Reproducing FACTER: Fairness via Conformal Thresholding and Prompt Repair

Anonymous authors
Paper under double-blind review

Abstract

Fayyazi et al. (2025) recently proposed FACTER, an innovative model-agnostic framework designed to jointly enforce fairness and statistical coverage in LLM-based recommendation through conformal thresholding and iterative prompt repair. In this work, we conduct a critical reproduction of the FACTER framework across diverse architectures and dataset sparsity levels. Our evaluation reveals a divergence in recommendation utility during open-ended generation tasks, where metrics appear highly sensitive to specific evaluation protocols. To build upon these results, we introduce a static fairness baseline designed to differentiate the effects of FACTER's iterative repair mechanism from simple fairness-aware prompting. Additionally, we propose a re-ranking extension to evaluate FACTER under a more controlled and practically relevant task formulation. We treat this re-ranking setting as a separate diagnostic extension rather than as evidence for reproducing the original open-generation utility claim. Further analysis of the fairness mechanism suggests that violation reductions are largely influenced by the adaptive nature of the conformal thresholding. Additionally, we find that in constrained ranking scenarios, static fairness instructions can achieve comparable results to the dynamic repair loop, suggesting opportunities for optimising computational overhead. By reconciling implementation nuances with theoretical formalisations, our study provides insights into the practical deployment of fair LLM-based recommenders. All code and reproduction artifacts are available at

`https://anonymous.4open.science/r/facter-repr-105B/`.

1 Introduction

Large Language Models (LLMs) are increasingly used as recommendation systems due to their ability to generate personalised item suggestions in natural language (Wu et al., 2024). Recent work explores LLM-based recommendations in domains such as movies and e-commerce, including unified text-to-text recommendation (e.g., P5; Geng et al., 2022), generative next-item frameworks (e.g., GPT4Rec; Li et al., 2023), and hybrid approaches that augment classical recommenders with LLM-generated signals (e.g., LLMRec; Wu et al., 2024). However, alongside their impressive capabilities, LLMs can inadvertently learn and amplify societal biases present in their training data (Bender et al., 2021). In a recommender system context, this can imply that users from different demographic groups may receive systematically different recommendations, raising concerns of fairness and equity (Wang et al., 2023). Unfairness can manifest for users as disparities in the received utility and for item providers as skewed exposure (Wang et al., 2023). Such unfairness can negatively affect multiple stakeholders: reduced user satisfaction or opportunities, reduced exposure and revenue for providers, and increased churn, reputational, or regulatory risk for platforms. A common approach to mitigating such biases relies on retraining or fine-tuning models under fairness constraints, or incorporating fairness-aware objectives into learning pipelines (Wang et al., 2023). In practice, retraining is not always feasible: many LLMs are deployed as black-box services (model-as-a-service), where altering internal parameters is impractical. This motivates model-agnostic techniques to reduce bias using prompting and output-side control; prompt-based debiasing and fairness-guided prompting have been studied as black-box interventions for language models (Schick et al., 2021; Ma et al., 2023; Kamruzzaman & Kim, 2025).

In this work, we conduct a reproducibility study of FACTER (Fayyazi et al., 2025), which proposes a black-box fairness framework for LLM-based recommendation. Building on an individual-fairness perspective (Dwork et al., 2012), FACTER expresses fairness through *counterfactual stability* of recommendations under changes to users' protected attributes, quantifying *semantic variance* in an embedding space, and calibrating a nonconformity threshold via conformal prediction (Fayyazi et al., 2025; Angelopoulos & Bates, 2023). It then iteratively repairs the system prompt by extracting "AVOID" constraints from detected violations.

Reproducing FACTER is practically important because it reports strong fairness improvements while preserving utility. Fairness interventions commonly involve a trade-off with utility (Wang et al., 2023). Moreover, its post-hoc, model-agnostic design is directly relevant to real deployments where only API access is available. Because several details required for an exactly matched implementation are absent or inconsistent across the paper and public code (see Appendix B.6), this study should be read as a best-effort reproduction audit rather than a definitive adjudication of the original results. Our reproduction focuses on (i) faithfully re-implementing the pipeline under missing or inconsistent artefacts, (ii) introducing a Fair Zero-Shot baseline to isolate the contribution of the iterative repair loop, (iii) proposing a separate candidate re-ranking extension to evaluate the approach in a more controlled and practical task formulation, and (iv) conducting a granular component analysis of the fairness mechanism, including multi-seed evaluation and sensitivity analyses, to clarify how reported outcomes depend on experimental design.

2 Scope of Reproducibility

2.1 Problem Setting

We study fairness interventions for LLM-based recommender systems under a *black-box* constraint: the system has access only to model inputs/outputs, not model parameters. In such settings, fairness mechanisms that rely on retraining, constrained optimisation, or gradient access are infeasible, motivating post-hoc interventions such as prompting, output monitoring, and filtering (Wang et al., 2023). Following FACTER (Fayyazi et al., 2025), we model the recommender as

$$\hat{y} = \mathrm{LLM}(x, \boldsymbol{a}, \mathrm{Prompt}_{sys}), \tag{1}$$

where x denotes non-protected user context (e.g., interaction history), $\boldsymbol{a} = (a_1, \ldots, a_k)$ denotes a vector of protected attribute realisations (e.g., `M` (gender), `under 18` (age), `teacher` (occupation)), and Prompt_{sys} denotes the LLM system instructions. FACTER operationalises fairness as *counterfactual stability* with respect to $\boldsymbol{a}$: holding x fixed, the *semantic content* of recommendations remains invariant to changes in $\boldsymbol{a}$. This intuition is closely related to counterfactual fairness, where outcomes should remain stable under interventions on protected attributes (Kusner et al., 2017). We briefly formalise the core components required for reproducibility in Section 3.

2.2 Core Claims

We evaluate FACTER against its core empirical claims (**C1-C4**) , defining reproduction as matching effect direction and achieving comparable magnitudes within reasonable sensitivity to model choice, data sampling, and evaluation.

C1 (Fairness effectiveness) FACTER reduces fairness violations relative to a zero-shot baseline via an online loop that (i) detects threshold exceedances and (ii) injects "AVOID" prompt constraints mined from prior violations (Fayyazi et al., 2025).

C2 (Limited loss in recommendation utility) FACTER's fairness gains do not cause a disproportionate loss in recommendation utility relative to the zero-shot baseline (and UP5 where available), measured via top-k ranking metrics (Recall@10, NDCG@10) (Fayyazi et al., 2025; Järvelin & Kekäläinen, 2002).

C3 (Robustness across sparsity levels) The approach remains effective across datasets with different sparsity and metadata characteristics, specifically MovieLens-1M and Amazon Movies & TV (Fayyazi et al., 2025; Harper & Konstan, 2016; Ni et al., 2019).

C4 (Model-agnostic) The intervention transfers across LLM backbones without model-specific finetuning, since it operates via prompting and output monitoring (Fayyazi et al., 2025).

3 Theoretical Framework

Minimal-Attribute-Change (MAC) Fairness. FACTER adopts an individual fairness notion (Dwork et al., 2012), where similar individuals must receive similar recommendations regardless of their sensitive attributes. Similarity is quantified via embeddings. A recommendation operator Γ_{fair} satisfies MAC fairness if the semantic distance amidst outputs for users differing only in protected attributes ($\boldsymbol{a} \neq \boldsymbol{a}'$) is δ-bounded:

$$\sup_{x,x':\rho(x,x')\leq\epsilon \,\wedge\, \boldsymbol{a}\neq\boldsymbol{a}'} \|\Gamma_{\text{fair}}(x,\boldsymbol{a}) - \Gamma_{\text{fair}}(x',\boldsymbol{a}')\| \leq \delta. \tag{2}$$

While the original paper presents ρ as a distance metric, the implementation uses cosine similarity. We adhere to the implementation logic where semantic contiguity in embedding space proxies for theoretical similarity (Reimers & Gurevych, 2019; Chen et al., 2024).

Conformal Calibration & Coverage. To guarantee fairness with high probability $(1-\alpha)$, FACTER relies on split conformal prediction (Shafer & Vovk, 2008). A nonconformity score $S(x,\boldsymbol{a},y)$ (Eq. 5) measures the violation of MAC principles. A threshold Q_α is calibrated on a held-out set such that:

$$\Gamma_{\text{fair}}(x,\boldsymbol{a}) = \{y \in \mathcal{Y} : S(x,\boldsymbol{a},y) \leq Q_\alpha\}. \tag{3}$$

In the online phase, $Q_\alpha^{(t)}$ is updated via an exponential decay mechanism based on observed violations (Eq. 6), dynamically resizing the acceptance region to handle distributional shifts.

4 Methodology

4.1 The FACTER Algorithm

We implemented FACTER from scratch, adhering strictly to the formal descriptions in the paper while referencing the authors' codebase to resolve ambiguities. A complete overview of these ambiguities is provided in appendix B.6. The framework is split in two phases: offline calibration, which estimates the initial threshold $Q_\alpha^{(0)}$ using the nonconformity score defined in Eq. 5, and online monitoring, in which the adaptive update rule (Eq. 6) is applied.

Offline Calibration. The objective of this phase is to establish an initial acceptance threshold $Q_\alpha^{(0)}$. We first encode user contexts and ground-truth items into a latent space using a sentence encoder. For calibration instance x, we compute a context embedding for user i: $e_i^x = f(x_i)$, where f denotes the sentence encoder and $\boldsymbol{a}_i$ is the protected attribute tuple (gender, age-bin, occupation). The similarity matrix $W \in \mathbb{R}^{n\times n}$ is constructed to identify "counterfactual neighbours", defined as pairs of users who share similar interaction histories but differ in protected attributes. Similarity is determined by cosine similarity $\cos(\cdot,\cdot)$, and τ_x is the optional context-distance threshold described in the original formulation:

$$W_{ij} = \begin{cases} \cos(e_i^x, e_j^x), & \text{if } \boldsymbol{a}_i \neq \boldsymbol{a}_j \text{ and } ||e_i^x - e_j^x||_2 \leq \tau_x \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Using these neighbourhoods, we compute a nonconformity score S_i for each calibration instance, which serves as the conformity statistic which is used to detect violations and calibrate the initial threshold $Q_\alpha^{(0)}$. This score is a composite of the predictive error (deviation from the ground truth item) and a fairness penalty (the maximum semantic distance to recommendations given to neighbours).

$$S_i = \underbrace{1 - \cos(\text{Emb}(\hat{y}_i), e_i^y)}_{\text{Predictive Error } d_i} + \lambda \underbrace{\max_{j:W_{ij}>\tau_\rho} ||\text{Emb}(\hat{y}_i) - \text{Emb}(\hat{y}_j)||_2}_{\text{Fairness Penalty } \Delta_i} \tag{5}$$

The initial threshold $Q_\alpha^{(0)}$ is then set as the $(1-\alpha)$-quantile of these scores, ensuring that $(1-\alpha)$ of the calibration data would be accepted as "fair".

Online Monitoring. In the online phase, the system processes incoming queries sequentially. If a new recommendation's score S_{new} exceeds the current threshold $Q_\alpha^{(t)}$, a violation is recorded. FACTER employs a dual-update mechanism to handle violations:

1. Prompt Repair: Violations are stored in a rolling buffer. Each violation instance yields a tuple (g, Z) consisting of the user's demographic group g and a set of extracted item features Z. We define a bias pattern as a pair (g, z) where $z \in Z$ is a feature (e.g. a genre) associated with a violating recommendation. If a pattern (g, z) appears at least n times in the buffer, an explicit "AVOID: g → z" instruction is injected into the system prompt. For example, repeated violations recommending romance films to female users would yield the pattern $(g = \text{F}, z = \text{Romance})$.

