SCOREd: Student-Aware CoT Optimization for Recommendation Distillation

Anonymous authors
Paper under double-blind review

Abstract

Chain-of-thought (CoT) distillation in the recommendation domain is a necessary precursor to RL training, but raw teacher traces are ill-suited to this task. Large teachers approach the recommendation task with unusually high reasoning uncertainty, repeatedly rechecking their answers without revising them; supervised fine-tuning on such traces produces verbose students that never revise their initial guess. Moreover, teacher-generated reasoning traces may be too complex for small student LLMs to learn effectively. In this work, we propose **S**tudent-Aware **C**oT **O**ptimization for **Re**commendation **D**istillation (SCOReD), a CoT optimization framework tailored to recommendation that first parses each teacher trace into typed segments and uses the student LLM's attention to score the importance of each segment. SCOReD then selects an edit for each segment (KEEP, REWRITE, FUSE, or PRUNE) using a reward that balances the student's log probability of the target answer against the edited segment's length and perplexity. Therefore, SCOReD prunes redundant sections of the reasoning trace while preserving information-dense sections and adapts raw teacher traces to the student's output distribution. Training on SCOReD-optimized CoTs provides a cleaner learning signal to the student model and improves over baseline SFT by 2.88% NDCG and 2.96% Recall@3, while producing shorter reasoning traces.

1 Introduction

Recommendation systems form the computational backbone of modern digital platforms, personalizing user experiences at scale by automatically matching individuals with relevant content, products, and services based on historical behavior. Recent works have established LLM-powered recommendation as a promising alternative to conventional discriminative retrieval and ranking (Liu et al., 2025; Liang et al., 2026; Li et al., 2026). Deng et al. (2025) propose OneRec, an end-to-end generative LLM-based recommender that unifies retrieval and ranking by directly generating recommendation outputs. OneRec-Think (Liu et al., 2025) further demonstrates that Chain-of-Thought (CoT) reasoning over users' purchase histories can improve the accuracy of generative recommenders. However, reasoning-enabled generative recommenders pose practical deployment challenges for systems that serve millions of concurrent users. Latency and computational constraints therefore require deployed recommender LLMs to remain small. To this end, Liang et al. (2026) and Wang et al. (2024) distill the reasoning capabilities of large teachers into smaller students by supervised fine-tuning on teacher-generated reasoning traces. Yet the gap between teacher and student capabilities can make complex teacher-generated reasoning difficult for the student to learn, resulting in suboptimal distillation (Cao et al., 2026). Effective distillation therefore requires traces that are not only correct but also appropriate for the target student's capacity.

Distillation for recommendation is further complicated by noisy ground-truth labels derived from user behavior. First, during data collection, users choose among alternatives selected by a prior recommendation system, introducing exposure bias. Second, users lack complete knowledge of the item catalog, so their purchases may not reflect globally optimal choices. These behavioral labels can therefore differ from the preferences inferred from purchase history: users may buy weakly related items, overlook seemingly relevant ones, or express multiple competing interests within the same session. This uncertainty is reflected in the teacher LLM's CoT reasoning, which often amounts to a negotiation among ambiguous preferences, overlapping candidates, and noisy labels. The result is a dominant pathology: repetitive verification that rarely changes the final answer. The teacher

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.

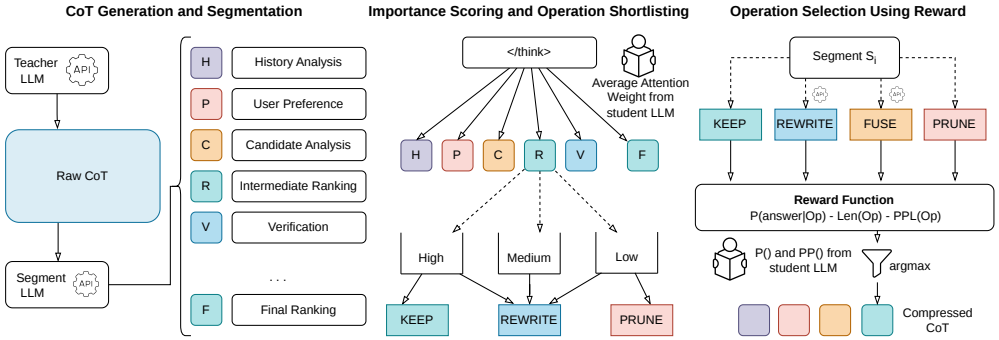


Figure 1: **SCOReD pipeline. Left:** A teacher LLM generates a raw CoT, which a segmentation LLM partitions into spans and labels by recommendation stage. **Middle:** The pre-SFT student scores span importance using attention from `</think>`, shortlisting KEEP/REWRITE, REWRITE/FUSE, or FUSE/PRUNE for high-, medium-, or low-importance spans, respectively. **Right:** SCOReD selects the operation maximizing a reward that favors the student's probability of the target answer while penalizing trace length and perplexity.

repeatedly revisits the same candidate decisions after having already formed an answer. On average, it rechecks its initial answer 9.35 times per trace, and in 79% of cases, these rechecks do not alter the final answer. Under naive supervised fine-tuning, this behavior becomes the dominant signal presented to the student: reason verbosely, verify repeatedly, and rarely revise.

To address this problem, we propose **Student-Aware CoT Optimization for Recommendation Distillation (SCOReD)**, a framework that adapts raw teacher traces to the capabilities of the target student. As illustrated in Figure 1, SCOReD treats compression as student-aware reasoning adaptation: rather than preserving the teacher's full deliberation or optimizing trace length alone, it modifies reasoning according to its utility to the student. This distinction is particularly important in recommendation, where useful evidence is often entangled with repeated verification and uncertainty. Existing CoT compression methods primarily focus on shortening reasoning traces (Xia et al., 2025; Griot et al., 2026; Lan et al., 2026), but removing an entire segment may discard useful evidence when that segment is informative yet repetitive or overly complex. SCOReD therefore includes a REWRITE operation that can preserve the function of such reasoning while expressing it in a more concise and student-compatible form.

Using traces generated by a Gemma-4-26B teacher (Gemma et al., 2026), SCOReD improves both distillation efficiency and recommendation quality. Compared to Supervised Fine-Tuning (SFT) on raw teacher traces, SCOReD improves Normalized Discounted Cumulative Gain (NDCG) from 0.773 to 0.801 for the Qwen-3-0.6B (Yang et al., 2025) student while also reducing the average trace length from 2.39K to 1.33K tokens, a 44.4% reduction. For the 1.7B student, it improves NDCG from 0.782 to 0.802 and reduces trace length from 2.53K to 1.47K tokens, a 41.9% reduction. SCOReD also improves MAP and Recall@3 over SFT on raw teacher traces for both student sizes. These results demonstrate that removing teacher redundancy can improve, rather than merely preserve, recommendation performance when the resulting reasoning is adapted to the target student.

Our main contributions are as follows:

1. We identify and quantify repetitive verification as a dominant source of redundancy in recommendation reasoning traces, showing that the teacher rechecks its initial answer an average of 9.35 times per trace and that 79% of these rechecks do not change the final answer.

2. We introduce SCOReD, a recommendation-specific, student-aware framework that combines semantic trace segmentation, student-attention-based operation shortlisting, and reward-guided deletion and rewriting to adapt teacher reasoning to the target student.

3. We show that SCOReD improves NDCG, MAP, and Recall@3 over SFT on raw teacher traces while reducing trace length by more than 40% for both 0.6B and 1.7B students.

2 Related Work

Generative recommendation. Generative recommenders replace separate retrieval and ranking stages with sequence generation. P5 (Geng et al., 2022) unifies several recommendation tasks in a text-to-text framework, M6-Rec (Cui et al., 2022) adapts a pretrained model across industrial recommendation tasks, TIGER (Rajput et al., 2023) autoregressively retrieves semantic item identifiers, and LLMRank (Hou et al., 2024) performs zero-shot listwise reranking. OneRec (Deng et al., 2025) further unifies retrieval and ranking within a single generator. More recent systems incorporate explicit reasoning: OneRec-Think (Liu et al., 2025) elicits reasoning over text-grounded item tokens, while GR2 (Liang et al., 2026) distills teacher-generated reranking rationales before reinforcement learning. Because logged targets are noisy and non-unique, these methods often condition teachers to justify predetermined items. This improves label alignment but risks post-hoc rationalization rather than faithfully capturing the teacher's decision process.

CoT distillation. CoT prompting improves reasoning by having a model generate intermediate steps before its answer (Wei et al., 2022). Distillation transfers this behavior to smaller models: Distilling Step-by-Step trains students on both final answers and teacher explanations (Hsieh et al., 2023), while other methods train them to imitate teacher-generated step-by-step solutions (Li et al., 2023; Magister et al., 2023). MiniLLM instead aligns the student with the teacher over complete generated responses rather than individual tokens (Gu et al., 2024). OneRec-Think (Liu et al., 2025) and GR2 (Liang et al., 2026) similarly train smaller recommenders on teacher reasoning traces before reinforcement learning. However, teachers may not naturally predict noisy logged targets, so recommendation methods often generate traces conditioned on the target or retain only target-compatible traces. These procedures can introduce post-hoc rationalization, selection bias, and supervision poorly matched to the student.

CoT compression. Prior works have compressed text at the different levels of granularity. At the token level, LLMLingua (Jiang et al., 2023) and LLMLingua-2 (Pan et al., 2024) prune low-importance tokens. TokenSkip (Xia et al., 2025) trains models on both original and token-compressed CoTs, allowing them to produce either verbose or concise reasoning at inference time. However, token pruning can produce fragmented text that is difficult for humans to interpret and audit. StepEntropy (Li et al., 2025), A*-Thought (Xu et al., 2025), and CRISP (Lan et al., 2026) therefore operate at the step or segment level, typically treating spans separated by a blank line as individual steps. Each step is then retained or removed according to its estimated importance. Finally, C3oT (Kang et al., 2024), Compress-Distill (Griot et al., 2026), and Yang et al. (2026) use LLMs to rewrite an entire teacher CoT into a shorter trace without first dividing it into fixed units.

3 Methodology

SCOReD consists of three stages: LLM-based chain-of-thought (CoT) segmentation, attention-based importance scoring, and per-segment edit selection based on each segment's importance score and relative advantage under a reward function.

3.1 Student-Aware CoT Optimization For Recommendation Distillation

Teacher CoT Generation and Selection We first generate $K_{rollout}$ reasoning traces using the teacher LLM, Google Gemma-4-26B, with the recommendation prompt shown below.

