Semantic ID tokenizer	Fixed residual k-means mapping; 3 codebooks with configured size 256
History / batch sizes	At most 20 items, training 256, evaluation 128
Beam width B	50
Prefix-trie mask	Shared across controlled CoLaGR variants
TCA4Rec adaptation	Base-initialized T5/SID; selected $(\alpha, \tau) = (0.05, 1)$
S^2GR adaptation	Fixed IDs; $K = 64$, temperature 0.1, regularization 1.0
MemGen-GR adaptation	Base and conventional item scores; selected $\alpha_{\text{ens}} = 0.3$
λ_{pref}	1.75 / 1.0 / 0.75 for Industrial / Musical / Video Games
Calibration α	3.0 / 1.5 / 3.0 for Industrial / Musical / Video Games
Collaborative leaf calibration	Candidate-preserving top-B reranking from training-data memory
Optimizer / schedule	AdamW, learning rate 0.003, weight decay 0.05; cosine decay with 10,000 warmup steps
Training / selection	150 epochs, validation every 3 epochs, select by validation NDCG@10
Memory retrieval	Window 5, at most 200 neighbors, last 10 history items, top-50 support

Calibration acts on generated candidates. Memory-supported candidates receive an additional score; unsupported candidates retain their generator scores. History items receive no memory contribution, and beam membership stays fixed. The fixed-candidate experiment therefore tests item ordering independently of candidate generation.

C Additional Experimental Results

C.1 Protocol and Statistical Scope

We evaluate four seed-2024 generators under observed prefixes with shared level-wise references. The test sets contain 50,985 Industrial, 57,439 Musical, and 94,762 Video Games examples, each supplying three pre-action states. This protocol measures supervised-context recoverability, separately from free-running prefix errors.

Readout fitting and uncertainty. We fit independent affine–softmax readouts on frozen training states, select by validation KL, and evaluate on test examples using a common architecture. All 36 fits select learning rate 0.003 and the epoch-25 budget limit, so the comparison measures common-budget recoverability. The 95% bootstrap intervals use 1,000 example resamples conditional on fitted predictions. Shuffle SDs describe permutation variability. Generator-seed variation is reported separately, and aggregate plots show point estimates.

C.2 Hyperparameter Sensitivity

Table C.1 reports validation NDCG@10 for the supervision-weight sweep. The selected weights are 1.75, 1.0, and 0.75 for Industrial, Musical, and Video Games. The domain-specific optima motivate selecting the supervision strength separately for each dataset.

Table C.2 places all selected calibration weights inside the evaluated range: 3.0 for Industrial and Video Games, and 1.5 for Musical. Larger coefficients do not improve the selected validation score under cumulative path-score calibration.

Tables C.1 and C.2 show dataset-specific supervision optima and calibration gains that flatten or decline at larger weights. Both coefficients are selected by validation NDCG@10. State supervision changes the generator during training, whereas calibration reorders a fixed candidate beam after decoding. Thus, their validation optima represent distinct uses of collaborative evidence.

Table C.1: Validation NDCG@10 across supervision weights. Bold marks the selected weights.

λ_{pref}	Industrial	Musical	Video Games
0.25	0.030560	0.036552	0.059137
0.5	0.030897	0.037499	0.059806
0.75	0.031756	0.037987	**0.060076**
1	0.032739	**0.039435**	0.059953
1.75	**0.033745**	0.038763	0.059933
2.5	0.033128	0.039044	0.059411
3.25	0.033489	0.038315	0.058608
4	0.033094	0.038639	0.057748
4.75	0.032789	0.038046	0.057310

Table C.2: Complete CoLeaf validation sweep under cumulative path-score calibration. Bold marks the validation-selected coefficient. The same evaluated grid is used in all three domains.

α	Industrial	Musical	Video	α	Industrial	Musical	Video
0	0.033745	0.039435	0.060076	1	0.037348	0.042912	0.067450
0.005	0.033765	0.039458	0.060122	1.5	0.038347	**0.043272**	0.068717
0.01	0.033773	0.039493	0.060169	2	0.038779	0.043249	0.069323
0.02	0.033812	0.039531	0.060229	3	**0.039202**	0.043160	**0.069807**
0.05	0.033958	0.039691	0.060547	4	0.039179	0.042775	0.069538
0.1	0.034198	0.039961	0.061072	5	0.038944	0.042292	0.069204
0.2	0.034664	0.040557	0.062001	6	0.038967	0.042049	0.068926
0.4	0.035544	0.041396	0.063714	7	0.038869	0.041818	0.068641
0.6	0.036303	0.042097	0.065193	8	0.038823	0.041666	0.068340
0.8	0.036846	0.042689	0.066420				

Table C.3: Efficiency of Base and CoLaGR-G. Train denotes hours for 150 epochs, memory is incremental peak GiB, and test throughput is users/s.

Dataset	Train (h)			Test (users/s)		Memory (GiB)	CoLeaf (s)
	Base	CoLaGR-G	Δ (%)	Base	CoLaGR-G	Base / CoLaGR-G	
Industrial	1.86	2.01	+8.0%	485.6	451.2	4.92 / 5.78	43
Musical	2.39	2.57	+7.2%	517.5	474.7	4.92 / 6.00	53
Video Games	3.28	3.54	+7.8%	672.1	603.6	4.89 / 6.52	80

C.3 Efficiency

Table C.3 times one seed-2024 run per model: 150 epochs, validation every three epochs, batches 256/128, and beam width 50. Models run sequentially after an idle-GPU check. Command-level times include periodic evaluation for training and checkpoint/data loading for testing. Training-memory peaks subtract the pre-command device baseline. Separate CoLeaf timing includes memory construction and file I/O on exported candidates.

Table C.3 shows that CoLaGR-G adds 7.2%–8.0% training time. Throughput drops from 485.6/517.5/672.1 to 451.2/474.7/603.6 users/s on Industrial/Musical/Video Games. CoLeaf adds 43–80 seconds per split and is timed separately from beam generation.

C.4 Teacher-Family Robustness

We replace the frozen collaborative teacher with GRU4Rec and BERT4Rec (Hidasi et al., 2016; Sun et al., 2019) to test whether the downstream latent guidance depends on one teacher family, and include the SASRec teacher used by the main experiments as the reference. The alternative teachers use their native sequential objectives, export their top-200 unseen items under the same protocol, and construct level-wise collaborative references with temperature 2.0. The downstream generator is unchanged: Industrial and Scientific, the same semantic ID assets and split, $\lambda_{\text{pref}} = 1.75$, seed 2024, and the same validation-based checkpoint protocol. This isolates teacher-family variation while keeping the generator-side comparison fixed.

