# RECDATAAGENT: USER-CONDITIONED DATA AUGMENTATION FOR SEQUENTIAL RECOMMENDATION

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

## ABSTRACT

Improving sequential recommendation requires not only learning from user behavior, but also deciding how that behavior becomes training supervision. With heterogeneous preferences, the value of an augmented view depends on whose history it represents and which signals it preserves. We therefore frame data augmentation as the optimization of supervision construction: user selection, method assignment, and exposure form a complete policy whose value is determined by the sequential recommendation outcomes of the resulting corpus. We introduce RecDataAgent, an agent that connects user understanding, executable data construction, and outcome-guided revision. It analyzes train-visible user context to propose augmentation policies, materializes them while preserving task targets and the construction budget, and retains each policy together with its execution trace and measured outcomes. This feedback allows both user partitions and method assignments to evolve, making data construction a decision process informed by the learning it supports. Experiments across sequential recommendation domains show that contextual assignment improves retrieval and ranking under matched method exposure, while outcome feedback strengthens policies beyond user analysis and execution checks. These findings support a shift from designing augmentation methods in isolation to optimizing how they jointly shape supervision for different users. RecDataAgent makes this shift concrete through a closed loop in which user context guides construction and sequential recommendation outcomes guide its revision. Anonymous code is available at https://anonymous.4open.science/r/RecDataAgent-382D.

## 1 INTRODUCTION

Recommendation LLMs learn preferences from language-formatted records (Geng et al., 2022; Bao et al., 2023). Data augmentation expands this supervision by constructing new records from logged behavior (Lee et al., 2026; Wang et al., 2025). Yet the same construction can preserve different preference signals for different users. A prefix may capture a stable interest but omit a recent shift, as illustrated in Figure 1. The construction determines how a history is transformed; its assignment determines whose preferences enter training through that view. This distinction raises a question beyond record generation: under a fixed augmentation budget, which users should receive which views?

Existing methods offer several ways to make this choice. CL4SRec (Xie et al., 2022) and CoSeRec (Liu et al., 2021) transform interaction histories, while GenPAS (Lee et al., 2026), verbalization-based augmentation (Shi et al., 2026), and Llama4Rec (Luo et al., 2026) construct language-oriented records. L2Aug (Wang et al., 2022a) and AsarRec (Zhang et al., 2026a) further adapt augmentation to user sequences or recommendation objectives. AutoAugment (Cubuk et al., 2019), Population Based Augmentation (Ho et al., 2019), and RandAugment (Cubuk et al., 2020) address transformation search, while DoReMi (Xie et al., 2023) optimizes data exposure. A natural approach is therefore to analyze user histories and assign suitable constructions within the budget. However, a plausible match between a user group and a construction does not establish the recommendation value of the resulting corpus. How should these assignments be judged and improved?

We argue that the complete assignment, rather than a method name alone, should be the unit of evaluation. Changing which users receive a view changes supervision even when the number of records from each method remains fixed. User features describe the population; recommendation outcomes reveal the utility of the corpus constructed from it. We express the assignment as a policy



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.

that specifies user-selection conditions, methods, and exposure. Evaluating this policy creates a further requirement: its outcome must remain linked to the decisions that produced the corpus. That link allows subsequent proposals to reconsider both the user partition and the assigned methods, instead of retaining a score without the construction it evaluates.

Building on this view, we introduce RecDataAgent. A language-model controller uses feature-analysis tools to turn train-visible user context into selection conditions and method assignments. The resulting policy specifies a corpus, which typed adapters materialize while preserving task targets and the construction budget. Downstream validation evaluates that corpus, and an archive retains the policy together with its execution trace and recommendation outcomes. Feedback from these trials then informs the next proposal, while evaluated alternatives remain available for final selection. Thus, user analysis determines the proposed assignment, execution makes it trainable, and recommendation feedback guides its revision. The same policy connects what the controller observes, what the recommender learns from, and what changes next.

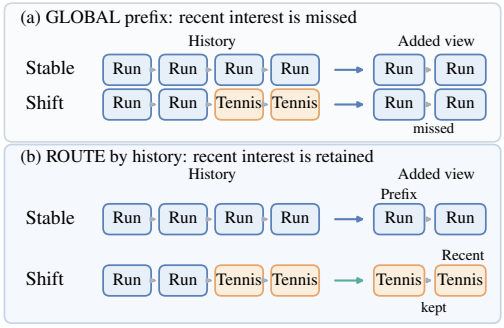


**Figure 1:** User-conditioned views under a shared augmentation budget. A global prefix view can miss a recent interest shift, whereas routing by train-visible history retains the recent signal for the same user.

Our experiments show that user assignment matters even when method exposure is held fixed. Across Amazon Beauty, Toys, and Sports, contextual routing outperforms random, history-length, and semantic assignments with the same method totals. The selected policies also improve Recall and NDCG at both 10 and 20 over the strongest locally evaluated augmentation baseline. Withholding recommendation feedback yields weaker final policies, supporting its role beyond user description and execution checks. Together, these results show that existing constructions can provide more useful supervision when their assignments are conditioned on user context and revised from measured outcomes. Our contributions are as follows:

- We formulate user-conditioned augmentation around the complete assignment, making user selection, method choice, and exposure part of the policy evaluated through recommendation outcomes.
- We introduce RecDataAgent, which converts multidimensional user analysis into executable policies and budgeted training corpora through a tool-using controller and typed adapters.
- We develop outcome-linked revision that allows user partitions and method assignments to change together, with evaluated alternatives retained for final selection.
- We show that contextual assignment and recommendation feedback improve augmentation across three domains, establishing their value beyond choosing a construction method alone.

## 2 Related Work

**Recommendation interfaces and augmentation.** Language-based recommendation turns histories, item content, and targets into interfaces learnable by pretrained language models. P5 (Geng et al., 2022) formulates recommendation as personalized text generation, TALLRec (Bao et al., 2023) aligns an LLM through recommendation tuning, and TIGER (Rajput et al., 2023) and GRLM (Zhang et al., 2026b) generate semantic or textual item identifiers. LLaRA (Liao et al., 2024) and E4SRec (Li et al., 2023) instead connect collaborative or ID-based signals with LLM representations. Sequential recommendation provides a broad augmentation vocabulary (Dang et al., 2024): CL4SRec (Xie et al., 2022) and CoSeRec (Liu et al., 2021) transform interaction sequences, DuoRec (Qiu et al., 2022) and ICLRec (Chen et al., 2022) construct semantic or intent-aware contrastive pairs, and EC4SRec (Wang et al., 2022b) uses explanation-guided perturbations. Recent generative methods sample histories and targets in GenPAS (Lee et al., 2026), learn verbalizations in From Logs to Language (Shi et al., 2026), augment conventional recommendation in Llama4Rec (Luo et al., 2026), and use LLMs as data enhancers in LLMSeR (Wang et al., 2025). These methods change the available supervision, but applying one method uniformly fixes its user population and exposure implicitly.

**Adaptive policy search and data curation.** Automated augmentation searches transformation compositions and strengths through AutoAugment (Cubuk et al., 2019), Population Based Aug-

mentation (Ho et al., 2019), and RandAugment (Cubuk et al., 2020). Recommendation methods introduce finer conditioning: CoSeRec (Liu et al., 2021) restricts transformations by sequence length, L2Aug (Wang et al., 2022a) learns an editing policy from recommender feedback, and AsarRec (Zhang et al., 2026a) learns user-conditioned transformation matrices. In parallel, data-allocation methods optimize which sources receive training opportunity: DoReMi (Xie et al., 2023) learns domain weights, Skill-It! (Chen et al., 2023) schedules skills from dependency estimates, and Data Mixing Laws (Ye et al., 2024) and RegMix (Liu et al., 2024) predict favorable mixtures from smaller runs. DataComp (Gadre et al., 2023), Data-Juicer (Chen et al., 2024), and JEST (Evans et al., 2024) systematize corpus evaluation, processing, and learnability-aware selection. These lines establish validation-guided policy search and exposure control, but their action spaces typically remain within one transformation family or operate over domains, examples, or corpora.

**Agents for training-data design.** Language agents provide the control loop needed to operationalize such data decisions. ReAct (Yao et al., 2023) couples reasoning with tool actions, Reflexion (Shinn et al., 2023) carries feedback into later attempts, and ADAS (Hu et al., 2025) and AgentSquare (Shang et al., 2025) search structured agent designs. Curation-Bench (Kang et al., 2026) applies an inspect–implement–train–evaluate loop directly to data policies and shows how downstream outcomes can revise later proposals. Recommendation augmentation adds execution constraints absent from general free-form curation: assignments must identify users, respect method applicability and exposure, preserve required task fields, and retain provenance. These agent methods motivate our controller.

Together, these lines of work supply methods, search ideas, and feedback mechanisms, but leave open the interface that joins them for user-conditioned recommendation augmentation. We therefore define that interface before describing the controller: the policy states the coupled decisions, the adapter executes them, and the archive returns their measured consequences to the next proposal.

## 3 Preliminaries and Problem Definition

The recommender trains on records, not on a method name. A method choice therefore leaves two decisions open: which users receive the resulting view and how much of the fixed training budget each user receives. We represent these coupled decisions as one policy, then make that policy executable and revise it from measured outcomes.

### 3.1 Sequential Recommendation and Augmentation Policy

For domain $\delta$, let $\mathcal{E}_\delta$ denote its users, let $s_e$ denote the train-visible history and item context of user $e \in \mathcal{E}_\delta$, and let $D_\delta^0$ denote the immutable base corpus. A candidate corpus $D$ is passed to the fixed downstream learner $G_\delta$, whose output is the vector of recommendation outcomes

$$\mathbf{y}_\delta(D) = G_\delta(D).$$

Here $\mathbf{y}_\delta(D)$ collects the evaluation metrics used for domain $\delta$; it is the outcome that later tells the planner whether the induced corpus is useful. The relevant object is consequently the corpus induced by a policy, rather than a construction view considered in isolation.