Candidate Ranking System Prompt

You are a recommendation assistant. You will receive a user's purchase history and a numbered list of candidate items. Your task is to rank all candidates from most to least likely to match the user's preferences, where preference is inferred from thematic, categorical, and ingredient/style similarity to past purchases.

Output format: Return only a Python-style list containing every candidate number exactly once, ordered from most to least preferred. Example for 10 candidates: [3, 6, 7, 2, 4, 9, 10, 5, 1, 8]. Do not include explanations, commentary, or any text outside the list in the final answer.

Our procedure is similar to the rejection-sampling approach of Liang et al. (2026). However, rather than requiring an exact match with the ground-truth ranking, we select the CoT associated with the highest NDCG score. Ties are broken in favor of the shorter CoT.

Recommendation-informed CoT Segmentation The recommendation prompt constrains only the format of the final output and imposes no structural requirements on the teacher model's reasoning process. This is important because Yueh-Han et al. (2026) show that reasoning LLMs successfully control the structure of their CoT in only 2.7% of cases. We therefore formulate structured reasoning as a post-hoc extraction problem and use a second LLM pass to identify the underlying reasoning stages. After generating the traces, we use Gemma-4-26B to partition each CoT into a sequence of contiguous segments, $S_1, \ldots, S_n$, using the prompt specified in Appendix H. Each segment is assigned one of six labels: Purchase History Review, User Interest Modeling, Candidate Analysis, Intermediate Ranking, Verification, or Final Ranking.

Importance Scoring via `</think>`-Attention We assign each segment an importance score based on the average attention its tokens receive from the `</think>` delimiter that terminates the CoT. Lan et al. (2026) observe that tokens generated *after* `</think>` attend almost exclusively to the delimiter itself rather than directly to the preceding reasoning trace. This behavior suggests that `</think>` functions as a learned summary representation of the CoT. We therefore use the average attention from `</think>` to the tokens in a segment as a proxy for that segment's contribution to the final answer. Based on these scores, we partition segments into three importance buckets: LOW, containing the bottom $\tau_{low}\%$ of segments by attention score; HIGH, containing the top $f\tau_{high}\%$; and MEDIUM, containing the remaining segments.

Reward-based Action Selection Each segment is processed sequentially. If a segment S_i has high semantic similarity to the preceding retained segment, as measured by BERTScore exceeding τ_{sim}, the two segments are FUSED using an LLM rewriter. Otherwise, the available actions are determined by the importance of S_i: HIGH-importance segments may be KEPT or REWRITTEN, MEDIUM-importance segments are REWRITTEN, and LOW-importance segments may be REWRITTEN or PRUNED. Let $\mathcal{A}_i$ denote the set of actions available for segment S_i. For each action $a \in \mathcal{A}_i$, let $\widetilde{S}_i^a = a(S_i)$ denote the resulting segment, and let $C_i = (prompt, S_{1:i-1})$ denote the prompt together with the previously processed segments. We evaluate each candidate edit using

$$R_i(a) = \log P(answer \mid C_i, \widetilde{S}_i^a) - \alpha\, Len(\widetilde{S}_i^a) - \beta\, PPL(\widetilde{S}_i^a \mid C_i). \tag{1}$$

The term $\log P(answer \mid C_i, \widetilde{S}_i^a)$ is the student model's log probability of the teacher's final ranking answer after conditioning on the prompt, the preceding segments, and the candidate edit. $Len(\widetilde{S}_i^a)$ is the number of tokens in the edited segment and penalizes unnecessarily long reasoning. $PPL(\widetilde{S}_i^a \mid C_i)$ is the perplexity of the edited segment under the student model, which penalizes edits that are difficult for the student to model. The coefficients α and β control the strengths of the length and perplexity penalties, respectively. We then select the highest-reward action for each segment:

$$a_i^* = \arg\max_{a \in \mathcal{A}_i} R_i(a). \tag{2}$$

3.2 POST-SFT OPTIMIZATION USING REINFORCEMENT LEARNING AND ON-POLICY DISTILLATION

We test two post-SFT methods initialized from the student trained on SCOReD-optimized CoTs: Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2026), which uses a sequence-level ranking reward, and On-Policy Self-Distillation (OPSD) (Zhao et al., 2026), to see if performance can be further improved over our strong SFT baseline.

Reinforcement Learning with DAPO. Let π_{θ_0} denote the SCOReD SFT policy. For each prompt x, we sample a group of responses $y_1, \ldots, y_G \sim \pi_\theta(\cdot \mid x)$ and compute a verifiable reward from

each parsed ranking:

$$r(y_g, y^*) = \begin{cases} \mathrm{NDCG}(y_g, y^*), & \text{if } y_g \text{ is a valid ranking,} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where y^* denotes the set of logged relevant items. DAPO constructs within-prompt relative advantages from these rewards and increases the likelihood of higher-reward trajectories. We initialize both the policy and reference model from π_{θ_0} and optimize the policy using LoRA.

On-Policy Self-Distillation with Privileged Information. Standard On-Policy Distillation (OPD) compares teacher and student token distributions along student-generated trajectories and therefore requires a shared vocabulary (Lu & Lab, 2025). Gemma-4-26B is stronger but vocabulary-incompatible with the Qwen student, whereas Qwen-3.6-35B-A3B shares its vocabulary but is not consistently stronger on reranking. We therefore use OPSD (Zhao et al., 2026), in which the student model also serves as the teacher under a privileged conditioning context.

Because the logged ground truth identifies relevant candidates rather than a complete ordering, we construct a privileged reference ranking $\widetilde{y}^*$ by stably moving the relevant items to the beginning of the teacher ranking while preserving the relative order of the remaining candidates. The student conditions only on the original prompt x, whereas the privileged teacher additionally conditions on $\widetilde{y}^*$. For each instance, the student samples an on-policy response $y \sim \pi_\theta^S(\cdot \mid x)$. Both policies then evaluate each student-generated prefix $y_{<t}$, and we minimize

$$\mathcal{L}_{\mathrm{OPSD}} = \mathbb{E}_x \mathbb{E}_{y \sim \pi_\theta^S(\cdot|x)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} \mathcal{D}\left(\mathrm{sg}\left[\pi^T(\cdot \mid x, \widetilde{y}^*, y_{<t}) \right], \pi_\theta^S(\cdot \mid x, y_{<t}) \right) \right], \tag{4}$$

where $\mathcal{D}$ denotes the token-distribution divergence and sg stops gradients through the privileged teacher distribution. OPSD thus provides dense guidance about the desired continuation on prefixes encountered under the student's own policy.

4 Experimental Setup

Dataset Construction We construct ranking datasets from four product categories from Amazon Reviews: Beauty, Books, Movies & TV, and Musical Instruments (Hou et al., 2026). For each category, we sample 22,363 user purchase histories with sequence lengths at least 5 and at most 50. Given an ordered sequence of N items, we use its final three items as positive candidates and the preceding $N-3$ items as purchase history. For each example, we uniformly sample seven negatives without replacement from the product catalog, excluding every item in the user's sequence. We shuffle these with the three positives to obtain $K_{candidate} = 10$. Items are represented by their title and full description. We use a user-disjoint 60:20:20 split for every category, yielding 13,417 training, 4,472 validation, and 4,474 test examples per dataset.

Model Selection We use Google Gemma-4-26B-A4B-it (Gemma et al., 2026) as the teacher LLM and Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B as student models (Yang et al., 2025). For each student, we use the same model to estimate segment importance. We use Gemma-4-26B for CoT segmentation and for the Rewrite and Fuse operations in our compression algorithm.

SCOReD Hyperparameter Selection We set the upper importance threshold $f_{\mathrm{high}} = 0.05$, the semantic-similarity threshold $f_{\mathrm{sim}} = 0.8$, the length-penalty weight $\alpha = 0.005$, and the perplexity-penalty weight $\beta = 0.1$. We sweep $f_{\mathrm{low}} \in \{0.0, 0.30, 0.50\}$ while holding all other hyperparameters fixed. This threshold has the largest effect on compression length because it directly controls which low-importance segments are eligible for pruning rather than being retained or rewritten.

Model Training For supervised fine-tuning, we perform full-parameter training using the `verl` library with a batch size of 64, a maximum sequence length of 8,192 tokens, and up to 10 epochs with early stopping based on validation set NDCG. We sweep the learning rate from 10^{-6} to 10^{-3} and select the best checkpoint using the validation set; a learning rate of 3×10^{-5} performs best. For DAPO (Yu et al., 2026) and OPSD (Zhao et al., 2026), we use LoRA ($r = 32$, $\alpha = 32$). For

DAPO, we use a batch of 16 prompts with eight rollouts per prompt and train for 6 epochs with a learning rate of 10^{-5}. For OPSD, we use a batch size of 32 and a learning rate of 5×10^{-6}, and train for 3,200 samples (approximately 0.28 epochs) before early stopping.

Evaluation Metrics Each example contains N candidates, of which K are relevant and $N - K$ are irrelevant. The ground truth requires only that relevant items rank ahead of irrelevant ones, with no preferred ordering within either group. Let $y_i \in \{0, 1\}$ denote the relevance of the item at rank i. We compute normalized discounted cumulative gain over the full ranking as

$$\mathrm{NDCG} = \frac{\sum_{i=1}^{N} y_i / \log_2(i+1)}{\sum_{i=1}^{K} 1/\log_2(i+1)},$$

which gives greater weight to relevant items near the top. Defining precision at rank i as $P(i) = \frac{1}{i}\sum_{j=1}^{i} y_j$, average precision is $\mathrm{AP} = \frac{1}{K}\sum_{i=1}^{N} P(i)y_i$, and MAP is the mean AP across examples. Recall@K is the fraction of the K relevant items appearing in the first K positions.

