CURE: Coupled User-Grouped Reinforcement Learning for Cross-Domain Recommendation with Non-Overlapping Users

Hyeongjun Yun[1,2] **Taesan Kim**[1] **Dongjoon Hong**[1] **Junui Hong**[1] **Jihoon Oh**[1]
Kijung Park[1] **MinCheol Cho**[1] **Kihyuk Song**[1] **Jaegul Choo**[2] **Chung Park**[1]*
[1]SK Telecom [2]KAIST

Abstract

Cross-Domain Recommendation is essential for enabling cross-sell in multi-service platforms; however, limited user overlap and strict cross-service data-sharing constraints often result in little to no ground-truth supervision for cross-domain learning. To bridge this gap, we propose an LLM-based framework for cross-domain recommendation in the non-overlapping user scenario. First, we propose an agentic pipeline that constructs cross-domain pseudo supervision, which is unavailable in the non-overlapping user scenario, by leveraging a user's source-domain history to generate a target-domain item recommendation. Using this dataset, we further propose Coupled User-GRouped REinforcement Learning (CURE), which induces coupling between in- and cross-domain gradient updates via user-level joint normalization, thereby calibrating synthetic cross-domain updates against reliable in-domain signals. We provide a theoretical analysis showing that user-level coupling stabilizes policy updates under imperfect pseudo supervision and improves cross-domain generalization. Experiments on Amazon Reviews and MovieLens show consistent gains over baselines in cross-domain recommendation, while also improving in-domain performance. An online A/B test demonstrates a statistically significant 56% lift in click-through rate over existing ML methods[1].

1 Introduction

Cross-Domain Recommendation (CDR) refers to recommender systems that integrate user behavioral signals from multiple service domains to improve recommendation performance in a target or newly introduced domain [1]. This paradigm is particularly important in real-world platforms that operate diverse services—such as the commerce, video, and music ecosystems of *Amazon*—where CDR serves as a core technology for cross-sell strategies.

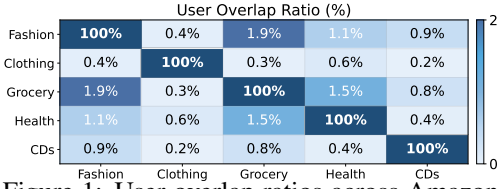


Figure 1: User overlap ratios across Amazon domains. Off-diagonal values are uniformly low, indicating minimal cross-domain overlap.

Despite its importance, real-world CDR faces a fundamental challenge arising from the imbalance of user behavioral data across service domains. Many users exhibit rich interaction histories in a small number of domains, while having little to no purchase or interaction records in others, often due to practical or legal constraints that limit the sharing of user information across service domains [2]. We refer to this setting as the cross-domain recommendation with **non-overlapping users** (Fig. 1) [3, 4]. In such cases, even when abundant user histories are available in a source domain, the absence of overlapping interactions in the target domain prevents the construction of ground-truth supervision for cross-domain recommendation. However, in contrast to this practical reality, most prior CDR

*Corresponding Author: cpark88kr@gmail.com, skt.cpark@sk.com
[1]We released our data and code to support reproducibility (https://github.com/cpark88/CURE).
40th Conference on Neural Information Processing Systems (NeurIPS 2026).

studies are developed under the assumption of overlapping scenarios, where users possess sufficient interaction histories across multiple domains and serve as effective bridges for cross-domain knowledge transfer [5]. Such assumptions rarely hold in real-world industrial systems, thereby limiting the direct applicability of existing CDR methods in practice.

Therefore, our goal is to train an LLM-based recommender that conditions on a user's source-domain history and flexibly recommends items from candidate sets in different target domains, enabling cross-domain recommendation in the non-overlapping user scenario. Moreover, our model generates natural-language rationales for its recommendations, leveraging the LLM's reasoning capability to enhance user engagement. To this end, we introduce two complementary components: (1) an Agentic Synthetic Data Generation Pipeline (ASP) and (2) a Coupled User-GRouped REinforcement Learning (CURE). ASP is a multi-agent pipeline that synthesizes high-quality cross-domain training data, which is otherwise unavailable in the non-overlapping user scenario. For in-domain cases (history and target item from the same domain), ASP generates rationales for ground-truth recommendations; for cross-domain cases (history and target item from different domains), it retrieves semantically related target-domain candidates, selects an item, and generates its rationale. The synthetic data exhibit statistically significant consistency with human judgments in post-hoc evaluation.

While ASP provides pseudo supervision for cross-domain recommendation, effectively integrating synthetic cross-domain signals with reliable in-domain interactions requires a principled coupling mechanism that enables cross-domain transfer without corrupting in-domain learning. Accordingly, we propose CURE, a reinforcement learning framework that explicitly aligns in- and cross-domain reward signals at the user level, built upon an adaptation of Group Relative Policy Optimization (GRPO). Specifically, for each user, we group in-domain (ground-truth) and cross-domain (synthetic) samples sharing the same context. Rewards are then normalized and compared within these user-specific groups, ensuring that all samples from the same user are placed in a single mini-batch prior to policy updates. Moreover, we design a recommendation-specific reward that encourages correct predictions and adaptively calibrates reward scales across domains, enabling stable within-user normalization. Intuitively, CURE **cures** noisy cross-domain supervision by anchoring it to reliable in-domain signals, even in challenging non-overlapping user scenarios.

Moreover, our theoretical analysis shows that CURE's user-level coupling anchors cross-domain updates to reliable in-domain signals, reducing the sensitivity of normalized policy updates to noisy pseudo supervision and yielding more stable cross-domain learning than uncoupled training (i.e., vanilla GRPO). We evaluate CURE through large-scale experiments on the Amazon Reviews dataset spanning five service domains and on MovieLens, where it generally outperforms state-of-the-art baselines in both in- and cross-domain recommendation tasks. Moving beyond a purely academic study, **our model was deployed in a large-scale production environment to validate its real-world impact**. Even in challenging non-overlapping scenarios, an online A/B test yielded a statistically significant 56% relative lift in click-through rate over the production baseline.

2 Preliminaries

Definition 1. Cross-Domain. Let the domain set be denoted as $\mathcal{D}$. A specific domain is denoted by $d \in \mathcal{D}$, and the set of items belonging to domain d is expressed as $\mathcal{V}^d$. Given two distinct domains $d, \tilde{d} \in \mathcal{D}$ with $d \neq \tilde{d}$, we treat d as the *source domain* where user histories are observed and $\tilde{d}$ as the *target domain* from which candidates and recommended items are drawn. We refer to this source (user history)–target (next item) formulation as the cross-domain recommendation setting.

Definition 2. Non-Overlapping User Sequence: We represent a user's interaction sequence in the source domain $d \in \mathcal{D}$ as $I^d = (i_1^d, i_2^d, \ldots, i_n^d)$, where interactions are ordered chronologically and n is the sequence length. Each interaction i^d is associated with an item's textual metadata (title and description) and, when available, user-generated feedback such as a review and a rating. Then, we set the ground-truth next item $i_{n+1}^d = i_+^d \in \mathcal{V}^d$ that the user interacted with following the historical interaction sequence I^d. **Note that in the non-overlapping user scenario, we assume that the user history I^d contains no items from other domains.** Based on this formulation, we construct an initial in-domain training dataset $\mathcal{S}^d = \{(I^d, i_+^d)\}$.

3 Methodology

3.1 Agentic Synthetic Data Generation Pipeline (ASP)

As shown in Fig. 2, given the in-domain training set $\mathcal{S}^d$, we use ASP to construct additional cross-domain training samples by conditioning on the source-domain user history I^d. We define **in-domain**

recommendation as the case where the user history, candidate items, and recommended item share the same domain, whereas **cross-domain recommendation** refers to the case where the user history domain differs from that of the candidate and recommended items, corresponding to the non-overlapping user scenario. ① For the in-domain case, given (I^d, i_+^d), we generate the rationale r_+^d that explains why the ground-truth item i_+^d is recommended to the user. ② For the cross-domain case, where no ground-truth exists, we use the same anchored user history I^d to generate the most semantically relevant item $i_+^{\tilde{d}}$ and its corresponding rationale $r_+^{\tilde{d}}$, where $\tilde{d} \in \mathcal{D} \setminus \{d\}$.

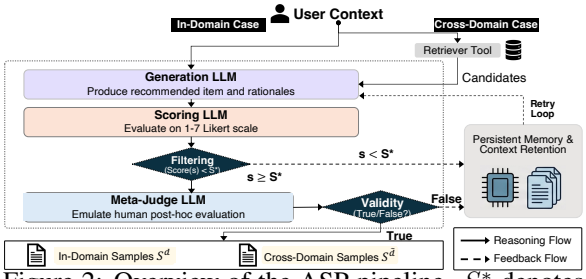


Figure 2: Overview of the ASP pipeline. S^* denotes the score threshold.

ASP is a multi-agent framework that comprises three LLM modules—(i) Generation, (ii) Scoring, and a (iii) Meta-Judge—along with a domain-specific item retrieval tool and a persistent memory component that retains prior context. The Generation LLM is responsible for producing recommended items and their corresponding rationales based on user histories. For in-domain samples where the ground-truth item i_+^d is available, the module only generates its corresponding rationale r_+^d. We then construct an in-domain candidate set C^d of size M by including i_+^d and randomly sampling the remaining M-1 items from the domain-specific item vocabulary $\mathcal{V}^d$, excluding i_+^d. In the cross-domain case, lacking explicit ground-truth and involving an extremely large candidate space, the LLM invokes the retrieval tool (see Appendix H.2) to identify M relevant item candidates (i.e., $C^{\tilde{d}}$) from domains different from the user history's domain before generating the most persuasive recommendation–rationale pair.

The Scoring LLM evaluates each generated sample in terms of plausibility and relevance using a 7-point Likert scale[2]. Samples with scores < 4 are discarded and returned to the Generation stage for refinement, accompanied by a prompt directing the model to regenerate a new sample distinct from the previous one. The Meta-Judge LLM emulates human post-hoc evaluation by jointly assessing the recommended item, its rationale, and intermediate scores, and outputs a binary validity label (True/False) based on overall consistency. When marked as False, they are sent back to the Generation stage for regeneration. The Meta-Judge LLM achieved 75.0% agreement with human majority-vote labels, with statistically significant chance-corrected agreement. Note that the three modules use different LLMs—`GPT-4.1` for generation, `Gemma-3-27B-it` for scoring, and `GPT-4.1-mini` for meta-judging—forming a diversified reasoning ensemble. Detailed descriptions of ASP, including tool specifications, operational costs, and human validation protocols, are provided in Appendix H.

Finally, given the initial in-domain dataset (I^d, i_+^d), our ASP generates two types of datasets for each user: ① an **in-domain ground-truth sample** $\mathcal{S}^d = \{(I^d, C^d, i_+^d, r_+^d)\}$, and ② a **cross-domain pseudo-labeled sample** $\mathcal{S}^{\tilde{d}} = \{(I^d, C^{\tilde{d}}, i_+^{\tilde{d}}, r_+^{\tilde{d}})\}$.

One may argue that ASP itself could serve as a highly accurate cross-domain recommender for non-overlapping users, since it directly generates both recommended items and rationales. However, ASP relies on multiple frontier-scale LLMs, retrieval and memory access, and iterative retry mechanisms, making its inference latency and cost impractical for online deployment. We therefore use ASP only as an offline pseudo-label construction pipeline: it generates high-quality cross-domain supervision once, which is then used to train a smaller LLM specialized for efficient recommendation serving. The average ASP generation cost is approximately $0.023 per sample, making the one-time construction of 150K pseudo-labeled samples practically manageable. After deployment, the served model is continually updated using newly collected feedback from real non-overlapping users, which serves as cross-domain supervision and makes the initial ASP construction cost reasonable in practice. This is also the strategy adopted in our real-world service: ASP is used offline for supervision generation, while the smaller recommender trained with our proposed CURE framework is used for online serving and continual updates. We provide the detailed cost and deployment breakdown in Appendix H.3.

[2]1: Very Weak – No connection, 2: Weak – Poorly justified, 3: Slightly Weak – Minimal relevance, 4: Neutral – Marginally related, 5: Slightly Strong – Generally aligned, 6: Strong – Well aligned and logical, 7: Very Strong – Highly coherent and contextually perfect.

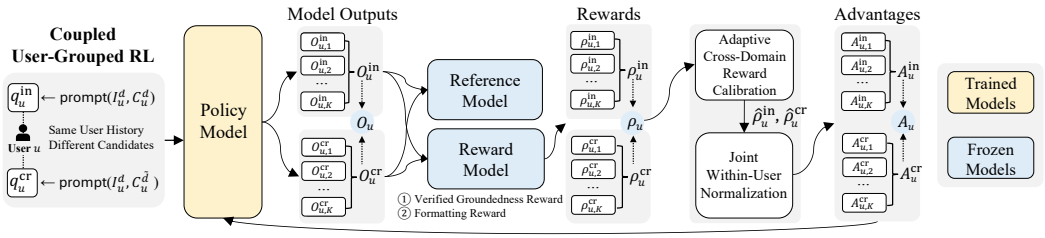


Figure 3: Overview of Coupled User-Grouped Reinforcement Learning (CURE)

3.2 Coupled User-Grouped Reinforcement Learning

In recommendation tasks, reinforcement-learning-based alignment is typically applied after supervised fine-tuning (SFT) on recommendation-specific training data. Using the previously constructed dataset of **in-domain sample** $\mathcal{S}^d = \{(I^d, C^d, i^d_+, r^d_+)\}$, we created instruction data for this task by prompting the model to generate both the rationale r^d_+ and the next item i^d_+, given the user history I^d. Details of SFT are provided in Appendix G.