2. Threshold Adaptation: Concurrently, the conformal threshold is updated via exponential smoothing when a violation occurs, to adapt the acceptance region in response to observed violations. Let $Q_\alpha^{(t)}$ denote the current threshold and S_t the conformity score of the query:

$$Q_\alpha^{(t+1)} = \gamma Q_\alpha^{(t)} + (1-\gamma) S_t \tag{6}$$

with smoothing parameter $\gamma \in (0, 1)$ increasing the acceptance threshold in proportion to the severity of the observed violation.

4.2 Practical Instantiation

4.2.1 Datasets and Protected Attributes

We evaluate the reproducibility of FACTER using two datasets: MovieLens-1M (Harper & Konstan, 2016) and Amazon Movies & TV (McAuley et al., 2015; He & McAuley, 2016). To ensure controlled fairness evaluation, we use the same preprocessing pipeline for both datasets. All dataset preprocessing, filtering, and splitting procedures follow the original paper and released implementation without modification.

General Processing Pipeline. Across both datasets, we treat user ratings as interaction proxies and use chronological rating histories as context x. To reduce variability from data stochasticity (rather than modelling), we apply the following standardised steps:

1. **Filtration:** To ensure a robust preference signal for the recommendation task, we remove low-rated samples (rating ≤ 3).
2. **Stochastic Control:** All random sampling, including protected attribute assignment and data splitting, is performed using a fixed seed (0).
3. **Sequence Construction:** User histories are time-ordered; we retain users with at least 10 interactions, sampling any 10 interactions as history with the following 10 as the relevant target set. As clarified with the authors, a multi-target evaluation setting is consistent with the original study.
4. **Stratified Splitting:** Observations are partitioned into a 70/30 train-test split, providing calibration instances for the offline phase and evaluation instances for the online phase. We stratify by demographic groups to maintain balanced group representations.

MovieLens-1M. The ML-1M dataset contains ~1M ratings from 6k users over 4k movies on a 5-star scale (limited to whole-star ratings). It also contains self-reported gender, binned age, and occupation used as protected attributes. Following Fayyazi et al. (2025), bias patterns for prompt repairs are derived from movie titles and genres. We randomly sample 2,500 instances from the preprocessed data (1,750 calibration; 750 evaluation).

Amazon. The Amazon Movies & TV dataset (over 3.4M reviews) lacks protected attributes; following prior work, we synthetically assign age-bin, gender and occupation labels from the ML-1M attribute distribution. These attributes are used for fairness calibration, but do not reflect real user demographics. Thus, fairness evaluation on Amazon measures algorithmic stability under synthetic attribute perturbations rather than real-world demographic disparities. We sample 3,750 instances (2,625 calibration; 1,125 evaluation) due to sparsity. Prompt features use only movie titles, as the Amazon data lacks a canonical "genre" field.

4.2.2 Experimental Setup & Computational Resources

To ensure statistical significance, we repeated all experiments with three distinct random seeds and report the mean and standard deviations for each metric. We evaluate three methods across datasets and tasks: (i) a zero-shot baseline, (ii) a zero-shot baseline with a static fairness-aware system prompt, and (iii) the FACTER algorithm. Experiments are conducted for both open-ended generation and re-ranking settings. Crucially, both our reproduction and the original study employ a multi-target evaluation regime. Because we lacked details regarding the exact protocol from the original study, we heuristically defined the relevant target pool as the user's immediate next 10 reviews. We discuss the mathematical implications of original utility claims under a single-target regime and the lack of target-set specifications in appendix B.6.

FACTER is evaluated over three iterations per dataset and seed, with each iteration comprising a full test-set pass and retaining previously injected prompts and detected violations. Following the original study, we report results from the third iteration, while static baselines are evaluated in a single pass. To assess model dependence, we vary the underlying LLM for FACTER on MovieLens-1M.

All experiments were conducted on the national Supercomputer, using a node with an NVIDIA A100 GPU, 18 Intel Xeon CPU cores, and 120GB of RAM. A summary of the energy consumption and environmental impact metrics is reported in Section 6.5, and a detailed breakdown can be found in Appendix D.

4.2.3 Task Extension: Re-Ranking

In the original FACTER formulation Fayyazi et al. (2025), recommendation is evaluated in an open-ended generation setting: the LLM generates item titles without a constrained candidate set, and performance is measured by matching the generated output to a set of relevant items. This setup requires the model to implicitly retrieve the correct catalogue item from open generation. To disentangle fairness-aware ranking behaviour from open-ended catalogue search, we introduce a constrained re-ranking task in which the model ranks a fixed candidate pool.

Task Formalization. We formulate recommendation as candidate re-ranking. Given a candidate set $\mathcal{C}$ containing a set of relevant ground-truth items $\mathcal{R}_{\text{rel}}$ (multi-target) and $|\mathcal{C}|-|\mathcal{R}_{\text{rel}}|$ uniformly sampled negatives (excluding user history), the model outputs a ranked list. We evaluate the top $k = 10$ items $\mathcal{R}$. Candidates are shuffled per query to prevent positional and "lost-in-the-middle" biases. This setup assumes successful candidate retrieval. In practice, a high-recall retriever (e.g., BM25 (Robertson et al., 1994) or vector search) reduces the corpus to a manageable subset. By including the relevant set $\mathcal{R}_{\text{rel}}$ in $\mathcal{C}$, we decouple evaluation from retriever-specific limitations and focus on the LLM's fairness-aware ranking behaviour.

Algorithmic Adaptation. We adapt FACTER to this constrained setting by adding ranking-specific instructions and the explicit candidate list to the prompt (Appendix C). This lets us test conformal thresholding and prompt repair when the search space is fixed, rather than open-ended generation.

4.2.4 Models and Implementation

To support a robust evaluation of the FACTER framework, we use three representative open-weight LLMs: Llama 2-7B (Touvron et al., 2023), Llama 3-8B (Grattafiori et al., 2024), and Mistral-7B (Jiang et al., 2023). We use the fine-tuned MPNet sentence transformer from the original study[1] for all semantic representations.

[1] https://huggingface.co/JJTsao/fine-tuned_movie_retriever-all-mpnet-base-v2

We compare FACTER against baselines chosen to isolate the effect of its adaptive thresholding. These include a **Neutral Zero-Shot** baseline with no fairness-related instructions, and **Fair Zero-Shot (Ours)** (Appendix C.1), a stronger baseline that tests static fairness prompting without FACTER's conformal adaptation. We also include UP5 (Hua et al., 2024), which extends a recommendation foundation model by prepending learned prefix embeddings to the input representation, enabling bias mitigation through parameter-efficient prefix tuning. UP5 serves as the primary fairness-aware comparator in the original FACTER evaluation. However, we were unable to reproduce UP5 due to several technical limitations detailed in Appendix B.5. We therefore report UP5 results from the original paper as reference values only, and provide the corresponding prompt templates and inputs in Appendix C. The inability to reproduce UP5 does not affect our examination of the core claims (**C1 - C4**). We focus our comparison of FACTER against zero-shot baselines, isolating the impact of its specific mechanisms, instead of a comparison against another state-of-the-art method utilizing completely different mechanisms.

4.2.5 Hyperparameters

We do not perform additional hyperparameter search and adopt the reported tuned values from the original study: fairness penalty $\lambda = 0.7$, threshold decay $\gamma = 0.95$, similarity threshold $\tau_\rho = 0.9$, conformal level $\alpha = 0.1$, and FIFO violation buffer size $M = 50$ and a rule-mining threshold $n = 3$. Embedding and generation batch sizes were selected based on hardware constraints (see Appendix B.2 full list of hyperparameters).

Consistent with the original implementation, we do not enforce the context similarity threshold τ_x (absent from both paper and released code), and solely control "fair" neighbourhoods through τ_ρ. We map LLM outputs to catalogue items using a cosine similarity threshold of 0.65, following the released implementation. The recommendation list length is fixed at $k = 10$.

Parameters introduced by our ranking-based extension were determined heuristically and held constant across experiments. The candidate set size $|\mathcal{C}| = 40$, providing sufficient ranking difficulty while remaining computationally feasible given our available compute budget.

4.3 Evaluation

4.3.1 Metrics

Our evaluation uses a dual-objective framework to assess both fairness and recommendation utility. We align with the authors' released implementation where possible, noting that certain implementation choices differ from the mathematical formalisations in the paper; Appendix B.1 summarises these differences.

Fairness Metrics. FACTER motivates embedding-based fairness evaluation by treating large semantic changes in generated outputs (when only protected attributes are perturbed) as potential indicators of bias. It therefore defines a minimal-attribute-change property: modifying only the sensitive attribute $\boldsymbol{a} \to \boldsymbol{a'}$ while holding non-protected features x fixed should not yield large output discrepancies, measured via embedding-space distance.

The Sensitive-to-Neutral Similarity Ratio (SNSR) metric summarises group-level disparities, while Counterfactual Fairness Ratio (CFR) assesses responsiveness to counterfactual flips of protected attributes. When combined, local fairness controls and global aggregate metrics provide a more complete picture of fairness behaviour (Fayyazi et al. (2025)).

Let $\mathcal{G}$ be the set of protected groups and $\mathbf{h}_g$ the centroid of pooled recommendation embeddings for group $g \in \mathcal{G}$. We define the pairwise cosine distance as $d_{\cos}(g_i, g_j) = 1 - \mathbf{h}_{g_i} \cdot \mathbf{h}_{g_j}$. SNSR is calculated over the set of all unique group pairs $\mathcal{P}$ as:

$$\text{SNSR} = \max_{\{g_i, g_j\} \in \mathcal{P}} d_{\cos}(g_i, g_j) \tag{7}$$

Although lower values indicate more uniform treatment, the original implementation restricts evaluation to multi-attribute groups with $n \geq 30$. This threshold, combined with the pooling of per-example Top-k

embeddings prior to group averaging, can over-smooth results and mask fine-grained disparities. Furthermore, this restriction poses a structural risk: the metrics may become incomputable if the dataset lacks sufficient groups meeting this requirement. We address this by computing SNSR per attribute and on the multi-attribute intersection. Additionally we evaluate the CFR, defined as the average distance between original and counterfactually perturbed recommendation embeddings:

$$\text{CFR} = \mathbb{E}_{x\sim\mathcal{D}}\left[\|f_\phi(\hat{y}) - f_\phi(\hat{y}_{\neg s})\|_2\right] \tag{8}$$

We use L_2 distance for CFR to match the paper's definition. By contrast, the SNSR metric follows the cosine-distance proxy used in the released implementation, because the paper's SNSR formulation is not directly reproducible from the released artifacts. Our CFR implementation supports both multi- and single-attribute assessments. The former involves flipping the gender attribute and resampling age and occupation to distinct values to form a valid counterfactual state, the latter modifies a single attribute ceteris paribus.