To measure the faithfulness of the generated reasoning to the model's final ranking, we additionally report *Leakage-Adjusted Simulatability* (LAS; Hase et al., 2020). LAS measures whether the reasoning is sufficient for an independent simulator LLM to recover the ranking produced by the trained student. Let f_{sim} be the simulator and f_{SFT} the trained student. Given prompt x, the student generates reasoning e and ranking y, i.e., $f_{\mathrm{SFT}}(x) = (e, y)$. Let $\Delta = \mathbf{1}[f_{\mathrm{sim}}(x, e) = y] - \mathbf{1}[f_{\mathrm{sim}}(x) = y]$ indicate whether the simulator requires the reasoning trace e to generate y. LAS is defined as

$$\mathrm{LAS} = \frac{1}{2}\left(\mathbb{E}\left[\Delta \mid f_{\mathrm{sim}}(e) = y\right] + \mathbb{E}\left[\Delta \mid f_{\mathrm{sim}}(e) \neq y\right]\right). \tag{5}$$

LAS therefore partitions examples into two subsets depending on whether the reasoning trace e alone causes the simulator f_{sim} to recover y. Hase et al. (2020) interpret the former case as the explanation "leaking" the answer y. Importantly, LAS measures consistency rather than correctness: even if the ranking y produced by the SFT model is incorrect, the example is still considered simulatable if the simulator recovers that same ranking. In our setting, we replace the trained T5 simulator used by Hase et al. (2020) with Qwen3-8B.

5 Results

Table 1: Comparison of CoT compression algorithms on the Amazon Review 2023 Dataset - Beauty domain. Error rate denotes the percentage of cases where the final ranking is not a permutation of the K candidates. Bold and underline denote the best and second-best results, respectively.

Method	Error Rate (%) ↓	NDCG ↑	MAP ↑	Recall@3 ↑	LAS ↑	Length (Token)
Qwen3-0.6B						
Baseline SFT	3.51	0.773	0.652	0.531	**0.660**	2.39K
Compress-Distill	8.74	0.734	0.620	0.506	0.654	1.24K
A*-Thought	<u>0.76</u>	<u>0.799</u>	<u>0.675</u>	**0.551**	0.592	0.85K
SCOReD	**0.40**	**0.801**	**0.676**	<u>0.549</u>	<u>0.656</u>	1.33K
Qwen3-1.7B						
Baseline SFT	2.82	0.782	0.660	0.540	0.634	2.53K
Compress-Distill	9.03	0.730	0.616	0.503	**0.690**	1.13K
A*-Thought	<u>0.72</u>	**0.802**	**0.680**	**0.556**	0.607	0.86K
SCOReD	**0.36**	**0.802**	<u>0.678</u>	<u>0.552</u>	<u>0.640</u>	1.47K
Reference						
Gemma-4-26B-A4B	0.00	0.803	0.678	0.553	1.000	2.92K
Qwen-3.6-35B-A3B	2.10	0.788	0.667	0.544	1.000	3.17K

Student-Aware Compression Improves Distillation Table 1 shows that compression is most effective when it accounts for the student model. SCOReD improves NDCG from 0.773 to 0.801 for Qwen3-0.6B and from 0.782 to 0.802 for Qwen3-1.7B relative to Baseline SFT, while also improving MAP and Recall@3. A*-Thought, which also incorporates signals from the pre-SFT student during compression, achieves comparable ranking performance.

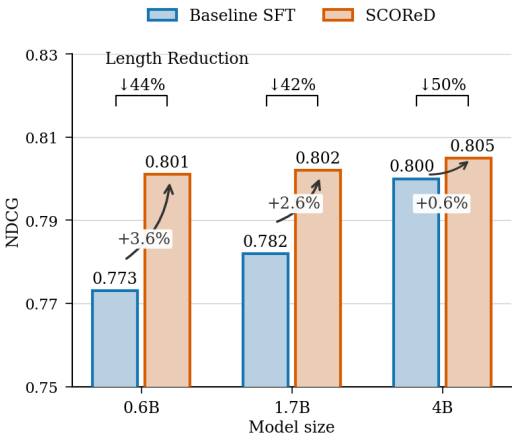
The effect is even more pronounced in output validity. SCOReD reduces the error rate from 3.51% to 0.40% for Qwen3-0.6B and from 2.82% to 0.36% for Qwen3-1.7B. In contrast, Compress-Distill, which shortens teacher traces with an LLM summarizer without explicitly accounting for the student model, increases the error rate to 8.74% and 9.03%, respectively, and degrades ranking performance. This shows that shorter traces alone are insufficient: compressing teacher reasoning without adapting to the student's distribution can make distillation substantially less effective. We analyze error cases in Appendix A and find that the dominant error type is incomplete ranking, in which the student omits candidate items.

A*-Thought is purely subtractive, removing low-importance segments while leaving the retained text unchanged, whereas SCOReD uses an LLM rewriter to reconstruct a coherent trace from the selected information. The qualitative examples in Appendix I show that the 0.6B A*-Thought student can produce locally inconsistent or factually unsupported reasoning despite generating a valid final ranking, while SCOReD generally preserves a clearer connection between its reasoning and recommendation. This difference is also reflected in LAS: SCOReD achieves higher simulatability than A*-Thought for both Qwen3-0.6B (0.656 vs. 0.592) and Qwen3-1.7B (0.640 vs. 0.607), indicating that its reasoning is more predictive of the ranking ultimately produced by the student. We note that more than 90% of student-generated CoTs contain the final ranking and fall into the leaking subset $f_{sim}(e) = y$ of Eqn. 5. Simulatability is also consistently higher within this subset. We report a detailed breakdown of LAS in Appendix G.

Table 2: Comparison of Baseline SFT and SCOReD across Amazon Review domains.

Dataset	Method	NDCG ↑	MAP ↑	R@3 ↑
Qwen3-0.6B				
Beauty	Baseline	0.773	0.652	0.531
	SCOReD	**0.801**	**0.676**	**0.549**
Music Instrument	Baseline	0.797	**0.680**	**0.558**
	SCOReD	**0.799**	**0.680**	0.555
Movies & TV	Baseline	0.634	0.474	0.316
	SCOReD	**0.654**	**0.493**	**0.337**
Books	Baseline	0.843	0.745	0.640
	SCOReD	**0.854**	**0.756**	**0.648**
Qwen3-1.7B				
Beauty	Baseline	0.782	0.660	0.540
	SCOReD	**0.802**	**0.678**	**0.552**
Music Instrument	Baseline	0.780	0.665	0.546
	SCOReD	**0.796**	**0.680**	**0.561**
Movies & TV	Baseline	0.646	0.485	0.328
	SCOReD	**0.661**	**0.498**	**0.340**
Books	Baseline	0.847	0.752	0.646
	SCOReD	**0.854**	**0.758**	**0.652**

Figure 2: Comparison of Baseline SFT and SCOReD across Qwen-3 model sizes.



SCOReD Generalizes Across Domains and Student Sizes Table 2 shows that SCOReD improves ranking performance over baseline SFT across Amazon Review domains and student sizes. The gains are strongest on Beauty, where NDCG improves from 0.773 to 0.801 for Qwen3-0.6B and from 0.782 to 0.802 for Qwen3-1.7B. Improvements also extend to Movies & TV and Books, while the 0.6B student shows the smallest improvement on the Music Instruments split.

Figure 2 examines how these gains vary with student capacity. SCOReD reduces CoT length by 42–50% across all three model sizes while improving NDCG by 3.6%, 2.6%, and 0.6% for Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B, respectively. The diminishing ranking gain with model size suggests that smaller students benefit more from student-specific adaptation of the teacher traces, whereas the more capable Qwen3-4B can learn effectively from the original Gemma-4-26B traces. Detailed results are provided in Appendix C.

6 Analysis and Ablations

6.1 Structure of Teacher Reasoning Traces

Across the 11,350 teacher CoTs in our training set, the segmentation procedure identifies 289,541 segments. We denote the six stages as Purchase History Review (**H**), User Interest Modeling (**P**), Candidate Analysis (**C**), Intermediate Ranking (**R**), Verification (**V**), and Final Ranking (**F**). Table 3 reports their frequencies and average lengths. Verification and intermediate ranking dominate the traces, together accounting for 63.3% of all segments. Final-ranking segments are substantially shorter, consistent with their role as concise conclusions.

Table 3: Reasoning Stages.

Stage	Share	Avg. chars
V (Verification)	33.9%	418
R (Intermediate Rank.)	29.4%	367
C (Candidate Analysis)	15.3%	459
F (Final Ranking)	7.3%	120
P (Interest Modeling)	7.1%	169
H (History Review)	7.0%	268

Table 4: Adjacent Stage-Transition Counts.

From \ To	H	P	C	R	V	F
H	0	12,130	4,204	1,343	2,620	104
P	126	0	13,943	4,343	1,736	321
C	912	2,475	0	23,749	15,436	1,558
R	2,390	1,345	7,862	0	70,414	2,337
V	4,768	4,415	17,433	52,407	0	16,687
F	880	95	849	3,233	8,076	0

Teacher Traces Follow a Two-Phase Reasoning Pattern Table 4 reveals two broad phases in the teacher's reasoning. The first is an initial ranking-construction phase, commonly following **H** $\rightarrow$ **P** $\rightarrow$ **C** $\rightarrow$ **R**. The teacher summarizes the purchase history, infers the user's preferences, evaluates the candidates, and proposes an initial ranking. The second phase consists of repeated refinement, primarily through **R** $\rightarrow$ **V** $\rightarrow$ **R** and **R** $\rightarrow$ **V** $\rightarrow$ **C** $\rightarrow$ **R**. A raw teacher CoT contains an average of 8.65 ± 4.17 verification segments, indicating that refinement occupies a substantial portion of the trace.

Most Refinement Does Not Change the Ranking Despite the prevalence of verification, only 47.15% of traces end with a ranking that differs from the initial ranking, and 92.89% of individual refinement loops leave the current ranking unchanged. Distilling the raw traces may therefore teach the student to repeatedly confirm an existing ranking rather than perform useful correction, while substantially increasing reasoning length.

Refinement Frequency Is Difficult to Predict from Input Features We next examine whether the amount of refinement can be anticipated from simple input properties. The number of verification segments is only weakly correlated with prompt length, purchase-history length, and textual overlap between purchased and candidate items, with Pearson correlations of -0.08, -0.05, and -0.11, respectively. These features therefore provide little information about how extensively the teacher will refine its ranking.

6.2 Effect of Compression Strength

SCOReD uses three gating hyperparameters: τ_{low} and τ_{high} define the low- and high-importance segments, while τ_{sim} determines when adjacent segments are fused. Because τ_{low} determines the proportion of segments eligible for pruning, it most directly controls compression. We therefore fix $\tau_{\text{high}} = 0.05$ and $\tau_{\text{sim}} = 0.8$ and sweep $\tau_{\text{low}} \in \{0.00, 0.30, 0.50\}$.

Figure 3 reports the actions selected by SCOReD at different compression strengths. Increasing τ_{low} shifts the action distribution toward PRUNE. At $\tau_{\text{low}} = 0.00$, neither student selects pruning, whereas at 0.50, PRUNE accounts for 42.57% and 38.66% of the edits for Qwen3-0.6B and Qwen3-1.7B, respectively. The proportion of KEEP actions remains nearly constant because τ_{high} is fixed.