Cross-domain samples generated by ASP should be aligned with the corresponding in-domain ground-truth signals from the same user, enabling the transfer of useful cross-domain information while stabilizing learning via reliable in-domain signals in the non-overlapping user scenario. Reinforcement Learning with Verifiable Rewards (RLVR) is well suited for enforcing this alignment, as it relies solely on reward-based feedback and iteratively refines outputs to satisfy in- and cross-domain consistency constraints [6, 7]. Building on this, as shown in Fig. 3, we propose coupled user-grouped reinforcement learning (CURE), an RLVR-based method implemented with Group Relative Policy Optimization (GRPO), which anchors each update on a shared user history and aggregates in-domain ground-truth and cross-domain ASP-generated pseudo item rewards within each user group.

3.2.1 Training Process

Using the in- and cross-domain samples, we construct two prompts that share the same anchor history for a given user u: $q^{\text{in}}_u = \texttt{prompt}(I^d_u, C^d_u), q^{\text{cr}}_u = \texttt{prompt}(I^d_u, C^{\tilde{d}}_u)$. For each prompt associated with user u, GRPO samples a group of K outputs from the old policy $\pi_{\theta_{\text{old}}}$: $O^{\text{in}}_u = \{o^{\text{in}}_{u,1}, \ldots, o^{\text{in}}_{u,K}\}, O^{\text{cr}}_u = \{o^{\text{cr}}_{u,1}, \ldots, o^{\text{cr}}_{u,K}\}$. Each output $o_{u,k} = (\hat{r}_{u,k}, \hat{i}_{u,k})$ consists of a generated rationale $\hat{r}_{u,k}$ and a recommended item $\hat{i}_{u,k}$. Our key design is to merge in- and cross-domain outputs into a single **user-level group**, $O_u = O^{\text{in}}_u \cup O^{\text{cr}}_u$, such that all outputs derived from the same anchor history I^d_u are jointly optimized within the same mini-batch. We optimize the LLM parameters θ using a user-level coupled GRPO objective:

$$\mathcal{J}_{\text{coupled GRPO}}(\theta) = \frac{1}{|O_u|} \sum_{o_j \in O_u} \frac{\pi_\theta(o_{u,j} | q_u^{T(o_{u,j})})}{\pi_{\theta_{\text{old}}}(o_{u,j} | q_u^{T(o_{u,j})})} A_{u,j}^{T(o_{u,j})}, \tag{1}$$

where o_j denotes the j-th element in an arbitrary ordering of the pooled in- and cross-domain outputs for a specific user u, and $T(o_j) \in \{\text{in}, \text{cr}\}$ denotes the scenario type associated with o_j. For analytical convenience, Eq. 1 omits auxiliary components of vanilla GRPO, such as clipping.

To compute the advantage $A_{u,j}^{T(o_j)}$, we evaluate a reward $\rho_{u,j}$ for each sampled output $o_{u,j}$ in the user group, where the reward function is described in detail in Section 3.2.2. Since in-domain rewards are computed from observed ground-truth signals with high reliability, whereas cross-domain rewards are derived from noisier pseudo supervision, their reward distributions—and hence raw scales—can differ. To prevent this mismatch from biasing the shared within-user normalization, we apply a lightweight training-time calibration to cross-domain rewards.

Adaptive Cross-Domain Reward Calibration. For each user group, we compute empirical moments (i.e., mean and standard deviation) of in- and cross-domain rewards, $(\mu^{\text{in}}, \sigma^{\text{in}})$ and $(\mu^{\text{cr}}, \sigma^{\text{cr}})$, respectively. We then define adaptive calibration coefficients $a := \frac{\sigma^{\text{in}}}{\sigma^{\text{cr}} + \epsilon}, b := \mu^{\text{in}} - a\,\mu^{\text{cr}}$ and transform each cross-domain reward $\rho^{\text{cr}}_{u,j}$ into $\hat{\rho}^{\text{cr}}_{u,j} := a\,\rho^{\text{cr}}_{u,j} + b$, where in-domain rewards are kept unchanged, i.e., $\hat{\rho}^{\text{in}}_{u,j} := \rho^{\text{in}}_{u,j}$. This transformation aligns the first two moments of cross-domain rewards to those of in-domain rewards, so that joint within-user normalization compares signals on a compatible scale.

Joint Within-User Normalization. We then normalize the calibrated rewards within each user group, following GRPO: $A_{u,j} := \frac{\hat{\rho}_{u,j} - \text{mean}(\hat{\boldsymbol{\rho}}_u)}{\text{std}(\hat{\boldsymbol{\rho}}_u) + \varepsilon}$, $\hat{\boldsymbol{\rho}}_u := \hat{\boldsymbol{\rho}}^{\text{in}}_u \cup \hat{\boldsymbol{\rho}}^{\text{cr}}_u$. This user-grouped normalization

explicitly couples in- and cross-domain samples, anchoring cross-domain updates to reliable in-domain signals rather than treating them as an independent auxiliary task. Accordingly, we enforce this coupling during training by grouping in- and cross-domain samples from the same user. This design induces explicit coupling, since when in- and cross-domain samples are standardized together via $\mu_u = \mathrm{mean}(\hat{\boldsymbol{\rho}}_u)$ and $\sigma_u = \mathrm{std}(\hat{\boldsymbol{\rho}}_u)$, the standardized advantage $A_{u,j}$ of any sample depends on all rewards in the same user group. For $|O_u| \geq 2$, $\sigma_u > 0$ and $\varepsilon = 0$, the Jacobian entry (i.e., the partial derivative) of $A_{u,j}$ with respect to the calibrated reward $\hat{\rho}_{u,k}$ is

$$\frac{\partial A_{u,j}}{\partial \hat{\rho}_{u,k}} = \frac{\mathbb{I}[j=k] - \frac{1}{|O_u|}}{\sigma_u} - \frac{(\hat{\rho}_{u,j} - \mu_u)(\hat{\rho}_{u,k} - \mu_u)}{|O_u|\,\sigma_u^3}, \tag{2}$$

where $\mathbb{I}[j = k]$ is simply an indicator: it equals 1 if k is the same sample as j, and 0 otherwise. Eq. (2) formalizes that, even without any explicit pairwise operation at the dataset or sample level (e.g., concatenation or direct cross-task pairing) between in- and cross-domain tasks, CURE creates a relational training signal: changing cross-domain rewards changes in-domain advantages, hence changes the effective in-domain gradient coefficients in Eq. (1). We train the model with two custom-designed reward functions tailored to recommendation: (1) Verified Groundedness Reward and (2) Formatting Reward.

3.2.2 Reward Design

① **Verified Groundedness Reward** (ρ_u^{vg}): The j-th output $o_{u,j} \in O_u$ contains a recommended item $\hat{i}_{u,j}$ and a rationale $\hat{r}_{u,j}$. We compute an item-matching indicator: $\mathbb{I}\big[\hat{i}_{u,j} = i_+\big]$, where i_+ denotes the available supervision: the ground-truth next item i_+^d for in-domain samples, or the ASP pseudo-label $i_+^{\tilde{d}}$ for cross-domain samples. This design anchors learning to verifiable supervision, with rewards defined as a binary indicator of exact item matching—using ground-truth targets for in-domain samples and pseudo-labeled targets for cross-domain samples.

② **Formatting Reward** (ρ_u^{fmt}): This reward assigns a binary signal indicating whether the generated output contains both a recommended item and its rationale in the required tagged format (i.e., `<item>` recommend item `</item>` `<think>` rationale `</think>`).

③ **Final Reward.** We define the final reward function as $\rho_{u,j} = \rho_{u,j}^{\text{vg}} + \rho_{u,j}^{\text{fmt}}$, where $\rho_{u,j}$ is further calibrated if $o_{u,j} \in O_u^{\text{cr}}$ before computing the joint normalized advantage.

4 Theoretical Analysis: In-Domain–Anchored Pseudo Cross-Domain Updates

While ASP provides valuable cross-domain supervision, pseudo labels are inherently imperfect in the non-overlapping user scenarios. To address this, our method computes GRPO advantages using user-specific mean and variance over both in- and cross-domain rewards. This coupled normalization contrasts with vanilla (uncoupled) GRPO, which normalizes advantages independently per sample using cross-domain rewards only. Our analysis isolates the common mean-shift component induced by noisy pseudo rewards and shows that user-level coupling attenuates this component.

4.1 A Simple Noise Model for Pseudo Rewards

Let $n_{\text{in}} := |O_u^{\text{in}}|$, $n_{\text{cr}} := |O_u^{\text{cr}}|$, and $N := n_{\text{in}} + n_{\text{cr}}$. We denote the set of in-domain outputs for user u as $O_u^{\text{in}} := \{o_{u,1}^{\text{in}}, \ldots, o_{u,n_{\text{in}}}^{\text{in}}\}$, and the set of cross-domain outputs as $O_u^{\text{cr}} := \{o_{u,1}^{\text{cr}}, \ldots, o_{u,n_{\text{cr}}}^{\text{cr}}\}$. We introduce an unobserved *oracle reward* $(\rho_k)^\star$, which represents the ideal reward value aligned with the ground-truth labels and is used solely for theoretical analysis. For in-domain outputs, rewards are computed from ground-truth labels and thus coincide with the oracle reward: $\rho_{u,k}^{\text{in}} = (\rho_{u,k}^{\text{in}})^\star,\ k \in \{1, \ldots, n_{\text{in}}\}$. For cross-domain outputs, rewards rely on ASP-generated pseudo supervision and are inherently noisy, since the pseudo labels are imperfect proxies of the oracle reward: $\rho_{u,j}^{\text{cr}} = (\rho_{u,j}^{\text{cr}})^\star + \varepsilon_j,\ j \in \{1, \ldots, n_{\text{cr}}\}$, where $\mathbb{E}[\varepsilon_j] = 0$ and $\mathrm{Var}(\varepsilon_j) \leq \sigma_\varepsilon^2$.

4.2 Coupled vs. Uncoupled (Vanilla GRPO) Gradient

The GRPO objective yields the following gradients:

$$\nabla_\theta \mathcal{J}_{\text{GRPO}} = \tfrac{1}{K} \textstyle\sum_{k=1}^{K} \frac{\pi_\theta(o_k|q)}{\pi_{\theta_{\text{old}}}(o_k|q)} A_k \cdot \nabla_\theta \log \pi_\theta(o_k \mid q). \tag{3}$$

The gradient term $\nabla_\theta \log \pi_\theta(o_k \mid q)$ adjusts the likelihood of the output o_k; the sign of A_k determines the update direction, while its magnitude scales the size of the update.

Uncoupled Cross-Domain Statistics (vanilla GRPO). Define the cross-domain-only mean over cross-domain rewards: $\mu_u^{\text{cr}} := \frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{\text{cr}}}\rho_{u,j}^{\text{cr}}$, $(\mu_u^{\text{cr}})^\star := \frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{\text{cr}}}(\rho_{u,j}^{\text{cr}})^\star$. The uncoupled gradient of GRPO uses $A_{u,j}^{\text{cr}} = (\rho_{u,j}^{\text{cr}} - \mu_u^{\text{cr}})/\sigma_u^{\text{cr}}$ and can be described as $g_{\text{unc}}(q_u^{\text{cr}}) := \mathbb{E}\left[\frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{\text{cr}}} A_{u,j}^{\text{cr}}\,\Delta_{u,j}^{\text{cr}}\right]$, where $\Delta_{u,j}^{\text{cr}} = \nabla_\theta \log \pi_\theta(o_{u,j}^{\text{cr}}|q_u^{\text{cr}})$. The importance ratio $\frac{\pi_\theta(o_{u,j}|q_u^{\text{cr}})}{\pi_{\theta_{\text{old}}}(o_{u,j}|q_u^{\text{cr}})}$ is omitted (Eq. 3), as it does not affect the directional analysis of the gradient and is unnecessary for the stability and variance arguments considered.

Coupled User-Grouped Statistics. Define user-grouped mean over in- and cross-domain rewards: $\mu_u^{\text{in+cr}} := \frac{1}{N}\sum_{k=1}^{N}\rho_{u,k}$, $(\mu_u^{\text{in+cr}})^\star := \frac{1}{N}\sum_{k=1}^{N}\rho_{u,k}^\star$, where $\rho_{u,k}^\star = \rho_{u,k}$ for $k \in \{1,...,n_{\text{in}}\}$ and $\rho_{u,j}^\star = (\rho_{u,j}^{\text{cr}})^\star$ for $j \in \{1,...,n_{\text{cr}}\}$. In addition, $\sigma_u^{\text{in+cr}}$ denotes the user-grouped standard deviation of in- and cross-domain rewards. CURE employs the joint-normalized advantage of the cross-domain sample $A_{u,j} := (\rho_{u,j}^{\text{cr}} - \mu_u^{\text{in+cr}})/\sigma_u^{\text{in+cr}}$ and its gradient term is described as $g_{\text{cpl}}(q_u^{\text{cr}}) := \mathbb{E}\left[\frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{\text{cr}}} A_{u,j}\Delta_{u,j}^{\text{cr}}\right]$.

