Fair RAG: End-to-End Fairness Across Retrieval and Generation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) used in Retrieval-Augmented Generation (RAG) can amplify demographic bias: retrievers may surface skewed context and generators can propagate that skew into decisions. Prior work typically treats fairness in retrieval or generation in isolation, leaving end-to-end fairness in RAG underexplored. We propose a post-hoc pipeline that jointly controls both stages: (i) a *Fair Greedy Reranker (FGR)* that builds prefix-balanced slates toward a target group mix; (ii) a *Residual Slate Bias Estimator (RSBE)* using signed, prefix-sensitive NDKL to quantify remaining skew; and (iii) *Confidence-Gated Logit Calibration (CGLC)* that converts the residual signal into small and margin-focused logit corrections without retraining. On an occupation classification task, our approach reduces retriever-side skew (lowest NDKL among baselines for both dense and sparse retrievers) and achieves the lowest generator-side disparity (e.g., Risk Difference) while largely preserving utility. The same calibration can be tuned to alternative fairness criteria (e.g., Equal Opportunity) with minimal utility loss.

1 Introduction

Large Language Models (LLMs) can handle a broad set of inference tasks: generation, classification, even structured prediction (Minaee et al., 2025). However, in their pre-trained form they often *hallucinate*, i.e., produce fluent yet incorrect or unsupported content, particularly on domain-specific or long-tail queries (I et al., 2024). This undermines reliability. **Retrieval-Augmented Generation (RAG)** (Lewis et al., 2020) alleviates this by retrieving relevant evidence (also known as slates) from an external knowledge corpus and conditioning the LLM on it, grounding the response in actual documents rather than only in parametric memory.

LLMs are already known to suffer from demographic bias (Haque et al., 2025b). Here, the term *demographic bias* means systematic differences in model behavior across protected and unprotected groups (e.g., female vs. male) that appear in downstream decisions like selection or classification (Garrido-Muñoz et al., 2021). Adding retrieval on top of such a model does not, by itself, make the system any more *fair*; it can actually introduce a second source of bias from the external knowledge base (Wu et al., 2025). Both main components of a RAG pipeline: the retriever and the generator can introduce or amplify this bias. The retriever may surface documents that overrepresent one group and underrepresent another because the corpus is historically imbalanced or similarity scores are skewed (Geyik et al., 2019), and the generator then conditions on this skewed slate and can further propagate it in the final response (Hu et al., 2025), even when the retrieval skew was modest. In effect, RAG can become a *bias amplifier*, carrying disparities along the chain database → retrieval → context → generation.

In this work, we argue that fairness in RAG must be enforced across the whole inference chain, not at a single point. We define fairness as **statistical parity** (demographic parity or equality of opportunity): for each prediction, the proportion of selected (positive) outcomes should be comparable across protected and unprotected groups, and we summarize disparity using risk difference and true positive rate gap (lower is better). Balancing the retrieved context helps, but it does not guarantee fair outcomes once the LLM makes the final decision, especially when its decision boundary is group-sensitive. Prior work on fair RAG typically intervenes either *after* generation (auditing or filtering the output) (Wu et al., 2025), or only on the retrieved list (debiasing the slate) (Kim and Diaz, 2025). Although (Zhang et al., 2025b) aims to jointly control both stages by fine-tuning the retriever on the LLM's fairness score distribution, it does not account for retriever-side fairness. Thus,

our contribution is a unified, three-part pipeline that (i) reshapes the retrieved slate toward a fair demographic mix, (ii) quantifies whatever bias still remains at the slate level, and (iii) uses that residual signal to impose a small logit correction on the LLM's output. In experiments, this pipeline achieves state-of-the-art level bias reduction in both dense and sparse retrieval settings, and consistently yields the lowest disparity across multiple LLMs, while keeping the added computation lightweight (linear in slate size) and practical as a post-hoc method, with no retraining of the retriever or the generator.

2 Background

2.1 RAG System

In a typical RAG workflow, given a user prompt (query) x, the retriever component searches an external knowledge base $\mathcal{D}$ to identify the most relevant documents or passages (Lewis et al., 2020). The query x is first encoded into a suitable representation, which is compared against document representations $d_i \in \mathcal{D}$ using a relevance scoring function $\text{sim}(x, d_i)$. The *top*-K most relevant results are then selected. The retrieved documents or slates $S_K = \{d_1, d_2, \ldots, d_K\}$, ensures factual grounding for inference which is then concatenated with the original query x and passed to the generator f, which performs knowledge inference to produce a grounded response $\hat{y} = f(x, S_K)$.