Stronger Compression Improves Distilled Students We next train each student on the traces produced under each setting. As shown in Table 5, all three recommendation metrics improve monotonically as τ_{low} increases. The strongest compression setting, $\tau_{\text{low}} = 0.50$, also reduces the average generated length by 26.9% for Qwen3-0.6B and 22.6% for Qwen3-1.7B relative to $\tau_{\text{low}} = 0.00$. Parse failures decrease from 1.30% to 0.40% and from 0.87% to 0.36%, respectively. CoT–answer

Figure 3: SCOReD edit action distribution.

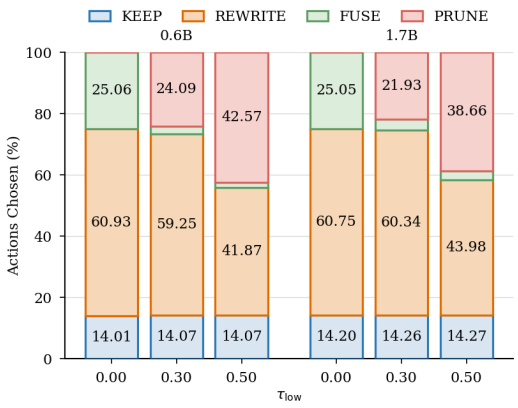


Table 5: Effect of τ_{low} on distilled students.

Trained Student	τ_{low}	Avg. tokens (K)	Error rate (%)	NDCG
0.6B	0.00	1.82K	1.30	0.793
	0.30	1.74K	0.72	0.799
	0.50	**1.33K**	**0.40**	**0.801**
1.7B	0.00	1.90K	0.87	0.797
	0.30	1.92K	0.58	**0.802**
	0.50	**1.46K**	**0.36**	**0.802**

agreement remains high across settings, indicating that the shorter generations remain consistent with their final answers. We therefore use $\tau_{\text{low}} = 0.50$ in subsequent experiments.

6.3 Post-SFT Optimization

Table 1 shows that the SCOReD 0.6B student approaches the Gemma-4 teacher's ranking performance (0.801 vs. 0.803 NDCG). We additionally test whether standard post-SFT optimization further improves the already strong SCOReD initialization; in our setting, neither DAPO (Yu et al., 2026) nor On-Policy Self Distillation (OPSD) (Zhao et al., 2026) provides consistent additional gains.

On-Policy Self-Distillation On-Policy Distillation (OPD) (Lu & Lab, 2025) requires a teacher that is both stronger than the student and vocabulary-compatible. We find that larger Qwen-3 variants do not consistently outperform the SCOReD 0.6B student on this task. We therefore evaluate On-Policy Self-Distillation with privileged information (OPSD), providing the ground-truth selection in the prompt. Since this supervision specifies selected items rather than a full ranking, we reorder the teacher's ranking to place ground-truth items first.

We find that OPSD does not improve over the SFT initialization. On the complete test set, the initial SFT checkpoint obtains the best NDCG, MAP, and Recall@3. OPSD checkpoints exhibit small, non-monotonic fluctuations and do not show a general improvement in ranking quality. Detailed OPSD results are reported in Appendix F.

Reinforcement Learning with DAPO Direct reinforcement learning with DAPO similarly fails to improve the SFT model. NDCG remains within a narrow range over six epochs, with no noticeable trend. We attribute this lack of improvement to challenges specific to the recommendation domain where the user-selected ground truth is subject to exposure bias and may not be the most logical selection from a given candidate set. The complete DAPO trajectory is shown in Appendix E.

7 Conclusion

We introduce SCOReD, a recommendation-specific framework for adapting teacher reasoning traces to the student distribution before supervised fine-tuning. We find that SCOReD consistently improves ranking performance across domains and student sizes, while LLM-based compression that shortens traces without accounting for the student can degrade both ranking quality and output validity. We find that the gains are largest for smaller students, where the capability gap between teacher and student is greatest, and diminish as student capacity increases. Our results suggest that effective reasoning distillation depends not only on the quality of the teacher trace, but also on how well that trace is matched to what the student can learn.

8 AI USE STATEMENT

In this work, we used generative AI tools to assist with literature review, code writing, and rephrasing portions of the manuscript. We did not use generative AI tools for research ideation or citation generation. AI-assisted code was reviewed and verified by one author. All AI-assisted text and technical content were reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

9 REPRODUCIBILITY STATEMENT

We release our code, datasets, and trained student outputs at `https://anonymous.4open.science/status/SCOReD-anonymous-submission-0B67`. The codebase includes scripts for teacher trace generation, SCOReD segmentation and compression, supervised fine-tuning, and evaluation. We also provide our implementations of Leakage-adjusted Simulatability (LAS) (Hase et al., 2020) and Compress-Distill (Griot et al., 2026). In addition, we release the traces generated by the 0.6B students corresponding to the results in Table 1.

REFERENCES

Jincheng Cao, Fanzhi Zeng, Leqi Liu, and Aryan Mokhtari. Curriculum learning-guided progressive distillation in large language models, 2026. URL `https://arxiv.org/abs/2605.11260`.

Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*, 2022.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

Team Gemma, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

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.

Maxime Griot, Paul Steven Scotti, and Tanishq Mathew Abraham. Compress-distill: Reasoning trace compression for efficient knowledge distillation. *arXiv preprint arXiv:2606.05988*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4351–4367, 2020.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, 2024.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking llms as semantic encoders, 2026. URL `https://arxiv.org/abs/2403.03952`.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *arXiv preprint arXiv:2412.11664*, 2024.

Yangsong Lan, Hongliang Dai, and Piji Li. Crisp: Compressing redundancy in chain-of-thought via intrinsic saliency pruning. *arXiv preprint arXiv:2604.17297*, 2026.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Ying Li, Shradha Sehgal, Arjun Rao, Rein Houthooft, Yaochen Zhu, and Ashish Rastogi. Genrec: An llm-backed recommendation ranker at netflix, 2026. URL https://arxiv.org/abs/2608.10257.

Zeju Li et al. Compressing chain-of-thought in llms via step entropy. *arXiv preprint arXiv:2508.03346*, 2025.

Mingfu Liang, Yufei Li, Jay Xu, Kavosh Asadi, Xi Liu, Shuo Gu, Kaushik Rangadurai, Frank Shyu, Shuaiwen Wang, Song Yang, Zhijing Li, Jiang Liu, Mengying Sun, Fei Tian, Xiaohan Wei, Chonglin Sun, Jacob Tao, Shike Mei, Hamed Firooz, Wenlin Chen, and Luke Simon. Generative reasoning re-ranker. *arXiv preprint arXiv:2602.07774*, 2026.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, Yifei Hu, Qigen Hu, Xinchen Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, Zexuan Cheng, Lu Ren, Huanjie Wang, Yi Su, Ruiming Tang, Kun Gai, and Guorui Zhou. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639*, 2025.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1773–1781, 2023.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Ruhle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 963–981, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.57.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukas Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Justin Samost, Maciej Kula, Maryam Fazel-Zarandi, Lixin Liu, and Ed H. Chi. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, 2023.

Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. Can small language models be good reasoners for sequential recommendation?, 2024. URL https://arxiv.org/abs/2403.04260.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Xiaoang Xu, Shuo Wang, Xu Han, Zhenghao Liu, Huijia Wu, Peipei Li, Zhiyuan Liu, Maosong Sun, and Zhaofeng He. A*-thought: Efficient reasoning via bidirectional compression for low-resource settings, 2025. URL https://arxiv.org/abs/2505.24550.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

H. Yang et al. Reasoning compression with mixed-policy distillation. *arXiv preprint arXiv:2605.08776*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Chen Yueh-Han, Robert McCarthy, Bruce W. Lee, He He, Ian Kivlichan, Bowen Baker, Micah Carroll, and Tomek Korbak. Reasoning models struggle to control their chains of thought, 2026. URL https://arxiv.org/abs/2603.05706.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

APPENDIX

A Error Analysis and Categorization

Table 6: Error analysis and categorization on the 4,474 test set rankings produced by 0.6B students.

Failure type	**Baseline SFT** (126 failures)	**Compress-Distill** (391 failures)	**A*-Thought** (34 failures)	**SCOReD** (18 failures)
Incomplete list ($< K$ indices)	112 (88.9%)	289 (73.9%)	21 (61.8%)	12 (66.7%)
Duplicate indices	1 (0.8%)	72 (18.4%)	9 (26.5%)	2 (11.1%)
No final list	6 (4.8%)	8 (2.0%)	1 (2.9%)	1 (5.6%)
Product names / invalid JSON*	0 (0.0%)	15 (3.8%)	0 (0.0%)	0 (0.0%)
Invalid index values	4 (3.2%)	5 (1.3%)	1 (2.9%)	1 (5.6%)
List too long (> 10 indices)	3 (2.4%)	2 (0.5%)	2 (5.9%)	2 (11.1%)
Total parse failures	126 (2.8%)	391 (8.7%)	34 (0.76%)	18 (0.40%)

B Effect of Compression Strength on SCOReD Student

Table 7: Results after training students on SCOReD traces generated with different values of τ_{low}. Best results before rounding are bolded.

Trained Student	τ_{low}	Avg. tokens (K)	Error Rate (%)	Recommendation Performance		
				NDCG	MAP	Recall@3
0.6B	0.00	1.823	1.30	0.793	0.670	0.548
	0.30	1.739	0.72	0.799	0.674	0.549
	0.50	**1.333**	**0.40**	**0.801**	**0.676**	**0.549**
1.7B	0.00	1.896	0.87	0.797	0.673	0.550
	0.30	1.915	0.58	0.802	0.678	0.551
	0.50	**1.467**	**0.36**	**0.802**	**0.678**	**0.552**

C Effect of Student LLM Size

Table 8: Comparison of Baseline SFT and SCOReD across Model Sizes

Method	Length (Token) ↓	Error Rat. (%) ↓	NDCG ↑	MAP ↑	Recall@3 ↑
Qwen3-0.6B					
Baseline SFT	2.39K	3.51	0.773	0.652	0.531
SCOReD	**1.33K**	**0.40**	**0.801**	**0.676**	**0.549**
Qwen3-1.7B					
Baseline SFT	2.53K	2.82	0.782	0.660	0.540
SCOReD	**1.47K**	**0.36**	**0.802**	**0.678**	**0.552**
Qwen3-4B					
Baseline SFT	2.69K	0.65	0.800	0.676	0.552
SCOReD	**1.35K**	**0.22**	**0.805**	**0.681**	**0.555**

D RANKING AND REASONING DYNAMICS DURING TRAINING

Figure 4 shows all-sample NDCG and error rate throughout training. SCOReD maintains higher all-sample NDCG while producing substantially fewer errors. Although the error rate of the Compress-Distill baseline decreases during training, it remains considerably higher than those of the other methods. These results show that minimizing trace length alone does not produce an effective distillation target. Aggressive one-shot compression can remove useful reasoning or weaken adherence to the required output format.