Let $\mathcal{R}_\delta$ be the catalog of typed data-construction methods available in domain $\delta$. A complete policy $\pi$ maps train-visible contexts to routes, methods, and exposures, producing an appended corpus $\widehat{D}_\delta(\pi)$ and the complete corpus

$$D_\delta(\pi) = D_\delta^0 \uplus \widehat{D}_\delta(\pi).$$

The symbol $\widehat{D}_\delta(\pi)$ denotes only the records added by $\pi$, whereas $D_\delta(\pi)$ is the corpus actually used by $G_\delta$. Applying the scalar utility functional $\mathcal{U}_\delta$ to the outcome vector gives

$$J_\delta(\pi) = \mathcal{U}_\delta(\mathbf{y}_\delta(D_\delta(\pi))) .$$

The complete policy and its induced corpus are the units of comparison: together they preserve the link between user assignment and recommendation outcome. This distinction separates user-conditioned augmentation from a method-only comparison, where the method totals can be fixed while the user–method mapping remains unspecified.

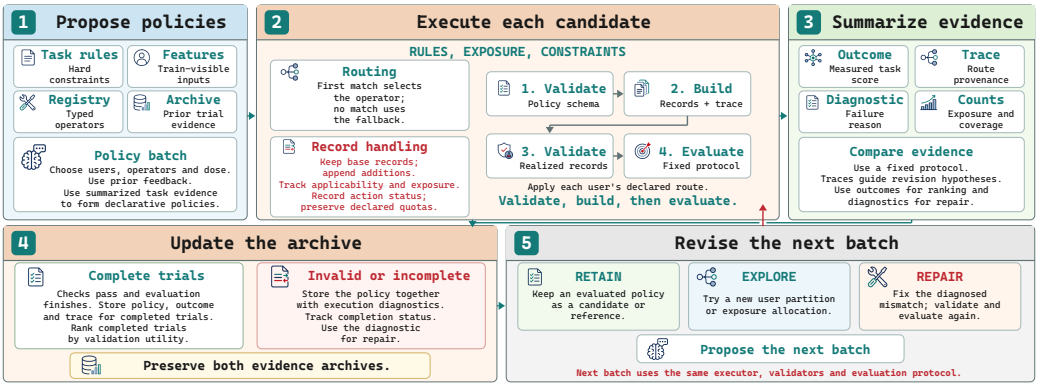


**Figure 2:** RecDataAgent as a typed policy-search loop. The controller proposes a complete user-conditioned policy; schema and run validators materialize only feasible records; downstream validation supplies outcome evidence; and the archive uses outcomes and construction diagnostics to form the next candidate batch.

**Feasible policy objective.** The search space $\Pi_\delta^{\text{feas}}$ contains the policies that satisfy domain $\delta$'s exposure, coverage, identity, label, and applicability constraints. RecDataAgent searches this feasible space using train-visible context, the method catalog, and an evidence archive:

$$\pi_\delta^\star \in \arg\max_{\pi \in \Pi_\delta^{\text{feas}}} J_\delta(\pi). \tag{1}$$

The superscript $\star$ marks the best feasible policy under the validation utility $J_\delta$. This objective identifies what to compare, but does not yet say how heterogeneous user context becomes a route, how a route becomes a training record, or how one measured policy changes the next proposal. The method follows these dependencies in that order.

## 4 Method

RecDataAgent answers three linked questions. Given train-visible context, which users should receive which views and exposure? Given that declarative assignment, how can it become a valid recommendation corpus without changing the task? Once the corpus is evaluated, how should its outcome change the next proposal? The order is substantive: contextual construction determines the assignment, typed execution determines the records that realize it, and evaluation supplies the evidence used for revision. Figure 2 shows this loop.

### 4.1 Contextual Policy Construction

The optimization objective compares complete policies, but it does not specify how user heterogeneity enters those policies. Users differ in history length, category concentration, semantic overlap, and popularity, so a construction method alone does not specify the corpus presented to the recommender. RecDataAgent first applies the feature map $\phi_\delta$ to each train-visible context, producing

$$z_e = \phi_\delta(s_e) \in \mathcal{Z}_\delta. \tag{2}$$

Here $z_e$ is the compact context of user $e$, and $\mathcal{Z}_\delta$ is the feature space for domain $\delta$. These features are useful only when they produce an executable partition. The controller therefore uses $z_e$ and joint-predicate counts to define an ordered route list:

$$\pi = (\mathbf{r}, \nu, \kappa), \qquad \mathbf{r} = \big((g_1, o_1, \eta_1), \ldots, (g_R, o_R, \eta_R), (g_0, o_0, \eta_0)\big), \tag{3}$$

where $g_j$ is the predicate for route $j$, $o_j$ is its typed construction operator, and $\eta_j$ contains that operator's parameters. The map $\nu$ specifies exposure, $\kappa$ collects the hard corpus constraints, and the final tuple $(g_0, o_0, \eta_0)$ is the remainder route. First-match routing converts the declarative route list into the action assigned to user $e$:

$$a_\pi(e) = \begin{cases} (o_j, \eta_j), & j = \min\{\ell : g_\ell(z_e) = 1\}, \\ (o_0, \eta_0), & \text{if no non-remainder predicate fires.} \end{cases} \tag{4}$$

The resulting $a_\pi(e)$ is the method and parameter pair assigned to $e$; it is not yet a training record. Exposure $\nu$ determines how many records each user contributes under the shared budget, while $\kappa$

determines whether the full assignment is admissible. Thus this stage resolves the user–method mapping left open by a method-only choice and hands the next stage a complete assignment to realize as records. For example, the Beauty policy assigns users with `cluster_id=1` to GenPAS prefix views and the remainder to AsarRec dual views, adding one record per user. The feature partition alone does not determine the training corpus: the method assignment and exposure complete the mapping from observed context to records.

### 4.2 Typed Corpus Materialization

The contextual policy now specifies who receives what, but it is still declarative while the recommender is trained on records. The domain adapter $\mathcal{I}_\delta$ packages the feature map, method catalog, validators, materializer, and learner needed to cross this representation boundary:

$$\mathcal{I}_\delta = \left(\phi_\delta, \mathcal{R}_\delta, V_\delta^{\mathrm{sch}}, V_\delta^{\mathrm{run}}, X_\delta, G_\delta\right). \tag{5}$$

The components have distinct roles: $\phi_\delta$ and $\mathcal{R}_\delta$ define available inputs and operators; $V_\delta^{\mathrm{sch}}$ validates the policy schema; $X_\delta$ materializes records; $V_\delta^{\mathrm{run}}$ validates realized records; and $G_\delta$ evaluates the resulting corpus. Before records are generated, $V_\delta^{\mathrm{sch}}$ checks that routes, methods, constraints, and the remainder action form a valid policy. Then $X_\delta$ applies $a_\pi(e)$ to the train-visible context, binds the resulting view to the matched base record, and emits a provenance trace $\tau$. Finally, $V_\delta^{\mathrm{run}}$ checks ownership, exposure, and task-field invariants before training. The original instruction and target stay attached to the base record; augmentation changes only the declared input view. This boundary resolves the remaining gap between an assignment and a training corpus. Because execution now yields a valid corpus and trace, the unresolved question is how its recommendation value should guide the next policy. Appendix C specifies the record contracts and accounting rules.

### 4.3 Outcome-Linked Evaluation and Archiving

Each materialized corpus is evaluated by the fixed downstream learner, which provides evidence about the complete policy rather than the construction view in isolation. At round $t$, the language-model planner $P_\theta$ receives task constraints $\mathcal{T}_\delta$, feature summaries $\mathcal{F}_\delta$, the method catalog $\mathcal{R}_\delta$, and the archive $\mathcal{H}_t$ of earlier trials:

$$C_t = (\mathcal{T}_\delta, \mathcal{F}_\delta, \mathcal{R}_\delta, \mathcal{H}_t), \qquad \Pi_t = P_\theta(C_t) = \{\pi_t^{(1)}, \ldots, \pi_t^{(K)}\}. \tag{6}$$

Here $C_t$ is the planner context at round $t$, $\mathcal{T}_\delta$ encodes task and corpus rules, $\mathcal{F}_\delta$ contains the feature summaries, and $\mathcal{H}_t$ supplies prior policy–outcome evidence. The planner returns a batch $\Pi_t$ of $K$ declarative candidates, each specifying routes, methods, parameters, and exposure. Each candidate passes through the same execution and evaluation path:

$$\pi_t^{(k)} \longrightarrow \text{schema check} \longrightarrow \text{materialize and run check} \longrightarrow \text{train and evaluate} \longrightarrow \text{archive update}.$$

For each completed candidate, the archive stores the policy $\pi$, its outcome vector $\mathbf{y}$, and its realized trace $\tau$ as a linked triple $(\pi, \mathbf{y}, \tau)$. Invalid or incomplete candidates contribute diagnostics for repair. Alongside scalar utility, the archive preserves the validation metric profile and construction trace. This keeps the evidence attached to the exact assignment and corpus that produced it; a score detached from those objects could rank candidates, but could not tell the planner what to retain or change.