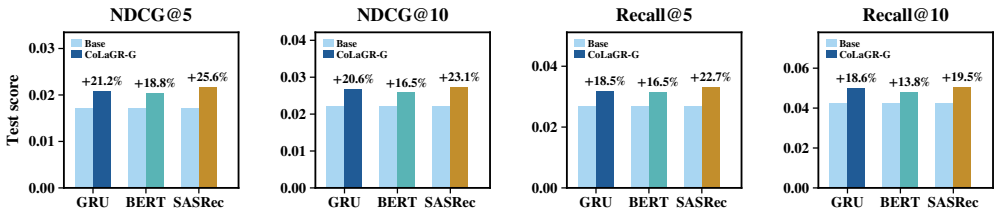


Figure C.1: Teacher-family robustness on Industrial and Scientific. Bars compare Base and CoLaGR-G, and labels show percentage gains over Base.

Table C.4: Teacher-family comparison on Industrial and Scientific. N and R denote NDCG and Recall, and bold highlights CoLaGR-G for each teacher.

Teacher	Variant	N@5	N@10	R@5	R@10
GRU4Rec	Base	0.0172	0.0221	0.0269	0.0421
	CoLaGR-G	**0.0208**	**0.0266**	**0.0319**	**0.0499**
BERT4Rec	Base	0.0172	0.0221	0.0269	0.0421
	CoLaGR-G	**0.0204**	**0.0257**	**0.0313**	**0.0479**
SASRec (default teacher)	Base	0.0172	0.0221	0.0269	0.0421
	CoLaGR-G	**0.0216**	**0.0272**	**0.0330**	**0.0503**

Table C.5: Complete seed-2024 component results. N and R denote NDCG and Recall.

Dataset	Variant	N@5	N@10	R@5	R@10
Industrial	Base	0.0172	0.0221	0.0269	0.0421
Industrial	Latent-only	0.0183	0.0236	0.0284	0.0450
Industrial	Direct	0.0187	0.0239	0.0285	0.0446
Industrial	Native-State Grounding	0.0214	0.0269	0.0330	0.0502
Industrial	CoLaGR-G	0.0216	0.0272	0.0330	0.0503
Industrial	CoLaGR	**0.0257**	**0.0309**	**0.0374**	**0.0536**
Musical	Base	0.0251	0.0319	0.0382	0.0593
Musical	Latent-only	0.0254	0.0324	0.0388	0.0608
Musical	Direct	0.0261	0.0333	0.0393	0.0616
Musical	Native-State Grounding	0.0272	0.0346	0.0410	0.0642
Musical	CoLaGR-G	0.0281	0.0353	0.0425	0.0648
Musical	CoLaGR	**0.0324**	**0.0395**	**0.0472**	**0.0694**
Video Games	Base	0.0387	0.0495	0.0588	0.0925
Video Games	Latent-only	0.0392	0.0502	0.0594	0.0938
Video Games	Direct	0.0404	0.0510	0.0616	0.0946
Video Games	Native-State Grounding	0.0412	0.0522	0.0623	0.0967
Video Games	CoLaGR-G	0.0414	0.0527	0.0629	0.0980
Video Games	CoLaGR	**0.0510**	**0.0618**	**0.0740**	**0.1073**

Table C.4 and Figure C.1 compare each teacher-conditioned CoLaGR-G generator with the shared Base control. NDCG@10 gains are 20.6%, 16.5%, and 23.1% for GRU4Rec, BERT4Rec, and SASRec. The corresponding Recall@10 gains are 18.6%, 13.8%, and 19.5%. These improvements support robustness across the tested teacher families on Industrial and Scientific.

C.5 Complete Ranking Results and Repeated Runs

Table C.5 reports component results at both cutoffs under shared splits and semantic ID assets. CoLaGR-G improves on Latent-only at both cutoffs in every domain, and CoLaGR further improves ranking. The same component-level pattern holds at both ranking cutoffs.

Table C.6 shows consistent gains across training seeds. CoLaGR-G raises mean NDCG@10 from 0.0230 to 0.0271 on Industrial, from 0.0317 to 0.0347 on Musical, and from 0.0494 to 0.0528 on

Table C.6: Three-seed results (2024–2026), reported as mean ± seed SD. For CoLaGR, each seed selects its CoLeaf coefficient on validation under cumulative-path calibration.

Dataset	Variant	NDCG@5	NDCG@10	Recall@5	Recall@10
Industrial	Base	0.0180 ± 0.0007	0.0230 ± 0.0008	0.0280 ± 0.0010	0.0436 ± 0.0013
Industrial	CoLaGR-G	0.0216 ± 0.0002	0.0271 ± 0.0003	0.0329 ± 0.0001	0.0498 ± 0.0004
Industrial	CoLaGR	**0.0259 ± 0.0002**	**0.0312 ± 0.0003**	**0.0373 ± 0.0002**	**0.0540 ± 0.0003**
Musical	Base	0.0249 ± 0.0002	0.0317 ± 0.0002	0.0380 ± 0.0005	0.0593 ± 0.0001
Musical	CoLaGR-G	0.0275 ± 0.0005	0.0347 ± 0.0005	0.0413 ± 0.0010	0.0638 ± 0.0010
Musical	CoLaGR	**0.0320 ± 0.0003**	**0.0390 ± 0.0005**	**0.0468 ± 0.0004**	**0.0684 ± 0.0011**
Video Games	Base	0.0386 ± 0.0001	0.0494 ± 0.0002	0.0591 ± 0.0003	0.0925 ± 0.0008
Video Games	CoLaGR-G	0.0417 ± 0.0002	0.0528 ± 0.0001	0.0632 ± 0.0003	0.0980 ± 0.0003
Video Games	CoLaGR	**0.0510 ± 0.0002**	**0.0618 ± 0.0002**	**0.0740 ± 0.0005**	**0.1074 ± 0.0004**

Table C.7: Structural-control results across seeds. Epoch denotes the validation-selected checkpoint, and metrics report generator test performance.