4.3 Anchoring Effect: Mean-Noise Attenuation

In our analysis, the *mean shift* refers to the deviation between the empirical normalization mean μ computed from noisy pseudo rewards and its oracle counterpart $\mu^\star$ (i.e., $\mu - \mu^\star$). This deviation induces a common bias term in all normalized advantages:

$$A_j = \frac{\rho_j - \mu}{\sigma} = \frac{\rho_j - \mu^\star}{\sigma} - \underbrace{\frac{\mu - \mu^\star}{\sigma}}_{\text{mean-shift}}, \tag{4}$$

The mean-shift term $(\mu - \mu^\star)$ is the portion of the normalized advantage that arises solely from replacing the oracle mean by the noisy mean. As a result, noise in pseudo rewards can systematically bias policy gradients and destabilize updates. Under the noise model in Section 4.1, we can describe the uncoupled and coupled mean-shifts as $b_u^{\text{unc}} := \mu_u^{\text{cr}} - (\mu_u^{\text{cr}})^\star = \frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{\text{cr}}}\varepsilon_j$, and $b_u^{\text{cpl}} = \mu_u^{\text{in+cr}} - (\mu_u^{\text{in+cr}})^\star = \frac{1}{N}\sum_{j=1}^{n_{\text{cr}}}\varepsilon_j$, respectively. Thus, coupling reduces the mean-shift noise by averaging the same cross-domain noise over a larger set ($N \geq n_{\text{cr}}$).

Lemma 1 (Mean-noise attenuation via coupling)**.** *Assume* $\{\varepsilon_j\}_{j\in\{1,\ldots,n_{cr}\}}$ *are independent with* $\mathbb{E}[\varepsilon_j] = 0$ *and* $\text{Var}(\varepsilon_j) \leq \sigma_\varepsilon^2$. *Then* $\text{Var}\big(b_u^{\text{cpl}}\big) = \left(\frac{n_{\text{cr}}}{N}\right)^2 \text{Var}\big(b_u^{\text{unc}}\big)$.

Lemma 1 shows that when we normalize in- and cross-domain rewards jointly within each user, the *variance* of the mean-shift induced by cross-domain pseudo rewards is scaled down by a factor of $(n_{\text{cr}}/N)^2$ compared to the uncoupled vanilla GRPO case.

Lemma 2 (Pooled standard deviation lower bound)**.** *Let* σ_u^{cr} *and* $\sigma_u^{\text{in+cr}}$ *denote the population standard deviations of the cross-domain rewards and the combined in-domain/cross-domain rewards for user* u*, respectively. Define* $\alpha = \frac{n_{\text{cr}}}{N}$*, where* n_{cr} *and* N *denote the number of cross-domain rewards and total rewards, respectively. Then* $\left(\sigma_u^{\text{in+cr}}\right)^2 \geq \alpha\left(\sigma_u^{\text{cr}}\right)^2$.

4.4 Stability: Reduced Update Perturbation

Theorem 1 (Coupling reduces sensitivity to pseudo-reward mean noise)**.** *Consider the advantage portion of the cross-domain policy update induced solely by the mean-shift term:*

$$U(b,\sigma) = \frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{cr}}\frac{\mu - \mu^\star}{\sigma}\Delta_j^{\text{cr}} = \frac{1}{n_{\text{cr}}}\sum_{j=1}^{n_{cr}}\frac{b}{\sigma}\Delta_j^{\text{cr}}.$$

Under the noise model in Section 4.1, the definitions of b_u^{unc} *and* b_u^{cpl} *imply* $b_u^{\text{cpl}} = \alpha b_u^{\text{unc}}$*, which is the deterministic relation underlying Lemma 1. Moreover, Lemma 2 gives* $\left(\sigma_u^{\text{in+cr}}\right)^2 \geq \alpha\left(\sigma_u^{\text{cr}}\right)^2$. *Therefore, the coupled normalization attenuates the mean-shift–induced update perturbation:*

$$\mathbb{E}\left[\left\|U(b_u^{\text{cpl}},\sigma_u^{\text{in+cr}})\right\|_2^2\right] \leq \alpha\mathbb{E}\left[\left\|U(b_u^{\text{unc}},\sigma_u^{\text{cr}})\right\|_2^2\right] \leq \mathbb{E}\left[\left\|U(b_u^{\text{unc}},\sigma_u^{\text{cr}})\right\|_2^2\right].$$

Theorem 1 conveys the following message. With user-level coupling, the normalization statistics are anchored to noise-free in-domain rewards, which reduces the mean shift induced by noisy

cross-domain rewards. Moreover, the pooled standard deviation cannot collapse relative to the cross-domain-only standard deviation by more than the factor characterized in Lemma 2. As a result, the mean-shift–induced update perturbation is attenuated under coupled normalization. Full proofs are deferred to Appendix K.3.

4.5 Bridge to Experiments

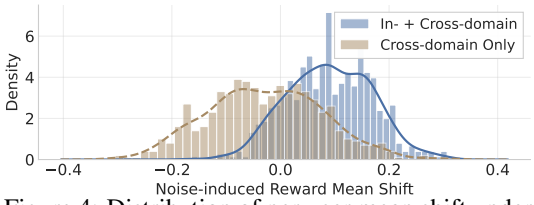


Figure 4: Distribution of per-user mean shift under controlled Gaussian perturbations of ASP-based cross-domain rewards.

Since oracle cross-domain rewards are unobservable in the non-overlapping setting, we conduct a controlled perturbation study. Specifically, we treat the original ASP-based cross-domain rewards as reference rewards and inject Gaussian noise $\mathcal{N}(0,1)$ into them, while keeping in-domain rewards unchanged. For each user, we measure the mean shift under (i) uncoupled cross-domain-only normalization (i.e., vanilla GRPO) and (ii) coupled in- and cross-domain normalization, and visualize their user-wise distributions. As shown in Fig. 4, coupling yields a smaller mean-shift–induced advantage perturbation than uncoupled training, reflecting a consistently reduced variance of the mean shift. This indicates that noise-free in-domain rewards anchor user-level normalization and suppress the propagation of pseudo-reward noise.

5 Empirical Analysis

In this section, we address the following research questions: **RQ1** (Cross-Domain Performance). Does our method achieve superior cross-domain recommendation performance over baselines under non-overlapping user scenario? **RQ2** (In-Domain Robustness). Does our model improve in-domain recommendation performance? **RQ3** (Ablation Study). Which design choices within CURE contribute most to the performance gains? **RQ4** (Online Impact). Does the proposed model improve CTR and CVR compared to production baselines in an online A/B test? **RQ5** (Pseudo-Label Reliability). How well do ASP pseudo-labels align with the held-out target-domain ground truth under the source-to-target evaluation protocol?

5.1 Datasets

We evaluate on five Amazon Reviews [8] domains—*Amazon* ***Fashion***, ***Clothing*** *Shoes and Jewelry*, ***Grocery*** *and Gourmet Food*, ***Health*** *and Household*, and ***CDs*** *and Vinyl*—and MovieLens 32M [9] (***ml-32m***), a large-scale movie recommendation benchmark; detailed dataset statistics are in Appendix A. During ASP-based data generation, we construct semantically related domain pairs—such as Fashion ↔ Clothing, Grocery ↔ Health, and CDs ↔ ml-32m—and also consider their reversed directions. Overall, this results in six in-domain (e.g., Fashion ↔ Fashion) and six cross-domain settings (e.g., Fashion ↔ Clothing), whose samples are jointly aggregated and used to train the model.

5.2 Baselines and Evaluation Settings

We compare our approach with two major classes of baselines: collaborative-filtering-based methods (**CF-Rec**) and LLM-based recommendation methods. For the backbone model, we use `Gemma-3-4B-it` [10], considering the efficiency and resource requirements of practical deployment. Recommendation accuracy is measured under a *leave-one-out* evaluation protocol, where each test candidate set contains the held-out next item along with items that the user has not previously interacted with. We report standard top-K ranking metrics, including *Hit Ratio* (**HR**) and *Normalized Discounted Cumulative Gain* (**NDCG**). Additional implementation details are provided in Appendix B to support reproducibility.

5.3 Cross-Domain Scenario (R1)

Table 1: **Cross-Domain** Recommendation Results

Model / Domain Pair	Metric	Rec-SAVER	Rec-R1	Intuitor	SumRec	S-DPO	CAPO	VUG	ALCDR	TCPLP	CURE
Fashion → Clothing	HR@1	4.40	6.73	7.18	4.53	4.79	5.82	6.80	7.80	6.40	**9.83**
	HR@5	17.08	19.34	15.65	22.12	17.66	18.11	19.10	**25.00**	21.50	22.12
	N@5	10.88	13.16	11.65	13.60	11.37	12.00	13.10	16.50	15.00	**16.55**
Clothing → Fashion	HR@1	4.40	4.51	3.59	5.73	6.37	**7.70**	4.60	4.90	6.10	6.42
	HR@5	17.36	17.88	9.26	18.98	18.29	17.88	17.70	18.80	20.10	**20.25**
	N@5	11.25	11.55	6.74	12.48	**15.79**	13.17	11.50	12.00	12.50	13.21
Grocery → Health	HR@1	2.58	4.72	3.77	3.02	2.96	3.96	4.80	2.30	4.20	**5.03**
	HR@5	11.64	12.77	10.31	11.64	11.76	12.52	12.90	11.60	11.20	**13.58**
	N@5	7.09	8.88	7.32	7.36	7.61	8.52	**8.90**	6.80	7.60	8.08
Health → Grocery	HR@1	4.21	4.11	7.24	4.93	3.59	4.52	4.00	4.20	5.80	**7.65**
	HR@5	14.37	14.58	15.55	15.04	15.66	15.50	14.70	14.40	12.50	**16.79**
	N@5	9.49	9.23	11.69	10.25	9.70	10.23	9.30	9.50	11.60	**12.55**

We evaluate the cross-domain scenario, where a user history from a source domain is given and the goal is to identify the ground-truth next item from a candidate set in a different target domain. To construct a strictly non-overlapping cross-domain evaluation setting, we select users whose interaction history appears only in the source domain and whose final interaction belongs to the target domain. We then use the source-domain

Table 2: **In-domain** performance on ***Amazon***. Best and second-best results are shown in **bold** and underlined, respectively.

Domain		Fashion			Clothing			Grocery			Health			CDs			MovieLens		
Types	**Models**	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5
CF Rec	SASRec	2.84	14.49	8.53	3.37	16.74	9.89	4.22	18.39	11.20	2.67	13.88	8.15	3.26	16.34	9.61	2.76	10.24	6.49
	BERT4Rec	1.59	5.11	3.39	2.19	12.14	7.02	4.27	18.97	11.47	2.45	12.85	7.53	5.34	22.16	13.65	2.26	12.84	7.38
	NextItNet	1.76	6.25	4.03	3.35	13.75	8.45	2.89	15.00	8.97	2.70	13.36	7.94	3.80	16.20	9.88	2.25	10.41	6.24
	SINE	2.30	9.13	5.69	4.71	19.14	11.89	3.08	15.11	8.96	2.92	14.19	8.43	6.03	26.91	16.36	1.93	11.60	6.61
	GRU4Rec	3.34	13.66	8.46	3.12	13.33	8.12	3.82	17.13	10.28	3.88	17.84	10.84	4.29	24.00	13.73	2.65	13.07	7.75
	FOSSIL	2.22	7.27	4.73	3.28	13.78	8.43	3.08	15.02	8.87	2.92	15.36	9.00	4.03	17.61	10.71	1.89	11.11	6.33
	STAMP	2.84	23.51	12.76	3.45	13.89	8.57	3.69	16.80	10.11	2.91	19.55	10.94	6.24	31.23	18.73	1.94	9.55	5.65
LLM Rec	RecSAVER	12.72	39.60	26.92	8.09	26.82	17.70	10.03	27.92	19.48	4.51	18.29	11.47	10.67	35.22	23.18	2.23	11.88	7.09
	SumRec	8.06	28.65	18.55	3.97	18.74	11.82	3.84	15.19	9.89	3.27	15.10	9.31	6.00	27.79	17.07	0.79	7.46	3.94
	S-DPO	5.22	23.78	14.38	4.16	19.37	11.75	6.57	18.47	12.91	2.76	15.71	9.28	6.70	29.66	18.28	1.63	8.02	4.76
	CAPO	9.47	28.22	18.87	7.11	23.12	15.09	6.90	19.47	13.62	4.83	18.21	11.73	10.08	28.44	19.66	2.81	13.13	8.10
	Rec-R1	10.16	34.88	22.69	8.42	**32.47**	20.74	7.16	19.96	14.23	2.92	9.69	6.45	8.71	32.59	21.03	1.84	9.73	5.75
	Intuitor	10.76	39.65	25.11	7.43	29.01	18.40	4.24	29.93	17.17	6.72	19.18	13.11	6.48	**37.04**	21.65	2.01	8.49	5.31
	TWiCERec	12.16	30.49	21.71	5.62	21.88	14.06	5.65	17.95	11.94	3.90	16.64	10.00	6.10	23.82	15.00	1.57	7.49	4.41
	VUG	9.70	28.60	19.70	8.60	19.90	14.90	7.10	20.10	14.10	3.00	9.80	6.60	8.80	32.40	21.10	1.90	9.80	5.90
	ALCDR	8.00	29.40	19.50	4.70	20.20	13.00	8.10	26.50	18.00	5.20	15.10	10.30	12.20	31.40	22.80	2.30	11.90	7.10
	TCPLP	9.40	33.80	22.00	5.40	24.50	15.00	10.90	**31.50**	**21.40**	4.80	18.40	11.60	10.90	35.60	23.80	2.70	13.00	8.00
	CURE	**16.75**	**43.12**	**30.90**	**9.47**	30.97	**20.80**	**11.02**	24.54	18.29	**6.86**	**20.91**	**14.22**	**17.51**	36.79	**27.73**	**3.17**	**13.36**	**8.19**

Table 3: Ablation study on grouping and calibration strategies in **cross-domain** recommendation. The top two methods are highlighted in **bold** and underline, respectively.