### 4.4 Archive-Guided Revision and Reporting

The archive now supplies the missing comparison history. Let $\mathcal{H}^+$ denote the archive after the current batch has been evaluated. A subsequent proposal can target a specific weakness while retaining the assignments that produced the result: the planner may keep effective routes, change the feature partition or assigned method, adjust exposure, or repair an execution issue. Each candidate returns to the same materialization, learner, and utility, keeping comparisons on a common evaluation path. Final selection applies the fixed validation utility to the completed archive:

$$\pi_\delta^{\mathrm{sel}} \in \operatorname*{arg\,max}_{\pi:\,(\pi,\mathbf{y},\tau)\in\mathcal{H}^+} J_\delta(\pi).$$

The superscript $\mathrm{sel}$ marks the policy retained for reporting. Candidate generation and reporting are distinct: $P_\theta$ proposes from accumulated evidence, while the selector applies $J_\delta$ across $\mathcal{H}^+$. Thus a fixed budget becomes a user-conditioned assignment, then records and outcomes, and finally a policy–outcome pair that determines what changes next. The experiments instantiate this chain with the fixed candidate and evaluation protocol described in Section 5.

**Table 1:** Dataset and constructed-corpus statistics. Base and added records describe the principal agent-run configurations.

| Domain | Users | Items | Interactions | Avg. length | Sparsity | Base records | Added records |
|---|---|---|---|---|---|---|---|
| Beauty | 22,363 | 12,101 | 194,687 | 8.71 | 99.9281% | 34,464 | 22,363 |
| Toys | 19,412 | 11,924 | 167,597 | 8.63 | 99.9276% | 31,336 | 19,412 |
| Sports | 35,598 | 18,357 | 296,337 | 8.32 | 99.9547% | 53,955 | 35,572 |

# 5 EXPERIMENTS

We evaluate the resulting corpora, isolate user assignment and outcome feedback, and examine how the archive supports transfer and selection. A Beauty case study then connects the policy's routes to the user features behind them.

## 5.1 EVALUATION SETUP

**Datasets and task.** We evaluate sequential recommendation on Amazon Beauty, Toys, and Sports (Table 1), predicting the next item's five characteristic words from the user's history. Augmented corpora retain the base records as an exact prefix and append at most one record per eligible user. Matched comparisons share each domain's appended-record budget; the 26 Sports users unsupported by the evaluated methods retain their base records.

**Baseline methods.** Locally evaluated L2Aug, GenPAS, and AsarRec retain their method-specific constructions under a common learner; RecDataAgent assigns available constructions using user context. These form the matched augmentation comparison in Table 2. Published SASRec (Kang & McAuley, 2018), BERT4Rec (Sun et al., 2019), HSTU (Zhai et al., 2024), OneRec-Think (Liu et al., 2025), and GRLM (Zhang et al., 2026b) rows retain their original protocols and provide architectural context.

**Training and evaluation.** Local methods use Qwen3-4B-Instruct-2507, the same split, and one epoch of full-parameter BF16 fine-tuning: cutoff 2,048 tokens, cosine learning-rate decay from $10^{-4}$, and effective global batch size 80. Deterministic decoding uses 20 beams, 20 returned sequences, and at most 30 new tokens. Recall (R) measures retrieval; normalized discounted cumulative gain (NDCG, abbreviated N) measures ranking quality. Hit rate (HR) describes early-rank retrieval performance. Higher is better throughout.

**Evaluation aggregation.** Training data supply the base corpus, augmentation inputs, and features. Each domain evaluates three candidates per round for three rounds; validation Recall@10 from the seed-42 reference run selects one policy from all nine. Table 2 reports means over training seeds 40–44 for the local methods and selected policy, with one policy per domain across all six metrics. Appendix A gives aggregation, tie-breaking, and the selection trace.

**Complete policies improve recommendation across domains.** Table 2 shows gains over the strongest local augmentation baseline in all twelve domain–metric comparisons formed by Recall/NDCG at 10 and 20. Recall@10 improves by 4.59% on Beauty (0.0820 versus 0.0784), 12.35% on Toys (0.0828 versus 0.0737), and 18.30% on Sports (0.0569 versus 0.0481). The complete-policy advantage therefore appears in each evaluated domain, including the largest and sparsest population, Sports.

Baseline strengths vary: AsarRec leads on Beauty, GenPAS on Toys, and Sports favors AsarRec at Recall@10 but GenPAS at Recall@20. Thus, no single construction dominates all domains and ranking depths. The complete policy improves over these differing strengths; the following controls examine whether user assignment, rather than method totals alone, contributes to the advantage.

The joint Recall/NDCG pattern is important for interpreting the corpus change. A gain in Recall alone would show that more targets enter the returned list; higher NDCG at both cutoffs and leading HR@5 show that early ranking also benefits. The selected policy therefore improves both aspects of the recommendation profile, rather than exchanging early-rank quality for broader retrieval.

**User assignment matters even when method exposure is fixed.** Table 3 replaces contextual routing while preserving coverage, method exposure, and training. RANDOM uses a frozen hash

**Table 2:** Recommendation performance across the three Amazon domains. L2Aug, GenPAS, AsarRec, and RecDataAgent share the local training and evaluation protocol; their entries are five-seed means. Base-model and generative-recommender rows provide published reference results under their source protocols.

| Domain | Group | Method | Top-5 | | Top-10 | | Top-20 | |
|---|---|---|---|---|---|---|---|---|
| | | | HR@5 | NDCG@5 | Recall@10 | NDCG@10 | Recall@20 | NDCG@20 |
| Beauty | Base Models | SASRec | 0.0364 | 0.0233 | 0.0550 | 0.0289 | 0.0766 | 0.0338 |
| | | BERT4Rec | 0.0211 | 0.0133 | 0.0357 | 0.0181 | 0.0493 | 0.0212 |
| | | HSTU | 0.0388 | 0.0258 | 0.0593 | 0.0322 | 0.0832 | 0.0381 |
| | Generative Recommenders | OneRec-Think | 0.0507 | 0.0365 | 0.0732 | 0.0441 | 0.0995 | 0.0505 |
| | | GRLM | 0.0399 | 0.0274 | 0.0591 | 0.0343 | 0.0828 | 0.0403 |
| | Data Policy Recommenders | L2Aug | 0.0468 | 0.0321 | 0.0677 | 0.0390 | 0.0949 | 0.0460 |
| | | GenPAS | 0.0508 | 0.0351 | 0.0730 | 0.0421 | 0.1019 | 0.0495 |
| | | AsarRec | 0.0546 | 0.0383 | 0.0784 | 0.0456 | 0.1058 | 0.0526 |
| | | RecDataAgent | **0.0583** | **0.0407** | **0.0820** | **0.0488** | **0.1087** | **0.0555** |
| Toys | Base Models | SASRec | 0.0404 | 0.0271 | 0.0570 | 0.0327 | 0.0762 | 0.0375 |
| | | BERT4Rec | 0.0196 | 0.0120 | 0.0302 | 0.0154 | 0.0408 | 0.0180 |
| | | HSTU | 0.0333 | 0.0227 | 0.0522 | 0.0281 | 0.0701 | 0.0329 |
| | Generative Recommenders | OneRec-Think | 0.0527 | 0.0375 | 0.0732 | 0.0441 | 0.0995 | 0.0505 |
| | | GRLM | 0.0440 | 0.0304 | 0.0626 | 0.0364 | 0.0850 | 0.0420 |
| | Data Policy Recommenders | L2Aug | 0.0484 | 0.0338 | 0.0679 | 0.0402 | 0.0916 | 0.0461 |
| | | GenPAS | 0.0534 | 0.0368 | 0.0737 | 0.0433 | 0.0987 | 0.0497 |
| | | AsarRec | 0.0457 | 0.0320 | 0.0654 | 0.0383 | 0.0882 | 0.0441 |
| | | RecDataAgent | **0.0592** | **0.0410** | **0.0828** | **0.0487** | **0.1108** | **0.0557** |
| Sports | Base Models | SASRec | 0.0183 | 0.0097 | 0.0272 | 0.0127 | 0.0381 | 0.0155 |
| | | BERT4Rec | 0.0093 | 0.0059 | 0.0158 | 0.0080 | 0.0218 | 0.0094 |
| | | HSTU | 0.0245 | 0.0158 | 0.0312 | 0.0179 | 0.0436 | 0.0212 |
| | Generative Recommenders | OneRec-Think | 0.0264 | 0.0179 | 0.0377 | 0.0219 | 0.0528 | 0.0258 |
| | | GRLM | 0.0236 | 0.0155 | 0.0376 | 0.0201 | 0.0522 | 0.0235 |
| | Data Policy Recommenders | L2Aug | 0.0245 | 0.0162 | 0.0387 | 0.0207 | 0.0567 | 0.0252 |
| | | GenPAS | 0.0311 | 0.0204 | 0.0460 | 0.0255 | 0.0668 | 0.0302 |
| | | AsarRec | 0.0327 | 0.0219 | 0.0481 | 0.0273 | 0.0661 | 0.0319 |
| | | RecDataAgent | **0.0385** | **0.0265** | **0.0569** | **0.0324** | **0.0779** | **0.0378** |

order; LENGTH sorts history lengths; SEMANTIC uses item-word overlap and concentration with a hash tie-breaker. All fill the same method totals, isolating the user–method mapping.

RecDataAgent leads all six metrics in each domain. Recall@10 reaches 0.0820, 0.0828, and 0.0569, versus the strongest controls' 0.0717, 0.0771, and 0.0480. With coverage and method totals fixed, the advantage shows that allocation quality depends on who receives each view, not only how much of each method is used. The best simple rule also changes by domain: Length on Beauty, Random on Toys, and a Length–Semantic tie on Sports. Richer contextual assignments outperform these individual summaries.

**Validation outcomes improve policy revision.** The no-feedback planner retains task context, features, the method catalog, schema checks, and construction diagnostics. The feedback-conditioned planner additionally receives aggregate validation outcomes, rankings, incumbent labels, and outcome-derived mechanism statements. Both follow Appendix A.1; Table 4 matches their base corpus, appended-record count, and per-policy training and evaluation budget.

Feedback-conditioned policies lead all six metrics. Recall@10 increases from 0.0770 to 0.0820 on Beauty, 0.0756 to 0.0828 on Toys, and 0.0456 to 0.0569 on Sports, with corresponding NDCG gains. Since both planners already receive user features and execution diagnostics, the comparison supports the value of measured recommendation outcomes for constructing stronger policies, beyond describing users or repairing invalid proposals.