Dataset	Variant	Seed	Epoch	NDCG@5	NDCG@10	Recall@5	Recall@10
Industrial	Native-State Grounding	2025	138	0.021495	0.026909	0.032009	0.048936
Industrial	Native-State Grounding	2026	120	0.022133	0.027652	0.033245	0.050466
Musical	Native-State Grounding	2025	108	0.026414	0.033771	0.040042	0.062884
Musical	Native-State Grounding	2026	132	0.027590	0.034696	0.041783	0.063894
Video Games	Native-State Grounding	2025	150	0.041175	0.052339	0.062198	0.096885
Video Games	Native-State Grounding	2026	144	0.041454	0.052782	0.063010	0.098267

Table C.8: Collaborative grounding across state realizations. Native-State Grounding uses native pre-action states; CoLaGR-G uses decision-aligned CP states.

Dataset	Variant	N@5	N@10	R@5	R@10
Industrial	Native-State Grounding	0.0214	0.0269	0.0330	0.0502
	CoLaGR-G	0.0216	0.0272	0.0330	0.0503
Musical	Native-State Grounding	0.0272	0.0346	0.0410	0.0642
	CoLaGR-G	0.0281	0.0353	0.0425	0.0648
Video Games	Native-State Grounding	0.0412	0.0522	0.0623	0.0967
	CoLaGR-G	0.0414	0.0527	0.0629	0.0980

Video Games, with higher mean Recall@10 in every domain. CoLaGR further reaches 0.0312, 0.0390, and 0.0618 NDCG@10, respectively. All CoLaGR seeds preserve the generated 50-item candidate set and select α on validation data.

Table C.7 extends the Native-State Grounding placement control with seeds 2025 and 2026. Its NDCG@10 ranges are 0.0269–0.0277 on Industrial, 0.0338–0.0347 on Musical, and 0.0523–0.0528 on Video Games, showing consistent performance across these seeds.

C.6 State Realizations of Collaborative Grounding

Native-State Grounding applies the same two readouts at native pre-action states without CP tokens. Its comparable gains across domains (Table C.8) show that grounding extends beyond CP positions; CoLaGR uses them as a uniform decision-aligned interface.

C.7 Training-Time User–Reference Correspondence

We permute training references once within groups sharing the same complete observed SID, with one donor supplying all three levels to preserve cross-level consistency.

Validation and test remain matched. Over 99.7% of training references change in every domain. Table C.9 shows that breaking this correspondence consistently degrades recommendation.

Table C.9: Training-time user–reference correspondence. Drop denotes relative degradation from matched to within-SID shuffled supervision.

Dataset	NDCG@10			Recall@10		
	Matched	Shuffled	Drop	Matched	Shuffled	Drop
Industrial	0.027200	0.024612	9.5%	0.050250	0.046327	7.8%
Musical	0.035285	0.031398	11.0%	0.064817	0.058706	9.4%
Video Games	0.052720	0.049920	5.3%	0.098046	0.093339	4.8%

Table C.10: Within-action KL variation and group coverage. Singleton rates count action groups.

Dataset	l	KL	95% interval	Groups	Min/med/max	Singleton%
Industrial	1	1.0064	[1.0027, 1.0098]	256	8/139/1962	0.00
Industrial	2	0.7106	[0.7089, 0.7120]	254	2/80/6942	0.00
Industrial	3	0.8415	[0.8401, 0.8430]	256	7/141/3071	0.00
Musical	1	1.0140	[1.0104, 1.0175]	255	7/162/1261	0.00
Musical	2	0.7739	[0.7719, 0.7757]	254	1/105/7018	1.18
Musical	3	0.8118	[0.8104, 0.8131]	256	5/148/2169	0.00
Video Games	1	1.0575	[1.0551, 1.0598]	256	2/234/3199	0.00
Video Games	2	0.8638	[0.8621, 0.8655]	255	2/165/10747	0.00
Video Games	3	0.9040	[0.9027, 0.9053]	256	6/238/4406	0.00

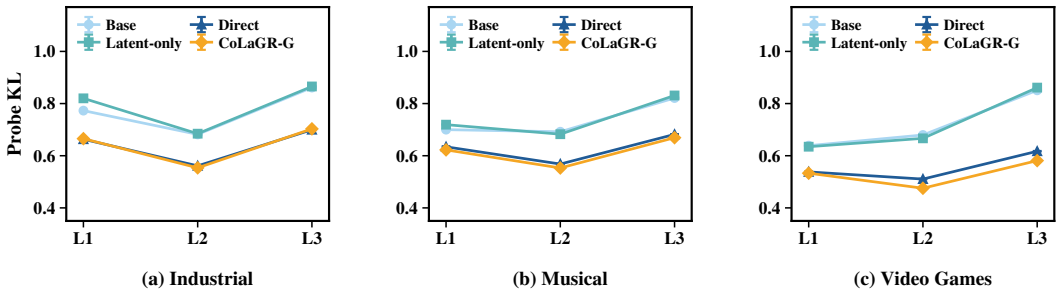


Figure C.2: Level-wise reference recoverability. Error bars show 95% bootstrap intervals conditional on each fitted readout.

C.8 Action-Conditioned Reference Ambiguity

For each level, examples with the same observed action form a group. We compute the empirical centroid of their level-wise references and average the KL divergence to that centroid over examples. Centroids and divergences use the same test pool, so this is a descriptive within-group statistic rather than held-out action-only prediction. Bootstrap resampling keeps the computed per-example divergences fixed and does not re-estimate centroids. Table C.10 includes group coverage: singleton groups contribute zero by construction. Only Musical L2 has a nonzero singleton-group fraction.

Table C.10 shows positive reference variation in every dataset–level pair. Singleton groups are absent except for 1.18% of Musical L2 groups, so the observed ambiguity is not confined to isolated action classes. Action identity alone does not determine the empirical collaborative reference.

C.9 Level-Wise Reference Recoverability

Figure C.2 summarizes all 36 fits, with dataset-specific results below. KL and Jensen–Shannon divergence compare the reference and predicted distributions. Modal agreement compares their highest-probability codes and is distinct from hard-action accuracy. CoLaGR-G lowers average KL relative to Base, while its comparison with Direct depends on the domain, level, and metric. Industrial Direct, for example, has slightly lower KL at L1/L3 and higher average modal agreement.