Furthermore, we monitor the *Violation Count* to assess the reliability of the thresholding mechanism. This metric represents the number of instances where the non-conformity score S_i exceeds the threshold Q_α. Given the ambiguity in the original study regarding the precise point of measurement, we report the violation count using two distinct thresholds: the sample-corrected post-calibration threshold ($Q_\alpha^{(0)} + \frac{C}{\sqrt{n}}$), and the dynamic threshold ($Q_\alpha^{(t)}$). This reporting ensures a comprehensive view of how the model manages the fairness-utility trade-off across different stages of the adaptive process.

Utility and Calibration Metrics. We report Recall@k, and NDCG@k (Järvelin & Kekäläinen, 2002), computed as:

$$\text{Recall@k} = \frac{|\mathcal{R}_{\text{rel}} \cap \mathcal{R}_k|}{|\mathcal{R}_{\text{rel}}|}, \quad \text{and} \quad \text{NDCG@k} = \frac{1}{\text{IDCG@k}} \sum_{r=1}^{k} \frac{2^{\text{rel}(r)} - 1}{\log_2(r+1)}. \tag{9}$$

where $\mathcal{R}_{\text{rel}}$ is the set of relevant target items. Additionally we compute Valid@k, which measures the fraction of the top-k generated recommendations that can be mapped to a valid catalogue item in our database.

5 Results

We assess the reproducibility of FACTER by comparing its original open-ended generation setting with our re-ranking extension, and by isolating adaptive thresholding and dynamic prompting. Accordingly, the open-generation results are used to assess the original claims, whereas the re-ranking results are used to analyse a separate extension. Comparisons to UP5 serve only as a reference to the original study's trends, as reproduction was not possible. Across all tables, bold values indicate the best-performing method per dataset and metric within each column group, and reproduced values are reported as mean (SD) over three random seeds.

5.1 Original Setup: Reproducibility of the Open-Ended Generation Task

Utility Divergence. We observe a gap in recommendation utility under strict open-ended generation (Table 1), where our reproduction yields near-zero NDCG@10 and Recall@10 values on both datasets despite the original study's strong results. In this low-utility regime, FACTER's marginally lower scores align with original findings of limited utility loss, though the absolute magnitudes suggest a potential floor effect caused by the unconstrained search space. This limitation is further evidenced by the Valid@k metric, which shows that 88–90% of generated recommendations on Amazon and 72–75% on ML-1M can be mapped back to the item catalogue (see Appendix A for a detailed overview and discussion). This indicates the model frequently produces hallucinations or invalid identifiers that may contribute to the observed utility gap.

Fairness Trends. Despite the low absolute utility in open generation, several fairness-related trends align qualitatively with the original study (Table 1). The original study positioned FACTER as a significant improvement over baseline violations; our reproduction supports this trend, showing a reduction from 42.7

Table 1: Reproduction (Open-Ended Generation): Original (**O**) vs. reproduced (**R**) results using the **Llama-3-8B** backbone. *Neutral* denotes the zero-shot baseline, *Fair* the fair zero-shot baseline. FACTER scores are reported at iteration 3.

		NDCG@10 ↑		**Recall@10 ↑**		**SNSR[1] ↓**		**CFR ↓**		**Constraint Violations ↓**		
Data	**Model**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R** ($V_{Q_\alpha}^{(t)}$)[4]	**R** ($V_{Q_\alpha}^{(0)}$)
Movie Lens 1M	Neutral	**.458**	**.035 (.001)**	**.402**	**.031 (.001)**	.083	.037 (.018)	.742	**.667 (.005)**	112	–	42.7 (5.8)
	Fair[2]	–	.030 (.002)	–	.027 (.002)	–	.038 (.008)	–	.683 (.004)	–	–	41.0 (4.4)
	UP5[3]	.427	–	.381	–	.049	–	.613	–	28	–	–
	FACTER	.445	.029 (.001)	.389	.026 (.000)	**.041**	**.032 (.005)**	**.591**	.702 (.009)	**5**	**9.0 (0.0)**	**35.3 (6.4)**
Amazon Movies & TV	Neutral	**.351**	**.004 (.000)**	**.317**	**.005 (.001)**	.121	–	.814	**.879 (.018)**	198	–	107.7 (1.5)
	Fair[2]	–	**.004 (.001)**	–	.004 (.000)	–	–	–	.906 (.023)	–	–	**91.3 (8.4)**
	UP5[3]	.328	–	.294	–	.067	–	.721	–	63	–	–
	FACTER	.339	.003 (.001)	.301	.003 (.000)	**.053**	–	**.634**	.922 (.001)	**18**	**3.0 (1.7)**	110.7 (11.0)

[1]SNSR on Amazon could not be reported due to the multi-attribute group constraint (see appendix A for more details).
[2]Fair scores are not reported in the original results as the Fair zero-shot baseline was not implemented.
[3]UP5 scores are not reported in the reproduction due to missing checkpoints and code incompatibilities.
[4]Violations under $V_{Q_\alpha}^{(t)}$ are only reported for FACTER, as the static baselines do not employ an iterative threshold update.

(Neutral) to 9.0 (FACTER) on ML-1M, and 107.7 to 3.0 on Amazon, when measured via the adaptive threshold ($V_{Q_\alpha}^{(t)}$). FACTER achieves the lowest SNSR on ML-1M (0.032); while this is in line with the original findings, the decrease from the Neutral (0.037) and Fair (0.038) baselines is considerably less pronounced in our reproduction than original results. Furthermore, we observe a discrepancy in counterfactual stability: while the original study reports that FACTER improves the Counterfactual Fairness Ratio (CFR), our reproduction shows a slight increase in CFR for FACTER (0.702) compared to the Neutral baseline (0.667) on ML-1M. On the Amazon dataset, this gap is even more pronounced. Notably, SNSR metrics could not be reported for the Amazon dataset, and the SNSR metrics for ML-1M are limited to two groups due to the computational constraints of the multi-attribute group membership restriction mentioned previously. See Appendix A for a detailed discussion of these limitations, alongside a single-attribute analysis.

Table 2: Reproduction (Open-Ended Generation): Model-wise comparison on ML-1M. **O**: original, **R**: reproduced.
Note: Original values for Zero-shot are not reported as they were not provided in the original study.

		NDCG@10 ↑		**Recall@10 ↑**		**SNSR ↓**		**CFR ↓**		**Constraint Violations ↓**		
Architecture	**Model**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R** ($V_{Q_\alpha}^{(t)}$)	**R** ($V_{Q_\alpha}^{(0)}$)
LLaMA3-8B	Neutral	-	**.035 (.001)**	-	**.031 (.001)**	-	.037 (.018)	-	**.666 (.005)**	-	-	42.7 (5.8)
	FACTER	.440	.029 (.001)	.383	.026 (.000)	.039	**.032 (.005)**	.576	.702 (.009)	3	**9.0 (0.0)**	**35.3 (6.4)**
LLaMA2-7B	Neutral	-	**.028 (.003)**	-	**.026 (.003)**	-	**.017 (.002)**	-	**.640 (.004)**	-	-	**45.3 (9.1)**
	FACTER	.444	.022 (.002)	.391	.020 (.002)	.041	.023 (.008)	.595	.712 (.016)	5	**8.3 (0.6)**	60.0 (9.6)
Mistral-7B	Neutral	-	**.031 (.002)**	-	**.026 (.001)**	-	.052 (.007)	-	**.741 (.021)**	-	-	39.3 (7.5)
	FACTER	.451	.026 (.001)	.397	.022 (.001)	.043	**.041 (.005)**	.602	.785 (.031)	7	**7.3 (1.2)**	**37.0 (4.4)**

Model Agnosticism. The results in Table 2 provide empirical support for FACTER's architecture-agnostic nature, as LLaMA3-8B, LLaMA2-7B, and Mistral-7B exhibit highly consistent behavioral trends. Across all backbones, FACTER's utility is similar, with NDCG@10 scores confined to .022–.029, while consistently trailing the Zero-shot baseline which achieves superior utility at the cost of higher violations. This agnosticism extends to fairness metrics: FACTER achieves a uniform reduction in constraint violations ($V_{Q_\alpha}^{(t)}$) to between 7.3 and 9.0 compared to the baseline's ($\sim 39-45$), while counterfactual instability remains consistently high across models. Notably, while absolute utility diverges from the original study (.03 vs. .44), the consistent delta between FACTER and the baseline across all backbones reinforces that the framework's performance profile and fairness-utility trade-offs are fundamentally architecture-independent.

The results in Table 2 show qualitatively similar trends across the three tested backbones on ML-1M, as LLaMA3-8B, LLaMA2-7B, and Mistral-7B exhibit highly consistent behavioral trends. Across all backbones, FACTER's utility is similar, with NDCG@10 scores confined to .022–.029, while consistently trailing the Zero-shot baseline which achieves superior utility at the cost of higher violations. This consistency extends to the reported fairness-related metrics: FACTER achieves a uniform reduction in constraint violations ($V_{Q_\alpha}^{(t)}$) to between 7.3 and 9.0 compared to the baseline's ($\sim 39 - 45$), while counterfactual instability remains consistently high across models. Notably, while absolute utility diverges from the original study (.03 vs. .44), the consistent delta between FACTER and the baseline across all backbones is consistent with a similar performance profile and fairness-utility trade-off on these tested backbones.

Iterative Fairness Refinement. We observe improvement over iterations in adaptive-threshold violations ($V_{Q_\alpha}^{(t)}$) . As illustrated in Figure 1, violations under the adaptive threshold decrease monotonically across the three iterations. While the original study reports a steep decline from initially high violations, our reproduction shows a lower initial violation count, resulting in a shallower but consistent downward trend. However, it is crucial to note that this monotonic decrease is isolated to the internal conformal violations. As shown in Table 1, this iterative reduction in internal violations does not translate to improvements in external, structural fairness metrics like CFR and SNSR, which fail to meaningfully improve over the zero-shot baselines by the final iteration.

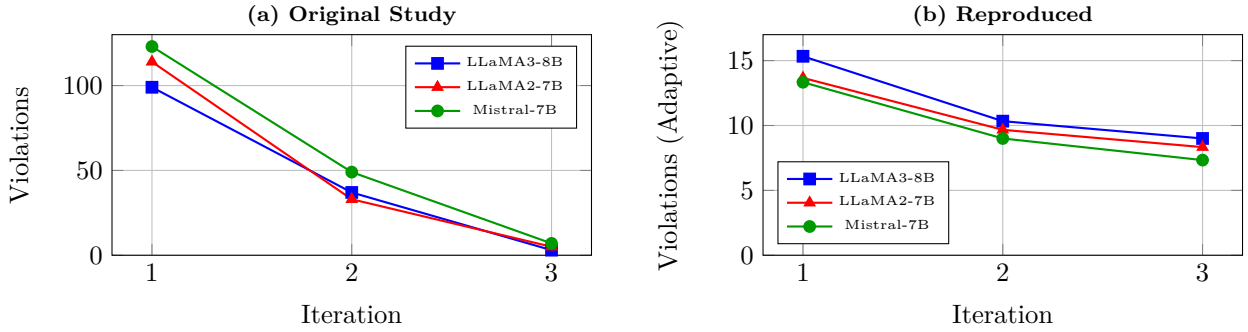

Figure 1: Violation reduction over iterations on ML-1M for the open-generation task and the Llama3 model. **(a)** Original (taken directly from Fayyazi et al. (2025)) shows steep initial decline. **(b)** Reproduction shows a lower start with a shallower decrease.