This comparison separates two roles in the loop: execution diagnostics establish whether an assignment can be materialized, whereas recommendation outcomes indicate how useful the resulting

**Table 3:** Routing controls under matched domain budgets. Random, Length, and Semantic replace the contextual route while preserving user coverage, method exposure, and downstream training; RecDataAgent uses the selected policy.

| Domain | Routing | HR@5 | NDCG@5 | R@10 | N@10 | R@20 | N@20 |
|---|---|---|---|---|---|---|---|
| Beauty | Random | 0.0503 | 0.0355 | 0.0711 | 0.0423 | 0.0968 | 0.0488 |
| | Length | 0.0502 | 0.0347 | 0.0717 | 0.0417 | 0.0981 | 0.0484 |
| | Semantic | 0.0459 | 0.0319 | 0.0685 | 0.0392 | 0.0939 | 0.0457 |
| | RecDataAgent | **0.0583** | **0.0407** | **0.0820** | **0.0488** | **0.1087** | **0.0555** |
| Toys | Random | 0.0550 | 0.0376 | 0.0771 | 0.0447 | 0.1001 | 0.0505 |
| | Length | 0.0537 | 0.0376 | 0.0750 | 0.0445 | 0.0982 | 0.0502 |
| | Semantic | 0.0527 | 0.0359 | 0.0747 | 0.0430 | 0.1008 | 0.0497 |
| | RecDataAgent | **0.0592** | **0.0410** | **0.0828** | **0.0487** | **0.1108** | **0.0557** |
| Sports | Random | 0.0317 | 0.0215 | 0.0474 | 0.0265 | 0.0648 | 0.0309 |
| | Length | 0.0319 | 0.0217 | 0.0480 | 0.0268 | 0.0669 | 0.0316 |
| | Semantic | 0.0331 | 0.0228 | 0.0480 | 0.0276 | 0.0669 | 0.0323 |
| | RecDataAgent | **0.0385** | **0.0265** | **0.0569** | **0.0324** | **0.0779** | **0.0378** |

**Table 4:** Final-policy performance with and without validation-derived planner feedback, at matched corpus size and per-policy training budget.

| Domain | Policy | HR@5 | NDCG@5 | R@10 | N@10 | R@20 | N@20 |
|---|---|---|---|---|---|---|---|
| Beauty | RecDataAgent | **0.0583** | **0.0407** | **0.0820** | **0.0488** | **0.1087** | **0.0555** |
| | No feedback | 0.0544 | 0.0380 | 0.0770 | 0.0453 | 0.1056 | 0.0525 |
| Toys | RecDataAgent | **0.0592** | **0.0410** | **0.0828** | **0.0487** | **0.1108** | **0.0557** |
| | No feedback | 0.0537 | 0.0364 | 0.0756 | 0.0435 | 0.1027 | 0.0503 |
| Sports | RecDataAgent | **0.0385** | **0.0265** | **0.0569** | **0.0324** | **0.0779** | **0.0378** |
| | No feedback | 0.0308 | 0.0211 | 0.0456 | 0.0259 | 0.0631 | 0.0303 |

corpus is. The performance difference supports retaining both channels rather than treating successful execution as sufficient feedback for policy construction.

**Source experience supports new assignments.** For Beauty-to-Toys and Toys-to-Sports transfer, the planner combines source policy–outcome records with target-domain features; the Sports stage also retains Beauty evidence. Model weights and routing thresholds remain domain-specific. Experience thus enters through planner context to construct new assignments, rather than copying a source route.

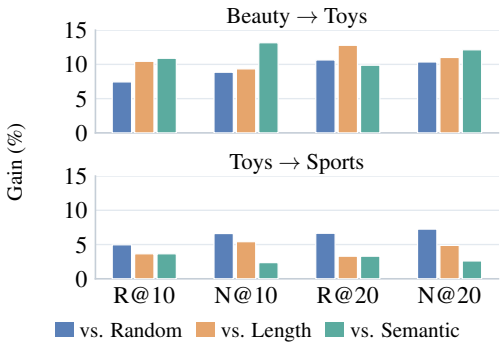


**Figure 3:** Relative gains of individual archived transfer-policy runs over the routing controls in Table 3.

Figure 3 shows the best individual archived transfer run for each target relative to the routing controls in Table 3; five-seed means are reported separately in Table 2. Both target policies improve on all three controls across the displayed retrieval and ranking metrics. These cases illustrate effective target-specific assignments constructed with source experience. The archive keeps the underlying constructions and outcomes available for both reuse and final selection.

**The archive preserves alternatives for final selection.** Table 5 shows one policy per round and the reference selected from all nine candidates. Their differing metric profiles explain why revision and final selection serve distinct purposes; Appendix A.1 gives the selection trace.

Beauty's displayed Recall@10 moves from 0.0791 to 0.0679 and then 0.0783; among the three displayed round policies, the third has the highest NDCG@10 and NDCG@20. Sports' second policy trades recall for NDCG@10, while later Toys policies have lower recall than the first. The selected reference leads all six reported metrics in every domain. These trajectories favor retaining alternatives over replacing the archive with the latest proposal: the final selection identifies a policy that outperforms the displayed round policies across all six reported metrics.

Beauty also illustrates why selection is performed at the policy level. Although its third-round policy recovers from the second round, the selected reference has higher values on all six reported metrics,

**Table 5:** Feedback-guided policy sequence and selected reference. Rounds 1–3 are successive policies in the revision sequence; Ref. is the final policy selected from the complete archive and reported in Table 2.

| Domain | Round | HR@5 | NDCG@5 | R@10 | N@10 | R@20 | N@20 | Outcome profile |
|---|---|---|---|---|---|---|---|---|
| Beauty | 1 | 0.0552 | 0.0382 | 0.0791 | 0.0459 | 0.1066 | 0.0528 | Round reference |
| | 2 | 0.0443 | 0.0293 | 0.0679 | 0.0369 | 0.0954 | 0.0439 | Lower scores |
| | 3 | 0.0562 | 0.0393 | 0.0783 | 0.0464 | 0.1075 | 0.0538 | Recovery |
| | Ref. | **0.0583** | **0.0407** | **0.0820** | **0.0488** | **0.1087** | **0.0555** | Validation-selected |
| Toys | 1 | 0.0530 | 0.0367 | 0.0749 | 0.0437 | 0.1012 | 0.0504 | Round reference |
| | 2 | 0.0521 | 0.0355 | 0.0725 | 0.0421 | 0.0956 | 0.0479 | Lower scores |
| | 3 | 0.0490 | 0.0333 | 0.0693 | 0.0399 | 0.0951 | 0.0464 | Lower scores |
| | Ref. | **0.0592** | **0.0410** | **0.0828** | **0.0487** | **0.1108** | **0.0557** | Validation-selected |
| Sports | 1 | 0.0328 | 0.0220 | 0.0496 | 0.0275 | 0.0692 | 0.0324 | Round reference |
| | 2 | 0.0332 | 0.0226 | 0.0493 | 0.0278 | 0.0682 | 0.0325 | Recall–NDCG trade-off |
| | 3 | 0.0316 | 0.0216 | 0.0472 | 0.0266 | 0.0675 | 0.0318 | Lower scores |
| | Ref. | **0.0385** | **0.0265** | **0.0569** | **0.0324** | **0.0779** | **0.0378** | Validation-selected |

including NDCG@10 (0.0488 versus 0.0464) and NDCG@20 (0.0555 versus 0.0538). Thus, the displayed results show no retrieval–ranking trade-off between these two policies. Keeping each metric vector attached to its policy lets the fixed selection rule choose a complete construction instead of assembling a row from different candidates.

**The Beauty route distinguishes concentration from breadth.** Figure 4 in Appendix A.2 makes the selected policy's user partition inspectable. Panel B locates the two groups in projected feature space; Panel C identifies their contrasting profiles. Cluster 1 (4,188 users) has greater category concentration and item-word overlap, whereas Cluster 0 (18,175 users) has greater category breadth and switching. The policy assigns GenPAS prefix views to Cluster 1 and AsarRec dual views to the remainder, with one appended record per user. Thus the route uses multiple behavioral dimensions to assign different views under the same exposure budget. The feature analysis explains the assignment, while Table 3 provides the performance comparison for the complete policy.

The profile contrast is primarily behavioral rather than a simple history-length or popularity split. In Panel C, the groups' median shifts for history length and the popularity-related coordinates are close to zero, while concentration, switching, and TID overlap differ more strongly. This makes the case complementary to the routing controls: it identifies the multidimensional distinctions used by the selected policy, beyond the single summaries tested by Length and Semantic. Panel B visualizes the fitted groups; Panel C gives their interpretation in the original features.

## 6 Conclusion

We studied recommendation augmentation as a question of assignment as well as construction. RecDataAgent makes user-selection conditions, method choices, and exposure explicit in one executable policy. Its controller connects multidimensional user analysis to corpus construction, while typed execution and an outcome-linked archive allow the resulting decisions to be evaluated and revised together. Across Amazon Beauty, Toys, and Sports, the selected policies improve retrieval and ranking over the strongest local augmentation baselines. Routing controls show that the user–method mapping matters even at fixed method exposure, and the feedback comparison shows that measured recommendation outcomes help construct stronger final policies. Transfer cases illustrate how archived experience can inform new domain-specific assignments, while retaining evaluated alternatives keeps final selection from depending only on the latest proposal. These findings shift the focus from choosing an augmentation method in isolation to improving the assignment that turns it into supervision. User context guides which views to construct for whom; recommendation outcomes guide how those choices should change. RecDataAgent makes this connection executable under a shared construction budget.