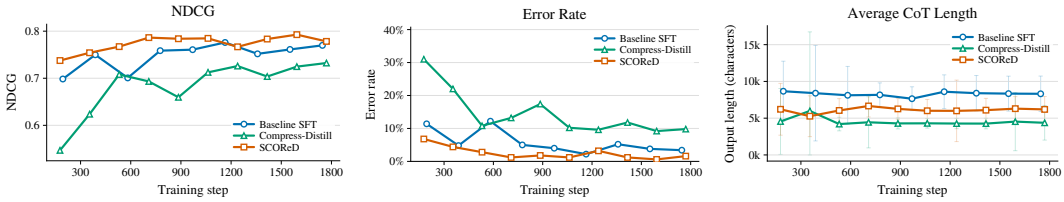


Figure 4: Validation set performance during training.

E DAPO RESULTS

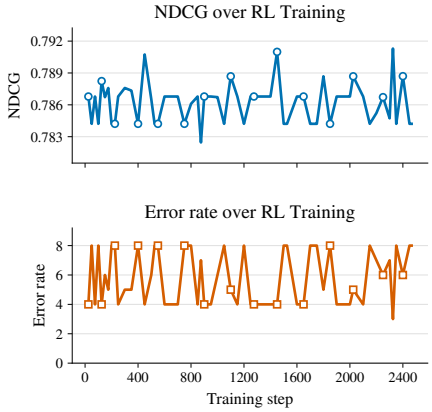


Figure 5: DAPO training over six epochs using NDCG maximization as the reward.

Figure 5 shows DAPO training for six epochs using NDCG as the reward, initialized from the SCOReD ($\tau_{low} = 0.00$) 0.6B student. NDCG oscillates within a narrow range and does not exhibit a sustained upward trend. Parse failures similarly fluctuate throughout optimization without a consistent reduction.

These results indicate that direct reinforcement learning does not improve upon the SCOReD SFT initialization. The limited movement in NDCG is consistent with the small remaining performance gap between the student and teacher: most sampled rankings are already similar in quality, reducing the frequency and magnitude of informative advantages.

F ON-POLICY SELF DISTILLATION RESULTS

Table 9 reports OPSD performance on the test set. The SCOReD ($\tau_{low} = 0.00$) 0.6B student initialization at checkpoint 0 achieves the highest NDCG, MRR, MAP, Recall@3, and Recall@5, and the lowest parse-failure rate. Although later checkpoints achieve small improvements on isolated Hit@k metrics, these improvements are inconsistent across metrics and checkpoints. By checkpoint 100, NDCG decreases from 0.7923 to 0.7878, while the parse-failure rate increases from 1.39% to 1.81%.

Table 9: OPSD results on Amazon Reviews 2023 – Beauty test set. Checkpoint 0 is the SCOReD SFT initialization.

Metric / Checkpoint	**0 (SCOReD)**	**25**	**50**	**75**	**100**
Training samples	0	800	1600	2400	3200
NDCG	**0.7923**	0.7908	0.7898	0.7907	0.7878
MRR	**0.7722**	0.7721	0.7702	0.7719	0.7689
MAP	**0.6691**	0.6667	0.6660	0.6667	0.6639
Hit@1	0.6444	**0.6451**	0.6417	0.6442	0.6419
Hit@3	0.8847	0.8831	**0.8858**	0.8809	0.8804
Hit@5	0.9555	**0.9569**	0.9564	**0.9569**	0.9535
Recall@1	0.2148	**0.2150**	0.2139	0.2147	0.2140
Recall@3	**0.5470**	0.5419	0.5432	0.5450	0.5417
Recall@5	**0.7269**	0.7252	0.7245	0.7240	0.7202
Parse failure rate (%)	**1.3858**	1.4528	1.5646	1.4975	1.8105

G LEAKAGE-ADJUSTED SIMULATABILITY

Leakage-Adjusted Simulatability (LAS; Hase et al., 2020) measures whether the reasoning is sufficient for an independent simulator LLM to recover the ranking produced by the trained student, while accounting for "leakage": cases where the reasoning directly reveals the ranking. Let f_{sim} be the simulator and f_{SFT} the trained student. Given prompt x, the student generates reasoning e and ranking y, i.e., $f_{\text{SFT}}(x) = (e, y)$. Let $\Delta = \mathbf{1}[f_{\text{sim}}(x, e) = y] - \mathbf{1}[f_{\text{sim}}(x) = y]$ indicate whether access to the reasoning trace e is necessary for the simulator to recover y. LAS is defined as

$$\text{LAS} = \frac{1}{2}\left(\mathbb{E}\left[\Delta \mid f_{\text{sim}}(e) = y\right] + \mathbb{E}\left[\Delta \mid f_{\text{sim}}(e) \neq y\right]\right).$$

LAS therefore partitions examples into two subsets depending on whether the reasoning trace e alone causes the simulator f_{sim} to recover y. Following Hase et al. (2020), we refer to the former as "leaking" examples. Importantly, LAS measures consistency rather than correctness: even if the ranking y produced by the SFT model is incorrect, the example is still considered simulatable if the simulator recovers that same ranking. In our setting, we replace the trained T5 simulator used by Hase et al. (2020) with Qwen3-8B.

Table 10 shows that the large majority of reasoning traces fall into the leaking subset, with $f_{\text{sim}}(e) = y$ for over 90% of parsed examples across all methods. Simulatability is correspondingly high within this subset, whereas agreement is substantially lower when the reasoning alone does not reveal the final ranking. LAS gives these two subsets equal weight, preventing the much larger leaking subset from dominating the reported score.

Table 10: Detailed results for Leakage-adjusted Simulatability.

Model	Method	|Test|	|Parsed|	$\lvert f_{sim}(e) = y \rvert$	$\lvert f_{sim}(e) \neq y \rvert$	$\mathbb{E}\left[\Delta \mid f_{\text{sim}}(e) \neq y\right]$	$\mathbb{E}\left[\Delta \mid f_{\text{sim}}(e) \neq y\right]$	LAS
0.6B	Baseline SFT	4,474	4,363	4,318	45	0.9866	0.3333	0.6600
	Compress-Distill	4,474	4,354	3,984	370	0.9618	0.3459	0.6539
	A* Thought	4,474	4,366	4,118	248	0.9330	0.2500	0.5915
	SCOReD	4,474	4,364	4,160	204	0.9587	0.3529	0.6558
1.7B	Baseline SFT	4,474	4,369	4,323	46	0.9849	0.2826	0.6338
	Compress-Distill	4,474	4,363	3,974	389	0.9731	0.4062	0.6896
	A* Thought	4,474	4,363	4,083	280	0.9287	0.2857	0.6072
	SCOReD	4,474	4,357	4,126	231	0.9586	0.3203	0.6395

H SCORED PROMPTS

SCOReD Recommendation-informed Segmentation Prompt

System:
You are an expert data annotator. Your task is to segment a reranking reasoning chain into contiguous blocks belonging to one of the following stages:

H: Purchase History Analysis (summarizing or categorizing past purchases)
P: User Preference / Theme Modeling (inferring categories, brands, ingredients, benefits, or use cases from history)
C: Candidate Analysis (describing candidate items and their relevance before constructing the ranking)
R: Ranking Construction / Tiering (building tiers, assigning rank positions, or forming a preliminary ordered list)
V: Verification / Pairwise Comparison / Self-Correction (checking close calls, comparing candidates, expressing doubt, or revising the order)
F: Final Ranking (the final conclusion, final order, or final ranked list inside the reasoning chain)
Common reranking patterns include candidate summaries, high/mid/low tiers, pairwise comparisons like 'candidate 5 vs candidate 6', repeated 'Wait' checks, and one or more final ordered lists.

CRITICAL RULES:
1. You will be given the reasoning chain with line numbers.
2. Output your response STRICTLY as a JSON array of objects. Each object must have 'stage' (string, one of H, P, C, R, V, F), 'start_line' (int), and 'end_line' (int).
3. The segments must cover all lines from first_line to final_line continuously, with no gaps or overlaps.
4. Segment at HIGH GRANULARITY. Break down long blocks when the reasoning shifts between history, preferences, candidate analysis, tiering, verification, and final ranking.
5. Do not label every ordered list as F. Use F only for final conclusions or final ranking statements; use R for intermediate tiering and draft rankings.
6. Output ONLY the valid JSON array. Do not include markdown formatting like ```json.

SCOReD REWRITE Prompt

System:
You are an expert at condensing reasoning steps. Your task is to rewrite the given reasoning step to be more concise while preserving all essential information and logical
flow.

Rules:
1. Keep all key facts, numbers, identifiers (like "Candidate N"), product names, and logical connections.
2. Remove redundant phrases and verbose expressions.
3. Maintain the logical correctness.
4. Output ONLY the condensed step, no explanations.

User:
Rewrite this reasoning step more concisely: {step}

SCOReD FUSE Prompt

System:
You are an expert at merging reasoning steps. Your task is to combine two consecutive reasoning steps into a single, coherent step while preserving all essential
information.

Rules:
1. Preserve all key facts, numbers, identifiers (like "Candidate N"), product names, and calculations.
2. Maintain logical flow and correctness.
3. Remove redundant information that appears in both steps.
4. The merged step should be shorter than the sum of both steps.
5. Output ONLY the merged step, no explanations.

User:
Merge these two reasoning steps into a single coherent step:

SCOReD FUSE Prompt *(continued)*

Step 1: {step1}
Step 2: {step2}

Compress-Distill/LLM Summarizer System Prompt

You are a distillation agent. Given the teacher's original reasoning chain, you will rewrite it to be more pedagogical for a student model. You will:

1) Consolidate reasoning about candidates.
2) Remove redundant doubts that repeat a previous step's reasoning.
3) Keep refinement/doubts that are logical. If the intermediate verdict differs from the final answer, keep both to teach the student model backtracking.
4) Do not shorten Purchase History Analysis, User Preference Modeling, initial Candidate Analysis, or Final Ranking.
5) You may enrich Purchase History Analysis, User Preference Modeling, and initial Candidate Analysis by adding more details from later steps.
6) Do not change the name of the steps.