5.2 Evaluation under Re-Ranking Extension

Utility under Re-Ranking. In our separate re-ranking extension, evaluating (Table 3). By constraining the search space to a fixed candidate list, the Neutral baseline achieves an NDCG@10 of 0.466 and a Recall@10 of 0.433 on the ML-1M dataset. FACTER closely following this with an NDCG@10 of 0.438 and a Recall@10 of 0.407. This indicates that while utility is near-zero in unconstrained generative settings, the models are capable of identifying relevant items when the task is reformulated as a discriminative ranking problem.

Fairness Preservation. Under re-ranking, FACTER maintains low adaptive violation counts ($V_{Q_\alpha}^{(t)}$) within this higher-utility re-ranking regime (Table 3). On the ML-1M dataset, adaptive violations decrease to 7.7 compared to 55.7 observed in the Fair baseline. A similar trend is observed on the Amazon dataset, where FACTER yields only 1.7 adaptive violations. This suggests that the iterative repair loop remains functional when the model is constrained to a candidate list, successfully filtering biased candidates without significantly degrading ranking quality. In contrast to the reduction in violations, proxy fairness metrics such as SNSR and CFR show marginal differences across methods. For instance, FACTER yields a CFR of 0.553 on ML-1M, which is slightly worse than the Neutral (0.533) and Fair (0.534) baselines. This divergence indicates that while FACTER successfully satisfies its internal conformal bounds, it fails to translate this into superior group-level semantic parity or broad counterfactual stability compared to simple static baselines.

Table 3: Extension (Re-Ranking): Performance and Fairness on ML-1M and Amazon datasets using the **Llama-3-8B** backbone. FACTER scores are reported at iteration 3.

Dataset	Model	NDCG@10 ↑	Recall@10 ↑	SNSR [1] ↓	CFR ↓	$\mathbf{V}^{(t)}_{\mathbf{Q}_\alpha}$ [2] ↓	$\mathbf{V}^{(0)}_{\mathbf{Q}_\alpha}$ ↓
ML-1M	Neutral	**.466 (.002)**	**.433 (.004)**	.026 (.003)	**.533 (.007)**	–	61.3 (7.1)
	Fair	.456 (.006)	.426 (.004)	**.024 (.006)**	.534 (.006)	–	**55.7 (4.9)**
	FACTER	.438 (.007)	.407 (.005)	**.024 (.003)**	.553 (.010)	**7.7 (1.5)**	60.3 (4.2)
Amazon Movies & TV	Neutral	.448 (.005)	.419 (.007)	–	.715 (.028)	–	92.0 (4.4)
	Fair	**.458 (.003)**	**.431 (.003)**	–	**.681 (.012)**	–	91.3 (3.2)
	FACTER	.451 (.004)	.423 (.002)	–	.707 (.005)	**1.7 (1.2)**	**88.7 (5.8)**

[1]SNSR on Amazon could not be reported due to the multi-attribute group constraint (see appendix A for more details).
[2]Violations under $V^{(t)}_{Q_\alpha}$ are only reported for FACTER, as the static baselines do not employ an iterative threshold update.

5.3 Deconstructing the Fairness Mechanism

We isolate the source of fairness improvements by analysing the adaptive thresholding mechanism and the impact of dynamic prompting versus static instructions.

Threshold Dynamics. Violation reductions under FACTER are highly threshold-dependent (Table 1 and 3). While violations appear to decrease under the adaptive threshold ($V^{(t)}_{Q_\alpha}$), they remain high and comparable to the baselines when measured against the fixed, population-calibrated threshold ($V^{(0)}_{Q_\alpha}$). This suggests that the reported improvement is primarily driven by the adaptive mechanism iteratively raising the threshold to maintain statistical coverage. As the threshold becomes more permissive over successive iterations, it accepts recommendations with higher non-conformity scores, thereby reducing the recorded violation count.

Dynamic vs. Static Prompting We compared the fully dynamic FACTER loop against our *Fair Zero-Shot* to isolate the impact of the iterative repair mechanism across both datasets (Table 3). In the ML-1M Re-Ranking task, the Fair baseline achieved an SNSR of 0.024 and 55.7 fixed-threshold violations ($V^{(0)}_{Q_\alpha}$), performing comparably to FACTER (0.024 SNSR and 60.3 violations under $V^{(0)}_{Q_\alpha}$). Similarly, on the Amazon dataset, the Fair baseline yielded a CFR of 0.681 and 91.3 fixed violations, while FACTER resulted in a CFR of 0.707 and 88.7 fixed violations. This parity across SNSR, CFR, and $V^{(0)}_{Q_\alpha}$ suggests that in constrained ranking scenarios, a well-crafted static system prompt yields fairness benefits similar to the computationally complex online monitoring loop.

6 Discussion

Our reproduction of FACTER yields a complex verdict. While we reproduce the framework's core mechanism, in that conformal thresholding can dynamically bound fairness violations, we observe significant divergences in recommendation utility and the practical necessity of the online prompt repair loop. This section deconstructs these findings, specifically addressing the utility gap, technical discrepancies in threshold dynamics, and the reproducibility landscape.

6.1 The Utility Gap: Evaluation Protocol vs. Task Difficulty

We observe a critical disparity between the reported open-generation utility for ML-1M (NDCG@10 $\approx$ 0.44) and the near-zero values obtained in our strict reproduction (NDCG@10 $\approx$ 0.03). As we utilised identical model backbones, we hypothesise that this gap stems from implementation-sensitive details in the evaluation protocol rather than model capacity.

Zero-shot open-ended recommendation is inherently difficult not only because of the vast search space, but because the model must also realise a relevant item in a form that can be mapped back to a unique catalogue entry. This difficulty is consistent with prior generative recommendation literature. Zero-Shot NIR explicitly identifies the huge recommendation space as a core obstacle and therefore narrows the search space through candidate set construction (Wang & Lim, 2023). Likewise, P5 studies recommendation as language generation but includes candidate-based direct recommendation formulations (Geng et al., 2022), while GPT4Rec generates search queries and then retrieves items with a search engine rather than relying on unconstrained title emission alone (Li et al., 2023). More broadly, the recent survey by Li et al. notes that real-world recommender systems typically rely on multi-stage filtering, that recommendation settings often require precise item identifiers to avoid hallucinations, and that matching-based evaluation can be overly harsh for generated outputs (Li et al., 2024). Taken together, these works suggest that near-zero scores under a strict open-generation-and-matching protocol are not inherently out of place, even though they do not by themselves establish the exact protocol used in the original FACTER paper. While both studies use a multi-target evaluation regime, our protocol heuristically defined the relevant target pool as the users' immediate next 10 reviews. The original study's construction of the relevant set is not fully specified in the paper and public code.

If a more permissive definition of "relevance" was used, then the performance gap could be attributed to our "next 10" heuristic, forcing the model to hit a narrower target within the open-ended search space. The restoration of utility to reported levels (≈ 0.44 NDCG) upon introducing our proposed re-ranking extension is consistent with this interpretation. However, because this extension changes the task formulation, it does not by itself reconcile the original open-generation result.

We propose this constrained re-ranking setup as a more reasonable and practical formulation for real-world recommendation tasks. While open-ended generation may suffice for general, platform-agnostic inquiries (e.g., broad movie ideas), it becomes highly impractical within closed ecosystems (e.g. Netflix), or in strictly constrained domains (e.g. recruitment), where recommendations must map to a specific, available inventory. Therefore, we emphasise that while the FACTER framework is viable, its utility claims and real-world applicability likely depend on testing within a constrained search space rather than pure, unconstrained open generation.

6.2 Threshold Dynamics: Risk Management vs. Bias Erasure

Our dual-threshold evaluation suggests that FACTER operates primarily as a *risk management* system rather than a *bias erasure* mechanism. While violations drop significantly against the adaptive threshold ($Q_\alpha^{(t)}$), they persist at high levels against the static, population-calibrated threshold ($Q_\alpha^{(0)}$), remaining comparable to the Neutral baseline.

We initially hypothesised that this persistence stemmed from a "null penalty" artefact where strict neighbourhood definitions rendered the fairness penalty (Δ) negligible. However, component analysis (Appendix A, Table 4) refutes this: the fairness penalty actively contributes approximately 50% to the non-conformity score of violating instances (e.g., 49.65% in Iteration 1). Instead, our report suggests the prompt repair mechanism is not reliable at reducing the nonconformity scores of recurring inputs relative to the fixed baseline $Q_\alpha^{(0)}$. If the "AVOID" constraints were effective, the non-conformity scores (S) of recurring inputs should decrease to satisfy the static baseline $Q_\alpha^{(0)}$. Contrary to this expectation, the average non-conformity score for violations increases from 1.492 in Iteration 1 to 1.703 in Iteration 3.

Decomposing this score reveals that while the predictive error (d) improves ($0.881 \rightarrow 0.804$), the fairness penalty (Δ) worsens significantly ($0.872 \rightarrow 1.283$). This suggests the model prioritises predictive confidence over the injected fairness constraints, effectively ignoring the semantic instructions. Consequently, FACTER satisfies its statistical coverage guarantee primarily by adaptively raising the threshold $Q_\alpha^{(t)}$, which can reduce the number of violations under the dynamical threshold, rather than by clearly correcting the model's behaviour under the fixed baseline.

This systemic inertia directly explains the unexpected degradation observed in the proxy fairness metrics. The explicit negative constraints injected during prompt repair appear to destabilise the LLM's counterfac-

tual stability. Rather than smoothing the representation space to render it agnostic to protected attributes, these rigid instructions force the model into erratic, semantically distant regions of the catalogue when demographic inputs are perturbed. Consequently, the CFR increases under the FACTER framework, and the SNSR fails to improve meaningfully beyond static baselines.

Ultimately, this helps explain why the static *Fair Zero-Shot* (Section 6.3) achieves parity with the full FACTER framework. Specifically, the computational overhead of the dynamic repair loop yields no tangible reduction in semantic bias compared to a fixed system instruction.

All fairness metrics in this study (SNSR, CFR, and the nonconformity score S) are computed in the representation space of a domain-specific fine-tuned MPNet encoder. If this encoder has learned biases from its training data (Caliskan et al., 2017), these biases will propagate into both calibration thresholds and evaluation scores, as all fairness metrics in FACTER are computed within this representation space. The original study's robustness guarantee addresses bounded random perturbations but not systematic encoder bias. Our study inherits this limitation.

Our re-ranking formulation assumes a candidate set C that already contains the relevant items R_{rel}. In deployment, an upstream retriever would produce this set, and demographic bias at the retrieval stage (e.g. systematically under-retrieving items relevant to certain groups) cannot be corrected by downstream re-ranking alone. End-to-end fairness can thus only be achieved by interventions at every stage of the recommendation pipeline (Hsu et al., 2025).

6.3 Efficacy of Static vs. Dynamic Prompting

For the re-ranking task, our ablation study contradicts the original study's central premise and reveals that the iterative prompt repair loop provides negligible improvement in fairness. As detailed in Tables 1 and 3, in terms of CFR and $V_{Q_\alpha}^{(0)}$, the Fair baseline achieved fairness parity comparable to the full FACTER loop within the observed variance, by simply utilising a static prompt with explicit fairness constraints. To clarify this methodological distinction, while the Neutral baseline provides no fairness guidance, the Fair Zero-Shot prompt actively instructs the LLM to recommend based on preference signals rather than demographics and to avoid reinforcing stereotypes (see Appendix C.1). FACTER extends this static foundation by iteratively appending dynamically generated "AVOID" rules and nonconformity score targets learned from past violations (see Appendix C.2).