## AI Use Statement

AI tools assisted with manuscript writing and LaTeX formatting, discovery, and research execution. The authors reviewed the manuscript and remain responsible for its contents.

## ETHICS STATEMENT

This work studies data selection for sequential recommendation with language models. Responsible use should consider biases in training data and reward signals, as well as potential misuse of improved model capabilities.

## REPRODUCIBILITY STATEMENT

A reference implementation of the selector is available at: https://anonymous.4open.science/r/RecDataAgent-382D. The experimental setup is described in Section 5; Appendices A and C provide experimental and implementation details, respectively.

## REFERENCES

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1007–1014, 2023. doi: 10.1145/3604915.3608857. URL https://arxiv.org/abs/2305.00447v3.

Daoyuan Chen et al. Data-Juicer: A One-Stop Data Processing System for Large Language Models. *arXiv preprint arXiv:2309.02033*, 2024. URL https://arxiv.org/abs/2309.02033.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-It! A Data-Driven Skills Framework for Understanding and Training Language Models. *arXiv preprint arXiv:2307.14430*, 2023. URL https://arxiv.org/abs/2307.14430.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2202.02519*, 2022. URL https://arxiv.org/abs/2202.02519.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Cubuk_AutoAugment_Learning_Augmentation_Strategies_From_Data_CVPR_2019_paper.html.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/1909.13719v2.

Yizhou Dang, Enneng Yang, Yuting Liu, Guibing Guo, Linying Jiang, Jianzhe Zhao, and Xingwei Wang. A Survey of Data Augmentation Approaches for NLP-based Sequential Recommendation. *arXiv preprint arXiv:2409.13545*, 2024. URL https://arxiv.org/abs/2409.13545.

Talfan Evans, Shreya Pathak Parthasarathy, Hamza Merzic, Olivier J. Henaff, Mateusz Damoc, Alex Braun, George Zhu, Kaifeng Zhang, Chuhan Duan, Sylvain Gelly, Thomas Mesnard, Michal Valko, and Soham De. Data Curation via Joint Example Selection Further Accelerates Multimodal Learning. *arXiv preprint arXiv:2406.17711*, 2024. URL https://arxiv.org/abs/2406.17711.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2304.14108v5. Datasets and Benchmarks Track.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 299–315, 2022. doi: 10.1145/3523227.3546767. URL https://arxiv.org/abs/2203.13366v7.

Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2731–2741. PMLR, 2019. URL https://proceedings.mlr.press/v97/ho19b.html.

Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/36b7acf6f6010652b3f2a433774a66fe-Paper-Conference.pdf.

Feiyang Kang, Hanze Li, Adam Nguyen, Mahavir Dabas, Jiaqi W. Ma, Frederic Sala, Dawn Song, and Ruoxi Jia. Can Generalist Agents Automate Data Curation? *arXiv preprint arXiv:2606.04261v1*, 2026. URL https://arxiv.org/abs/2606.04261v1.

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

Geon Lee, Bhuvesh Kumar, Mingxuan Ju, Tong Zhao, Kijung Shin, Neil Shah, and Liam Collins. Sequential Data Augmentation for Generative Recommendation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 303–312, 2026. doi: 10.1145/3773966.3778000. URL https://doi.org/10.1145/3773966.3778000.

Xinhang Li, Chong Wang, Bei Zhu, Shanghua Wang, Zhenzhen Zhang, Yong Yu, Weinan Chen, and Jun Wang. E4SRec: An Elegant Effective Efficient Extensible Solution of Large Language Models for Sequential Recommendation. *arXiv preprint arXiv:2312.02443*, 2023. URL https://arxiv.org/abs/2312.02443.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Large Language-Recommendation Assistant. *arXiv preprint arXiv:2312.02445*, 2024. URL https://arxiv.org/abs/2312.02445.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data Mixture as Regression for Language Model Pre-training. *arXiv preprint arXiv:2407.01492*, 2024. URL https://arxiv.org/abs/2407.01492.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, et al. OneRec-Think: In-Text Reasoning for Generative Recommendation. *arXiv preprint arXiv:2510.11639*, 2025. URL https://arxiv.org/abs/2510.11639.

Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian McAuley, and Caiming Xiong. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. *arXiv preprint arXiv:2108.06479v1*, 2021. URL https://arxiv.org/abs/2108.06479v1.

Sichun Luo, Yuxuan Yao, Bowei He, Wei Shao, Jian Xu, Yinya Huang, Aojun Zhou, Xinyi Zhang, Yuanzhang Xiao, Hanxu Hou, Mingjie Zhan, and Linqi Song. Integrating Large Language Models into Recommendation via Mutual Augmentation and Adaptive Aggregation. *IEEE Journal on Selected Topics in Signal Processing*, 20(1):77–89, 2026. doi: 10.1109/JSTSP.2026.3653160. URL https://doi.org/10.1109/JSTSP.2026.3653160.

Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. *arXiv preprint arXiv:2110.05730*, 2022. URL https://arxiv.org/abs/2110.05730.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.05065v3.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. AgentSquare: Automatic LLM Agent Search in Modular Design Space. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0ae94013da7cd459402fd77874e09ee3-Paper-Conference.pdf.

Yucheng Shi, Ying Li, Yu Wang, Yesu Feng, Arjun Rao, Rein Houthooft, Shradha Sehgal, Jin Wang, Hao Zhen, Ninghao Liu, and Linas Baltrunas. From Logs to Language: Learning Optimal Verbalization for LLM-Based Recommendation at Industry Scale. *arXiv preprint arXiv:2602.20558v2*, 2026. URL https://arxiv.org/abs/2602.20558v2.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

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.

Jianling Wang, Ya Le, Bo Chang, Yuyan Wang, Ed H. Chi, and Minmin Chen. Learning to Augment for Casual User Recommendation. In *Proceedings of the ACM Web Conference 2022*, 2022a. doi: 10.1145/3485447.3512147. URL https://arxiv.org/abs/2204.00926v1.

Lei Wang, Ee-Peng Lim, Zhiwei Liu, and Tianxiang Zhao. Explanation Guided Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2209.01347*, 2022b. URL https://arxiv.org/abs/2209.01347.

Xinping Wang, Yuqing Cai, Kai Lu, Jing Geng, Xiaomin Li, Lichen Cai, and Zhenhua Ling. LLMSeR: Large Language Model as Data Enhancer for Sequential Recommendation. *arXiv preprint arXiv:2503.12547*, 2025. URL https://arxiv.org/abs/2503.12547.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.10429v4.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629v3.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, Xipeng Qiu, and Xuanjing Huang. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance. *arXiv preprint arXiv:2403.16952*, 2024. URL https://arxiv.org/abs/2403.16952.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. *arXiv preprint arXiv:2402.17152*, 2024. URL https://arxiv.org/abs/2402.17152.

Kaike Zhang, Qi Cao, Fei Sun, Xinran Liu, Huawei Shen, and Xueqi Cheng. AsarRec: Adaptive Sequential Augmentation for Robust Self-supervised Sequential Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2026a. URL https://arxiv.org/abs/2512.14047v2. Author manuscript, arXiv:2512.14047v2.

Zhiyang Zhang, Junda She, Kuo Cai, Bo Chen, Shiyao Wang, Xinchen Luo, Qiang Luo, Ruiming Tang, Han Li, Kun Gai, and Guorui Zhou. Unleashing the Native Recommendation Potential: LLM-Based Generative Recommendation via Structured Term Identifiers. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 19659–19673, 2026b. doi: 10.18653/v1/2026.findings-acl.984. URL https://aclanthology.org/2026.findings-acl.984/.

# A EXPERIMENTAL DETAILS

This appendix expands the evaluation setup and conclusion-driven analyses in Section 5. It follows a complete policy from planner inputs through corpus construction, training, evaluation, and final reporting. The protocol and selection trace specify how the reported scores are obtained; the Beauty case study then connects train-visible feature structure to an executable route and a concrete augmented record.

**Controller inputs and execution.** The controller uses train-visible context and catalogued interfaces to produce declarative policies. Its outcome channel comprises aggregate validation metrics, rankings, and incumbent labels; held-out labels, per-user predictions, and model weights remain on the evaluator side. Source paths serve as provenance identifiers, with file access handled by the adapter. Construction methods produce views from source context, and the materializer binds those views to base records while preserving the instruction and target. The resulting examples retain the original recommendation task. Schema approval precedes materialization, and record checks precede evaluation. The trace records intended allocation, realized exposure, and check status as distinct fields. Inapplicable actions retain their recorded status and contribute zero added records, with their quotas left unfilled. Completed trials supply utility observations; invalid or incomplete attempts supply diagnostics for repair. The executor follows the declared routes and remainder action.

**Result reporting.** The matched augmentation comparisons use L2Aug, GenPAS, AsarRec, and RecDataAgent; the architectural reference rows retain their published protocols. Figure 3 describes individual archived transfer-policy runs, while Table 2 reports five-seed means for the local rows. Transfer operates through planner evidence, with model weights and routing thresholds remaining domain-specific. The no-feedback comparison evaluates complete policies at matched corpus size and per-policy training budget. The historical Beauty design log in Table 9 records routing, exposure, and Recall@10. Its short labels also appear in Table 7, where the reported scores differ.

## A.1 EXPERIMENTAL PROTOCOL

**Evaluation protocol.** The evaluation uses Amazon Beauty, Toys, and Sports sequential recommendation. Each example provides a preceding item sequence and predicts the next item's five characteristic words. The fixed populations contain 22,363 Beauty users, 19,412 Toys users, and 35,598 Sports users. The principal comparisons use Qwen3-4B-Instruct-2507 with one epoch of full-parameter BF16 training, cutoff length 2,048, cosine decay from $10^{-4}$, and effective global batch size 80. DeepSpeed ZeRO-3 is used for training; hardware-dependent micro-batches vary while the effective global batch remains fixed. The five training seeds are $\{40, 41, 42, 43, 44\}$ and the reference data seed is 42. The matched local rows are L2Aug, GenPAS, AsarRec, and RecDataAgent. They retain their method-specific constructions under the shared backbone and training horizon.