Domains / Types	Fashion → Clothing		Clothing → Fashion		Grocery → Health		Health → Grocery	
	HR@1	NDCG@5	HR@1	NDCG@5	HR@1	NDCG@5	HR@1	NDCG@5
(A) Base	3.33	5.62	2.43	4.17	2.45	4.93	2.31	4.89
(B) Only SFT	6.34	14.70	3.70	10.82	4.59	**8.99**	6.93	10.89
(C) GRPO with In-Domain	8.41	14.41	4.11	10.51	4.40	6.63	5.34	10.22
(D) Ungrouped+No Calibration	6.73	13.16	4.51	11.55	3.71	6.13	4.11	9.23
(E) Grouped+No Calibration	7.37	16.51	6.31	10.15	4.53	6.45	6.52	10.40
(F) Grouped+Calibration	**9.83**	**16.55**	**6.42**	**13.21**	**5.03**	8.08	**7.65**	**12.55**

history as input and the last target-domain item as the cross-domain ground truth for evaluation (details in Appendix A)[3]. Notably, even in Amazon Reviews, overlapping users are extremely sparse ($< 1\%$), and this evaluation is only possible due to the scale of the public dataset; in real-world industrial settings, cross-domain ground truth is almost entirely unavailable. As shown in Table 1, **our method shows generally strong performance, outperforming most baselines across the majority of domain pairs.** This setting directly corresponds to cross-sell applications, where histories in one service domain are leveraged to recommend items in another domain to drive additional revenue.

5.4 In-Domain Scenario (R2)

In the in-domain scenario, we are given a user history from a source domain and aim to identify the ground-truth next item from a candidate set in the same target domain. As shown in Table 2, **We observe generally consistent in-domain performance gains across most domains compared to competitive CF-Rec and LLM-Rec baselines**. While our primary objective is to improve cross-domain recommendation under the non-overlapping user scenario, the results also show that our model remains highly competitive in in-domain recommendation, demonstrating its general applicability. We further verify backbone robustness by replacing the main backbone with QWEN-4B-INSTRUCT-2507 [11], where CURE remains competitive across both in- and cross-domain settings; detailed results are reported in Section 5.8. We also evaluate rationale quality using an independent LLM judge over 600 samples, where CURE achieves the highest proportion of high-quality rationales; details are provided in Appendix E.

5.5 Ablation Study (R3)

As shown in Fig. 5 and Table 3, we conduct ablation studies to isolate the contribution of CURE in both in-domain and cross-domain settings. We compare six variants: (A) **Base** (`Gemma-3-4B-it`); (B) **+SFT** (Base fine-tuned with SFT on in-domain data); (C) **+SFT+GRPO (in-domain only)** (GRPO applied using only in-domain samples); (D) **+SFT+GRPO (in+cross, ungrouped, no calibration)**

[3]We exclude MovieLens from this evaluation because it comes from a separate data source with no user overlap with Amazon, making such cross-domain ground-truth construction infeasible.

(GRPO using both in-domain and ASP-generated cross-domain samples without user-level grouping); (E) **+SFT+GRPO (in+cross, grouped, no calibration)** (GRPO with user-level grouping but without adaptive reward calibration); and (F) **Ours (in+cross, grouped, calibration)**.

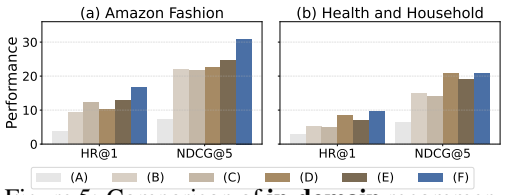


Figure 5: Comparison of **in-domain** recommendation performance across model variants

(1) Recommendation-oriented adaptation is essential. The Base model (A) is consistently outperformed by all recommendation-adapted variants, demonstrating that additional training aligned with recommendation objectives is necessary beyond general-purpose instruction tuning.

(2) Reward-based RL generally provides additional gains beyond SFT. Variants with GRPO (C–F) generally improve over SFT-only training (B), although the gains are not uniform across all cases. This suggests that directly optimizing recommendation rewards can complement SFT: while SFT improves format adherence, GRPO further encourages outputs that match the target item, often leading to better top-k recommendation performance.

(3) Cross-domain pseudo supervision yields limited and inconsistent gains under naive mixing. Adding ASP-generated cross-domain samples (D) yields limited and inconsistent gains over single-domain-only RL (C), with C sometimes performing better in both in- and cross-domain settings.

(4) User-level grouping is beneficial. User-level grouping (E) generally improves over the ungrouped variant (D) in both in- and cross-domain settings. This trend is consistent with our theoretical analysis, which shows that user-level coupling anchors noisy cross-domain updates to reliable in-domain signals, thereby attenuating the effect of pseudo-reward noise and stabilizing optimization.

(5) Joint calibration and normalization complete CURE. By further incorporating adaptive reward calibration and joint within-user normalization, our full method (F) achieves the largest and most consistent gains. This confirms that CURE's effectiveness stems not only from grouping samples by user, but from explicitly calibrating cross-domain rewards against reliable in-domain signals.

We further provide an extended ablation study on ASP submodules in Appendix D. Across the reported settings, adding the Meta-Judge improves the average HR@1 from 9.8 to 16.8 (Amazon Fashion), suggesting that it helps improve the reliability of ASP-generated supervision.

5.6 Online A/B Test (R4)

Table 4: Online A/B Test Results

Model	# Impression	# Clicks	CTR	CVR	Latency
Popularity	243.95K	22.86K	9.37%	1.55%	94ms/call
ML model	177.44K	16.03K	9.04%	1.93%	120ms/call
GRPO	9.55K	1.23K	12.88%	1.59%	132ms/call
Ours	112.77K	15.86K	14.07%	2.90%	138ms/call

We deployed our model in a real-world production environment and conducted an online A/B test from August to December 2025. As baselines, we considered a popularity-based control, a traditional machine learning (ML) model, and a vanilla GRPO-based model. Users were assigned via stable user-level hash randomization, with comparable populations and identical routing logic across models, while LLM-Rec traffic was limited due to current computational cost. As shown in Table 4, our model achieved a +9.24% CTR lift over the GRPO-based LLM-Rec. The CTR gain is statistically reliable under a one-sided two-proportion z-test ($Z = 3.20$, $p < 0.001$), and the 95% confidence interval for the CTR gap is [0.48%, 1.88%]. **Notably, this online test was conducted on real non-overlapping users across diverse service domains.** These results demonstrate that our model delivers robust gains under real production constraints, supporting its suitability for real-world deployment. Further details are provided in Appendix I.

5.7 Study on Pseudo-Label Reliability of ASP (R5)

We evaluate ASP-generated cross-domain pseudo-labels using the same source-to-target protocol as in Section 5.3. ASP receives only source-domain user histories and target-domain candidates, and is considered correct if its final output after generation, scoring, and meta-judging matches the held-out target item. ASP achieves meaningful HR@1 scores under this challenging cross-domain setting: 25.00% for Fashion→Clothing, 14.88% for Clothing→Fashion, 17.65% for Grocery→Health, and 31.43% for Health→Grocery. These results indicate that ASP provides reliable cross-domain pseudo-labels, supporting its use as a teacher for CURE; Table 1 reports the same-protocol comparison.

Table 5: Cross-domain backbone robustness results. For each domain pair and backbone, the best result among the compared methods is highlighted in **bold**.

Cross-Domain Pair	Backbone	Rec-SAVER		Rec-R1		CURE	
		HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
Fashion → Clothing	Qwen	21.9	16.9	22.5	14.1	**27.9**	**17.4**
Fashion → Clothing	Gemma	17.1	10.9	19.3	13.2	**22.1**	**16.6**
Clothing → Fashion	Qwen	17.0	12.4	17.2	12.0	**20.6**	**12.5**
Clothing → Fashion	Gemma	17.4	11.3	17.9	11.6	**20.3**	**13.2**
Grocery → Health	Qwen	14.7	10.1	16.2	10.8	**21.9**	**12.2**
Grocery → Health	Gemma	11.6	7.1	12.8	8.9	**13.6**	**8.9**
Health → Grocery	Qwen	22.5	16.1	21.6	16.0	**26.9**	**17.1**
Health → Grocery	Gemma	14.4	9.5	14.6	9.2	**16.8**	**12.6**

Table 6: In-domain backbone robustness results. For each dataset and backbone, the best result among the compared methods is highlighted in **bold**.

In-Domain Dataset	Backbone	Rec-SAVER		Rec-R1		CURE	
		HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
Fashion	Qwen	31.4	23.0	33.8	24.5	**39.5**	**26.3**
Fashion	Gemma	39.6	26.9	34.9	22.7	**43.1**	**30.9**
Grocery	Qwen	**23.1**	**16.5**	23.0	**16.5**	**23.1**	16.3
Grocery	Gemma	**27.9**	**19.5**	25.9	18.6	24.5	18.3
Clothing	Qwen	19.5	13.7	19.8	14.4	**26.4**	**18.1**
Clothing	Gemma	26.8	17.7	**32.5**	20.7	31.0	**20.8**
Health	Qwen	18.4	12.3	18.5	13.3	**23.5**	**14.1**
Health	Gemma	18.3	11.5	9.7	6.5	**20.9**	**14.2**
CDs	Qwen	34.1	25.0	35.1	**27.0**	**39.3**	24.2
CDs	Gemma	35.2	23.2	32.6	21.0	**36.8**	**27.7**
MovieLens	Qwen	15.4	9.8	13.3	8.0	**15.7**	**9.8**
MovieLens	Gemma	11.9	7.1	9.7	5.8	**13.4**	**8.2**

5.8 Backbone Robustness Analysis

To examine whether the performance gains of CURE are tied to a single backbone, we conduct an additional backbone robustness study using QWEN-4B-INSTRUCT-2507 [11] in addition to the GEMMA-3-4B-IT [10] backbone used in the main experiments. Across both cross-domain and in-domain settings, CURE remains consistently competitive and often outperforms Rec-SAVER and Rec-R1 under both backbone choices. This suggests that the effectiveness of CURE is not limited to a specific LLM backbone.

5.9 Robustness under Sparse-Transfer and Cold-Start-like Scenarios

Table 7: Dataset Statistics

Dataset	Domain	#Users	#Items
Amazon Review	All Beauty	8,367	3,049
	Musical Instruments	10,846	4,213

We clarify that CURE's stability mainly comes from reward normalization over multiple rollouts per input instance, rather than from relying on many historical interactions per user. Thus, although pure cold-start recommendation is outside our primary scope, the method is less sensitive to limited user history than approaches that depend heavily on dense user profiles. To further evaluate this point, we conduct additional sparse-transfer stress tests under a strictly user-disjoint setting. Specifically, we evaluate zero-shot transfer on unseen target domains summarized in Table 7: *All Beauty*, using models trained on Fashion/Clothing domains, and *Musical Instruments*, using models trained on CDs/MovieLens domains. As shown in Table 8, CURE improves over strong LLM-based recommenders in nearly all cases, demonstrating robust transferability even without user overlap.

Table 8: Sparse-transfer stress test on unseen target domains under a strictly user-disjoint setting. The best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Rec-SAVER			Rec-R1			CURE		
	HR@1	HR@5	NDCG@5	HR@1	HR@5	NDCG@5	HR@1	HR@5	NDCG@5
All Beauty	17.30	40.98	30.35	<u>18.94</u>	<u>46.99</u>	<u>33.65</u>	**25.87**	**51.18**	**39.30**
Musical Instruments	4.83	<u>15.17</u>	<u>10.33</u>	**5.89**	13.59	<u>10.33</u>	<u>5.70</u>	**16.42**	**11.39**

6 Conclusion

We address cross-domain recommendation in the non-overlapping user setting by generating cross-domain pseudo supervision and training an LLM recommender via coupled user-grouped RL. The resulting in- and cross-domain coupling delivers consistent gains on Amazon and MovieLens, with a significant CTR lift in an online A/B test.

Acknowledgments and Disclosure of Funding

We would like to thank SK Telecom for providing the GPU cluster and associated research infrastructure that enabled this work.

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST)), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00555621).

References

[1] Chung Park, Taesan Kim, Hyungjun Yoon, Junui Hong, Yelim Yu, Mincheol Cho, Minsung Choi, and Jaegul Choo. Pacer and runner: Cooperative learning framework between single-and cross-domain sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2071–2080, 2024.

[2] Hao Liu, Lei Guo, Lei Zhu, Yongqiang Jiang, Min Gao, and Hongzhi Yin. Mcrpl: A pretrain, prompt, and fine-tune paradigm for non-overlapping many-to-one cross-domain recommendation. *ACM Transactions on Information Systems*, 42(4):1–24, 2024.

[3] Hyunsoo Kim, Jaewan Moon, Seongmin Park, and Jongwuk Lee. Mergerec: Model merging for data-isolated cross-domain sequential recommendation. *arXiv preprint arXiv:2601.01753*, 2026.

[4] Weixin Chen, Yuhan Zhao, Li Chen, and Weike Pan. Leave no one behind: Fairness-aware cross-domain recommender systems for non-overlapping users. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 226–236, 2025.

[5] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. Contrastive cross-domain sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 138–147, 2022.

[6] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[7] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[8] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

[9] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[10] Gemma Team. Gemma 3. 2025. URL `https://goo.gle/Gemma3Report`.

[11] Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[12] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[14] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. Recbole 2.0: Towards a more up-to-date recommendation library. In *CIKM*, pages 4722–4726. ACM, 2022.

[15] Alicia Y Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. Leveraging llm reasoning enhances personalized recommender systems. *arXiv preprint arXiv:2408.00802*, 2024.

[16] Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation. *Advances in Neural Information Processing Systems*, 37:27463–27489, 2024.

[17] Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

[18] Jiacheng Lin, Tian Wang, and Kun Qian. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289*, 2025.

[19] Manato Tajiri and Michimasa Inaba. Refining text generation for realistic conversational recommendation via direct preference optimization. *arXiv preprint arXiv:2508.19918*, 2025.