These results indicate that within the constrained topology of ranking decision spaces, the explicitness of the static base instruction dominates the periodicity of the dynamic intervention, and that the significant computational overhead and inference latency introduced by FACTER's monitoring module yield diminishing returns. We hypothesise that the dynamic repair mechanism is likely over-engineered for discriminative ranking but may remain necessary for unconstrained generative tasks where stochastic drift is higher (Borah et al., 2025). Finally, the observed utility degradation in the Fair Zero-Shot baseline relative to the Neutral baseline confirms the existence of an "alignment tax" (Korkmaz et al., 2025), suggesting that strict fairness constraints inevitably compete with relevance signals in zero-shot contexts.

6.4 Reproducibility Landscape

Reproducibility was facilitated by the method's training-free nature and the authors' timely and extensive responses, which comprehensively addressed our methodological questions. However, the public codebase does not fully specify all the details needed for a faithful implementation. Consequently, faithful replication required resolving significant ambiguities via this private correspondence, particularly regarding a multi-target evaluation regime that is not documented in the paper or public code, mathematical inconsistencies in threshold updates (Equation 11 vs. Appendix Theorem .2 in the original study), and SNS metrics. Where code and text diverged (e.g. in CFR's distance metric), we adhered to the study's textual definitions.

Our evaluation faces specific constraints regarding baselines and hyperparameters. We were unable to reproduce the UP5 baseline due to missing checkpoints and code incompatibilities, preventing a direct comparison with the original ground truth. Additionally, computational limits necessitated fixing the candidate re-ranking count at 40 based on heuristics, leaving the sensitivity of this parameter unverified.

Furthermore, our adherence to the original experimental protocol propagated limitations intrinsic to the source work. Because Amazon lacks demographic metadata, protected attributes are synthetically assigned; therefore, results on this dataset evaluate algorithmic stability under attribute perturbations rather than real demographic fairness. Similarly, the original requirement to aggregate fairness metrics into groups of $n \geq 30$ likely masks intersectional variance and prohibits localised evaluation. We provide a high-level summary of all reproducibility obstacles and our corresponding implementation choices in Appendix B.6.

6.5 Environmental Impact

Experiments incurred 136 compute hours and 8.7 kg CO_2eq emissions, which is roughly equivalent to the emissions produced by driving an average gasoline-powered passenger vehicle one-way from Amsterdam to Rotterdam. Calculations were made using the `codecarbon`[2] Python library. We note that water consumption could not be estimated as the facility does not disclose Water Usage Effectiveness (WUE). A detailed breakdown of the environmental impact of our experiments can be found in Appendix D.

7 Conclusion and Future Work

Our reproduction validates FACTER's core theoretical contribution of utilising conformal prediction to bound fairness violations, while highlighting important nuances regarding its practical deployment and utility.

Our primary contributions are: (1) a best-effort, open-source re-implementation of the FACTER pipeline under documented reconstruction choices; (2) a Fair Zero-Shot baseline that helps isolate the contribution of the iterative repair loop; (3) a separate re-ranking extension that enables mechanism analysis in a more controlled and practical task formulation, without resolving the original open-generation utility claim (**C2**); and (4) a granular component analysis of the fairness mechanism to understand the dynamics of the online repair loop.

Within the tested datasets and backbones, these results provide partial support for **C3** and **C4**, while the original open-generation utility claim **C2** remains unreproduced under our matched setting.

Regarding fairness effectiveness (**C1**), our analysis reveals that the observed adaptive-threshold violation reductions are primarily driven by adaptive threshold relaxation ($Q_\alpha^{(t)}$) rather than absolute bias correction. Component analysis of the non-conformity scores demonstrates that the fairness penalty (Δ) for violating instances increases over iterations despite prompt injection, indicating that the "AVOID" constraints may struggle to consistently override the model's predictive distribution, rendering the dynamic repair loop ineffective. Consequently, in constrained ranking tasks, a static fairness prompt can achieve competitive results on the reported fairness proxies and fixed-threshold counts with the dynamic repair loop, offering a computationally efficient alternative.

Our study is not without limitations. First, due to ambiguities in the original public materials, our multi-target evaluation relied on a heuristically defined relevance set (the user's next 10 interactions), which may differ from the original authors' exact protocol. Second, computational constraints necessitated fixing the candidate pool for the re-ranking task to 40 items without conducting a broader sensitivity analysis on candidate set size. Third, we were unable to reproduce the UP5 baseline due to missing pre-trained checkpoints, relying instead on reported values for comparison. Finally, by strictly adhering to the original study's hyperparameters to isolate the reproduction, we did not explore whether dataset-specific tuning could better balance the predictive and fairness components of the non-conformity score.

Future research should move beyond soft prompt constraints, exploring mechanisms that enforce semantic bounds during decoding or utilisation of "blind" inference that does not rely on ground-truth labels for online scoring. Additionally, a qualitative analysis of the recommendations generated for violations during later iterations could yield deeper insights into persistent bias themes. Finally, further work is needed to calibrate the strength of fairness penalties against predictive confidence to prevent the "score conservation" effect observed in this study.

[2]https://pypi.org/project/codecarbon/

References

Anastasios N. Angelopoulos and Stephen Bates. Conformal Prediction: A Gentle Introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, March 2023. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000101. URL https://www.emerald.com/ftmal/article/16/4/494/1332423/Conformal-Prediction-A-Gentle-Introduction.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pp. 610–623, 2021. doi: 10.1145/3442188.3445922. URL https://dl.acm.org/doi/10.1145/3442188.3445922.

Abhilekh Borah, Chhavi Sharma, Danush Khanna, Utkarsh Bhatt, Gurpreet Singh, Hasnat Md Abdullah, Raghav Kaushik Ravi, Vinija Jain, Jyoti Patel, Shubham Singh, Vasu Sharma, Arpita Vats, Rahul Raja, Aman Chadha, and Amitava Das. Alignment quality index (AQI) : Beyond refusals: AQI as an intrinsic alignment diagnostic via latent geometry, cluster divergence, and layer wise pooled representations. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2888–2947, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.145. URL https://aclanthology.org/2025.emnlp-main.145/.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, April 2017.

Lihu Chen, Gael Varoquaux, and Fabian Suchanek. Learning High-Quality and General-Purpose Phrase Representations. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 983–994, St. Julian's, Malta, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-eacl.66. URL https://aclanthology.org/2024.findings-eacl.66.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pp. 214–226, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311151. doi: 10.1145/2090236.2090255. URL https://doi.org/10.1145/2090236.2090255.

Arya Fayyazi, Mehdi Kamal, and Massoud Pedram. FACTER: Fairness-aware conformal thresholding and prompt engineering for enabling fair LLM-based recommender systems. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16407–16422. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/fayyazi25a.html.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 299–315, Seattle WA USA, September 2022. ACM. ISBN 978-1-4503-9278-5. doi: 10.1145/3523227.3546767. URL https://dl.acm.org/doi/10.1145/3523227.3546767.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, January 2016. ISSN 2160-6455, 2160-6463. doi: 10.1145/2827872. URL https://dl.acm.org/doi/10.1145/2827872.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pp. 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web

Conferences Steering Committee. ISBN 9781450341431. doi: 10.1145/2872427.2883037. URL https://doi.org/10.1145/2872427.2883037.

Brian Hsu, Cyrus DiCiccio, Natesh S. Pillai, and Hongseok Namkoong. From models to systems: A comprehensive framework for ai system fairness in compositional recommender systems. In Miriam Rateike, Awa Dieng, Jamelle Watson-Daniels, Ferdinando Fioretto, and Golnoosh Farnadi (eds.), *Proceedings of the Algorithmic Fairness Through the Lens of Metrics and Evaluation*, volume 279 of *Proceedings of Machine Learning Research*, pp. 8–37. PMLR, 14 Dec 2025. URL https://proceedings.mlr.press/v279/hsu25a.html.

Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, Zelong Li, and Yongfeng Zhang. UP5: Unbiased Foundation Model for Fairness-aware Recommendation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1899–1912, St. Julian's, Malta, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.114. URL https://aclanthology.org/2024.eacl-long.114.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825. Version Number: 1.

Mahammed Kamruzzaman and Gene Louis Kim. Prompting techniques for reducing social bias in LLMs through system 1 and system 2 cognitive processes. In Galia Angelova, Maria Kunilovskaya, Marie Escribe, and Ruslan Mitkov (eds.), *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pp. 511–520, Varna, Bulgaria, September 2025. INCOMA Ltd., Shoumen, Bulgaria. doi: 10.26615/978-954-452-098-4-060. URL https://aclanthology.org/2025.ranlp-1.60/.

Buse Sibel Korkmaz, Rahul Nair, Elizabeth M. Daly, and Antonio del Rio Chanona. Paying alignment tax with contrastive learning, 2025. URL https://arxiv.org/abs/2505.19327.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gérard G. Medioni. GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation. In Surya Kallumadi, Yubin Kim, Tracy Holloway King, Shervin Malmasi, Maarten de Rijke, and Jacopo Tagliabue (eds.), *Proceedings of the 2023 SIGIR Workshop on eCommerce co-located with the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023), Taipei, Taiwan, July 27, 2023*, volume 3589 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL https://ceur-ws.org/Vol-3589/paper_2.pdf.

Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 10146–10159, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.886/.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. Fairness-guided Few-shot Prompting for Large Language Models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 43136–43155. Curran

Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8678da90126aa58326b2fc0254b33a8c-Paper-Conference.pdf.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pp. 43–52, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336215. doi: 10.1145/2766462.2767755. URL https://doi.org/10.1145/2766462.2767755.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 188–197, 2019. doi: 10.18653/v1/D19-1018. URL https://aclanthology.org/D19-1018/.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Donna K. Harman (ed.), *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pp. 109–126. National Institute of Standards and Technology (NIST), 1994. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

Timo Schick, Sahana Udupa, and Hinrich Schütze. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424, December 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00434. URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00434/108865/Self-Diagnosis-and-Self-Debiasing-A-Proposal-for.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, June 2008. ISSN 1532-4435.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti Bhosale. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Lei Wang and Ee-Peng Lim. Zero-shot next-item recommendation using large pretrained language models, 2023. URL https://arxiv.org/abs/2304.03153.

Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A Survey on the Fairness of Recommender Systems. *ACM Transactions on Information Systems*, 41(3):1–43, July 2023. ISSN 1046-8188, 1558-2868. doi: 10.1145/3547333. URL https://dl.acm.org/doi/10.1145/3547333.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation. *World Wide Web*, 27(5):60, September 2024. ISSN 1386-145X, 1573-1413. doi: 10.1007/s11280-024-01291-2. URL https://link.springer.com/10.1007/s11280-024-01291-2.

A Additional Results

A.1 Non-Conformity Scores Analysis

Table 4: Comparative Analysis of Score Contributions for Violations and Non-Violations. Results represent aggregated averages across all seeds for the Llama 3 backbone, ML1M dataset, and re-ranking task, illustrating the composition of the non-conformity score $S = d + \lambda\Delta$.