Table C.11 shows lower probe KL with collaborative supervision at every Industrial level. CoLaGR-G has lower KL than Direct at L2, while Direct is slightly lower at L1 and L3. Their aggregate recoverability is consequently close, rather than uniformly ordered across levels.

Table C.11: Industrial common-readout diagnostics. Agreement measures modal-code recovery.

Variant	l	KL	95% interval	JS	Agreement
Base	1	0.7729	[0.7701, 0.7757]	0.1831	0.3191
Base	2	0.6818	[0.6800, 0.6834]	0.1590	0.8562
Base	3	0.8613	[0.8595, 0.8630]	0.2103	0.3388
Latent-only	1	0.8201	[0.8170, 0.8228]	0.1909	0.3015
Latent-only	2	0.6845	[0.6828, 0.6861]	0.1589	0.8478
Latent-only	3	0.8656	[0.8637, 0.8674]	0.2108	0.3344
Direct	1	0.6632	[0.6608, 0.6658]	0.1576	0.3895
Direct	2	0.5612	[0.5596, 0.5628]	0.1313	0.8670
Direct	3	0.6997	[0.6981, 0.7013]	0.1711	0.3812
CoLaGR-G	1	0.6654	[0.6630, 0.6680]	0.1585	0.3817
CoLaGR-G	2	0.5535	[0.5520, 0.5550]	0.1304	0.8657
CoLaGR-G	3	0.7030	[0.7013, 0.7046]	0.1735	0.3377

Table C.12: Musical common-readout diagnostics. Agreement measures modal-code recovery.

Variant	l	KL	95% interval	JS	Agreement
Base	1	0.7000	[0.6978, 0.7024]	0.1665	0.2184
Base	2	0.6919	[0.6902, 0.6936]	0.1648	0.5398
Base	3	0.8212	[0.8196, 0.8228]	0.2036	0.1470
Latent-only	1	0.7191	[0.7169, 0.7213]	0.1692	0.2105
Latent-only	2	0.6825	[0.6809, 0.6841]	0.1628	0.5561
Latent-only	3	0.8310	[0.8295, 0.8328]	0.2058	0.1428
Direct	1	0.6342	[0.6321, 0.6363]	0.1502	0.2598
Direct	2	0.5681	[0.5668, 0.5694]	0.1366	0.5699
Direct	3	0.6818	[0.6806, 0.6830]	0.1693	0.1988
CoLaGR-G	1	0.6224	[0.6205, 0.6244]	0.1479	0.2574
CoLaGR-G	2	0.5530	[0.5517, 0.5543]	0.1337	0.5566
CoLaGR-G	3	0.6692	[0.6679, 0.6704]	0.1681	0.2045

Table C.13: Video Games common-readout diagnostics. Agreement measures modal-code recovery.

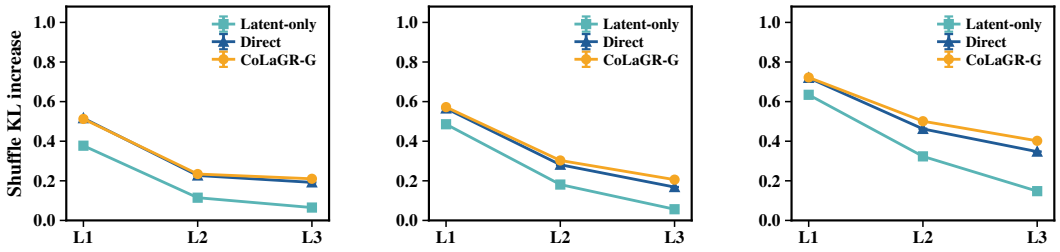
Variant	l	KL	95% interval	JS	Agreement
Base	1	0.6386	[0.6369, 0.6402]	0.1527	0.3059
Base	2	0.6795	[0.6781, 0.6810]	0.1622	0.5010
Base	3	0.8508	[0.8493, 0.8522]	0.2109	0.2010
Latent-only	1	0.6346	[0.6329, 0.6362]	0.1513	0.3021
Latent-only	2	0.6667	[0.6652, 0.6681]	0.1593	0.5128
Latent-only	3	0.8612	[0.8596, 0.8628]	0.2134	0.2019
Direct	1	0.5380	[0.5366, 0.5395]	0.1298	0.3613
Direct	2	0.5104	[0.5094, 0.5116]	0.1231	0.6330
Direct	3	0.6171	[0.6161, 0.6182]	0.1542	0.3291
CoLaGR-G	1	0.5330	[0.5317, 0.5344]	0.1289	0.3584
CoLaGR-G	2	0.4753	[0.4742, 0.4764]	0.1161	0.6509
CoLaGR-G	3	0.5812	[0.5802, 0.5822]	0.1471	0.3508

Table C.12 shows the lowest KL and JS for CoLaGR-G at all Musical levels. Modal agreement does not always give the same ordering, distinguishing reconstruction of the reference distribution from prediction of its single highest-mass code.

Table C.13 shows lower KL for CoLaGR-G than Direct at all Video Games levels, with the clearest separation at L2 and L3. Both supervised variants improve substantially on Base and Latent-only, supporting collaborative grounding of intermediate states. On Video Games, the largest CoLaGR-G versus Direct KL difference occurs at L2, whereas the Musical gains are smaller and spread across all three decisions. Thus, the aggregate reduction persists beyond the first semantic branch as the candidate set narrows.

Table C.14: Within-action shuffle, averaged equally over the three levels.

Dataset	Variant	Real KL	Shuffled KL	ΔKL	Increase
Industrial	Latent-only	0.7901	0.9757	0.1857	23.5%
Industrial	Direct	0.6414	0.9531	0.3117	48.6%
Industrial	CoLaGR-G	0.6407	0.9597	0.3190	49.8%
Musical	Latent-only	0.7442	0.9853	0.2411	32.4%
Musical	Direct	0.6280	0.9664	0.3383	53.9%
Musical	CoLaGR-G	0.6148	0.9751	0.3603	58.6%
Video Games	Latent-only	0.7208	1.0894	0.3685	51.1%
Video Games	Direct	0.5552	1.0649	0.5098	91.8%
Video Games	CoLaGR-G	0.5299	1.0715	0.5417	102.2%



(a) Industrial (b) Musical (c) Video Games

Figure C.3: Level-wise within-action shuffle penalties. Error bars denote permutation SD.