[20] Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. Beyond the overlapping users: Cross-domain recommendation via adaptive anchor link learning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1488–1497, 2023.

[21] Lei Guo, Chenlong Song, Feng Guo, Xiaohui Han, Xiaojun Chang, and Lei Zhu. Semantic-enhanced co-attention prompt learning for non-overlapping cross-domain recommendation. *ACM Transactions on Information Systems*, 43(5):1–27, 2025.

[22] Jiayi Liao, Xiangnan He, Ruobing Xie, Jiancan Wu, Yancheng Yuan, Xingwu Sun, Zhanhui Kang, and Xiang Wang. Rosepo: Aligning llm-based recommenders with human values. *arXiv preprint arXiv:2410.12519*, 2024.

[23] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL `https://arxiv.org/abs/1908.10084`.

Appendix

In this appendix, we provide additional implementation details, experimental analyses, and theoretical proofs to support our results. We first describe the datasets and evaluation settings (Section A), followed by implementation details and baseline methods (Sections B and C). We then present extended ablation studies and rationale quality evaluations (Sections D and E), along with a Jacobian derivation (Section F). Next, we detail the supervised fine-tuning procedure and the Agentic Synthetic Data Generation Pipeline (ASP), including its implementation, computational costs, and human evaluation (Sections G and H). Finally, we report the online A/B test results, computational complexity, and proofs supporting our theoretical analysis (Sections I to K). Below is the table of contents for the appendix.

A Dataset **14**

B Reproducibility and Implementation Details **16**

C Baselines and Prior Methods **17**

D Extended Ablation Study on ASP Submodules **18**

E Rationale Quality Evaluation **18**

F Derivation of Jacobian of A_j with respect to $\hat{\rho}_k$ **19**

G Supervised Fine-Tuning before CURE **20**

H Implementation Details of Agentic Synthetic Data Generation Pipeline (ASP) **21**

H.1 System Prompts for Each LLM in ASP 21

H.2 Implementation Details of ASP Tools 22

H.3 ASP Construction Cost and Deployment Efficiency 22

H.4 Human Evaluation Protocol 23

I Online A/B Test Details **23**

J Time Complexity **24**

K Proofs for Section 4 (Theoretical Analysis) **24**

K.1 Proof of Lemma 1 24

K.2 Proof of Lemma 2 25

K.3 Proof of Theorem 1 26

A Dataset

Training Dataset. We conduct experiments on five **Amazon Review 2023**[4] [8] domains and the **MovieLens** dataset[5] [9], which represent heterogeneous domains spanning e-commerce reviews and movie viewing behavior, as shown in Table 9. The Amazon Review dataset consists of domains with practical relevance to real-world applications, including (1) *Amazon* ***Fashion***, (2) ***Clothing*** *Shoes and Jewelry*, (3) ***Grocery*** *and Gourmet Food*, (4) ***Health*** *and Household*, and (5) ***CDs*** *and Vinyl*.

[4] https://amazon-reviews-2023.github.io

[5] https://grouplens.org/datasets/movielens

MovieLens 32M (***ml-32m***) is a standard large-scale movie recommendation benchmark. **We assume that these domains are operated as distinct platforms in practice**; for instance, *Fashion* can be viewed as an online apparel shopping service, whereas *Grocery* corresponds to an online grocery marketplace. **In the authors' company, for example, both a mobile device sales platform and an e-commerce platform coexist, yet leveraging customer information across such services typically requires explicit user consent at the time of sign-up, which is often difficult to obtain in practice.**

During Agentic Synthetic Data Generation Pipeline (ASP)-based data generation, we construct semantically related domain pairs (source ↔ target)—such as Fashion ↔ Clothing, Grocery ↔ Health, and CDs ↔ ml-32m—and also consider their reversed directions. Overall, as shown in Table 9, this results in six in-domain settings and six cross-domain settings (twelve in total), whose samples are jointly aggregated and used to sequentially train the model via SFT followed by CURE.

Specifically, SFT is performed using only in-domain samples, whereas CURE jointly optimizes both in-domain and cross-domain samples. As a result, for each user, the training data used in CURE necessarily contains paired in-domain and cross-domain samples constructed from the same user history, enabling explicit user-level coupling during optimization. Consequently, the in-domain and cross-domain settings share identical user-history statistics (as reported in Table 9), differing only in the domain from which the next item to be predicted is drawn.

Table 9: Statistics of Training Dataset from ASP. User and item statistics are computed from source-domain user histories and are therefore **identical** for in- and cross-domain cases; the two settings differ only in the target labels (ground-truth vs. pseudo-labels).

Dataset	Domain	#Users	#Items	Avg Length	Avg Item Purchase	Sparsity
In-Domain Case (Ground-Truth Label)	**Fashion → Fashion**	11,028	59,004	6.54	1.23	0.9994
	Clothing → Clothing	13,766	107,353	11.83	1.65	0.9994
	Grocery → Grocery	11,334	6,149	10.76	21.02	0.9926
	Health → Health	21,152	74,784	8.49	2.50	0.9994
	CDs → CDs	17,035	83,774	13.71	1.97	0.9993
	ml-32m → ml-32m	10,623	12,259	48.96	42.43	0.9960
Cross-Domain Case (ASP-Generated Pseudo Label)	**Fashion → Clothing**	11,028	59,004	6.54	1.23	0.9994
	Clothing → Fashion	13,766	107,353	11.83	1.65	0.9994
	Grocery → Health	11,334	6,149	10.76	21.02	0.9926
	Health → Grocery	21,152	74,784	8.49	2.50	0.9994
	CDs → ml-32m	17,035	83,774	13.71	1.97	0.9993
	ml-32m → CDs	10,623	12,259	48.96	42.43	0.9960

Cost of Agentic Synthetic Data Generation Pipeline (ASP). Our ASP pipeline incurs non-trivial costs from LLM API usage and GPU computation. It comprises three LLM modules—a *Generation*, a *Scorer*, and a *Meta-Judge*—and we intentionally instantiate them with different backbones (`GPT-4.1` for generation, `Gemma-3-27B-it` for scoring, and `GPT-4.1-mini` for meta-judging), forming a diversified reasoning ensemble. At an average LLM API cost of $0.022975 per sample, generating approximately 150K training samples incurred about $3.4K in total, which is manageable given that ASP is used only for one-time offline data construction. While ASP can be relatively expensive when relying on commercial LLM APIs, the resulting training data enables substantial performance gains over prior methods; in production deployments, such improvements can translate to revenue uplift that typically outweighs the one-time data generation cost. **Despite these costs, we publicly release the entire ASP-generated dataset to support and accelerate future research in non-overlapping cross-domain recommendation.** It consists of three splits: an SFT training set, a CURE training set, and a held-out test set for evaluation.

Dataset Link: https://drive.google.com/drive/folders/1uRyKuDbfqsOvr31d1UzJUoLm2SThp3eA?usp=drive_link

Cross-Domain Evaluation Dataset. To evaluate cross-domain recommendation performance under a strictly source-to-target transfer setting, we construct a dedicated evaluation dataset from users whose historical interactions appear only in the source domain, while their final interaction belongs

to the target domain. For each such user, we use the source-domain interaction history as the input and the final target-domain item as the ground-truth label. This design allows us to evaluate whether a model can recommend a target-domain item without observing any target-domain history in the input sequence.

Table 10: Statistics of Evaluation Datasets in Cross-Domain Scenarios (Source → Target)

Dataset	Domain	#Users	#Items	Avg Length	Avg Item Purchase	Sparsity
Amazon Review	**Fashion → Clothing**	1,546	12,459	8.56	6.67	0.9957
	Clothing → Fashion	1,728	11,278	6.95	6.63	0.9962
	Grocery → Health	1,590	9,854	7.23	6.75	0.9957
	Health → Grocery	1,948	11,860	7.13	6.78	0.9965

This setting reflects a practical non-overlapping transfer scenario: from the model's perspective, the user provides only source-domain behavioral evidence, and the target-domain item is observed solely as the evaluation label. In our Amazon Reviews data, such cross-domain transition cases are sparse, consistent with the observation that most users do not actively engage across multiple service domains (Fig. 1). The constructed evaluation set is used exclusively for performance assessment and is not included in training. We exclude MovieLens from this evaluation because it comes from a separate data source with no shared user trajectory with Amazon, making source-to-target ground-truth construction infeasible. In addition, our online A/B test in Section I further evaluates the deployed model in real service environments, where users are naturally non-overlapping across multiple domains and performance is measured from actual user feedback. This provides a more practical validation of CURE under real non-overlapping scenarios beyond the limited offline overlapping-user evaluation.

B Reproducibility and Implementation Details

All experiments were conducted on a cluster of four NVIDIA H100 GPUs. To ensure reproducibility, we make our implementation code publicly available through anonymous links.

Code Link: https://github.com/cpark88/CURE

Table 11: Training hyperparameters for our model

Hyperparameter	Value
Batch size	4
Epochs	2
Optimizer	Paged AdamW (8-bit)
Learning rate	5.0×10^{-5} (cosine schedule, min LR applied)
Warm-up ratio	0.05
Gradient accumulation steps	8
Max sequence length	12,000
Weight decay	0.01 (default)
Gradient clipping	1.0 (default)
LoRA rank (r)	8
LoRA α	16
LoRA dropout	0
LoRA target modules	v-, q-, o-, gate-, up-, down-projections
Precision	bf16
Gradient checkpointing	True
Parallelism	DeepSpeed ZeRO-2
Hardware	4 × NVIDIA H100 80GB GPUs

Hyperparameters Table 11 summarizes the hyperparameter configurations used throughout our experiments, including the learning rate, batch size, optimizer, warm-up ratio, gradient clipping threshold, weight decay, and training epochs. We adopt `Gemma-3-4B-it` [12] as the LLM backbone for all models. Training begins with supervised fine-tuning (SFT), where the model is optimized using only the in-domain samples. In contrast, CURE jointly leverages the in-domain samples item together with cross-domain samples during training. For CURE, we train the model for 2 epochs with a batch size of 4, employing the 8-bit paged AdamW optimizer and a cosine learning rate schedule with an initial learning rate of 5.0×10^{-5} that decays to a predefined minimum. We apply gradient accumulation with a factor of 8, set the warm-up ratio to 0.05, and allow a maximum input length of 12k tokens. Training is conducted using DeepSpeed ZeRO-2 with bf16 mixed precision and gradient checkpointing enabled. Parameter-efficient fine-tuning is performed via LoRA [13], with rank $r = 8$, scaling factor $\alpha = 16$, and zero dropout, applied to the q-, v-, o-, $gate$-, up-, and $down$-projection layers. The number of output generations per sample in GRPO is set to 16. Accordingly, within a single mini-batch, we obtain 32 outputs per user, corresponding to in-domain and cross-domain samples. We then compute the mean and standard deviation of the rewards across these 32 outputs. For SFT, we largely follow

the same hyperparameter settings as CURE (Table 11), except that the batch size is increased to 12. This SFT model serves as the initial checkpoint for all RL-based methods, including CURE and RL-based baselines such as Rec-R1. All baselines and our model use identical user contexts and candidate sets within each prompt.

Evaluation Protocol We adopt a standard leave-one-out evaluation strategy, where the final interaction in each user sequence is reserved for testing, the penultimate interaction for validation, and all earlier interactions for training. During evaluation, each inference prompt explicitly includes a candidate set consisting of the ground-truth target item and randomly sampled unseen negative items. Unless otherwise specified, we use 200 candidate items per request, consisting of the ground-truth item and unseen negative candidates. We include popularity-based unseen items as hard negatives (50%) to complement random sampling and better reflect realistic candidate competition. All baselines use the same candidate set construction. DPO-based baselines require explicit construction of negative samples and were therefore trained with additional negatives. GRPO-based baselines (e.g., Rec-R1) instead leverage all samples generated by ASP, consistent with the training setup used in our model.

C Baselines and Prior Methods

Collaborative filtering recommenders (CF-Rec) model user preferences based on co-occurrence patterns between users and items, leveraging historical interaction signals such as clicks, purchases, or ratings. In this paradigm, each item is typically treated as a discrete identifier, and the model learns latent representations by capturing interaction statistics at the ID level, without explicitly utilizing item semantics or content information. As a result, CF-Rec performs well when sufficient user–item interaction data are available within a domain. However, in non-overlapping settings where users or items do not share interactions across domains, such co-occurrence signals are absent, rendering CF-Rec inapplicable to scenarios like ours. **Accordingly, we include CF-Rec baselines for in-domain recommendation (Table 2), where collaborative signals are available, but exclude them from cross-domain evaluation (Table 1) due to the absence of shared interactions.**

For CF-Rec, we implemented all methods within the RecBole framework [14][6]. We summarize the LLM-based baselines used in our experiments and describe their implementations below.

■ **Rec-SAVER** [15] fine-tunes LLMs with paired recommended items and rationales, showing that rationale-supervised instruction tuning can improve recommendation quality in both zero-shot (i.e., non-overlapping cross-domain scenario) and fine-tuning regimes. We implemented this baseline from scratch.

■ **S-DPO** [16] proposes a ranking-centric objective for LLM recommendation by reformulating Direct Preference Optimization (DPO) with a softmax over multiple negative samples. This design better separates preferred items from negatives and emphasizes hard negatives during optimization. We implemented this baseline entirely from scratch.

■ **Intuitor** [17] is a GRPO-style reinforcement learning approach that uses the model's own prediction confidence (self-certainty) as an intrinsic training signal, thereby avoiding reliance on external reward models or additional labels. We implemented Intuitor following the official repository[7].