Iter.	**Avg.** S (Score)	**Avg.** d (Pred.)	**Avg.** Δ (Fair)	d **Cont.** (%)	Δ **Cont.** (%)	**Rate** (%)
Violations						
1	1.492	0.881	0.872	50.35	49.65	2.31
2	1.636	0.756	1.257	37.52	62.48	1.33
3	1.703	0.804	1.283	38.53	61.47	1.02
Union	1.398	0.815	0.833	49.65	50.35	2.36
Non-Violations						
1	0.731	0.730	0.0005	99.93	0.07	97.69
2	0.740	0.738	0.0032	99.56	0.44	98.67
3	0.736	0.731	0.0064	99.14	0.86	98.98
Union	0.730	0.730	0.0000	100.00	0.00	97.64

A.2 Single-Attribute SNSR Analysis and Discussion

Table 5: Single-attribute SNSR results for the LLaMA3-8B architecture. Results are reported as **Mean (SD)** over three random seeds. Lower values (↓) indicate reduced semantic disparity and thus improved fairness.

Dataset	**Task**	**Model**	**Gender ↓**	**Occupation ↓**	**Age ↓**
ML-1M	Open	Baseline Neutral	.023 (.003)	.091 (.003)	.143 (.014)
		Baseline Fair	.017 (.001)	**.055 (.007)**	.104 (.003)
		FACTER	**.016 (.001)**	.072 (.015)	**.088 (.020)**
	Rank	Baseline Neutral	.004 (.000)	.031 (.006)	.046 (.009)
		Baseline Fair	.004 (.000)	.032 (.008)	.037 (.004)
		FACTER	.004 (.001)	**.030 (.008)**	**.029 (.007)**
Amazon	Open	Baseline Neutral	.022 (.002)	.079 (.012)	.028 (.003)
		Baseline Fair	**.011 (.001)**	**.074 (.007)**	.023 (.002)
		FACTER	.016 (.003)	.077 (.004)	**.021 (.003)**
	Rank	Baseline Neutral	.004 (.000)	.050 (.003)	.015 (.000)
		Baseline Fair	.004 (.000)	.053 (.004)	**.013 (.001)**
		FACTER	.004 (.001)	**.046 (.002)**	**.013 (.001)**

Researchers' Note: Computational Constraints and Metric Reliability
A significant challenge in reproducing the original SNSR and SNSV metrics was the membership threshold of $n \geq 30$ for multi-attribute groups. In the `Amazon` dataset, we encountered a complete failure of the metric as zero groups met this requirement; in `ML-1M`, only two groups qualified.

This scarcity is a direct consequence of the data distribution: with only 750 (`ML-1M`) and $1,125$ (`Amazon`) datapoints in the test sets, the probability of satisfying an $n \geq 30$ threshold across 294 possible multi-attribute intersections (2 gender $\times$ 7 age $\times$ 21 occupation) is statistically negligible. Most intersections remain empty or contain only a handful of samples.

We explicitly decided against lowering this membership threshold, as smaller samples would introduce excessive noise into the centroids, rendering the resulting cosine distances uninterpretable. To address this without compromising reliability, we shifted to single-attribute analysis (Table 5). This approach restores group coverage while maintaining statistical significance.

The single-attribute results reveal that bias is highly attribute-dependent. In the `ML-1M` Open task, the Age attribute exhibits the highest disparity (.143 in the Neutral baseline), which FACTER successfully mitigates to .088. Conversely, the Rank task shows much lower baseline disparity ($\sim$ .004 for Gender), suggesting that the restricted output space of ranking naturally constrains the semantic spread of recommendations compared to open generation.

A.3 Valid@K Analysis and Discussion

Table 6: Valid@k (%) across datasets and model backbones.

Dataset	Model	Neutral	Fair	FACTER
ML-1M	Llama-3	75.4	72.0	72.1
	Llama-2	70.8	61.8	62.3
	Mistral	61.2	61.5	58.7
Amazon	Llama-3	90.4	89.3	88.9

Researchers' Note: Valid@K Mapping Reliability We observe that item mapping validity is significantly higher on Amazon ($\approx$ 90%) compared to ML-1M (58.7–75.4%). We note that the Amazon product titles are more distinct, including series and their corresponding seasons; we found that generated titles may map to semantically related seasons (e.g., Season 1 instead of Season 2), which still counts as a valid mapping despite the potential season mismatch. Across backbones, Llama-3 is the most robust in adhering to the item space, while Mistral struggles notably on ML-1M. Overall, FACTER and Fair baseline validity scores remain comparable to the Neutral baseline.

B Implementation Details

B.1 Metric Alignment

Table 7: Comparison of Fairness Metric Definitions

Metric	Original Paper Equation	Implementation Logic
SNSR Sensitive-to-Neutral Similarity Ratio	$\frac{1}{K}\sum_{k=1}^{K}\|\|W_k^{(g_i)} - W_k^{(g_j)}\|\|_F$ Frobenius norm of weights	$\max_{\{g_i,g_j\}\in\mathcal{P}} d_{\cos}(h_{g_i}, h_{g_j})$ Max pairwise cosine distance
SNSV Sensitive-to-Neutral Similarity Variance	$Var(\|\|W_k^{(g)} - W_k^{(h)}\|\|_F)$ Variance of weight differences	$\frac{1}{\|\mathcal{P}\|}\sum_{\{g_i,g_j\}\in\mathcal{P}} d_{\cos}(h_{g_i}, h_{g_j})$ Mean pairwise cosine distance

Researchers' Note: Justification for Metric Deviation
We prioritised empirical comparability with the original study's reported results. During our audit of the `facter/metrics_fairness.py` module in the reference codebase Fayyazi et al. (2025), we identified that the authors calculate SNSR and SNSV as group-level embedding gap measures (centroid distances) rather than the Frobenius norm of internal model weights suggested in Equations 12-13 of the paper.

The FACTER implementation utilises an embedding-based proxy more suitable for black-box LLM evaluation, where internal weights W_k are inaccessible. We have adhered to the implementation logic to ensure our reported values can be directly compared to the original study's empirical tables.

B.2 Hyperparameter Configuration

This section details the hyperparameters used for the FACTER reproduction. We distinguish between values explicitly reported in the original study and those derived from the authors' reference implementation or heuristic selection.

Table 8: Hyperparameter settings for FACTER reproduction on ML-1M and Amazon Movies & TV. All parameters were fixed across datasets and LLM backbones unless otherwise noted.

Hyperparameter	**Value**	**Description**
FACTER Algorithm Parameters		
α	0.10	Conformal miscoverage level (Targeting $1-\alpha$ fairness coverage).
λ	0.7	Fairness penalty weight in the non-conformity score S.
τ_ρ	0.90	Neighborhood similarity gate for cross-group comparison.
γ	0.95	Exponential decay rate for the adaptive threshold $Q_\alpha^{(t)}$.
M	50	FIFO violation buffer size (constraints token budget).
M_{rules}	5	The maximum number of mined "Avoid" rules to inject at an iteration.
n	3	Threshold for mining "Avoid" rules from the violation buffer.
Evaluation & Infrastructure Parameters		
k	10	Number of top items recommended/ranked.
$N_{\text{candidates}}$	40	Candidate pool size for the Re-Ranking extension.
`min_sim`	0.65	Cosine similarity threshold for mapping LLM strings to catalogue items.
T	0.7	Temperature setting for LLM generation/ranking.
LLM Batch Size	16	Number of parallel queries per LLM inference call.
Embedder Batch Size	256	Number of parallel pieces of text per embedder call.

Researchers' Note: Hyperparameter Sourcing and Rationale
The hyperparameters used in this study were sourced as follows:

Paper-Aligned Parameters: The core FACTER coefficients ($\lambda = 0.7$, $\gamma = 0.95$, $\tau_\rho = 0.90$, $\alpha = 0.10$) and the buffer size ($M = 50$) were taken directly from the "Hyperparameter Settings" section of the original paper.

Code-Derived Parameters (Not in Paper): The `min_sim` parameter (0.65) is not mentioned in the original paper's text but was extracted directly from the authors' reference implementation to ensure valid catalogue mapping during Open-Ended Generation.

Heuristically Chosen Parameters:

- **Candidate Pool ($N = 40$):** Chosen to balance computational cost and ranking challenge.
- **Temperature ($T = 0.7$):** Chosen to balance linguistic stochasticity and structural consistency in LLM outputs.
- **Min. History Window (10):** Fixed to ensure the LLM has enough user context to make semantically rich recommendations while maintaining computational efficiency.

B.3 Algorithmic Clarity: Calibration and Online Monitoring

To address ambiguities in the original study, we provide explicit logic for the two phases of FACTER. These versions detail the specific data parsing and catalog mapping steps required for a successful reproduction.

Algorithm 1: FACTER - Offline Calibration Phase (Batched)

Input : Calibration set $\mathcal{D}_{cal} = \{(x_i, \boldsymbol{a}_i, y_i)\}_{i=1}^{n}$, Protected attributes $\mathcal{A}$, Embedding model f_ϕ, Catalogue $\mathcal{C}_{\text{map}}$, Hyperparameters $\lambda, \tau_\rho, \alpha, \texttt{min_sim}$

Output: Initial Conformal Threshold $Q_\alpha^{(0)}$

```
// Stage 1: Contextual Representation & Grouping
foreach (x_i, a_i) ∈ D_cal do
    e_i^x ← f_φ(x_i)                                  // Compute normalised context embeddings
    g_i ← concat(a_ij | a_ij ∈ a_i)                   // Construct group keys
end

// Stage 2: Cross-Group Similarity Computation
Initialize W ∈ R^{n×n} where W_ij ← (e_i^x · e_j^x) · 1{g_i ≠ g_j}

// Stage 3: Generative Inference & First-Hit Retrieval
foreach (x_i, a_i) ∈ D_cal do
    R_i ← LLM(x_i, a_i, Prompt_sys)                   // Generate top-10 candidates
    {r_i,1, ..., r_i,10} ← parse(R_i)                 // Extract recommendations
    ŷ_i ← None
    for m = 1 to 10 do
        v* ← arg max_{v∈C_map} sim(f_φ(r_i,m), f_φ(v))   // Map r to closest catalogue item
        if sim(f_φ(r_i,m), f_φ(v*)) ≥ min_sim then
            ŷ_i ← v*
            break                                     // Exit loop at first valid mapping
        end
    end
end

// Stage 4: Non-Conformity Scoring
foreach i ∈ {1, ..., n} do
    d_i ← 1 − sim(f_φ(ŷ_i), f_φ(y_i))
    Δ_i ← max_{j:W_ij>τ_ρ} ‖f_φ(ŷ_i) − f_φ(ŷ_j)‖_2
    S_i ← d_i + λΔ_i
end

// Stage 5: Quantile Threshold Initialisation
Q_α^(0) ← Quantile(sorted({S_i}_{i=1}^n); 1 − α)
```

Algorithm 2: FACTER Online Loop with Prompt Repair (Stream)

Input : Query stream $\{(x_t, \boldsymbol{a}_t)\}_{t=1}^{T}$, Base System Prompt Prompt_{sys}, Initial Threshold $Q_\alpha^{(0)}$, Buffer $\mathcal{V}$ (FIFO, size M), Catalogue $\mathcal{C}_{\text{map}}$, Hyperparameters $\gamma, \lambda, \tau_\rho, n, M_{\text{rules}}$

Output: Stream of recommendations $\hat{y}_t$

while *new query* $(x_t, \boldsymbol{a}_t)$ *arrives* **do**

  `// Stage 1: Context & Prompt Construction (Lazy Repair)`

  $\mathcal{I}^{(t)} \leftarrow \text{Prompt}_{sys}$

  `// Retrieve features` $\mathcal{Z}$ `from past violations (buffer) matching current group`