For the reported local results, deterministic decoding uses batch size 1, 20 beams, 20 returned sequences, and at most 30 new tokens. Local results aggregate six disjoint evaluation shards, with checks covering aggregate identity, protocol consistency, and shard completeness. This gives each local method the same decoding width and a complete domain-level evaluation. Training data provide the base corpus, augmentation inputs, and user features, while aggregate validation outcomes supply the evidence used for policy revision and selection.

**Main-table aggregation.** For each domain, every metric in the locally evaluated rows of Table 2—L2Aug, GenPAS, AsarRec, and RecDataAgent—is the arithmetic mean across the five training seeds $\{40, 41, 42, 43, 44\}$. The same selected policy supplies all six metrics in its domain row. Consequently, the main comparison describes a complete ranking profile for one corpus policy, with the seed-level outcomes combined by the same arithmetic operation for each metric.

**Search budget and stopping rule.** For each domain, policy search evaluates three candidates per round for three rounds, yielding a fixed budget of nine candidates. After Round 3, the completed archive supplies the candidate set for final selection. The reference row in Table 5 reports the selected policy. The selection utility is validation Recall@10 from the fixed seed-42 reference run, computed as the user-count-weighted micro-average over complete validation shards. This criterion is fixed before candidate generation and governs candidate ranking, incumbent updates, and final selection; exact

**Table 6:** Round-level selection trace, pairing seed-42 validation Recall@10 with the reported Test Recall@10 from Table 5. Selection roles follow the fixed search protocol.

| Domain | Entry | Validation R@10 | Test R@10 | Selection role |
|---|---|---|---|---|
| Beauty | Round 1 | 0.0824 | 0.0791 | Displayed round |
| | Round 2 | 0.0701 | 0.0679 | Displayed round |
| | Round 3 | 0.0822 | 0.0783 | Displayed round |
| | REF. | **0.0881** | **0.0800** | Final selected |
| Toys | Round 1 | 0.0803 | 0.0749 | Displayed round |
| | Round 2 | 0.0745 | 0.0725 | Displayed round |
| | Round 3 | 0.0719 | 0.0693 | Displayed round |
| | REF. | **0.0901** | **0.0828** | Final selected |
| Sports | Round 1 | 0.0523 | 0.0496 | Displayed round |
| | Round 2 | 0.0535 | 0.0493 | Displayed round |
| | Round 3 | 0.0584 | 0.0472 | Displayed round |
| | REF. | **0.0631** | **0.0569** | Final selected |

**Table 7:** Beauty round-policy ledger. Each displayed row reports formal Full-evaluation metrics for one completed policy. The bold row is the final reference policy used in the main comparison.

| Round | Candidate | R@10 | N@10 | R@20 | N@20 | Role |
|---|---|---|---|---|---|---|
| 1 | `research-1` | 0.0791 | 0.0459 | 0.1066 | 0.0528 | |
| 2 | `research-2` | 0.0679 | 0.0369 | 0.0954 | 0.0439 | |
| 3 | `research-3` | 0.0783 | 0.0464 | 0.1075 | 0.0538 | |
| REF. | `main-reference` | **0.0800** | **0.0463** | **0.1087** | **0.0535** | **Final Ref.** |

ties favor the earlier round and then the lower candidate index. Candidate-specific hypothesis metrics describe the proposed comparisons. The main table reports five-seed means for the selected policies.

**Round-level selection trace.** Table 6 links the selection utility to the corresponding Test Recall@10 in Table 5. Rounds 1–3 identify one displayed policy per round; REF. identifies the final choice from all nine candidates, including the six candidates outside the displayed sequence. The selected reference leads the displayed round policies on both validation and Test Recall@10 in every domain. This agreement connects the fixed selection criterion to a favorable final recommendation outcome, while the full trace retains the alternative policies for comparison.

**Beauty policy results.** Table 7 identifies the three displayed Beauty policies as `research-1`, `research-2`, and `research-3`. The ledger connects each displayed policy to its formal Full-evaluation outcomes under deterministic 20-beam decoding. The third displayed policy recovers both recall and NDCG after the second policy's lower scores. The final reference has higher Recall@10 and Recall@20 than all three displayed policies, while `research-3` retains slightly higher NDCG at both cutoffs. Keeping the four metrics together exposes the ranking trade-off behind this example of validation-based selection.

**Candidate corpus budget.** Each candidate retains one complete copy of the base corpus. The principal configurations append 22,363, 19,412, and 35,572 records in Beauty, Toys, and Sports, respectively, as summarized in Table 1; the same domain-specific budgets are used in the matched feedback comparison. The resulting complete corpora contain 56,827, 50,748, and 89,527 records. Every eligible user contributes at most one appended record, and the 26 unsupported Sports users remain base-only. These settings preserve the exact base-record prefix and fixed method applicability across local comparisons. Within each domain, the controller must express its choices through the appended views and their assignments under this common budget.

### A.2 FEATURE-GUIDED CASE STUDY

**Planner-visible feature prompt.** The planner received the following feature block as part of its Beauty research prompt. Here $L$ is the train-visible history length, a leaf category is the final normalized category component, and each item TID contains five words.

**USER_FEATURES (name = computation; planner-visible meaning).**

1. `history_length` $= L$; available sequential context.
2. `p_law_item_actual_entropy` $=$ actual_entropy(item-ID sequence); the supplied P-Law statistic, equal to $\ln L$ for this population.
3. `category_actual_entropy` $=$ actual_entropy(ordered leaf-category sequence); ordered category complexity.
4. `category_unique_ratio` $= \#\text{unique leaf categories}/L$; category breadth.
5. `category_top_share` $= \max_c \#c/L$; concentration in the most frequent category.
6. `category_shannon_entropy_norm` $= -\sum_c p_c \ln p_c / \ln L$; normalized category-frequency diversity.
7. `category_switch_rate` $= \#\{c_t \neq c_{t-1}\}/(L-1)$; adjacent category switching.
8. `tid_unique_ratio` $= \#\text{unique TID words}/(5L)$; TID vocabulary breadth.
9. `tid_top_share` $= \max_w \#w/(5L)$; concentration of the most frequent TID word.
10. `tid_shannon_entropy_norm` $= -\sum_w p_w \ln p_w / \ln(5L)$; normalized TID-word frequency diversity.
11. `adjacent_tid_jaccard_mean` $= \text{mean}_t\, |T_{t-1} \cap T_t| / |T_{t-1} \cup T_t|$; local semantic-word continuity.
12. `early_late_tid_jaccard` $= |T_{\text{early}} \cap T_{\text{late}}| / |T_{\text{early}} \cup T_{\text{late}}|$; persistence between the first and second history halves.
13. `late_new_tid_ratio` $= |T_{\text{late}} \setminus T_{\text{early}}| / |T_{\text{late}}|$; recently introduced TID words.
14. `item_log_popularity_mean` $= \text{mean}_i \ln(1 + \text{freq}_{\text{train}}(i))$; average item popularity.
15. `item_log_popularity_std` $= \text{std}_i \ln(1 + \text{freq}_{\text{train}}(i))$; within-history popularity range.
16. `long_tail_item_ratio` $= \#\{i : \text{freq}_{\text{train}}(i) \leq 8\}/L$; share of items in the frozen training-frequency tail.
17. `popularity_trend` $= \text{mean}_{\text{late}} \ln(1 + \text{freq}) - \text{mean}_{\text{early}} \ln(1 + \text{freq})$; movement toward more or less popular items.

**CLUSTERING.**

Retain all 17 features in the planner input. Fit the user clusters with the following 14 nonredundant coordinates:

- `history_length`, `category_actual_entropy`, `category_unique_ratio`, `category_top_share`.
- `category_shannon_entropy_norm`, `category_switch_rate`, `tid_unique_ratio`, `tid_top_share`.
- `adjacent_tid_jaccard_mean`, `early_late_tid_jaccard`.
- `item_log_popularity_mean`, `item_log_popularity_std`.
- `long_tail_item_ratio`, `popularity_trend`.

Apply median/IQR RobustScaler normalization. Compare KMeans $K = 2, \ldots, 8$ with `random_state=42` and `n_init=20`; among candidates whose smallest cluster contains at least 2% of users, select the highest sampled silhouette.

**Meaning of `cluster_id`.** The resulting `cluster_id` is the categorical integer label produced by this frozen Beauty-specific KMeans fit. Figure 4 connects the cluster-count selection, projected feature geometry, and group-level feature profiles. In Panel A, $K = 2$ achieves the highest sampled silhouette, 0.386, among the evaluated counts; Panels B and C show the corresponding user partition and feature contrasts.