■ **TWiCERec** combines rationale-aware instruction tuning with collaborative filtering guidance, injecting CF-derived interaction patterns to address the limited collaborative exposure of LLM-based recommenders. Concretely, it uses confidence-weighted reinforcement learning where rewards depend on both next-item correctness and CF-model confidence, improving recommendation performance and rationale quality. This model is also trained in an LLM-based manner and showed competitive performance in our zero-shot cross-domain recommendation setting under the non-overlapping scenario. Our implementation follows the authors' official repository[8].

■ **Rec-R1** [18] applies GRPO to recommendation, where the reward is based on whether the generated item matches the ground-truth next item. We built this baseline using a public GRPO implementation[9].

[6] https://github.com/RUCAIBox/RecBole
[7] https://github.com/sunblaze-ucb/Intuitor
[8] https://github.com/cpark88/TWiCE-Rec
[9] https://github.com/huggingface/open-r1

■ **C-APO**[10] introduces Coherent Preference (CP) as a complementary signal to Revealed Preference (RP), motivated by the observation that observed behaviors can be noisy and may not always reflect consistent intent. Its conflict-aware DPO formulation jointly optimizes RP- and CP-derived orderings and dynamically balances them when the two signals agree or disagree, yielding robust recommendations and more logically consistent rationales. We implemented this baseline with reference to the official repository[11].

■ **SumRec$_{\text{DPO}}$** [19] trains with DPO using LLM-generated dialogue summaries paired with recommendation supervision, originally designed for multi-turn conversational recommendation. To fit our single-turn sequential setting, we adapt the training pipeline by applying SFT and DPO on single-turn inputs constructed from user histories. We implemented this baseline from scratch.

■ **VUG** [4] addresses fairness-aware cross-domain recommendation by mitigating the performance gap between overlapping and non-overlapping users. It generates virtual source-domain user representations for non-overlapping target-domain users using dual attention over user-user and item-item similarities, together with a limiter that aligns generated users with the source-domain distribution while preserving user-specific characteristics. Since VUG is originally designed as a model-agnostic CDR framework rather than an LLM-based recommender, we adapt it as a non-overlapping CDR baseline under our evaluation protocol. We implemented this baseline following the original paper.

■ **ALCDR** [20] formulates cross-domain recommendation as adaptive anchor-link learning, arguing that directly transferring knowledge through overlapping users may be noisy when user preferences differ across domains. It learns task-relevant cross-domain anchor links via optimal transport and then aggregates intra-domain and inter-domain collaborative signals with graph-based and attention-based modules. Since ALCDR is originally designed for conventional CDR rather than LLM-based recommendation, we adapt it as a non-overlapping CDR baseline under our evaluation protocol. We reproduce this baseline using the training procedure described in the original work.

■ **TCPLP** [21] addresses many-to-one non-overlapping cross-domain sequential recommendation by representing items with textual descriptions rather than atomic IDs. It leverages natural language semantics as a bridge across disjoint domains and learns transferable knowledge through domain-shared and domain-specific prompts encoded by a co-attention network. The model is trained with a two-stage pre-training and prompt-tuning strategy to first capture multi-domain prior knowledge and then adapt it to the target domain. Since TCPLP is originally designed for non-overlapping cross-domain sequential recommendation, we adapt it as a strong semantic-enhanced CDR baseline under our evaluation protocol. We implemented this baseline following the paper's methodology.

D Extended Ablation Study on ASP Submodules

To further analyze the contribution of each component in the Agentic Synthetic Data Generation Pipeline (ASP), we conduct an additional ablation study focused on the *Meta-Judge* module. Specifically, we compare the full ASP pipeline with a variant that removes the Meta-Judge while keeping the remaining pseudo-label construction process unchanged. After constructing pseudo-labels under each setting, we train the same downstream CURE policy model and evaluate its recommendation performance.

As shown in Table 12, removing the Meta-Judge consistently degrades performance in both cross-domain and in-domain recommendation tasks. This suggests that the Meta-Judge plays an important role in filtering noisy pseudo-labels and improving the reliability of ASP-generated supervision.

E Rationale Quality Evaluation

A potential concern is that optimizing recommendation performance may lead to reward hacking, where the model produces plausible-looking but low-quality rationales. To examine this, we conduct an additional rationale-quality evaluation on 600 samples from two domains, with 300 samples each from Grocery and Health. We first verify item validity and then evaluate the generated rationales using an independent LLM (i.e., GPT-4O) that is different from the rationale-generation model.

[10] This work has not yet been formally published but is publicly available on OpenReview (link: https://openreview.net/forum?id=WYfDoB44xy).

[11] https://github.com/cpark88/C-APO

Table 12: Extended ablation study on the Meta-Judge module in ASP. We compare downstream CURE performance when pseudo-labels are constructed with and without the Meta-Judge.

Cross-Domain Evaluation				
ASP Variant	**Fashion → Clothing**		**Grocery → Health**	
	HR@1	**NDCG@5**	**HR@1**	**NDCG@5**
No Meta-Judge	6.7	13.2	2.5	6.9
Including Meta-Judge	**9.8**	**16.6**	**4.1**	**8.9**
In-Domain Evaluation				
ASP Variant	**Amazon Fashion**		**Health**	
	HR@1	**NDCG@5**	**HR@1**	**NDCG@5**
No Meta-Judge	9.8	19.6	2.9	6.5
Including Meta-Judge	**16.8**	**31.0**	**6.9**	**14.2**

Table 13: Rationale quality evaluation on Grocery and Health. Scores denote the percentage of generated rationales assigned to each quality level. Higher values are better for scores 2 and 3, while lower values are better for scores 0 and 1. The final row reports the proportion of high-quality rationales, i.e., scores 2–3.

Score	**Grocery**					**Health**				
	SFT	**Rec-R1**	**TWiCERec**	**Gemma-12B**	**CURE**	**SFT**	**Rec-R1**	**TWiCERec**	**Gemma-12B**	**CURE**
0	5.67	5.00	5.00	4.67	**4.33**	**1.33**	3.67	7.00	4.33	1.67
1	2.00	1.33	4.67	7.33	**1.00**	4.00	**2.00**	7.33	5.33	**2.00**
2	6.67	4.00	7.00	20.00	5.33	10.00	4.67	11.33	15.67	5.33
3	85.67	**89.67**	83.33	68.00	89.33	84.67	89.67	74.33	74.67	**91.00**
Score 2–3	92.34	93.67	90.33	88.00	**94.66**	94.67	94.34	85.66	90.34	**96.33**

Each rationale is rated on a 4-point scale: 0 indicates an incorrect or hallucinated rationale, 1 indicates a correct but weak rationale, 2 indicates a correct and reasonable rationale, and 3 indicates a correct and persuasive rationale. To validate the evaluator, we compare its judgments against 250 human-labeled samples and observe substantial agreement (QWK = 0.75, $p < 0.0001$). As shown in Table 13, CURE achieves the highest proportion of high-quality rationales, measured by the percentage of samples rated as 2 or 3, in both domains.

F Derivation of Jacobian of A_j with respect to $\hat{\rho}_k$

Consider a user group with $n := |O|$ sampled outputs and calibrated rewards $\hat{\rho} = (\hat{\rho}_1, \ldots, \hat{\rho}_n)$. Define the within-group mean and (population) standard deviation as $\mu = \frac{1}{n}\sum_{i=1}^{n} \hat{\rho}_i, \quad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\rho}_i - \mu)^2}$, and the standardized advantage (assuming $\epsilon = 0$) as $A_j = \frac{\hat{\rho}_j - \mu}{\sigma}$. We derive the Jacobian entry $\frac{\partial A_j}{\partial \hat{\rho}_k}$ for $n \geq 2$ and $\sigma > 0$.

Step 1: Derivative of the mean. The partial derivative of the mean with respect to $\hat{\rho}_k$ is given by $\frac{\partial \mu}{\partial \hat{\rho}_k} = \frac{\partial}{\partial \hat{\rho}_k}\left(\frac{1}{n}\sum_{i=1}^{n} \hat{\rho}_i\right) = \frac{1}{n}$. Let $u_j := \hat{\rho}_j - \mu$. Then, we have $\frac{\partial u_j}{\partial \hat{\rho}_k} = \mathbb{I}[j = k] - \frac{1}{n}$.

Step 2: Derivative of the standard deviation. Let $v := \frac{1}{n}\sum_{i=1}^{n}(\hat{\rho}_i - \mu)^2, \ \sigma = \sqrt{v}$. By the chain rule, $\frac{\partial \sigma}{\partial \hat{\rho}_k} = \frac{1}{2\sigma}\frac{\partial v}{\partial \hat{\rho}_k}$. Denoting $s_i := \hat{\rho}_i - \mu$, we have $v = \frac{1}{n}\sum_i s_i^2$ and

$$\frac{\partial v}{\partial \hat{\rho}_k} = \frac{1}{n}\sum_{i=1}^{n} 2 s_i \frac{\partial s_i}{\partial \hat{\rho}_k} = \frac{2}{n}\sum_{i=1}^{n} s_i \left(\mathbb{I}[i = k] - \frac{1}{n}\right) \tag{5}$$

$$= \frac{2}{n}\left(s_k - \frac{1}{n}\sum_{i=1}^{n} s_i\right) = \frac{2}{n} s_k = \frac{2}{n}(\hat{\rho}_k - \mu), \tag{6}$$

where we used $\sum_i s_i = \sum_i (\hat{\rho}_i - \mu) = 0$. Therefore,

$$\frac{\partial \sigma}{\partial \hat{\rho}_k} = \frac{1}{2\sigma} \cdot \frac{2}{n}(\hat{\rho}_k - \mu) = \frac{\hat{\rho}_k - \mu}{n\sigma}. \tag{7}$$

Step 3: Jacobian entry. Since $A_j = u_j/\sigma$,

$$\frac{\partial A_j}{\partial \hat{\rho}_k} = \frac{1}{\sigma}\frac{\partial u_j}{\partial \hat{\rho}_k} - \frac{u_j}{\sigma^2}\frac{\partial \sigma}{\partial \hat{\rho}_k} \tag{8}$$

$$= \frac{\mathbb{I}[j=k] - \frac{1}{n}}{\sigma} - \frac{(\hat{\rho}_j - \mu)}{\sigma^2} \cdot \frac{\hat{\rho}_k - \mu}{n\sigma} \tag{9}$$

$$= \frac{\mathbb{I}[j=k] - \frac{1}{n}}{\sigma} - \frac{(\hat{\rho}_j - \mu)(\hat{\rho}_k - \mu)}{n\sigma^3}, \tag{10}$$

which matches Eq. 2.

G Supervised Fine-Tuning before CURE

Training Supervised fine-tuning (SFT) is a common preliminary step for LLM-based recommender systems, serving to adapt general-purpose language models to recommendation-specific behaviors. This procedure typically involves two stages: first, converting recommendation data into instruction-style prompt–output pairs, and second, fine-tuning the model on these pairs. For each user u, we construct an input prompt q_u^{in} that encodes the user's interaction history I_u^d, a candidate item set C_u^d, and an explicit instruction describing the sequential recommendation task. The corresponding output y_u consists of the selected target item $i_+^d \in C_u^d$ along with its associated rationale r_+^d, yielding a supervised training pair (q_u^{in}, y_u). Approximately 10% of the SFT examples are later reused during the CURE stage to ensure continuity between the two training phases.

In the second stage, the LLM-based recommender f_θ is optimized on these prompt–output pairs using a causal language modeling objective [22]. This training objective encourages the model to generate the target output autoregressively, predicting each token conditioned on the input prompt and previously generated tokens. Under this formulation, recommendation is framed as a sequence generation problem, where the model learns to produce both the recommended item and its rationale. Formally, given training pairs (q_u^{in}, y_u), the optimization objective is defined as:

$$\max_\theta \sum_{(q_u^{\text{in}}, y_u)} \sum_{t=1}^{|y_u|} \log(P_\theta(y_{u,t}) | q_u^{\text{in}}, y_{u,<t}), \tag{11}$$

where $|y_u|$ denotes the length of the output sequence, $y_{u,t}$ represents the t-th token, and $y_{u,<t}$ denotes all preceding tokens.

Despite its effectiveness, this SFT objective primarily conditions on observed target items and does not explicitly account for user preference alignment, which lies at the core of recommendation problems.

Prompt Example The SFT prompt is constructed to guide the model toward identifying the most suitable item among a set of candidate recommendations based on the user's purchase history, while simultaneously producing a natural-language rationale that justifies the selection.

An example prompt for Supervised Fine-Tuning

#Prompt q_u^{in}
Based on **[Purchase History]**, use your logical reasoning process to identify the most suitable item for this customer from the **[Candidate List]**. Then, include your reasoning inside the `<think></think>`tag and the recommended item inside the `<item></item>`tag.
[Purchase History] (1) Title: Bela fleck & abigail washburn (2) Title: Love me or leave me: from the sound track 1955 film,...(truncated) $\leftarrow I_u^d$
[Candidate List]: (omitted for brevity) $\leftarrow C_u^d$

#Response y_u
`<think>`Given your appreciation for classic American music and musical storytelling-evident from your enjoyment Love Me or Leave Me (1955), ... (truncated) `</think>` $\leftarrow r_+^d$
`<item>`Love Me or Leave Me `</item>` $\leftarrow i_+^d$

H Implementation Details of Agentic Synthetic Data Generation Pipeline (ASP)

H.1 System Prompts for Each LLM in ASP

In ASP, the individual LLMs are interconnected using LangGraph and operate as coordinated agents. Each LLM is equipped with a role-specific system prompt tailored to its designated function. The code-level prompts for each LLM, constructed with LangGraph, are specified as follows.