I Qualitative Comparison of 0.6B Students

This section showcases the reasoning traces and recommendations generated by different 0.6B students for a test case. The reasoning traces have been manually annotated to highlight factual errors and hallucinations.

Table 11: Ranking performance for ground-truth relevant items $\{3, 7, 8\}$. Since there is a single example, MAP is equal to AP.

Method	Ranking	NDCG	AP	Recall@3
Baseline SFT	[3, 9, 2, 7, 6, 1, 5, 8, 10, 4]	0.8194	0.6250	0.3333
Compress-Distill	[3, 9, 6, 7, 2, 5, 10, 8, 1, 4]	0.8194	0.6250	0.3333
A*-Thought	[3, 6, 1, 7, 9, 2, 5, 10, 8, 4]	0.8127	0.6111	0.3333
SCOReD	[3, 6, 7, 9, 2, 5, 8, 10, 1, 4]	0.8603	0.6984	0.6667

User Prompt

User's purchase history:
Purchase 1
Title: Laura Mercier Loose Setting Powder - Ivory - 29g/1oz
Description: A sheer silky powder that looks & feels natural Comprises the finest French cashmere talc Light reflecting ingredient ...
———

Purchase 2
Title: Trish McEvoy Lash Curling Mascara Jet Black
Description: This revolutionary tubular curling mascara forms water-resistant tubes 360 around lashes, dries instantly and will ...
———

Purchase 3
Title: Maybelline New York Shine Free Oil Control Pressed Powder, Ivory - 1 ea
Description: Oil blotting ingredients in this pressed powder keep your face fresh for hours. 100% oil free for a no shine, matte finish. Won't clog pores or cause blemishes. suitable for acne prone skin...
———

Purchase 4
Title: Acqua Di Gio By Giorgio Armani For Men. Eau De Toilette Spray 1.7 Oz.
Description: Inspired by beauty of Pantellerie, where he spent his vacation, Armani created the aroma of Aqua di Gio for men and women. ...
———

User Prompt *(continued)*

Purchase 5
Title: Infiniti Pro by Conair Instant Heat 20 Ceramic Flocked Rollers
Description: Lock-in smooth, frizz-free waves and curls. The INFINITI PRO CONAIR™ Ceramic Flocked Hot Rollers let you quickly ...
———

Candidate items:
Candidate 1
Title: Creative Nail Shellac Fedora, 0.25 Fluid Ounce
Description: Patent-pending formulations of solvents, monomers and polymers goes on like polish and wears like gel. A UV-cured color coat featuring UV3 technology.
———

Candidate 2
Title: AcneFree Severe Acne Treatment System
Description: Significantly better faster than Proactiv, plus 4 times more powerful. With time-released 10% Benzoyl Peroxide & Retinol. Accelerates healing fast - ...
———

Candidate 3
Title: Maybelline Dream Fresh BB Cream 100 Light Sheer Tint, 1.0 ounces
Description: Dream fresh BB cream 8-in-1 skin perfector is the 1-step wonder for a good skin day, everyday. ...
———

Candidate 4
Title: World Pride Pocket Nail Cuticle Nipper Pack Contains Nail Trimmer, Pack of 3
Description: Package included: 1pc of Nail Cuticle Nipper; 2pcs of Nail Files
———

Candidate 5
Title: WEN reg; Sweet Almond Mint Styling Creme 6oz
Description: Achieve all your styling options and the benefits of a leave-in conditioner, polishing gloss and gel in one product. ...
———

Candidate 6
Title: The Brush Guard–Foundation Pack
Description: The Brush Guard is an entirely new brush care product which allows brushes to be washed and then dried in the best way possible: ...
———

Candidate 7
Title: Neutrogena Ultra Gentle Daily Cleanser, 12 Ounce
Description: This foaming cleanser works in a new, skinfriendly way to leave you with clean yet naturally soft skin. The nonstripping formula thoroughly removes excess oil, dirt, and makeup without overdrying or irritating skin.
———

Candidate 8
Title: Nexxus Color Assure Shampoo, 33.8Ounce Bottle
Description: Hair is made up of protein. Every touch, every treatment, every style you love—it all rewrites your hair's protein. When protein is damaged, hair's look and feel is compromised. ...
———

Candidate 9
Title: Red Door Revealed By Elizabeth Arden For Women. Eau De Parfum Spray 3.4 Ounces
Description: Introduced in 2003. Fragrance notes: An inviting blend of sweet fruits and florals, with lower notes of moss, woods and amber. ...
———

Candidate 10
Title: Kanekalon Jumbo Braid Extension Hair Jet Black
Description: Perfect for braiding and dreadlock hair extensions. Professional quality, easy sealing Kanekalon fiber. ...
———

Baseline SFT Student CoT

<think>
* Purchase 1: Laura Mercier Loose Setting Powder (Face Makeup/Powder)
* Purchase 2: Trish McEvoy Lash Curling Mascara (Eye Makeup/Mascara)
* Purchase 3: Maybelline New York Shine Free Oil Control Pressed Powder (Face Makeup/Powder/Oil control/Acne-prone focus mentioned (Error: the purchase only indicates oil control; acne-prone skin is not established.))
* Purchase 4: Acqua Di Gio By Giorgio Armani (Fragrance/Eau de Toilette)
* Purchase 5: Infiniti Pro by Conair Ceramic Flocked Rollers (Hair Tools)

Summary: Beauty/Cosmetics (Makeup/Face, Eye Makeup, Fragrance, Hair Care).

* Candidate 1: Creative Nail Shellac Fedora (Nail Polish/Gel). Category: Cosmetics/Nails.
* Candidate 2: AcneFree Severe Acne Treatment System (Skincare/Acne treatment).
* Candidate 3: Maybelline Dream Fresh BB Cream (Face Makeup/Tinted moisturizer). Category: Face Makeup/Skincare.
* Candidate 4: World Pride Pocket Nail Cuticle Nipper Pack (Nail Tool). Category: Personal Grooming/Nails.
* Candidate 5: WEN® Sweet Almond Mint Styling Creme (Hair Product/Styling). Category: Hair Care.
* Candidate 6: The Brush Guard–Foundation Pack (Brush Care tool). Category: Makeup Accessories/Hygiene.
* Candidate 7: Neutrogena Ultra Gentle Daily Cleanser (Skin Cleanser). Category: Skincare.
* Candidate 8: Nexxus Color Assure Shampoo (Shampoo/Hair Care). Category: Hair Care.
* Candidate 9: Red Door Revealed By Elizabeth Arden (Fragrance/Eau de Parfum). Category: Fragrance.
* Candidate 10: Kanekalon Jumbo Braid Extension Hair (Hair Extensions/Braids). Category: Hair Care.

* *Group A: High Similarity (Directly related categories)*

* Candidate 3 (Maybelline BB Cream): Very high relevance. User bought Maybelline products previously (Purchases 1, 3). (Error: only Purchase 3 is Maybelline; Purchase 1 is Laura Mercier.) Focus is on face/skin perfection/makeup.
* Candidate 9 (Elizabeth Arden Perfume): High relevance. User bought perfume (Purchase 4).
* Candidate 2 (AcneTreatment): Medium-High relevance. User bought "Oil Control" and mentions suitability for "acne prone skin" (Purchase 3).
* Candidate 7 (Neutrogena Cleanser): Medium relevance. Part of skincare routine often paired with makeup/oils/cleansing.
* Candidate 6 (Brush Guard): Medium relevance. Related to foundation/powders (makeup usage context).
* Candidate 1 (Creative Nail Shellac): Low-Medium relevance. Cosmetic category but nail-specific.
* Candidate 5 (WEN Styling Creme): Low-Medium relevance. Hair care/styling (matches Purchase 5 theme).
* Candidate 8 (Nexxus Shampoo): Low-Medium relevance. Hair care.
* Candidate 10 (Kanekalon Braids): Low relevance. Hair accessory/extensions.
* Candidate 4 (Cuticle Nippers): Low relevance. Nail grooming.

* *Refining Group A Ranking:*
1. Candidate 3 (BB Cream): Direct brand overlap (Maybelline) + direct product type overlap (face makeup/tint).
2. Candidate 9 (Perfume): Direct product type overlap (fragrance).
3. Candidate 2 (Acne Treatment): Matches specific need identified in Purchase 3 ("suitable for acne prone skin").
4. Candidate 7 (Cleanser): Standard part of facial maintenance.
5. Candidate 6 (Brush Guard): Accessory for applying/fathering powders (like P1, P3).

* *Ranking Candidates based on closeness:*

* Rank 1: Candidate 3 (Maybelline BB Cream) - Brand affinity (Maybelline) + Functionality (Face/-Foundation).
* Rank 2: Candidate 9 (Red Door Perfume) - Category affinity (Fragrance).
* Rank 3: Candidate 2 (Acne Treatment) - Specific problem-solving (Acne) matching Purchase 3 description.
* Rank 4: Candidate 7 (Neutrogena Cleanser) - Routine consistency (Facial cleansing).
* Rank 5: Candidate 6 (Brush Guard) - Utility for makeup application (Powders).
* Rank 6: Candidate 1 (Nail Shellac) - General cosmetic/beauty interest.
* Rank 7: Candidate 5 (Styling Creme) - Hair care/styling (Matches Purchase 5).

Baseline SFT Student CoT *(continued)*

* Rank 8: Candidate 8 (Shampoo) - Basic personal care/hair.
* Rank 9: Candidate 10 (Braid Extensions) - Hair extension/accessories.
* Rank 10: Candidate 4 (Cuticle Nipper) - More niche nail/grooming.

Wait, let me re-evaluate Candidate 1 vs Candidate 5/8/10.
User buys: Powder (Face), Mascara (Eyes), Powder (Face), Fragrance (Scent), Hair rollers (Hair).
Total themes: Facial cosmetics, Scent, Hair accessories/tools.