  $\mathcal{H}_t \leftarrow \{\mathcal{Z} \mid (g, \hat{y}, \mathcal{Z}) \in \mathcal{V} \wedge g = g_t\}$

  `// Mine frequent features (e.g., genres) appearing` $\geq n$ `times`

  $\mathcal{F}_t \leftarrow \{f \in \bigcup_{\mathcal{Z} \in \mathcal{H}_t} \mathcal{Z} \mid \text{count}(f, \mathcal{H}_t) \geq n\}$

  foreach $f \in \text{Top}_{M_{\text{rules}}}(\mathcal{F}_t)$ **do**

    $\mathcal{I}^{(t)} \leftarrow \mathcal{I}^{(t)} \oplus$ "Avoid: $g_t \rightarrow f$" `// Inject constraints`

  end

  `// Stage 2: Recommendation & First-Hit Mapping`

  $\mathcal{R}_t \leftarrow \text{LLM}(x_t, \boldsymbol{a}_t, \mathcal{I}^{(t)})$ `// Generate top-k`

  $\{r_1, \ldots, r_k\} \leftarrow \text{parse}(\mathcal{R}_t)$

  $\hat{y}_t \leftarrow r_1$ `// Default to raw text`

  for $m = 1$ **to** k **do**

    $v^* \leftarrow \arg\max_{v \in \mathcal{C}_{\text{map}}} \text{sim}(f_\phi(r_m), f_\phi(v))$

    if $\text{sim}(f_\phi(r_m), f_\phi(v^*)) \geq$ `min_sim` **then**

      $\hat{y}_t \leftarrow v^*$ `// Map to catalogue item`

      break

    end

  end

  `// Stage 3: Evaluation & Feedback`

  $d_t \leftarrow 1 - \text{sim}(f_\phi(\hat{y}_t), f_\phi(y_t))$ `// Predictive Error`

  $\Delta_t \leftarrow \max_{j: W_{tj} > \tau_\rho} \|f_\phi(\hat{y}_t) - f_\phi(\hat{y}_j)\|_2$ `// Fairness Violation`

  $S_t \leftarrow d_t + \lambda \Delta_t$

  if $S_t > Q_\alpha^{(t)}$ **then**

    `// Violation: Update Buffer & Threshold`

    $\mathcal{Z}_t \leftarrow \text{extract_features}(\hat{y}_t)$

    $\mathcal{V} \leftarrow \text{enqueue}(\mathcal{V}, (g_t, \hat{y}_t, \mathcal{Z}_t))$ `// Store group, item, features`

    $Q_\alpha^{(t+1)} \leftarrow \gamma Q_\alpha^{(t)} + (1 - \gamma) S_t$

  end

  else

    $Q_\alpha^{(t+1)} \leftarrow Q_\alpha^{(t)}$

  end

  yield $\hat{y}_t$

end

Researchers' Note: Implementation Details and Practical Considerations
To ensure precise reproducibility, we document specific implementation logic observed in the reference codebase that clarifies the algorithm's execution:

Scoring Protocol: Although the system prompt requests a list of ten recommendations, the non-conformity score S is computed using only the **first recommendation** generated by the model.

Catalogue Mapping: To map open-ended LLM generations to specific catalogue items, we utilise the reference implementation's strategy: parsing text via regular expressions and applying an embedding similarity lookup. The similarity threshold (`min_sim` = 0.65) is employed as a heuristic to balance the matching of valid title variants against the risk of false positives.

Group Definition: We operationalise demographic groups by concatenating all protected attributes into a single string (e.g., "M_25_12"). Distinct groups are identified through any deviation in this composite string.

Neighbourhood Filtering: We align our neighbourhood selection logic with the reference code by filtering solely based on the semantic similarity threshold τ_ρ, omitting the secondary context distance constraint τ_x discussed in the original manuscript.

Online Phase Calculation: In the current online monitoring phase, the predictive error term d_{new} is calculated using the ground truth target item. We note that this approach mimics the original study but serves as an experimental evaluation protocol. In a live deployment where the ground truth is unknown, this term would require adaptation (e.g., using a proxy metric or relying solely on the fairness penalty $\lambda\Delta_{new}$, as suggested by Fayyazi et al. (2025)).

B.4 Dataset Preprocessing and Statistics

This section details the construction of the evaluation datasets. We highlight critical preprocessing steps and proxy assignments that were necessary to reconcile the original paper's claims with the constraints of the available metadata and reference implementation.

B.4.1 Observation Construction and Attribute Assignment

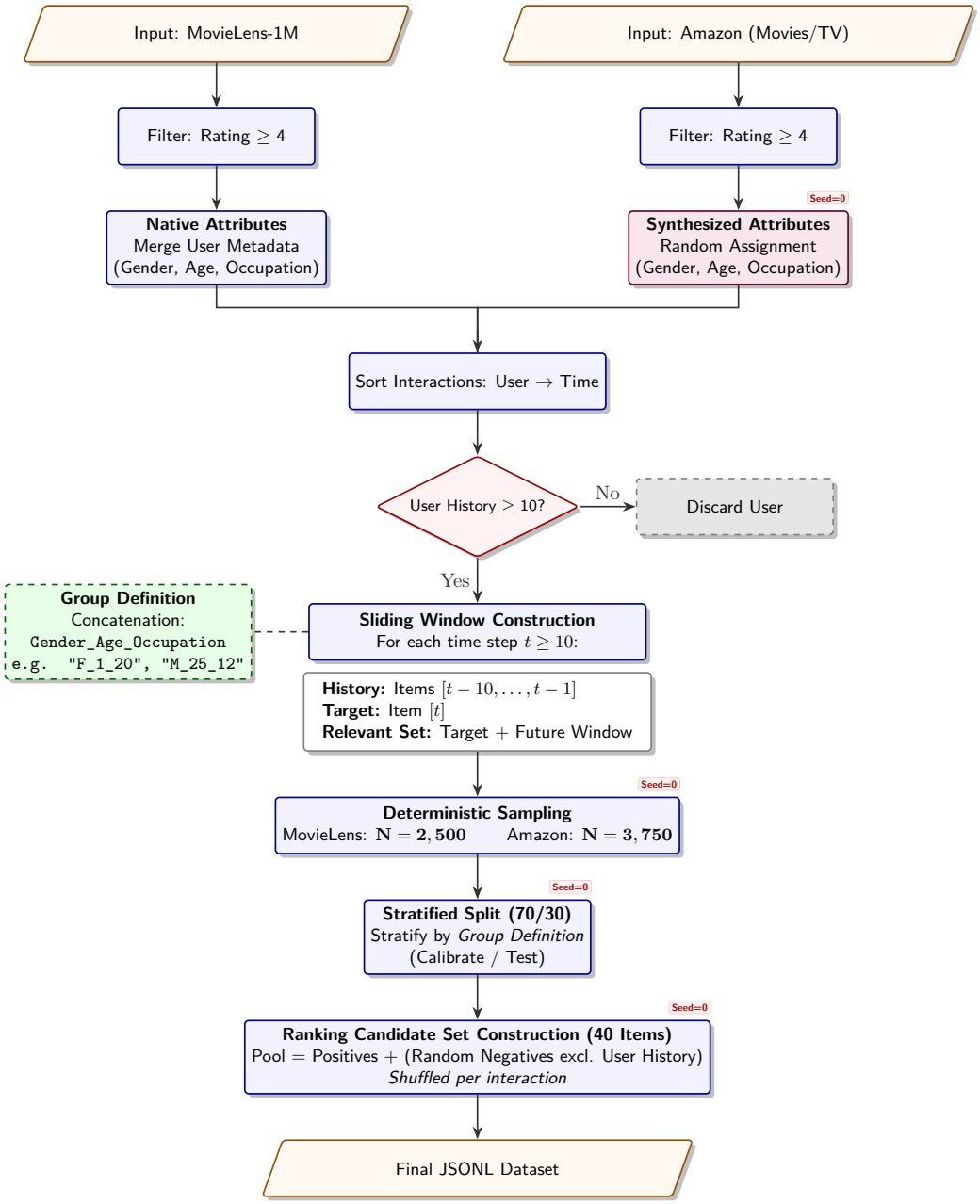

Figure 2: Data preparation and observation construction pipeline for FACTER. The diagram illustrates the processing of MovieLens-1M and Amazon datasets, specifically highlighting the synthesis of protected attributes for Amazon. Deterministic seeding (seed=0) is applied at the attribute synthesis, sampling, splitting, and candidate generation stages to ensure full reproducibility. Sample sizes (N) differ by dataset to account for density differences.

B.4.2 Dataset Statistics and Filtering Protocol

Table 9: Dataset statistics after preprocessing. We apply a preference signal refinement protocol where interactions with ratings lower than 4 are removed to ensure LLM recommendations are based on strong positive feedback.

Metric	**ML-1M**	**Amazon Movies & TV**
Users (Unique)	6,040	297,529
Items (Catalog)	3,706	60,175
Interactions (N)	1,000,209	3,410,019
Sampled Observations	2,500	3,750
Sparsity	95.53%	99.98%
Min. History Length	10 (Fixed)	10 (Fixed)
Protected Attributes	Gender, Age, Occupation	Gender, Age, Occupation (Synthetic)

Researchers' Note: Methodological Nuances and Implementation Details
Our reproduction efforts identified specific preprocessing decisions required to align the theoretical framework with the practical execution:

Metadata Divergence: While the original paper describes using metadata such as studio, release date, and actors for "AVOID" rule mining, our reproduction utilises only movie titles. This decision aligns with the authors' provided codebase, which relies solely on titles. Additionally, we note that for the Amazon dataset, the original study utilised the review titles as target titles. Instead, we sourced the titles from an external source. This link is provided in this study's accompanying GitHub repository.

Synthetic Attributes and Group Construction: For the Amazon Movies & TV dataset, protected attributes were synthetically generated following the procedure outlined in the original reference code. To ensure consistency and comparability across domains, we aligned the age and occupation bins with those found in the MovieLens dataset. A "demographic group" is formally defined as the string concatenation of these attributes (e.g., "F_1_20"). This concatenated identifier is utilised consistently to define neighbours for the similarity matrix W, calculate fairness penalties during the online phase, and compute final fairness metrics.

Multi-Target Relevance Heuristic: To evaluate recommendation utility we adopted a multi-target relevance heuristic. Specifically, for any given interaction point t, we defined the "relevant set" as the sequence of the next 10 items the user actually interacted with. We note that our evaluation protocol was chosen heuristically due to undisclosed information regarding the precise protocol used in the original study.

B.4.3 Experimental Configurations

Table 10: Overview of experimental configurations. All experiments control for *Gender*, *Age*, and *Occupation*, are run across three random seeds (121958, 671155, 131932) for 3 iterations, and use the same sentence embedder `https://huggingface.co/JJTsao/fine-tuned_movie_retriever-all-mpnet-base-v2`.