H.1.1 Generator LLM

The Generation LLM generates a recommended item and an accompanying post-hoc rationale conditioned on a user's historical interactions and cross-domain candidate items, ensuring logical consistency with the user's history and adherence to a predefined output schema.

```
GEN_PROMPT = ChatPromptTemplate.from_messages([
    ("system",
     "You are a recommendation expert tasked with generating a rationale for the next item "
     "purchased by a customer, based on their prior user history among candidate items. "
     "Note that the user history and candidate items come from different domains,
     ↪  representing a cross-domain setting. "
     "Given user's history and candidate items, produce a target recommended item and a
     ↪  post-hoc rationale that is logically consistent "
     "with the history and explains why the target item is recommended in English. "
     "Keep it natural, specific, and within seven sentences. Return content matching the "
     "enforced schema."),
    ("user",
     "User history:\\n{historyList}\\n\\nCandidate items:\\n{candidates}")
])
```

H.1.2 Scoring LLM

The Scoring LLM serves as a strict evaluator that assesses the logical coherence between a user's interaction history and the generated recommendation rationale. Given the user history, the target recommended item, and its rationale, it assigns a discrete coherence score on a 7-point scale, accompanied by a brief justification. The evaluation protocol is intentionally conservative, encouraging high scores only when the rationale is strongly aligned with the user's historical context, and penalizing weak or unpersuasive explanations to ensure reliable downstream supervision.

```
EVAL_PROMPT = ChatPromptTemplate.from_messages([
    ("system",
     "You are a recommendation expert and a strict evaluator. "
     "Score the logical coherence between the user history and the recommendation rationale
     ↪  on a scale of 1 - 7 (
     "1: Very Weak -- No connection, "
     "2: Weak -- Poorly justified, "
     "3: Slightly Weak -- Minimal relevance, "
     "4: Neutral -- Ambiguous or marginally related, "
     "5: Slightly Strong -- Generally aligned, "
     "6: Strong -- Well aligned and logical, "
     "7: Very Strong -- Highly coherent and contextually perfect). "
     "And explain the justification behind the coherence score (1--7) you assigned to the
     ↪  recommendation."
     "You should give scores of 6 or higher only with great caution. "
     "If the rationale lacks persuasiveness, you must assign a score of 3 or lower!!! "
     "Return only a value matching the enforced schema."),
    ("user",
     "User history:\\n{history_list}\\n\\nTarget
     ↪  item:\\n{target_item}\\n\\nRationale:\\n{rationale}")
])
```

H.1.3 Meta-Judge LLM

The Meta-Judge LLM acts as a conservative human proxy that verifies the overall plausibility of the coherence evaluation produced by the Scoring LLM. Given the user history, target item, generated

rationale, coherence score, and its justification, it determines whether the assigned score—high or low—is appropriate and internally consistent. The Meta-Judge LLM adopts a conservative rejection policy under uncertainty, ensuring that only reliable and well-grounded evaluations are retained for downstream use.

```
META_PROMPT = ChatPromptTemplate.from_messages([
    ("system",
     "You are a recommendation expert, acting as a human evaluator. "
     "Given a user history, target item, rationale, justification of coherence score, "
     "and coherence score (1--7) "
     "(1: Very Weak -- No connection, "
     "2: Weak -- Poorly justified, "
     "3: Slightly Weak -- Minimal relevance, "
     "4: Neutral -- Ambiguous or marginally related, "
     "5: Slightly Strong -- Generally aligned, "
     "6: Strong -- Well aligned and logical, "
     "7: Very Strong -- Highly coherent and contextually perfect), "
     "decide whether the rationale and score are plausible and valid overall. "
     "Be conservative: prefer False if unsure. "
     "If the score is 6 or higher, the rationale should be reasonably persuasive; "
     "if the score is low, the rationale should lack persuasiveness and show little to no
     ↪  relevance between the user history and the target item. "
     "You are judging whether the score itself - whether high or low  - is appropriate. "
     "Return only a value matching the enforced schema."),
    ("user",
     "User history:\\n{historyList}\\n\\nTarget item:\\n{target_item}"
     "\\n\\nRationale:\\n{rationale}"
     "\\n\\nScore:\\n{score}"
     "\\n\\nJustification of score:\\n{eval_reason}")
])
```

H.2 Implementation Details of ASP Tools

We design a retriever tool within ASP to construct cross-domain recommendation data by searching for candidate items in the target domain that are semantically aligned with a given user history. This tool is activated specifically in the cross-domain setting, where the user's interaction history originates from a source domain and recommendations are generated for a different target domain. Concretely, we first encode the user history using a sentence embedding model such as SBERT [23]. We then perform a semantic search over pre-chunked embeddings of all items in the target domain, retrieving the top-N most semantically similar items from a large candidate pool (often consisting of tens of thousands of items). This retrieval step follows a retrieval-augmented generation (RAG) paradigm and yields a target-domain candidate set. Conditioned on the source-domain user history and the retrieved target-domain candidate set, the Generation LLM is prompted to select the most appropriate item from the candidates and to generate an accompanying recommendation rationale grounded in the user's historical preferences. In addition, we introduce a memory tool to support iterative refinement during generation. If an output is assigned a score below a predefined threshold S^* by the Scoring LLM, or is judged as False by the Meta-Judge LLM, the previously generated outputs are appended to the prompt via the memory tool. This mechanism explicitly instructs the Generation LLM to produce a new output that is distinct from prior attempts, encouraging diversity and avoiding repeated failure modes. The retry process is capped at a maximum of 10 iterations. If a sample fails to pass the scoring and meta-judging criteria within this limit, it is discarded and excluded from the final training data.

H.3 ASP Construction Cost and Deployment Efficiency

We emphasize that ASP is used only as an *offline data construction pipeline*, rather than as a serving-time recommender. The actual online recommender is the resulting CURE policy model, which is substantially smaller and more efficient to serve. Thus, the computational overhead of ASP is incurred only once during pseudo-label generation and does not affect deployment-time latency.

Table 14 reports the per-sample cost breakdown of ASP. Although ASP uses a multi-LLM pipeline for generation, scoring, and meta-judging, the total cost remains confined to offline data construction. The LLM API cost was about \$0.022975 per sample, resulting in a total cost of approximately \$3.4K for generating 150K samples. We consider this cost reasonable, as the synthetic data is required only

Table 14: Per-sample cost breakdown of the ASP pseudo-label construction pipeline. ASP is used only offline and is not part of online serving.

Component	Model Type	Latency	API Cost
Generator	GPT-4.1	5.5s	$0.0225
Scoring	Gemma-3-27B	7.0s	– (Local)
Meta-judge	GPT-4.1-mini	2.5s	$0.000475
Total ASP	**Multi-LLM**	**15.0s**	**$0.022975**

Table 15: Deployment efficiency comparison between offline ASP construction and online CURE serving.

Metric	ASP (Offline)	CURE (Online Serving)
Inference Latency	15.0s / sample	0.14s / sample via vLLM
GPU Memory	21GB+	8GB

once for initial training; after deployment, the model can be updated using continuously collected user feedback logs. In contrast, Table 15 shows that CURE can be served with low latency and modest GPU memory when deployed with standard inference engines such as vLLM.

H.4 Human Evaluation Protocol

To assess the alignment between the LLM-based meta-judge and human evaluators, we conduct a statistical agreement analysis on binary (True/False) judgments. For each sample, we collect annotations from eight human[12] judges and define the human reference label via majority voting. We asked eight domain experts to independently provide binary (True/False) judgments on 300 samples using the same criteria as the Meta-Judge LLM, and compared the resulting majority-vote labels against the Meta-Judge LLM outputs. On this evaluation dataset, the Meta-Judge LLM achieved a simple agreement rate of 75.0% (95% Wilson confidence interval: [0.70, 0.80]), indicating a high level of consistency with human judgments. To account for agreement expected by chance, we further computed Cohen's Kappa between the LLM predictions and the human reference labels. The resulting chance-corrected agreement was statistically significant ($\kappa = 0.4775$; bootstrap standard error = 0.1624; 95% bootstrap confidence interval: [0.1190, 0.7441]), suggesting moderate agreement beyond chance. To contextualize this result, we additionally measured inter-annotator reliability among the eight human experts using Fleiss' κ. The obtained value of 0.51 reflects strong consistency across human judgments and serves as an empirical upper bound on the agreement achievable by an automated evaluator under the same criteria. Together, these results indicate that the Meta-Judge LLM operates close to the level of human consensus while maintaining statistically meaningful agreement beyond chance. Overall, these results demonstrate that the LLM meta-judge exhibits human-aligned judgment behavior at a level approaching human–human agreement, despite inherent annotation noise.

I Online A/B Test Details

We validated our approach in a real-world production deployment and ran an online A/B test as a pilot study during August–December 2025. In the live setting, each user was shown a single (top-1) recommended item together with a short rationale explaining the suggestion. We served a customized variant of our model that satisfies product and business constraints. We compared against three baselines: (i) a popularity-based model, (ii) a conventional machine-learning (ML) recommender, and (iii) an LLM model trained with GRPO only, which generates both the recommendation and a rationale. Traffic was assigned at the user level using a stable hash-based randomization into four buckets (Popularity, ML, GRPO, Ours).

Table 4 reports the main online A/B results. Relative to the ML baseline, our method achieves a +55.7% lift in CTR (14.07% vs. 9.04%), corresponding to an absolute gain of +5.03%p. This

[12]A total of eight domain experts participated in the evaluation: one service marketer (male, age 34); three service engineers (male, ages 41, 33, and 29); and four data scientists (male, ages 37, 33, 31, and 30). This group reflects diverse professional roles spanning product, engineering, and modeling.

improvement is strongly supported by a one-sided two-proportion z-test ($Z = 42.23$, $p < 0.001$), with a 95% confidence interval of [4.79%p, 5.27%p] for the absolute CTR difference. We observe similarly significant gains over the popularity-based model (14.07% vs. 9.37%; +4.70%p, $Z = 41.92$, $p < 0.001$, 95% CI = [4.46%p, 4.93%p]). Importantly, these improvements are obtained while maintaining low inference latency, indicating that the approach is practical for production deployment.

To rule out the possibility that the uplift is driven solely by the novelty of displaying rationales, we additionally include an LLM baseline trained with GRPO only, which also presents rationales to users but does not apply our CURE training method. Since both systems expose rationales, this comparison isolates the incremental benefit of our proposed optimization. Even under this stricter control, our method yields a statistically significant CTR increase over GRPO (14.07% vs. 12.88%), i.e., +1.18%p ($Z = 3.20$, $p < 0.001$), with a 95% confidence interval of [0.48%p, 1.88%p] for the absolute difference. Finally, we also measure conversion rate (CVR), defined as purchases per impression, and observe consistent improvements over all baselines.

The online experiment was conducted with randomized (user-sticky) bucket assignment and all variants were served concurrently under identical eligibility and placement conditions. While the traffic allocation was intentionally unbalanced for risk control, randomization ensures that CTR comparisons remain unbiased, with unequal sample sizes primarily affecting statistical power.

J Time Complexity

Theoretical Analysis Let $C_{\texttt{LLM}}$ denote the computational cost of a single forward–backward pass of the backbone LLM, and let B be the mini-batch size. For each prompt, GRPO samples K candidate outputs. In our training setup, for each user we construct N prompts that share the same anchor history (1 in-domain and $N-1$ cross-domain), resulting in a user-level output group of size $|O| = N \cdot K$.

The policy computes log-probabilities and rewards for all sampled outputs, yielding a per-batch complexity of $\mathcal{O}\big(B \cdot |O| \cdot C_{\texttt{LLM}}\big) = \mathcal{O}\big(B \cdot NK \cdot C_{\texttt{LLM}}\big)$, which matches the standard GRPO complexity under the same sampling budget. CURE does not introduce additional LLM forward or backward passes beyond those required for sampling.

User-level grouping, joint normalization, and advantage computation operate on scalar rewards and incur an $\mathcal{O}(B \cdot |O|)$ cost, which is negligible relative to the backbone LLM computation. These components therefore do not affect the overall asymptotic complexity, resulting in the same computational complexity as vanilla GRPO.

Empirical Analysis We measure wall-clock training time using four NVIDIA H100 GPUs under identical optimization settings. Consistent with the theoretical analysis, CURE exhibits training throughput comparable to standard GRPO. In practice, training time is dominated by backbone LLM inference, and the additional operations introduced by user-level coupling and reward calibration have negligible impact.