Let's refine ranking again:
1. Candidate 3 (Maybelline BB Cream) -> Strongest link via brand and function.
2. Candidate 9 (Elizabeth Arden Perfume) -> Strong link via category.
3. Candidate 2 (Acne Treatment) -> Link via concern expressed in Purchase 3.
4. Candidate 7 (Neutrogena Cleanser) -> Logical follow-up to makeup/skincare users.
5. Candidate 6 (Brush Guard) -> Useful for someone using multiple powders (P1, P3).
6. Candidate 1 (Nail Shellac) -> Broadening into nails.
7. Candidate 5 (Styling Creme) -> Links to hair (P5).
8. Candidate 8 (Shampoo) -> Links to hair (P5).
9. Candidate 10 (Braid Extensions) -> Links to hair (P5).
10. Candidate 4 (Cuticle Nipper) -> Less relevant/niche.

Comparing Candidate 1 (Nails) vs Candidate 5 (Hair):
User has purchased two hair-related things (rollers) (Error: there is only one hair-related purchase, the rollers.) and three face/body-related things (powder, mascara, powder, perfume).
Actually, count:
P1: Face
P2: Eyes
P3: Face
P4: Body/Scent
P5: Hair
Total = 3 Face, 1 Eye, 1 Body, 1 Hair. (Error: this double-counts categories; there are only five purchases, and only P1/P3 are Face.)

Therefore, Face-centric products should definitely lead.
Candidate 3 (Face) > Candidate 9 (Body/Scent) > Candidate 2 (Face-Specific Concern) > Candidate 7 (Face-Cleaning) > Candidate 6 (Face-Accuracy) > ...?

Is Candidate 1 higher? Usually, if they buy makeup, they might want nail polish too.
Is Candidate 5 higher? They did buy hair rollers.

Re-ordering logic:
1. Candidate 3 (Maybelline BB Cream) - Most similar to existing Maybelline and face-focused purchases.
2. Candidate 9 (Perfume) - Same category as P4.
3. Candidate 2 (Acne Treatment) - Specifically addresses concerns noted in P3.
4. Candidate 7 (Cleanser) - Essential for maintaining face products.
5. Candidate 6 (Brush Guard) - Maintains accuracy/application of face products.
6. Candidate 1 (Nail Shellac) - Moving towards general beauty/color.
7. Candidate 5 (Hair Styling) - Moves into hair.
8. Candidate 8 (Shampoo) - Basic hair.
9. Candidate 10 (Extensions) - Specialized hair.
10. Candidate 4 (Nail nipper) - Niche tool.

Check Candidate 1 again: "goes on like polish". Fits the aesthetic of Laura Mercier and Maybelline.
Check Candidate 5: Styling creme. Fits the hair roller profile.
Check Candidate 8: Shampoo. Generic hair.
Check Candidate 10: Extensions. Hair.

Final sequence attempt: 3, 9, 2, 7, 6, 1, 5, 8, 10, 4.

One detail: Candidate 2 (Acne) says "maximum strength... for people of all ages..." while Purchase 3 explicitly mentions "suitable for acne prone skin." That makes Candidate 2 very strong.

Double-check Candidate 6 (Brush guard): Used for Foundation/Direct makeup application. Since the user uses loose setting powder (P1, P3), a brush protector is highly practical.

Order:
3 (Brand Match + Type Match)

Baseline SFT Student CoT *(continued)*

9 (Category Match)
2 (Concern Match)
7 (Routine Support)
6 (Tool Support)
1 (Beauty Category expansion)
5 (Hair Category expansion)
8 (Hair Category basics)
10 (Hair Category specialty)
4 (Grooming Category tool)

Looking at Candidate 1 (Shellac) vs Candidate 5 (Styling Creme)...
User likes premium names (Laura Mercier, Trish McEvoy, Maybelline, Armani). Candidate 1 (Creative Nail) sounds slightly more aligned with that "beauty enthusiast/prestige" vibe than generic shampoo.

Checking Candidate 10 (Braids) vs Candidate 8 (Shampoo)... both hair. Shampoo is more fundamental.

Refined List: [3, 9, 2, 7, 6, 1, 5, 8, 10, 4]

Wait, I'll double check Candidate 1 (Nails) position. Many makeup buyers expand to nails. But the primary driver here seems to be the core interests found in the first few purchases: Face/Eye/Scent/Hair.

Candidate 3: Maybelline BB Cream (High)
Candidate 9: Perfume (Medium-High)
Candidate 2: Acne Treatment (Medium-High due to P3 mention)
Candidate 7: Cleanser (Medium)
Candidate 6: Brush Guard (Medium-Low)
Candidate 1: Nail Shellac (Low-Medium)
Candidate 5: Styling Creme (Low)
Candidate 8: Shampoo (Low)
Candidate 10: Braid Extensions (Very low)
Candidate 4: Cuticle Nipper (Very low)

Does Candidate 1 go above 7?
7 is a daily necessity for anyone wearing lotions/powders.
1 is decorative.
Users buying professional-grade stuff usually prioritize utility/routine.

Let's verify Candidate 2 again. "Attacks the 3 stages of severe acne".
Purchase 3: "...suitable for acne prone skin."
This connection is quite tight.

Final verification of indices:
1: Shellac
2: Acne
3: BB Cream
4: Nipper
5: Styling Creme
6: Brush Guard
7: Cleanser
8: Shampoo
9: Perfume
10: Braid

Sequence: 3, 9, 2, 7, 6, 1, 5, 8, 10, 4.
</think>

[3, 9, 2, 7, 6, 1, 5, 8, 10, 4]

Compress-Distill 0.6B Student CoT

<think>
Purchase History Analysis:
* *Makeup/Cosmetics:*
* Purchase 1: Laura Mercier Loose Setting Powder (Face setting)
* Purchase 2: Trish McEvoy Lash Curling Mascara (Eye makeup)

Compress-Distill 0.6B Student CoT *(continued)*

* Purchase 3: Maybelline New York Shine Free Oil Control Pressed Powder (Face setting/oil control)
* *Fragrance:*
* Purchase 4: Acqua Di Gio (Giorgio Armani) (Eau de Toilette/Fresh perfume)
* *Hair Tools/Styling:*
* Purchase 5: Infiniti Pro by Conair Instant Heat 20 Ceramic Flocked Rollers (Hair styling tool)

Initial Candidate Analysis:
* Candidate 1: Creative Nail Shellac Fedora (Nail Polish/Gel) — Category: Beauty/Nails
* Candidate 2: AcneFree Severe Acne Treatment System (Skincare/Acne treatment) — Category: Beauty/Skincare
* Candidate 3: Maybelline Dream Fresh BB Cream (Skin tint/makeup) — Category: Makeup
* Candidate 4: World Pride Pocket Nail Cuticle Nipper Pack (Nail Care) — Category: Beauty/Nails
* Candidate 5: WEN Sweet Almond Mint Styling Creme (Hair styling/care) — Category: Beauty/Haircare
* Candidate 6: The Brush Guard—Foundation Pack (Makeup brush cleaner) — Category: Makeup Accessories
* Candidate 7: Neutrogena Ultra Gentle Daily Cleanser (Facial cleansing/skin prep) — Category: Beauty/Skincare
* Candidate 8: Nexxus Color Assure Shampoo (Hair washing) — Category: Beauty/Haircare
* Candidate 9: Red Door Revealed By Elizabeth Arden (Eau De Parfum/Fragrance) — Category: Fragrance
* Candidate 10: Kanekanon Jumbo Braid Extension Hair (Hair styling/extensions) — Category: Beauty/Hairspray

Verification/ Self-Correction:

Step 1: Grouping Candidates into Categories
To organize the search, I will first group the candidates based on their broad category relevance to the user's history:
* *Group 1: Makeup (High Relevance):* Cand 3 (BB Cream), Cand 6 (Brush Guard).
* *Group 2: Skincare (Medium-High Relevance):* Cand 2 (Acne Treatment), Cand 7 (Cleanser).
* *Group 3: Fragrance (Direct Match):* Cand 9 (Perfume).
* *Group 4: Haircare/Styling (Medium Relevance):* Cand 5 (Styling Creme), Cand 10 (Braids), Cand 8 (Shampoo).
* *Group 5: Nails/Miscellaneous (Lower Relevance):* Cand 1 (Nail Polish), Cand 4 (Nail Tool).

Step 2: Initial Ranking Attempt
Based on these groups, my initial ranking is:
[3, 9, 6, 7, 2, 5, 10, 8, 1, 4]

Step 3: Refinement and Backtracking

*(a) *Refining the Top Tier (Brand vs. Product Type):*
I need to decide between Candidate 3 (Maybelline BB Cream) and Candidate 9 (Elizabeth Arden Perfume).
* Candidate 3 offers a double signal: it matches the "Makeup" category (P1, P2, P3) AND the specific brand "Maybelline" (P3).
* Candidate 9 is a direct category match for "Fragrance" (P4), but it lacks the explicit brand loyalty seen with Maybelline.
Verdict:* Keep 3 at Rank 1 and 9 at Rank 2.

*(b) *Re-evaluating Accessory Utility (Rank 3 vs. 4):*
Let's compare Candidate 6 (Brush Guard) and Candidate 7 (Neutrogena Cleanser).
* Initially, I might place the general skincare/cleanser higher (Cand 7), but Candidate 6 is a specialized accessory designed for cleaning foundation brushes. Since the user heavily invests in multiple powders and mascaras, they are highly engaged with application/protection. Therefore, the utility of the Brush Guard (Cand 6) makes it a stronger secondary priority than basic skincare (Cand 7).
Updated Order:* 3 > 9 > 6 > 7.

*(c) *Comparing Skincare Needs (Cand 7 vs. Cand 2):*
Between the facial cleanser (Cand 7) and acne treatment (Cand 2), which should come first?
* The user purchased oil-control pressed powder (P3), implying a focus on facial maintenance.
* While the powder needs a gentle cleanser (Cand 7), the acne-treatment system (Cand 2) aligns well with the functional profile of the Maybelline oil-control powder.
* However, standard daily facial care (Cand 7) typically precedes intensive medicated treatments unless a specific medical concern like acne is established.

Compress-Distill 0.6B Student CoT *(continued)*

Decision:* Prioritize the routine (Cand 7) before the treatment (Cand 2).

*(d) *Finalizing Haircare Stamping (Cand 5 vs. Cand 10 vs. Cand 8):*
For the remaining hair products, I must determine if styling, accessories, or wash matters.
* Candidate 5 (Styling Creme) is a high-use item similar to hair rollers (P5).
* Candidate 10 (Braid Extensions) is a specialty/styling item.
* Candidate 8 (Shampoo) is a fundamental wash.
* Given the user bought specific styling tools, the styling-focused items (5 and 10) should rank above the shampoo (8).