C.10 Within-Action Shuffle Control

We independently permute collaborative references within observed-action groups in the training, validation, and test splits, then refit and select the readout using the same procedure for five permutation seeds (2024–2028). This preserves action identity while disrupting individual state–reference correspondence. Tables C.14 and C.15 report the across-level means and all level-wise results. Here ΔKL is shuffled minus real KL, and Increase expresses this difference as a percentage of real KL. CoLaGR-G has the largest average shuffle penalty in each domain, but the effect is also present for Direct and Latent-only. The penalty measures sensitivity to correspondence under the fitted-readout protocol, not a calibrated quantity of mutual information.

Figure C.3 isolates sensitivity to state–reference correspondence within a fixed observed-action group. All three variants show positive penalties at every level, and the collaboratively supervised variants retain stronger deeper-level contrasts than Latent-only.

Table C.15 shows lower KL for real pairings in every listed cell. The absolute penalty is generally largest at L1, while collaboratively supervised variants retain larger deeper-level contrasts than Latent-only. The SD describes variation across the five permutations.

The across-level means also reveal why the magnitude of the penalty should be read within each domain. On Video Games, CoLaGR-G has both the lowest real-pairing KL and the largest absolute shuffle penalty; on Industrial, its real KL and penalty are close to those of Direct. The control therefore tests correspondence within a fixed evaluation setting, not a common scale of reference information across domains.

The level-wise rows qualify the aggregate ordering. On Industrial, Direct has a slightly larger L1 penalty, but CoLaGR-G has larger penalties at L2 and L3; its average lead comes from the deeper decisions. On Musical and Video Games, CoLaGR-G exceeds Direct at every level, with the clearest separation after L1. The result therefore reflects preservation across the hierarchy, not sensitivity at only the first branch. The within-action permutation preserves the reference multiset in each action group. Thus, an action-only readout faces the same empirical target distribution after shuffling, whereas a state-dependent readout loses each history's matching reference. Refitting the readout on shuffled references rules out a penalty caused merely by evaluating a matched-reference probe on new targets.

Table C.15: All within-action shuffle results. ± denotes permutation SD.

Dataset	Variant	l	Real KL	Shuffled KL	ΔKL
Industrial	Latent-only	1	0.8201	1.1974 ± 0.0035	0.3773
Industrial	Latent-only	2	0.6845	0.7992 ± 0.0014	0.1147
Industrial	Latent-only	3	0.8656	0.9305 ± 0.0008	0.0649
Industrial	Direct	1	0.6632	1.1794 ± 0.0031	0.5162
Industrial	Direct	2	0.5612	0.7881 ± 0.0021	0.2269
Industrial	Direct	3	0.6997	0.8918 ± 0.0007	0.1921
Industrial	CoLaGR-G	1	0.6654	1.1779 ± 0.0027	0.5125
Industrial	CoLaGR-G	2	0.5535	0.7878 ± 0.0012	0.2343
Industrial	CoLaGR-G	3	0.7030	0.9133 ± 0.0009	0.2103
Musical	Latent-only	1	0.7191	1.2048 ± 0.0009	0.4857
Musical	Latent-only	2	0.6825	0.8636 ± 0.0018	0.1811
Musical	Latent-only	3	0.8310	0.8875 ± 0.0009	0.0565
Musical	Direct	1	0.6342	1.1996 ± 0.0012	0.5654
Musical	Direct	2	0.5681	0.8493 ± 0.0020	0.2812
Musical	Direct	3	0.6818	0.8502 ± 0.0008	0.1684
Musical	CoLaGR-G	1	0.6224	1.1946 ± 0.0020	0.5722
Musical	CoLaGR-G	2	0.5530	0.8558 ± 0.0015	0.3028
Musical	CoLaGR-G	3	0.6692	0.8750 ± 0.0007	0.2059
Video Games	Latent-only	1	0.6346	1.2690 ± 0.0015	0.6344
Video Games	Latent-only	2	0.6667	0.9901 ± 0.0035	0.3235
Video Games	Latent-only	3	0.8612	1.0090 ± 0.0008	0.1478
Video Games	Direct	1	0.5380	1.2579 ± 0.0018	0.7199
Video Games	Direct	2	0.5104	0.9725 ± 0.0018	0.4620
Video Games	Direct	3	0.6171	0.9645 ± 0.0010	0.3474
Video Games	CoLaGR-G	1	0.5330	1.2550 ± 0.0019	0.7220
Video Games	CoLaGR-G	2	0.4753	0.9764 ± 0.0025	0.5010
Video Games	CoLaGR-G	3	0.5812	0.9832 ± 0.0008	0.4020

The control distinguishes recovering an action-group mean reference from recovering the reference associated with an individual history. Permutation retains the same observed code and the same collection of references within each group, so action identity alone cannot explain the advantage of the real pairing. Refitting the readout also lets it adapt to the shuffled targets under the same training budget. The remaining KL increase therefore supports history-specific reference recoverability beyond a shared action. This is the distinction required by collaborative continuity: users following the same branch can still carry different preferences into that decision.

The comparison with Latent-only further connects this diagnostic to collaborative grounding. Intermediate computation already preserves some correspondence, as its positive shuffle penalties show, while supervised variants retain stronger contrasts at deeper levels. CoLaGR-G has the largest domain-averaged penalty, with a clearer separation from Direct on Video Games than on Industrial. Together with the training-time correspondence experiment in Appendix C.7, these results link the representation-level diagnostic to the importance of matching supervision to the user. The two controls provide complementary evidence about reference preservation and recommendation quality, rather than identifying a causal effect of probe KL on ranking.

C.11 Action-Conditional Mean-Anchored Probes

Table C.16 tests recoverability beyond action identity rather than replacing the unconditional diagnostic. We estimate $\widehat{q}(c \mid Y)$ from the first 50,000 training examples, select its smoothing strength from $\{0, 0.1, 1, 5, 20, 100\}$ by validation KL, and fit $r(c \mid Z, Y) = \mathrm{softmax}(\log \widehat{q}(\cdot \mid Y) + W\widetilde{Z} + b)_c$. Here $\widetilde{Z}$ is the state standardized feature-wise using training-set statistics.

For each fit, W, b start at zero. Validation selects the learning rate from $\{0.001, 0.003\}$ and the epoch from 0–25, with patience 4. Including epoch zero keeps the fitted action-only predictor available instead of relying on an underfit action-only affine head.