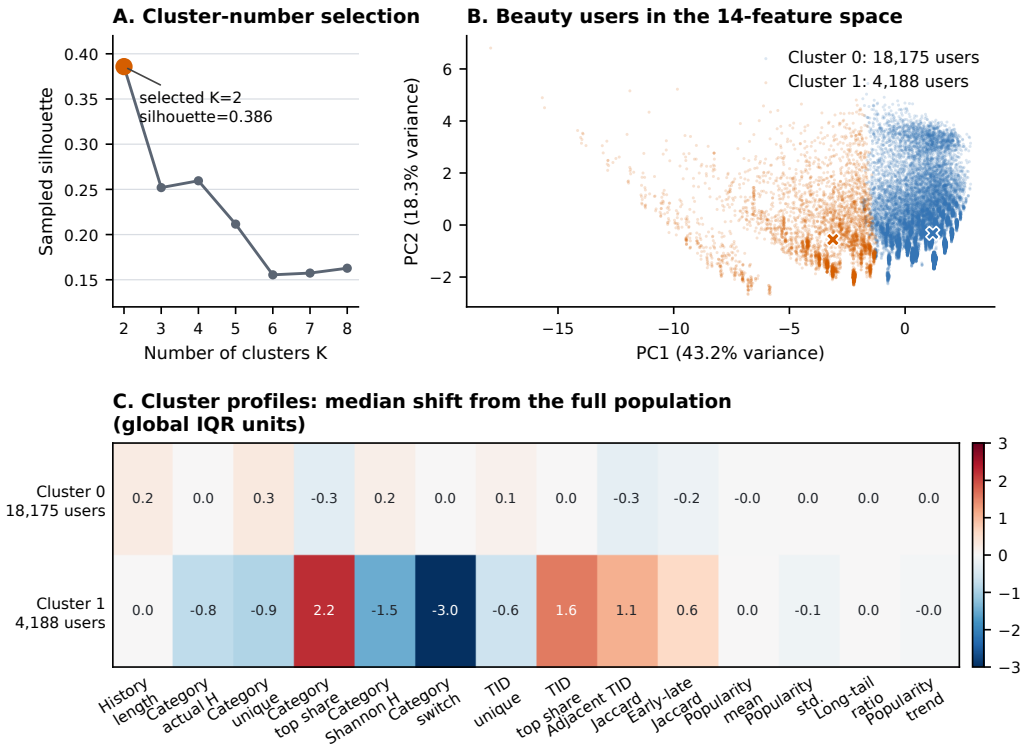


**Figure 4:** Beauty features underlying the selected route. **A:** Sampled silhouette scores select $K = 2$ under the minimum-cluster-size condition specified above. **B:** Principal component analysis (PCA) projects the 14 clustering features; crosses mark projected group medians. **C:** Group medians minus population medians, in population interquartile-range (IQR) units. H denotes category entropy; TID denotes the five-word item text identifier.

**Representative augmentation example.** Consider a train-visible Beauty history of length four assigned to Cluster 1. Its feature profile belongs to the group with stronger category concentration and adjacent TID-word overlap, for which the selected policy supplies a GenPAS prefix view. The following excerpt shows how that history is presented to the recommender after augmentation.

```
Item text ID: [nail, art, pearl, stone, small]
Title: Islandoffer Color Pearl Nail Art Stone Small Wheel
Rhinestones Beads.

<genpas_augmented_sequence>
Item ID: 6066; Item text ID: [nail, art, pearl, stone, small]
Item ID: 4171; Item text ID: [nail-art, 3d, colorful,
slices, diy]
Item ID: 1270; Item text ID: [nail-art, rhinestones,
white-pearl, round, 1.5mm]
```

The augmentation contributes an additional structured view of the observed history under the record-binding convention in Appendix A. The excerpt retains the item identifiers and five-word item representations, making the selected prefix directly readable as recommendation context. Its repeated nail-art vocabulary is consistent with the concentrated feature group to which the user was assigned. The materializer attaches this view to the matched base input while retaining the original instruction and prediction target, connecting the group-level route to the actual training record.

### A.3 Planner Prompt and Feature-Guided Policy

**Prompt organization.** The formal planner input combines the task, train-visible data and features, clustering analysis, method definitions, applicability constraints, released validation evidence, and the JSON response contract. Table 8 summarizes these components. The accompanying *RecDataAgent Planner Protocol and Prompt Specification* provides an author-edited English specification of the formal interface, including the feedback-conditioned and no-feedback settings.

**Table 8:** Planner-input specification for feature-guided policy construction. Task context, user features, and execution context support proposal construction; validation feedback supplies the revision signal.

| Component | Content and role |
|---|---|
| Task and data context | Recommendation task, train-visible histories, item information, and the unchanged base corpus. |
| Features and user groups | Feature definitions, distribution summaries, clustering, and executable route predicates. |
| Method and execution context | Baseline methods, applicability, exposure constraints, and the response schema. |
| Validation feedback | Completed validation outcomes, rankings, and incumbent labels for policy revision. |

The no-feedback planner uses the task, feature, and execution components. Feedback-conditioned search adds the outcome channel. Both conditions follow Appendix A.1. This organization keeps the train-visible construction context available in both settings and makes validation-derived evidence a distinct planner input. The resulting policies are then materialized and evaluated through the same typed execution interface.

**Language and presentation.** The production planner prompt was issued in Chinese. The feature definitions and operative instructions are organized here as an English protocol specification, with the Beauty policy providing a concrete case study.

**Recommendation instruction.**

```
Based on the user's historical product interaction sequence,
predict the next product's characteristic words.
Each product is represented by exactly 5 characteristic words
enclosed in square brackets []. The historical sequence shows
the user's interaction pattern.
```

**Operative prompt rendering.** **Role and task.** Act as a recommendation-data research agent for Beauty sequential recommendation with Qwen3-4B-Instruct-2507. Propose exactly three augmentation hypotheses with substantively different mechanisms. Separate observed statistics, mechanism hypotheses, and candidate evaluation objectives.

**Data and input.** The base corpus contains 34,464 JSON records, each with exactly three string fields: `instruction`, `input`, and `output`. Its first 22,363 records are recommendation examples and its remaining 12,101 records are item-description-to-five-word-TID examples. The augmentation source contains 22,363 users with train-visible histories of length 3–18. Policy-generation inputs comprise `user_idx`, `sequence_item_ids`, `sequence_tids`, derived train-visible features, and the train-visible catalog.

**Base corpus and appended views.** Each candidate dataset retains one unchanged copy of the complete base corpus in its original order. An augmentation creates a new record by copying the matched base record's `instruction` and `output` and appending the selected method view to its `input`. Each selected user contributes one appended row, giving coverage histogram `{"1": N}` for $N$ unique users.

**Available methods and routes.** Use the listed methods with their actual signatures. The available method implementations are `asarrec_dual_view`, `genpas_prefix_view`, and `l2aug_sequence_view`. A feature route contains one to eight conjunctive numeric predicates using <, <=, >, >=, ==, or inclusive `between`; `cluster_id` may only be compared for equality to an existing integer. Within a candidate, routes must be mutually exclusive. The selected streams are combined by the declared route assignment, and any uncovered selected user must go to an explicit remainder stream.

**Coverage conditions.** Execution checks method applicability for every selected user, preserves unique-user exposure, and routes the complete remainder explicitly. A missing predicate or method triggers a minimal capability request and dry-run before materialization.

**Evidence checks.** Exact joint-predicate counts, interface signatures, and applicability are verified through `code_evidence`. Each dry-run reports covered users, remainder users, cross-route overlap, and method applicability.

**Feedback and selection.** Use aggregate validation outcomes to revise proposals. Apply the selection and reporting protocol in Appendix A.1. Candidate-specific hypotheses describe comparisons within that protocol.

**Response protocol.** Return a complete JSON object whose root contains exactly `action`, `reason`, `experiments`, and `requests`. The action is either `execute` or `needs_capability`. For `execute`, return exactly three experiments and an empty request list; for `needs_capability`, return an empty experiment list and at least one concrete request.

```
{
  "action": ..., "reason": ...,
  "experiments": [{
    "name": ..., "total_exposure": ...,
    "operators": [{"name": ..., "params": ...}],
    "rationale": ..., "risks_and_checks": ...,
    "coverage": {
      "unique_users": N,
      "exposure_histogram": {"1": N}
    },
    "comparison": {
      "reference": ..., "primary_metric": ...,
      "hypothesis": ..., "changed": [...], "fixed": [...],
      "interpretation": ...
    }
  }],
  "requests": [{
    "kind": ..., "target": ...,
    "details": ..., "why_needed": ...
  }]
}
```

Each request `kind` must be either `implement_function` or `code_evidence`; each method entry contains exactly `name` and `params`. The primary metric must be one of Recall@10, NDCG@10, Recall@20, or NDCG@20. The `primary_metric` field annotates the metric emphasized by a candidate's proposed comparison.

**Feature-guided policy definition.**

```
genpas_prefix_view(
  source_id="beauty-all-train-visible-v1",
  policy_id="genpas_full_available_seed42_v1",
  output_stream="cgeom_genpas",
  applicability="require_all")

asarrec_dual_view(
  source_id="beauty-all-train-visible-v1",
  policy_id="asarrec_full_available_seed42_v1",
  output_stream="cgeom_asarrec",
  applicability="require_all")

feature_route_select(
  routes=[{
    stream: "cgeom_genpas",
    query: {all: [{
      feature: "cluster_id", op: "==", value: 1
    }]},
    applicability: "require_all"
  }],
  remainder_stream="cgeom_asarrec",
  merge="concatenate")

emit_standard_artifacts()
```

**Alternative hypotheses.** The response also proposed routes based on zero early-versus-late TID-word overlap and on joint long-tail exposure with decreasing popularity. These alternatives express different train-visible distinctions through the same route-and-remainder structure. The cluster-geometry policy shown above is used for the Beauty results in Table 2. Together with the routing controls in Table 3, this case connects a concrete multidimensional partition, its realized method assignments, and the recommendation profile of the complete corpus.

## B Historical Policy-Revision Traces

This section presents the historical Beauty policy-revision log. Each entry records the route definition, method exposure, and observed Recall@10 together, making changes to the complete construction policy explicit.