K Proofs for Section 4 (Theoretical Analysis)

K.1 Proof of Lemma 1

Let $n_{\text{in}} := |O_u^{\text{in}}|$, $n_{\text{cr}} := |O_u^{\text{cr}}|$, and $N := n_{\text{in}} + n_{\text{cr}}$. Recall the noise model in Section 4.1: for cross-domain rewards,

$$\rho_{u,j}^{\text{cr}} = (\rho_{u,j}^{\text{cr}})^\star + \varepsilon_j, \qquad j \in \{1, \dots, n_{\text{cr}}\},$$

where $\mathbb{E}[\varepsilon_j] = 0$ and $\text{Var}(\varepsilon_j) \le \sigma_\varepsilon^2$. In-domain rewards are treated as noise-free:

$$\rho_{u,k}^{\text{in}} = (\rho_{u,k}^{\text{in}})^\star, \qquad k \in \{1, \dots, n_{\text{in}}\}.$$

Step 1: Express the mean shifts in terms of noise. By definition,

$$\mu_u^{\text{cr}} := \frac{1}{n_{\text{cr}}} \sum_{j=1}^{n_{\text{cr}}} \rho_{u,j}^{\text{cr}}, \qquad (\mu_u^{\text{cr}})^\star := \frac{1}{n_{\text{cr}}} \sum_{j=1}^{n_{\text{cr}}} (\rho_{u,j}^{\text{cr}})^\star.$$

Substituting $\rho_{u,j}^{\mathrm{cr}} = (\rho_{u,j}^{\mathrm{cr}})^\star + \varepsilon_j$ gives

$$b_u^{\mathrm{unc}} := \mu_u^{\mathrm{cr}} - (\mu_u^{\mathrm{cr}})^\star = \frac{1}{n_{\mathrm{cr}}} \sum_{j=1}^{n_{\mathrm{cr}}} \varepsilon_j. \tag{12}$$

Similarly, the user-grouped mean over in- and cross-domain rewards is

$$\mu_u^{\mathrm{in+cr}} := \frac{1}{N} \left(\sum_{k=1}^{n_{\mathrm{in}}} \rho_{u,k}^{\mathrm{in}} + \sum_{j=1}^{n_{\mathrm{cr}}} \rho_{u,j}^{\mathrm{cr}} \right),$$

and its oracle counterpart is

$$(\mu_u^{\mathrm{in+cr}})^\star := \frac{1}{N} \left(\sum_{k=1}^{n_{\mathrm{in}}} (\rho_{u,k}^{\mathrm{in}})^\star + \sum_{j=1}^{n_{\mathrm{cr}}} (\rho_{u,j}^{\mathrm{cr}})^\star \right).$$

Since in-domain rewards are noise-free, their contributions cancel in the difference. Therefore,

$$b_u^{\mathrm{cpl}} := \mu_u^{\mathrm{in+cr}} - (\mu_u^{\mathrm{in+cr}})^\star = \frac{1}{N} \sum_{j=1}^{n_{\mathrm{cr}}} \varepsilon_j. \tag{13}$$

Let

$$\alpha := \frac{n_{\mathrm{cr}}}{N}.$$

From (12) and (13), we have the deterministic relation

$$b_u^{\mathrm{cpl}} = \frac{1}{N} \sum_{j=1}^{n_{\mathrm{cr}}} \varepsilon_j = \frac{n_{\mathrm{cr}}}{N} \left(\frac{1}{n_{\mathrm{cr}}} \sum_{j=1}^{n_{\mathrm{cr}}} \varepsilon_j \right) = \alpha b_u^{\mathrm{unc}}. \tag{14}$$

Step 2: Compare the variances. Using (14),

$$\mathrm{Var}\big(b_u^{\mathrm{cpl}}\big) = \mathrm{Var}(\alpha b_u^{\mathrm{unc}}) = \alpha^2 \mathrm{Var}(b_u^{\mathrm{unc}}).$$

Since $\alpha = n_{\mathrm{cr}}/N$, this gives

$$\mathrm{Var}\big(b_u^{\mathrm{cpl}}\big) = \Big(\frac{n_{\mathrm{cr}}}{N}\Big)^2 \mathrm{Var}(b_u^{\mathrm{unc}}),$$

which proves the claimed equality.

For completeness, under independence and $\mathrm{Var}(\varepsilon_j) \le \sigma_\varepsilon^2$,

$$\mathrm{Var}(b_u^{\mathrm{unc}}) = \mathrm{Var}\left(\frac{1}{n_{\mathrm{cr}}} \sum_{j=1}^{n_{\mathrm{cr}}} \varepsilon_j \right) = \frac{1}{n_{\mathrm{cr}}^2} \sum_{j=1}^{n_{\mathrm{cr}}} \mathrm{Var}(\varepsilon_j) \le \frac{\sigma_\varepsilon^2}{n_{\mathrm{cr}}},$$

and

$$\mathrm{Var}\big(b_u^{\mathrm{cpl}}\big) = \mathrm{Var}\left(\frac{1}{N} \sum_{j=1}^{n_{\mathrm{cr}}} \varepsilon_j \right) = \frac{1}{N^2} \sum_{j=1}^{n_{\mathrm{cr}}} \mathrm{Var}(\varepsilon_j) \le \frac{n_{\mathrm{cr}}}{N^2} \sigma_\varepsilon^2.$$

□

K.2 Proof of Lemma 2

Let R_u^{cr} denote the set of cross-domain rewards for user u, and let $R_u^{\mathrm{in+cr}}$ denote the union of in- and cross-domain rewards. Let

$$\mu_u^{\mathrm{cr}} = \frac{1}{n_{\mathrm{cr}}} \sum_{x \in R_u^{\mathrm{cr}}} x, \qquad \mu_u^{\mathrm{in+cr}} = \frac{1}{N} \sum_{x \in R_u^{\mathrm{in+cr}}} x.$$

We use the population variance convention:

$$(\sigma_u^{\mathrm{cr}})^2 = \frac{1}{n_{\mathrm{cr}}} \sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{cr}})^2, \qquad (\sigma_u^{\mathrm{in+cr}})^2 = \frac{1}{N} \sum_{x \in R_u^{\mathrm{in+cr}}} (x - \mu_u^{\mathrm{in+cr}})^2.$$

By definition,

$$N(\sigma_u^{\mathrm{in+cr}})^2 = \sum_{x \in R_u^{\mathrm{in+cr}}} (x - \mu_u^{\mathrm{in+cr}})^2.$$

Keeping only the cross-domain terms gives

$$N(\sigma_u^{\mathrm{in+cr}})^2 \geq \sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{in+cr}})^2.$$

For any finite set, the sum of squared deviations is minimized at the empirical mean of that set. Therefore,

$$\sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{in+cr}})^2 \geq \sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{cr}})^2.$$

Equivalently, this follows from the variance decomposition

$$\sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{in+cr}})^2 = \sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{cr}})^2 + n_{\mathrm{cr}} (\mu_u^{\mathrm{cr}} - \mu_u^{\mathrm{in+cr}})^2.$$

Thus,

$$N(\sigma_u^{\mathrm{in+cr}})^2 \geq \sum_{x \in R_u^{\mathrm{cr}}} (x - \mu_u^{\mathrm{cr}})^2 = n_{\mathrm{cr}} (\sigma_u^{\mathrm{cr}})^2.$$

Dividing both sides by N yields

$$(\sigma_u^{\mathrm{in+cr}})^2 \geq \frac{n_{\mathrm{cr}}}{N} (\sigma_u^{\mathrm{cr}})^2 = \alpha (\sigma_u^{\mathrm{cr}})^2.$$

□

K.3 Proof of Theorem 1

We show that coupling attenuates the *mean-shift-induced* perturbation in the cross-domain policy update.

Step 1: Isolate the mean-shift component. Define the mean-shift terms of the uncoupled and coupled cases as

$$b_u^{\mathrm{unc}} := \mu_u^{\mathrm{cr}} - (\mu_u^{\mathrm{cr}})^\star, \qquad b_u^{\mathrm{cpl}} := \mu_u^{\mathrm{in+cr}} - (\mu_u^{\mathrm{in+cr}})^\star.$$

Consider the portion of the cross-domain normalized advantage that arises solely from replacing the oracle mean by the noisy empirical mean. This contributes the common mean-shift term b/σ to all cross-domain samples in the same user group and hence induces the following perturbation in the update:

$$U(b_u, \sigma_u) := \frac{1}{n_{\mathrm{cr}}} \sum_{j=1}^{n_{\mathrm{cr}}} \frac{b_u}{\sigma_u} \Delta_{u,j}^{\mathrm{cr}}.$$

Equivalently, defining

$$\bar{\Delta}_u^{\mathrm{cr}} := \frac{1}{n_{\mathrm{cr}}} \sum_{j=1}^{n_{\mathrm{cr}}} \Delta_{u,j}^{\mathrm{cr}},$$

we can write

$$U(b_u, \sigma_u) = \frac{b_u}{\sigma_u} \bar{\Delta}_u^{\mathrm{cr}}.$$

Accordingly, the uncoupled and coupled mean-shift perturbations are

$$U(b_u^{\mathrm{unc}}, \sigma_u^{\mathrm{cr}}) \qquad \text{and} \qquad U(b_u^{\mathrm{cpl}}, \sigma_u^{\mathrm{in+cr}}).$$

Step 2: Compare the perturbations using the mean-shift relation and the pooled standard deviation bound. From (14), we have

$$b_u^{\mathrm{cpl}} = \alpha b_u^{\mathrm{unc}}, \qquad \alpha = \frac{n_{\mathrm{cr}}}{N}.$$

Therefore,

$$\left\| U(b_u^{\mathrm{cpl}}, \sigma_u^{\mathrm{in+cr}}) \right\|_2^2 = \left\| \frac{\alpha b_u^{\mathrm{unc}}}{\sigma_u^{\mathrm{in+cr}}} \bar{\Delta}_u^{\mathrm{cr}} \right\|_2^2 = \frac{\alpha^2 (b_u^{\mathrm{unc}})^2}{(\sigma_u^{\mathrm{in+cr}})^2} \left\| \bar{\Delta}_u^{\mathrm{cr}} \right\|_2^2.$$

By Lemma 2,

$$(\sigma_u^{\mathrm{in+cr}})^2 \geq \alpha (\sigma_u^{\mathrm{cr}})^2.$$

Hence,

$$\left\| U(b_u^{\mathrm{cpl}}, \sigma_u^{\mathrm{in+cr}}) \right\|_2^2 \leq \frac{\alpha^2 (b_u^{\mathrm{unc}})^2}{\alpha (\sigma_u^{\mathrm{cr}})^2} \left\| \bar{\Delta}_u^{\mathrm{cr}} \right\|_2^2 = \alpha \frac{(b_u^{\mathrm{unc}})^2}{(\sigma_u^{\mathrm{cr}})^2} \left\| \bar{\Delta}_u^{\mathrm{cr}} \right\|_2^2.$$

The right-hand side is exactly

$$\alpha \left\| U(b_u^{\mathrm{unc}}, \sigma_u^{\mathrm{cr}}) \right\|_2^2.$$

Therefore,

$$\left\| U(b_u^{\mathrm{cpl}}, \sigma_u^{\mathrm{in+cr}}) \right\|_2^2 \leq \alpha \left\| U(b_u^{\mathrm{unc}}, \sigma_u^{\mathrm{cr}}) \right\|_2^2. \tag{15}$$

Step 3: Take expectations. Since $0 < \alpha \leq 1$, taking expectations on both sides of (15) gives

$$\mathbb{E}\left[\left\| U(b_u^{\mathrm{cpl}}, \sigma_u^{\mathrm{in+cr}}) \right\|_2^2 \right] \leq \alpha \mathbb{E}\left[\left\| U(b_u^{\mathrm{unc}}, \sigma_u^{\mathrm{cr}}) \right\|_2^2 \right] \leq \mathbb{E}\left[\left\| U(b_u^{\mathrm{unc}}, \sigma_u^{\mathrm{cr}}) \right\|_2^2 \right].$$

This proves the claimed attenuation of the mean-shift-induced update perturbation under coupled normalization. □

Limitations

ASP relies on LLM-generated pseudo supervision, which may introduce noise, although our validation results, ablation studies, and reward calibration show that it provides useful and robust training signals. Another limitation is that ASP incurs a non-trivial one-time generation cost; however, this cost is incurred offline and can be amortized over downstream training and deployment, making it increasingly practical in the current token-economy setting. Future work will investigate how such recommendation models can be continually updated using real-world user feedback collected after deployment.

Societal Impact

This work can improve recommendation quality in non-overlapping cross-domain settings without requiring explicit user identity matching across platforms. By avoiding direct sharing or alignment of user-level information across domains, it may also help reduce privacy risks in cross-platform recommendation scenarios. At the same time, as with any recommender system, deployment should consider potential risks related to privacy, fairness, popularity bias, user control, and over-personalization.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's main contributions and scope: ASP-based pseudo-supervision, CURE's coupled user-grouped RL framework, theoretical analysis, offline evaluation, and online A/B validation for non-overlapping cross-domain recommendation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated limitations section in the appendix that discusses the scope and limitations of the proposed approach.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper states the key assumptions for the theoretical analysis and provides the complete proofs in Appendix; the stated lemmas and theorem appear consistent with those assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper discloses the key information needed to reproduce the main experimental results, including datasets, domain pairs, baselines, evaluation protocol/metrics, backbone model, implementation details, and ablation settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper makes the code and datasets available through the appendix and provides sufficient guidance to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The appendix reports the essential experimental details needed to interpret and reproduce the results, including data splits, baselines, hyperparameters, evaluation metrics, and implementation settings such as the model and optimizer configuration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper includes statistical validation where practical. Offline experiments are run over three random seeds, while comprehensive error bars are constrained by the scale of baselines and the cost of LLM training and inference. The online A/B test additionally reports a statistical significance test.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix reports the main computational details, including GPU type and memory, providing enough information to understand the resource requirements of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper provides sufficient justification that the research conforms to the NeurIPS Code of Ethics: it uses public benchmark datasets, documents the human-evaluation and online-test protocols, and does not disclose any high-risk data collection, privacy violation, or harmful deployment procedure.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader societal implications of our work in the appendix, including both its potential benefits and possible risks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not introduce or release high-risk pretrained models, image-generation systems, or newly collected sensitive datasets. It relies on public recommendation benchmarks and provides code/data for reproducibility purposes, so safeguards specific to high-misuse-risk model or dataset releases are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit the creators of external assets, including the Amazon Reviews and MovieLens datasets, the Gemma and Qwen backbone models, and the Open-R1 codebase, and state the relevant terms of use in the paper or accompanying code.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We document the released assets, including code and data, and provide the necessary usage and reproduction instructions. Because we do not release a trained model, model-specific documentation is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: For the internal human evaluation, we describe the protocol and criteria; compensation details are not applicable because no paid crowdworkers were used.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: We state that the production A/B test followed internal review and privacy safeguards, used anonymized aggregate metrics, and involved minimal user risk.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We used LLMs only to support language editing and improve the clarity of the manuscript.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.