Table C.16: Action-conditioned reference recovery. Values are mean test KL across SID levels, with lower values indicating better recovery.

Dataset	Model	Action mean	State residual	Shuffled
Industrial	Base	0.892842	0.769395	0.892766
Industrial	Direct	0.892842	0.660035	0.892296
Industrial	CoLaGR-G	0.892842	0.643051	0.892436
Industrial	Native-State Grounding	0.892842	0.648884	0.892468
Musical	Base	0.890515	0.733754	0.890567
Musical	Direct	0.890515	0.647500	0.890550
Musical	CoLaGR-G	0.890515	0.613494	0.890544
Musical	Native-State Grounding	0.890515	0.617517	0.890545
Video Games	Base	0.970559	0.721199	0.970486
Video Games	Direct	0.970559	0.600601	0.970473
Video Games	CoLaGR-G	0.970559	0.549606	0.970457
Video Games	Native-State Grounding	0.970559	0.626986	0.970444

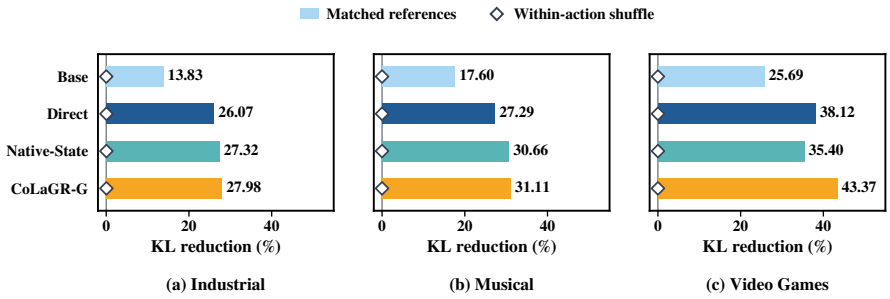


Figure C.4: Reference recovery beyond action identity. Bars show KL reduction with matched references, and diamonds show within-action shuffle controls.

Evaluation uses all validation/test examples. The action label is diagnostic side information and is never provided to the generator. A single independently generated within-action permutation per split supplies the within-action shuffle control.

Table C.16 and Figure C.4 show that CoLaGR-G lowers mean test KL from the action-only baselines of 0.8928, 0.8905, and 0.9706 to 0.6431, 0.6135, and 0.5496. Native-State Grounding reaches 0.6489, 0.6175, and 0.6270, showing that collaborative information remains recoverable when the two readouts are attached to the native decoder state. The explicit decision-aligned realization provides the largest additional recoverability gain on Video Games. Direct also retains substantial reference information, so these observations support an ordering of recoverability under this readout protocol, not exclusive representability or a causal explanation of ranking gains.

C.12 Conflict-Stratified Action Modeling

For each dataset and level, conflict is $\kappa_{u,l} = -\log \max\{q^{\mathrm{col}}_{u,l}(Y_{u,l}), 10^{-8}\}$, where $\kappa_{u,l}$ measures how little collaborative mass the observed action receives. Tertile thresholds are computed on the validation set and then applied to test examples to define the Low, Mid, and High conflict groups.

The action probabilities and ranks use the valid-prefix trie mask. Action NLL also clips probabilities below 10^{-8}. Aggregates weight level-specific metrics by their group sample counts, rather than averaging the three level means equally.

Figure C.5 exposes the local reversals behind the aggregate effect. At L3 in the high-conflict group, Direct/CoLaGR-G NLL is 1.5631/1.5837 on Industrial, 1.4731/1.4853 on Musical, and 1.4314/1.4598 on Video Games. CoLaGR-G nevertheless improves target rank and action accuracy in these cells. NLL, rank, and accuracy therefore capture distinct aspects of action prediction. Table C.17 aggregates the groups. The dataset-specific tables below retain all 54 level-wise rows. Conflict groups are fixed from the teacher reference and observed action before comparing models. Thus, Direct and CoLaGR-G are evaluated on the same examples within each group, making the L3 metric reversals directly comparable.

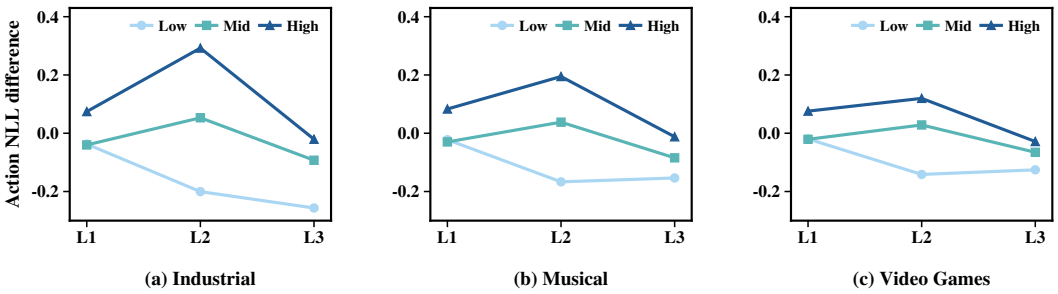


Figure C.5: Level-wise action NLL under reference–action conflict. Positive Direct minus CoLaGR-G differences favor CoLaGR-G.

Table C.17: Conflict metrics weighted by sample counts across levels. Prob., Rank, and Acc. refer to the observed hard action.

Dataset	Conflict	Variant	NLL	Prob.	Rank	Acc.
Industrial	Low	Direct	2.1527	0.2361	4.5182	0.4199
Industrial	Low	CoLaGR-G	2.3198	0.2561	5.9666	0.3744
Industrial	Mid	Direct	3.1674	0.1820	17.9458	0.2088
Industrial	Mid	CoLaGR-G	3.1942	0.2130	18.6452	0.2496
Industrial	High	Direct	3.9102	0.1716	40.4127	0.1902
Industrial	High	CoLaGR-G	3.7954	0.2000	36.7955	0.2278
Musical	Low	Direct	2.0786	0.2633	4.4779	0.4267
Musical	Low	CoLaGR-G	2.1937	0.2782	5.3719	0.3969
Musical	Mid	Direct	2.9328	0.2060	13.9731	0.2583
Musical	Mid	CoLaGR-G	2.9586	0.2288	14.4478	0.2765
Musical	High	Direct	3.7331	0.1831	34.4901	0.2121
Musical	High	CoLaGR-G	3.6448	0.2039	32.3900	0.2326
Video Games	Low	Direct	1.8488	0.3097	3.3209	0.4809
Video Games	Low	CoLaGR-G	1.9439	0.3322	3.8945	0.4572
Video Games	Mid	Direct	2.6741	0.2512	11.1279	0.3145
Video Games	Mid	CoLaGR-G	2.6933	0.2809	11.4956	0.3314
Video Games	High	Direct	3.6693	0.2053	32.4087	0.2414
Video Games	High	CoLaGR-G	3.6134	0.2288	30.6011	0.2578