**Table 9:** Historical Beauty design logs: routing, exposure, and Recall@10 recorded within the historical trace.

| Trace | Routing basis | Method / dose | R@10 |
|---|---|---|---|
| research-1 | GenPAS: $L = 3$–$6$; AsarRec: $L = 7$–$18$ | GenPAS 15,874; AsarRec 6,489; total 22,363 | 0.0771 |
| research-2 | AsarRec: $L = 3$–$5$, 10–18; GenPAS: $L = 6$–$9$ | AsarRec 17,610; GenPAS 4,753; total 22,363 | 0.0626 |
| research-3 | GenPAS: $L = 3$–$6$, 13–18; AsarRec: $L = 7$–$12$ | GenPAS 18,063; AsarRec 4,300; total 22,363 | 0.0736 |
| feedback candidate | No length route; stable round-robin | AsarRec 11,875; GenPAS 3,125; total 15,000; 3,125 repeated | 0.0752 |

**Policy components.** We treat route predicates and feature partitions as the *analytical basis*, and methods, exposure allocation, merge rule, and repetition as the *augmentation configuration*. This separates whom to route from which view to supply and at what dose.

**Routing and exposure changes.** The analytical basis evolves: the first trace uses two length segments, the second uses three with GenPAS in the middle, and the third moves $L = 13$–$18$ back to GenPAS while retaining AsarRec for $L = 7$–$12$. The associated recall profile declines and then partially recovers. Reading the boundaries and outcomes together makes the user–method assignment a concrete object of comparison: each entry expresses a different hypothesis about the users suited to each available view.

The augmentation configuration also changes. Across the three research traces, the method pair remains GenPAS plus AsarRec, but the exposure allocation changes from 15,874/6,489 to 4,753/17,610 and then to 18,063/4,300. The feedback candidate changes the budget to 15,000 and uses stable round-robin mixing with 3,125 repeated exposures. The log therefore captures both changes in the user partition and changes in the composition of the appended corpus. These are represented as separate policy fields even when they change in the same proposal.

**Feedback-conditioned proposal.** The recorded policy history compares earlier single-method results and uses archived outcomes to propose an AsarRec-heavy mixture with a smaller GenPAS component. That candidate materialized with 15,000 appended rows and one unchanged base-corpus copy. The example connects archived recommendation outcomes to an executable change in method proportions, mixing rule, and exposure budget. Recording these choices together makes the complete construction available for subsequent comparison and reuse.

# C Formal Method Details

This appendix records the construction interfaces, record contracts, accounting rules, and archive transitions used by the method in Section 4.

## C.1 Typed Construction and Feasibility

For a method $o$ with parameters $\eta$, the typed transformation is

$$o_\eta = (A_{o,\eta}, T_{o,\eta}, \Gamma_{o,\eta}), \tag{7}$$

where $A_{o,\eta} : \mathcal{Z}_\delta \to \{0, 1\}$ checks applicability, $T_{o,\eta}$ emits train-visible views, and $\Gamma_{o,\eta}$ returns a provenance trace for each emitted view. If applicability is false, no view is emitted and the inapplicable status is recorded. The interface is shared across sequence views, retrieved context, relabeled examples, and synthetic views.

The paired output has type $(T_{o,\eta}, \Gamma_{o,\eta}) : \mathcal{S}_\delta \to \mathsf{View}_\delta^* \times \mathsf{Trace}_\delta^*$, where $*$ denotes a finite collection. Let $B_\delta(e) = \{b \in D_\delta^0 : e(b) = e\}$ denote the multiset of base records owned by $e$. The allocation rule $\nu$ maps the context population and route quotas to nonnegative integer exposures $\nu_{\pi,e}$; $\kappa$ specifies their budget, coverage, identity, label, and applicability constraints. The policy-induced corpus is

$$\begin{aligned} \widehat{D}_\delta(\pi) &= \biguplus_{e \in \mathcal{E}_\delta} \mathrm{Bind}_\delta\Big(B_\delta(e), \mathsf{Expose}_{\nu_{\pi,e}}\big(T_{a_\pi(e)}(s_e)\big)\Big), \\ D_\delta(\pi) &= D_\delta^0 \uplus \widehat{D}_\delta(\pi). \end{aligned} \tag{8}$$

The exposure operator deterministically selects or repeats views; no record is emitted when the assigned exposure is zero. The binding operator attaches the view to the base record and leaves adapter-declared immutable fields unchanged.

Let $D_\delta^{\mathrm{src}} = \{(e, s_e, B_\delta(e))\}_{e \in \mathcal{E}_\delta}$ be the source package and $\Pi_\delta^{\mathrm{sch}} = \{\pi \in \Pi_\delta : V_\delta^{\mathrm{sch}}(\pi) = 1\}$ the schema-valid policies. Feasibility additionally requires the declared constraints and a valid realized record–trace pair:

$$\Pi_\delta^{\mathrm{feas}} = \left\{\pi \in \Pi_\delta : V_\delta^{\mathrm{sch}}(\pi) = 1,\ \pi \models \kappa,\ V_\delta^{\mathrm{run}}(\pi, X_\delta(\pi, D_\delta^{\mathrm{src}})) = 1\right\}. \tag{9}$$

The schema check runs before materialization; the run check runs after materialization and before training.

### C.2 Adapter and Record Contracts

For the fixed source package, the materializer and validator have typed signatures

$$\begin{aligned} X_\delta(\cdot, D_\delta^{\mathrm{src}}) : \Pi_\delta^{\mathrm{sch}} &\longrightarrow \mathsf{Record}_\delta^* \times \mathsf{Trace}_\delta^*, \\ V_\delta^{\mathrm{run}} : \Pi_\delta^{\mathrm{sch}} \times (\mathsf{Record}_\delta^* \times \mathsf{Trace}_\delta^*) &\longrightarrow \{0, 1\}. \end{aligned} \tag{10}$$

For every emitted record $r$, $\beta(r)$ is its bound base record and $\gamma(r)$ is its method trace. If $\mathcal{F}_\delta^{\mathrm{fix}}$ is the set of immutable task fields, the record contract is

$$\begin{aligned} e(r) = e(\beta(r)), \qquad f(r) = f(\beta(r)) \quad \forall f \in \mathcal{F}_\delta^{\mathrm{fix}}, \\ \mathrm{trace}(r) = (\beta(r), o, \eta, \gamma(r)). \end{aligned} \tag{11}$$

Thus record identity, instruction, target, and provenance remain checkable after augmentation.

For route sets $R_j(\pi)$ induced by first-match predicates,

$$R_j(\pi) = \{e : g_j(z_e) = 1\} \setminus \bigcup_{\ell < j} R_\ell(\pi), \qquad R_0(\pi) = \mathcal{E}_\delta \setminus \bigcup_{j=1}^{R} R_j(\pi). \tag{12}$$

These sets form a disjoint partition before applicability filtering. Realized exposure and coverage are measured at record and user level:

$$n_e(\pi) = \sum_{r \in \widehat{D}_\delta(\pi)} \mathbf{1}[e(r) = e], \quad N(\pi) = \sum_e n_e(\pi), \quad \rho(\pi) = \frac{|\{e : n_e(\pi) > 0\}|}{|\mathcal{E}_\delta|}. \tag{13}$$

The run validator checks that these realized quantities agree with the declared policy and corpus constraints.

### C.3 Archive Transition and Portability

The archive transition used by the controller is

$$\pi_t^{(k)} \xrightarrow{V_\delta^{\mathrm{sch}}} (v_t^{(k)}, d_t^{(k)}) \xrightarrow[v_t^{(k)}=1]{X_\delta} (\widehat{D}_t^{(k)}, \tau_t^{(k)}) \xrightarrow{V_\delta^{\mathrm{run}}} v_{t,\mathrm{run}}^{(k)} \xrightarrow[v_{t,\mathrm{run}}^{(k)}=1]{G_\delta} \mathbf{y}_t^{(k)} \xrightarrow{\mathrm{Update}} \mathcal{H}_{t+1}. \tag{14}$$

Here $d_t^{(k)}$ is a schema, execution, or infrastructure diagnostic, $\tau_t^{(k)}$ is the realized provenance trace, and $\mathbf{y}_t^{(k)}$ contains aggregate validation outcomes. A trial is complete after both validators pass and training and validation finish. Its status determines the update:

$$\begin{aligned} \mathcal{H}_{t+1}^+ &= \mathcal{H}_t^+ \cup \{(\pi_t^{(k)}, \mathbf{y}_t^{(k)}, \tau_t^{(k)}) : \mathrm{complete}_t^{(k)} = 1\}, \\ \mathcal{H}_{t+1}^- &= \mathcal{H}_t^- \cup \{(\pi_t^{(k)}, d_t^{(k)}) : \mathrm{complete}_t^{(k)} = 0,\ d_t^{(k)} \neq \varnothing\}. \end{aligned} \tag{15}$$

The completed archive stores measured utility; the diagnostic archive supplies repair information. Keeping these channels distinct lets the planner repair construction failures without treating them as recommendation outcomes.

**Policy-space inclusion.** Let $\Pi_\delta^{\text{global}}$ contain context-independent action and exposure rules, including a fixed mixture implemented by a deterministic merge action. The contextual policy class includes these configurations:

$$\Pi_\delta^{\text{global}} \subseteq \Pi_\delta^{\text{contextual}}. \tag{16}$$

A global rule uses one context-independent route. With two distinguishable contexts and compatible methods that emit different views, a contextual rule can assign different methods to the two groups; this gives the strict inclusion described in the original formulation. Feasible assignments additionally obey the domain constraints.

**Adapter and method extension.** For a new domain $\delta'$, the adapter supplies a context encoder, typed method catalog, schema and run validators, deterministic materializer, and aggregate evaluator. Replacing this adapter preserves the policy grammar, route semantics, archive update, and feedback protocol. Adding a method $\tilde{o}$ that satisfies the same contract gives

$$\mathcal{R}'_\delta = \mathcal{R}_\delta \cup \{\tilde{o}\} \quad \Longrightarrow \quad \Pi(\mathcal{R}_\delta) \subseteq \Pi(\mathcal{R}'_\delta). \tag{17}$$

Each call remains defined because the substituted adapter supplies its required input and output types. This is structural portability of the search interface; the new domain supplies its own features, constructions, and utility.