Final Optimized Sequence:
1. *Cand 3* (Strongest: Brand + Category match)
2. *Cand 9* (Category match: Fragrance)
3. *Cand 6* (Utility match: Makeup brush protection)
4. *Cand 7* (Routine match: Facial cleansing)
5. *Cand 2* (Need match: Functional facial skincare)
6. *Cand 5* (Style match: Hair styling)
7. *Cand 10* (Accessory match: Hair styling)
8. *Cand 8* (Basic match: Hair wash)
9. *Cand 1* (Low overlap: Nails)
10. *Cand 4* (Low overlap: Nails)
</think>

[3, 9, 6, 7, 2, 5, 10, 8, 1, 4]

A*-Thought 0.6B Student CoT

<think>
* Purchase 1: Laura Mercier Loose Setting Powder (Makeup/Face)
* Purchase 2: Trish McEvoy Lash Curling Mascara (Makeup/Eyes)
* Purchase 3: Maybelline New York Shine Free Oil Control Pressed Powder (Makeup/Face/Oil control)
* Purchase 4: Acqua Di Gio By Giorgio Armani For Men (Fragrance/Men's)
* Purchase 5: Infiniti Pro by Conair Instant Heat 20 Ceramic Flocked Rollers (Hair/Styling)

Summary: User buys cosmetics (face powders, mascara) and fragrances. One perfume (though it was "For Men", implying adult female use (Error: contradiction.)).

* Candidate 1: Creative Nail Shellac Fedora (Nail Polish/Cosmetics).
* Candidate 2: AcneFree Severe Acne Treatment System (Skincare/Acne treatment).
* Candidate 3: Maybelline Dream Fresh BB Cream (Face Makeup/Skin tint).
* Candidate 4: World Pride Pocket Nail Cuticle Nipper Pack (Nail Care tool).
* Candidate 5: WEN Sweet Almond Mint Styling Creme (Haircare/Styling).
* Candidate 6: The Brush Guard (Brush care/makeup accessory).
* Candidate 7: Neutrogena Ultra Gentle Daily Cleanser (Skincare/Cleanser).
* Candidate 8: Nexxus Color Assure Shampoo (Haircare/Shampoo).
* Candidate 9: Red Door Revealed By Elizabeth Arden (Fragrance/Women's).
* Candidate 10: Kanekalon Jumbo Braid Extension Hair (Hair extension/Haircare).

Let's re-evaluate Candidate 1 vs 6 vs 7.
P1/P3 are powders/tints.
P2 is mascara.
Candidate 3 = BB Cream (Very similar to P1, P2, P3).
Candidate 6 = Brush guard (Related to applying/cleaning up makeup, like P1/P3).
Candidate 7 = Cleanser (Cleansing makeup products is common for users of loose powders/specialized makeup).

Rank attempt 2 (Error: not second attempt.)
3, 6, 7, 9, 5, 2, 10, 8, 1, 4

Wait, let me reconsider Candidate 9 position. The user bought a high-end perfume (Laura Mercier) (Error: it is a powder). A luxury woman's perfume is quite likely for someone who also uses Trish McEvoy and Maybelline.
And Candidate 2? The user had "Oil Control" pressure powder. Acne treatment is very specific.
Let's try to group by category:

A*-Thought 0.6B Student CoT *(continued)*

Cosmetics (P1, P2, P3): 3, 6, 1
Skincare (None explicitly, but relates to makeup): 7, 2
Fragrance (P4): 9
Hair (P5): 5, 10, 8, 4

Ordering these groups:
Group Cosmetics: 3 (most frequent/type), 6 (accessory), 1 (nails/cosmetic)
Group Skincare: 7 (cleansing face), 2 (acne treatment)
Group Fragrance: 9
Group Hair: 5 (styling), 10 (extensions), 8 (shampoo), 4 (tools) (Error: 4 is nail clipper, not for hair.)

Final sequence: 3, 6, 1, 7, 9, 2, 5, 10, 8, 4.

One last thought: Is Candidate 1 closer to P1/P2/P3 than Candidate 7?
P1/P3 are powders (shellac can be seen as a specialized powder-based cosmetic (Error: Shellac is nail coat, not powder)).
Candidate 1 is nail shellac (color/coat).
Candidate 3 is BB cream (tint/light).
Actually, 3 and 1 are both about covering/blushuing the face (Error: C1 is a nail coat, not blush.), while 6 is about tools. All are close.

Comparing 3, 6, 1, 7...
3 is Maybelline (Brand Match!)
6 is Tool (Utility Match)
1 is Cosmetic (Category Match)
7 is Skincare (Maintenance Match)

This seems solid.

Checking Candidate 1 again... "Shellac... goes on like polish". Polished/Legal/Nails.
Checking Candidate 6... "Brush Guard... allows brushes to be washed and then dried".
</think>

[3, 6, 1, 7, 9, 2, 5, 10, 8, 4]

SCOReD 0.6B Student CoT

<think>
* Purchase 1: Laura Mercier Loose Setting Powder (Makeup/Face)
* Purchase 2: Trish McEvoy Lash Curling Mascara (Makeup/Eyes/Jewelry-like)
* Purchase 3: Maybelline New York Shine Free Oil Control Pressed Powder (Makeup/Face/Oil control/-Mattifying)
* Purchase 4: Acqua Di Gio By Giorgio Armani (Fragrance/Cologne)
* Purchase 5: Infiniti Pro by Conair Instant Heat Ceramic Flocked Rollers (Hair Tools)

Summary: User buys cosmetics (face and eyes), Skincare (specifically related to oil control/makeup application), Fragrance, and Hair Care/Styling.

* Candidate 1: Creative Nail Shellac Fedora (Nail Polish/Gel) -> Fits "Beauty".
* Candidate 2: AcneFree Severe AcneTreatment System (Skincare/Acne) -> Related to skincare/oily skin (similar to Purchase 3).
* Candidate 3: Maybelline Dream Fresh BB Cream (Makeup/Face/Tint) -> Very high affinity to P1, P2, P3 (makeup/beauty category).
* Candidate 4: World Pride Pocket Nail Cuticle Nipper (Nails/Skin tool) -> General beauty/nail care.
* Candidate 5: WEN Sweet Almond Mint Styling Creme (Hair styling) -> High affinity to P5 (hair care).
* Candidate 6: The Brush Guard–Foundation Pack (Makeup Tool/Hygiene) -> High affinity to makeup usage (P1, P2, P3).
* Candidate 7: Neutrogena Ultra Gentle Daily Cleanser (Skincare/Cleanser) -> High affinity to skincare/cleaning (related to removing makeup/powder).
* Candidate 8: Nexxus Color Assure Shampoo (Hair care/Shampoo) -> High affinity to P5 (hair care).
* Candidate 9: Red Door Revealed By Elizabeth Arden (Fragrance/Women) -> High affinity to P4 (fragrance).
* Candidate 10: Kanekalon Jumbo Braid Extension (Hair extension) -> High affinity to P5 (hair).

Re-evaluating Candidates 2 and 7 against P3 ("Oil Control"): While both address oil versus acne concerns, I'll prioritize the logical sequence: 3 (Maybelline + Face), 6 (Brushes + Makeup Accessory), 7

SCOReD 0.6B Student CoT *(continued)*

(Cleanser + Facial Care), 9 (Perfume), 2 (Acne treatment + Skin concern), 5 (Styling), 8 (Shampoo), 10 (Extensions/Braids link to P5 rollers), 1 (Nails), and 4 (Nails). Regarding other connections: P1/P2/P3 relate to cosmetics/application, while P4 relates to perfume, and P5 refers to hair styling tools. Comparing relevance, Candidate 10 connects specific hair textures (braids/extensions) to roller users, whereas Candidate 5 focuses on general hair styling. Additionally, comparing nail items (Candidates 1 and 4), shellac (1) is a long-term cosmetic similar to makeup, while nipper (4) is basic grooming.

Final Order Check:

Is Candidate 2 higher? As an acne treatment, it strongly matches the user's interest in oil control for acne-prone skin (P3) (Error: P3 is an oil-control powder; no direct evidence that the user has acne-prone skin.), though cleansing (7) is typically a safer baseline. Is Candidate 9 higher? Since they bought Armani Cologne, Red Door fits the fragrance niche perfectly.

Let me re-rank slightly:
3, 6, 7, 9, 2, 5, 8, 10, 1, 4.

Compare Candidate 1 vs. 4: Shellac (polishing/gel coating) differs significantly from talc powder but shares the "cosmetic" theme. Compare Candidate 10 vs. 8: Extensions align with the user's recent hair tool purchase.

Refined ranking: 3 (BB Cream), 6 (Brush guard), 7 (Neutrogena), 2 (AcneFree), 9 (Red Door Perfume), 5 (WEN Styling), 8 (Nexxus Shampoo), 10 (Kanekalon Braids), 1 (Shellac), 4 (Cuticle Nippers).

Comparing 2 and 9 again...
P3: Oil control for acne-prone skin.
2: Acne treatment.
They're very close.

Order seems solid: 3, 6, 7, 2, 9, 5, 8, 10, 1, 4.

Wait, let's reconsider if 9 should be closer to 2.

If we group by category:
Category: Makeup (3, 6)
Category: Skincare (7, 2)
Category: Fragrance (9)
Category: Hair (5, 8, 10)
Category: Nails (1, 4)

Interleaved ordering:
3 (Brand + Face), 9 (Fragrance), 2 (Skin treatment), 6 (Tool), 7 (Cleanser), 5 (Hair style), 8 (Hair wash), 10 (Hair extensions/braid links to P5), 1 (Nails), 4 (Nails).

This interleaved approach provides a balanced assessment of the categories' importance and their relation to the user's profile.

Rankings: #1 Candidate 3 (Maybelline), #2 Candidate 6 (Brush Guard), #3 Candidate 7 (Cleanser), #4 Candidate 9 (Fragrance), #5 Candidate 2 (Acne), etc.
(Error: Does not match the "Interleaved ordering" proposed above.)

Verification:
- User interests: Makeup (powders), Eyes (mascara), Oil control (powder), Fragrance (Armani), Hair Roller.
- Matches: 3 (Face makeup), 6 (Makeup accessory), 7 (Facial cleanser/skincare), 9 (Fragrance), 2 (Acne treatment/skincare), 5 (Hair styling).
- Misses: 8 (Shampoo/hair), 10 (Braids/hair), 1 (Nails/cosmetics), 4 (Nails/tools).
</think>

[3, 6, 7, 9, 2, 5, 8, 10, 1, 4]