Table C.17 separates confidence and ordinal behavior. Direct has lower aggregate NLL at low and medium conflict. CoLaGR-G has lower NLL, lower target rank, and higher action accuracy at high conflict in every domain. CoLaGR-G has a higher mean target probability across all aggregate groups. This need not imply lower mean NLL, which is more sensitive to low-probability predictions.

C.13 Fixed-Candidate Calibration

The diagnostic runner uses a transition window of 5, at most 200 neighbors per source item, the last 10 history items, and memory support truncated to 50 items. It preserves the generated beam of 50 items and exports each candidate together with its cumulative autoregressive log-probability $S_{\text{gen}}(u, j)$. The calibrated score is $S_{\text{gen}}(u, j) + \alpha S_{\text{leaf}}(u, j)$, using the normalized memory score in Eq. 37, which is zero outside the retained top-50 support. We select α on validation data: 3.0 for Industrial, 1.5 for Musical, and 3.0 for Video Games. Each is an interior maximum of its evaluated grid.

Five leaf-score permutations (seeds 2024–2028; Table C.22) preserve the candidate set, generation scores, and memory-score multiset while changing only item–memory-score correspondence. Table C.21 shows improvements at both ranking cutoffs with unchanged Recall@50. Thus calibration improves the ordering of recovered items without expanding the generated candidate set.

Table C.22 shows that matched evidence improves both metrics in every domain, whereas shuffled evidence falls below the generator-only ranking. Because the permutation preserves the candidates and the memory-score multiset, this contrast isolates the importance of assigning collaborative evidence to the correct item. The gain therefore depends on item-specific correspondence, rather than adding an arbitrary score perturbation to the generated beam.

Table C.18: Industrial: all level-wise conflict results under trie-masked action prediction.

l	Conflict	Variant	n	NLL	Prob.	Rank	Acc.
1	Low	Direct	15775	3.1678	0.0691	8.2391	0.2563
1	Low	CoLaGR-G	15775	3.2050	0.0797	10.7296	0.2470
1	Mid	Direct	17145	4.7782	0.0130	40.2781	0.0205
1	Mid	CoLaGR-G	17145	4.8184	0.0160	42.7030	0.0306
1	High	Direct	18065	6.0305	0.0046	94.5645	0.0042
1	High	CoLaGR-G	18065	5.9562	0.0061	89.9336	0.0081
2	Low	Direct	16704	2.0619	0.1724	3.1422	0.4510
2	Low	CoLaGR-G	16704	2.2623	0.1912	4.8014	0.3306
2	Mid	Direct	16954	3.2792	0.0640	10.4314	0.0702
2	Mid	CoLaGR-G	16954	3.2261	0.1142	10.3059	0.1685
2	High	Direct	17327	4.0444	0.0432	21.0874	0.0448
2	High	CoLaGR-G	17327	3.7521	0.0898	15.4072	0.1362
3	Low	Direct	16659	1.2826	0.4580	2.3743	0.5435
3	Low	CoLaGR-G	16659	1.5391	0.4883	2.6246	0.5389
3	Mid	Direct	17016	1.4330	0.4698	2.9311	0.5365
3	Mid	CoLaGR-G	17016	1.5258	0.5100	2.7140	0.5511
3	High	Direct	17310	1.5631	0.4746	3.2432	0.5298
3	High	CoLaGR-G	17310	1.5837	0.5127	2.7492	0.5486

Table C.18 localizes the Industrial high-conflict improvement: CoLaGR-G reduces NLL by 0.0743 at L1 and 0.2923 at L2. At both levels, the observed action also gains probability and rank. At L3, NLL is slightly higher despite better rank and accuracy. The aggregate gain therefore reflects a depth-dependent change, not a uniform reduction in action loss.

Table C.19: Musical: all level-wise conflict results under trie-masked action prediction.

l	Conflict	Variant	n	NLL	Prob.	Rank	Acc.
1	Low	Direct	18363	3.3310	0.0526	8.8824	0.1988
1	Low	CoLaGR-G	18363	3.3532	0.0586	10.2270	0.1947
1	Mid	Direct	18936	4.5456	0.0156	31.4796	0.0210
1	Mid	CoLaGR-G	18936	4.5754	0.0172	32.8949	0.0297
1	High	Direct	20140	5.8870	0.0056	83.9874	0.0049
1	High	CoLaGR-G	20140	5.8039	0.0066	80.9139	0.0071
2	Low	Direct	19119	1.7928	0.2128	2.4811	0.4592
2	Low	CoLaGR-G	19119	1.9596	0.2311	3.5830	0.3865
2	Mid	Direct	18783	2.9688	0.0959	7.9157	0.1715
2	Mid	CoLaGR-G	18783	2.9308	0.1388	7.9775	0.2176
2	High	Direct	19537	3.7913	0.0585	15.2457	0.0953
2	High	CoLaGR-G	19537	3.5966	0.0925	12.3639	0.1389
3	Low	Direct	18591	1.1356	0.5232	2.1811	0.6183
3	Low	CoLaGR-G	18591	1.2891	0.5435	2.4161	0.6074
3	Mid	Direct	19149	1.3026	0.5023	2.6029	0.5783
3	Mid	CoLaGR-G	19149	1.3871	0.5264	2.5527	0.5783
3	High	Direct	19699	1.4731	0.4881	2.9707	0.5398
3	High	CoLaGR-G	19699	1.4853	0.5161	2.6411	0.5561

Table C.19 shows high-conflict NLL reductions of 0.0831 and 0.1947 at Musical L1 and L2. CoLaGR-G improves high-conflict target rank and accuracy across all levels, including L3. The last level repeats the Industrial tradeoff: its mean target probability rises while mean NLL does not fall. The two metrics describe different aspects of the hard-action distribution. Low- and mid-conflict rows do not follow the high-conflict trend. At Musical L1, NLL rises in both lower-conflict groups even though it falls in the high-conflict group. The same L1 contrast appears on Industrial, localizing the NLL benefit to decisions with greater preference–action disagreement.