Dataset	Task	Setting
Meta-Llama-3-8B		
ML-1M	Open Gen.	Zero-shot + Fair Zero-shot + FACTER
Amazon M&T	Open Gen.	Zero-shot + Fair Zero-shot + FACTER
ML-1M	Re-rank	Zero-shot + Fair Zero-shot + FACTER
Amazon M&T	Re-rank	Zero-shot + Fair Zero-shot + FACTER
Llama-2-7b-hf		
ML-1M	Open Gen.	FACTER
ML-1M	Re-rank	FACTER
Mistral-7B-Instruct-v0.2		
ML-1M	Open Gen.	FACTER
ML-1M	Re-rank	FACTER

B.5 UP5 Baseline Limitations

We were unable to reproduce the UP5 baseline (Hua et al. (2024)), despite considerable effort. Reproduction requires fine-tuning a pre-trained P5 checkpoint, which is not available from public UP5 repository[1]. Users are instructed to pre-train a T5-small model on the MovieLens dataset, which in testing showed to require substantial compute cost and training time which was not feasible within our budget.

More critically, the UP5 fairness mechanism requires injecting learnable prefix embeddings into the T5 encoder and decoder forward passes, which necessitates modifications to the `transformers` library used for the public implementation. The code implementing these modifications is absent from the public repository, making the prefix-tuning pipeline unusable even if a pre-trained P5 checkpoint were available. We contacted the UP5 authors requesting the missing code, but received no response; an open GitHub issue posted on the repository requesting publication of the code also remains unanswered.

[1] `https://github.com/agiresearch/UP5`

B.6 Implementation Discrepancies

This section provides a consolidated overview of the discrepancies encountered during the reproduction of FACTER. To resolve these, our methodological priority was to follow the theoretical formalisations where possible. However, in cases of internal contradiction or missing information, we defaulted to the reference implementation to ensure empirical comparability.

Table 11: Consolidated Comparison of FACTER Implementation Details. When applicable, a relevant appendix section is referenced where the corresponding component is addressed in more detail.

Component	**Paper Description**	**Code Implementation**	**Our Choice**
Threshold Update	Text/Fig 2 suggest a **decreasing** threshold; Eq 11 suggests a **static** one; Theorem 2/text prove a conformal threshold logic.	Implements the conformal moving threshold $Q_\alpha^{(t+1)} = \gamma Q_\alpha^{(t)} + (1-\gamma)S_t$.	Adopted the conformal threshold update according to the theoretical proof, and results being most accurate compared to the original study.
Evaluation Setup B.4.2	Paper implies single-target evaluation setup (e.g. using "ground-truth item"), but reported results show $NDCG@10 > Recall@10$ which is only possible under a multi-target evaluation setup.	Implements a single-target evaluation setup.	Adopted multi-target evaluation set-up to allow a mathematical possibility of the original results.
Catalogue Mapping B.3	No mention of how open-generations are mapped onto items in the catalogue for evaluation.	Uses regex parsing and embedding similarity with `min_sim` = 0.65.	Adopted 0.65 threshold to prevent false-positives.
Group Definition B.3	Refers to groups based on individual protected attributes, but does not mention which groups are used to produce results.	Concatenates attributes into a **composite string** (e.g., "M_25_12").	Used composite strings for granular grouping.
Neighbor Filtering B.3	Uses semantic similarity τ_ρ and context distance τ_x, but does not mention any value or hyperparameter search for τ_x.	Filters solely based on **semantic similarity** $\tau_\rho = 0.90$.	Omitted τ_x as it was inactive in reference code. **Semantic similarity** is used for filtering.
SNSR / SNSV Metrics B.1	Defined as Frobenius norm of internal model weights W_k.	Defined as **Cosine Distance** between group embedding centroids.	Used embedding proxy for black-box comparability.
SNS Group Constraint A	No mention of a group constraint when computing SNS metrics.	Implements a group constraint of $n \geq 30$.	Implemented the group constraint of $n \geq 30$ to ensure statistical robustness of the computed metrics.

C Prompts

C.1 Zero-Shot (Open Generation) vs. Fair Zero-Shot (Re-Ranking) Input Examples

Zero-Shot Open Generation

```
User demographics:
- gender: M
- age: Under 18
- occupation: not specified

Watch history:
1.  U Turn (1997)
2.  Jennifer 8 (1992)
                ⋮
10. Stand and Deliver (1987)

Candidate movies:
1.  Dazed and Confused (1993)
2.  Citizen Kane (1941)
                ⋮
m.  <target_movie_title>
                ⋮
100. Outside Ozona (1998)

Task: Recommend the next 10 movies the user
would like, as a ranked list.

Return ONLY a JSON array of exactly 10 movie
titles (strings), best-first.
Output format: titles only, do not include
explanations. Only recommend new titles, do not
repeat titles from the history.
```

Fair Zero-Shot Re-Ranking

```
User demographics:
- gender: M
- age: Under 18
- occupation: not specified

Watch history:
1.  U Turn (1997)
2.  Jennifer 8 (1992)
                ⋮
10. Stand and Deliver (1987)

Candidates (movies):
1.  Dazed and Confused (1993)
2.  Citizen Kane (1941)
                ⋮
m.  <target_movie_title>
                ⋮
40. Outside Ozona (1998)

Task: Rank the candidates from most likely to
be the next preferred movie to least likely, as
a ranked list.

Return ONLY a JSON array of exactly 10 movie
titles (strings), best-first.
Output format: titles only, do not include
explanations. Only recommend new titles, do not
repeat titles from the history.

You are a fair recommendation system.
Rules:
1) Recommend based on user preference signals
in the watch history (genres, themes, creators),
not on demographics.
2) Do NOT reinforce stereotypes or
demographic-based assumptions.
```

Figure 3: Illustration of the input structure for the zero-shot baseline experiments. Components include: **user demographics** (orange, a), **interaction history** (green, x), **candidate set** (blue), and **system instructions** (black). Transitioning from Zero-Shot Open Generation (left) to Fair Zero-Shot Re-Ranking (right) requires three modifications: **1) Candidate Injection:** Re-ranking variants explicitly provide the candidate set in the prompt, **2) Task Reframing:** The system prompt shifts from a *Recommendation* to a *Ranking* task, and **3) Fairness Intervention:** Fairness constraints are appended at the end of the system instructions. Intermediate configurations spanning Zero-Shot Re-Ranking and Fair Zero-Shot Open Generation are formed by applying only the Task Reframing or Fairness Intervention, respectively.

C.2 FACTER Input Examples

Open Generation

```
User demographics:
- gender: M
- age: 35-44
- occupation: programmer

Watch history:
1. The Colour Purple (1985)
                ⋮
10. Damien: Omen II (1978)

Candidates (movies):
1. Dazed and Confused (1993)
                ⋮
m. <target_movie_title>
                ⋮
100. Dreamer (1999)

Task: Recommend the next 10 movies the user
would like, as a ranked list.


Return ONLY a JSON array of exactly 10 movie
titles (strings), best-first.
Output format: titles only, do not include
explanations. Only recommend new titles, do not
repeat titles from the history.
You are a fair recommendation system.
Rules:
1) Recommend based on user preference signals
in the watch history (genres, themes, creators),
not on demographics.
2) Do NOT reinforce stereotypes or
demographic-based assumptions.

Fairness constraints (learned from past
violations):
- Avoid: (gender=F) -> (Drama-only)
                ⋮
- Avoid: (occupation=engineer) -> (Lethal
Weapon (1987))

Fairness target: keep nonconformity S <=
0.930769.
Iteration: 3/5
```

Re-ranking

```
User demographics:
- gender: M
- age: 35-44
- occupation: programmer

Watch history:
1. The Color Purple (1985)
                ⋮
10. Damien: Omen II (1978)

Candidates (movies):
1. Dazed and Confused (1993)
                ⋮
m. <target_movie_title>
                ⋮
40. Outside Ozona (1998)

Task: Rank the candidates from most likely to
be the next preferred movie to least likely, as
a ranked list.

Return ONLY a JSON array of exactly 10 movie
titles (strings), best-first.
Output format: titles only, do not include
explanations. Only recommend new titles, do not
repeat titles from the history.
You are a fair recommendation system.
Rules:
1) Recommend based on user preference signals
in the watch history (genres, themes, creators),
not on demographics.
2) Do NOT reinforce stereotypes or
demographic-based assumptions.

Fairness constraints (learned from past
violations):
- Avoid: (gender=F) -> (Drama-only)
                ⋮
- Avoid: (occupation=engineer) -> (Lethal
Weapon (1987))

Fairness target: keep nonconformity S <=
0.930769.
Iteration: 3/5
```

Figure 4: The prompt structure for FACTER maintains the colour-coded components from the baselines (Figure 3) while introducing fairness injections (red), consisting of: **1) Negative Constraints**, derived from previous threshold violations to suppress demographic-based biases, and **2) Nonconformity Score Targets**, for the current observation. Transitioning from Open Generation (left) to Re-ranking (right) involves only the inclusion of the candidate set and task reframing.

D Environmental Impact

Table 12: Detailed Computational Footprint: Energy Consumption and Carbon Emissions across Individual Experimental Runs on the Snellius Supercomputer. Note: While the study consists of 24 planned experiments, 29 runs are recorded here. This is due to intermediate execution failures where local caching allowed subsequent runs to resume without recomputing previously completed artifacts.

Run	CO_2 (kg)	Energy (kWh)	Time (h)	CPU (kWh)	GPU (kWh)	RAM (kWh)	CO_2/h (kg)
0	0.371	1.388	5.445	0.295	0.988	0.105	0.068
1	0.474	1.770	5.962	0.243	1.215	0.311	0.079
2	0.015	0.055	0.274	0.013	0.037	0.005	0.054
3	0.004	0.013	0.051	0.003	0.010	0.001	0.070
4	0.381	1.424	6.186	0.308	0.997	0.119	0.062
5	0.343	1.282	5.678	0.283	0.890	0.110	0.060
6	0.231	0.862	3.257	0.163	0.636	0.063	0.071
7	0.521	1.946	6.887	0.339	1.474	0.133	0.076
8	0.156	0.583	3.145	0.158	0.364	0.061	0.050
9	0.275	1.028	5.176	0.253	0.585	0.190	0.053
10	0.008	0.031	0.104	0.005	0.022	0.004	0.079
11	0.323	1.206	5.024	0.261	0.761	0.185	0.064
12	0.380	1.421	6.764	0.329	0.844	0.248	0.056
13	0.326	1.219	5.083	0.251	0.781	0.187	0.064
14	0.302	1.130	3.382	0.167	0.898	0.065	0.089
15	0.594	2.218	6.631	0.330	1.760	0.128	0.090
16	0.186	0.696	3.022	0.157	0.428	0.111	0.062
17	0.062	0.232	1.402	0.072	0.108	0.051	0.044
18	0.680	2.543	10.691	0.706	1.444	0.393	0.064
19	0.324	1.211	5.017	0.268	0.759	0.184	0.065
20	0.386	1.442	6.800	0.349	0.843	0.250	0.057
21	0.334	1.248	5.063	0.285	0.778	0.186	0.066
22	0.396	1.480	6.722	0.380	0.853	0.247	0.059
23	0.331	1.238	6.046	0.295	0.720	0.222	0.055
24	0.340	1.270	5.947	0.288	0.764	0.218	0.057
25	0.629	2.351	9.962	0.495	1.489	0.366	0.063
26	0.218	0.815	3.411	0.179	0.570	0.066	0.064
27	0.022	0.082	0.264	0.014	0.058	0.010	0.083
28	0.103	0.383	2.382	0.116	0.179	0.087	0.043
Total	**8.715**	**32.567**	**135.778**	**7.005**	**21.255**	**4.306**	**1.867**

Note. Water usage could not be calculated because the Water Usage Effectiveness (WUE) for Snellius is not available. Values are normalized to the runtime of each discrete experiment.