Table C.20: Video Games: all level-wise conflict results under trie-masked action prediction.

l	Conflict	Variant	n	NLL	Prob.	Rank	Acc.
1	Low	Direct	31197	2.9816	0.0740	5.8669	0.2609
1	Low	CoLaGR-G	31197	3.0012	0.0803	6.6137	0.2581
1	Mid	Direct	31309	4.3595	0.0186	25.0408	0.0244
1	Mid	CoLaGR-G	31309	4.3804	0.0206	25.9971	0.0328
1	High	Direct	32256	5.9014	0.0055	80.1696	0.0033
1	High	CoLaGR-G	32256	5.8255	0.0064	77.5936	0.0059
2	Low	Direct	30597	1.5656	0.2809	2.1034	0.5268
2	Low	CoLaGR-G	30597	1.7070	0.3176	2.9175	0.4629
2	Mid	Direct	31353	2.5625	0.1565	5.9873	0.2660
2	Mid	CoLaGR-G	31353	2.5341	0.2152	6.1964	0.3034
2	High	Direct	32812	3.6905	0.0860	14.5942	0.1374
2	High	CoLaGR-G	32812	3.5708	0.1275	12.0735	0.1740
3	Low	Direct	31040	0.9893	0.5750	1.9622	0.6568
3	Low	CoLaGR-G	31040	1.1149	0.5997	2.1246	0.6518
3	Mid	Direct	31239	1.0969	0.5793	2.3431	0.6538
3	Mid	CoLaGR-G	31239	1.1623	0.6078	2.2802	0.6588
3	High	Direct	32483	1.4314	0.5242	2.9764	0.5828
3	High	CoLaGR-G	32483	1.4598	0.5521	2.6523	0.5926

Table C.20 shows lower high-conflict NLL for CoLaGR-G at Video Games L1 and L2. At L3, its target rank improves from 2.9764 to 2.6523 and accuracy from 0.5828 to 0.5926, despite slightly higher NLL. The same depth dependence appears across all three domains: earlier decisions gain in log loss under conflict, whereas the final decision shows a stronger ordinal than NLL improvement.

Table C.21: Matched-memory calibration: complete metric record. R@50 is unchanged and candidate mismatches are zero for every dataset.

Dataset	Variant	N@5	N@10	R@5	R@10	R@50
Industrial	CoLaGR-G	0.0216	0.0272	0.0330	0.0503	0.1153
Industrial	CoLaGR	0.0257	0.0309	0.0374	0.0536	0.1153
Musical	CoLaGR-G	0.0281	0.0353	0.0425	0.0648	0.1505
Musical	CoLaGR	0.0324	0.0395	0.0472	0.0694	0.1505
Video Games	CoLaGR-G	0.0414	0.0527	0.0629	0.0980	0.2233
Video Games	CoLaGR	0.0510	0.0618	0.0740	0.1073	0.2233

Table C.21 shows ranking gains at both cutoffs with unchanged Recall@50.

Table C.22: Fixed-beam leaf-score permutation results. Values are mean $\pm$ SD over five permutations, with unchanged candidates and Recall@50.

Dataset	Scoring	NDCG@10	Recall@10
Industrial	Generator only	0.0272	0.0503
	Matched evidence	0.0309	0.0536
	Shuffled evidence	0.0217 ± 0.0002	0.0447 ± 0.0003
Musical	Generator only	0.0353	0.0648
	Matched evidence	0.0395	0.0694
	Shuffled evidence	0.0323 ± 0.0003	0.0610 ± 0.0006
Video Games	Generator only	0.0527	0.0980
	Matched evidence	0.0618	0.1073
	Shuffled evidence	0.0408 ± 0.0001	0.0835 ± 0.0003

Table C.22 shows that shuffled evidence weakens fixed-candidate ranking in every domain. The shuffle keeps candidates, path scores, and the multiset of memory scores fixed. Its drop below generator-only ranking shows that item–score correspondence drives the gain.

Table C.23: Seed-2024 calibration across generator variants with validation-selected α.

Dataset	Generator	α	N@10 before	N@10 after	R@10 before	R@10 after
Industrial	Base	5.0	0.0221	0.0271	0.0421	0.0470
Industrial	Latent-only	4.0	0.0236	0.0291	0.0450	0.0504
Industrial	Direct	4.0	0.0239	0.0294	0.0446	0.0505
Industrial	Native-State Grounding	4.0	0.0269	0.0316	0.0502	0.0548
Industrial	CoLaGR-G	3.0	0.0272	0.0309	0.0503	0.0536
Musical	Base	2.5	0.0319	0.0368	0.0593	0.0643
Musical	Latent-only	2.5	0.0324	0.0371	0.0608	0.0660
Musical	Direct	2.0	0.0333	0.0377	0.0616	0.0662
Musical	Native-State Grounding	2.5	0.0346	0.0389	0.0642	0.0681
Musical	CoLaGR-G	1.5	0.0353	0.0395	0.0648	0.0694
Video Games	Base	3.0	0.0495	0.0594	0.0925	0.1026
Video Games	Latent-only	3.0	0.0502	0.0597	0.0938	0.1040
Video Games	Direct	3.0	0.0510	0.0610	0.0946	0.1062
Video Games	Native-State Grounding	2.0	0.0522	0.0614	0.0967	0.1063
Video Games	CoLaGR-G	3.0	0.0527	0.0618	0.0980	0.1073

Table C.23 applies CoLeaf to every structural control. All generators improve on both metrics. Native-State Grounding leads by 0.0006 NDCG@10 on Industrial, while CoLaGR-G leads on Musical and Video Games. For the audited Industrial Base checkpoint, $\alpha = 0$ reproduces uncalibrated generator evaluation, confirming evaluator consistency.

D LIMITATIONS

Recommendation from historical interactions inherits exposure and popularity biases, and sparse histories limit preference estimation. Offline ranking metrics provide an incomplete account of long-term user satisfaction, motivating evaluation under evolving preferences and online feedback. Deployment must also respect dataset licenses, privacy obligations, and data deletion requirements.