2.2 Bias in RAG Components

Despite the effectiveness, both stages of a RAG pipeline can show demographic bias across different demographic groups (Bender et al., 2021; Otterbacher, 2018). Pre-trained LLMs may encode disparities from training data (Kiritchenko and Mohammad, 2018), and external corpora can reflect historical imbalances (Zhang et al., 2025a). If the retriever surfaces context whose group distribution is skewed relative to a desired baseline (Geyik et al., 2019) and the generator conditions on that context, the output can amplify those disparities such that one demographic group a_i is systematically favored over another a_j (Zhang et al., 2025b) where $a \in \mathcal{A}$.

To measure the bias, each stage is evaluated with complementary metrics. On the retriever side, Skew@K (Geyik et al., 2019) assesses over or under-representation of groups among the *top*-K results relative to a desired representation, KL ranking bias quantifies how the *top*-K distribution departs from the desired distribution(Yang and Stoyanovich, 2017), and rank bias captures whether higher-visibility ranks disproportionately feature certain groups (Rekabsaz and Schedl, 2020). On the generator side, demographic parity (Dwork et al., 2011) checks whether outcome rates are independent of groups (summarized by the risk difference between groups), while equality of opportunity (Hardt et al., 2016) tests whether true positives are equally likely across groups (summarized by the TPR gap) (Zhang et al., 2018).

2.3 Research Gap

While many studies address bias correction in information retrieval (Seyedsalehi et al., 2025; Zerveas et al., 2022) and bias mitigation in LLM outputs (Dong et al., 2024; Dige et al., 2023; Haque et al., 2025a), they typically treat these problems in isolation. In RAG, however, retrieval and generation form a single inference pipeline, so bias can compound across stages (Zhang et al., 2025a). Even if generator f were unbiased in ideal conditions, bias introduced via S_K through selection effects, document imbalance, or representation frequency can propagate into $\hat{y}$; conversely, even with neutral retrieval, a biased f can still yield discriminatory outputs. Although some studies have examined bias mitigation within RAG (Ji et al., 2025; Zhang et al., 2025b; Kim et al., 2025), they largely focus on generator-side fairness and do not account for retriever-side bias. Thus, fairness in RAG must be analyzed as a joint property of both retrieval and generation stages, yet a unified end-to-end pipeline that corrects disparities in both stages remains absent from the literature.

3 Methodology

Our objective is to design an end-to-end *fair* RAG system that enforces fairness on both retrieval and generation. To the best of our knowledge, prior work typically addresses these two sources of bias in isolation. We (empirically) show that correcting only the retrieved slate is not enough: a balanced retrieved slate can still yield biased outputs. To address this, we propose a three-stage pipeline as shown in Figure 1, comprising (i) an efficient deterministic **fair reranker** for the retrieved slates, (ii) a **residual bias estimator** that quantifies any remaining imbalance in the displayed slate, and (iii) a **confidence-gated calibration** layer on the generator that uses this signal to debias the final decision.

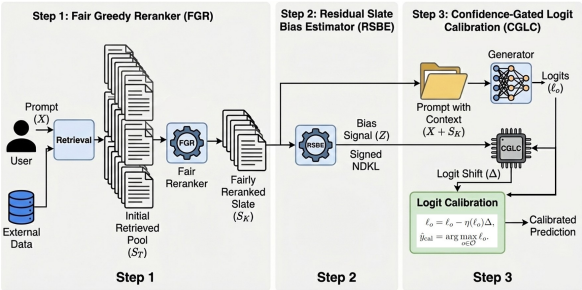

Figure 1: End-to-end fair RAG pipeline

Together, these components enable fairness-aware RAG throughout the entire inference pipeline without any fine-tuning.

3.1 Fair Greedy Reranker (FGR)

In neural retrievers, much of the effectiveness is inherited from the pre-trained backbone (Yao et al., 2025), so injecting *fairness* via additional training is costly and hard to control. A practical posthoc alternative is to retrieve a larger context pool of size $T \geq K$ first (instead of the final *top-K*), and then apply a fairness-aware reranker on this pool to select the final *top-K* results. Thus we propose a **Fair Greedy Reranker (FGR)** as follow:

Given a larger retrieved slate ordered by relevance from high to low, $S_T = \{d_1, \ldots, d_T\}$ and a target distribution $\mathbf{q} = \{q_1, \ldots, q_M\}$ over groups of a protected attribute $\mathcal{A} = \{a_1, \ldots, a_M\}$ with $q_m \geq 0$ for group a_m and $\sum_{a_m \in \mathcal{A}} q_m = 1$, FGR constructs a *fair slate* S_K of length $K \leq T$ that (i) approximately preserves the original retrieval order and (ii) keeps each retrieved list up to K (called each prefix) close to $\mathbf{q}$.

We construct the fair slate incrementally. At each step $k \in \{1, \ldots, K\}$, we pick the next high-relevance document while tracking, for each group $a_m \in \mathcal{A}$, how far the current prefix deviates from the target distribution. Let g_k denote the group of document d_k. We define the group-wise prefix deficit at position k as:

$$U_k[a_m] = q_m \cdot k - \#\{j \leq k-1 : \; g_j := a_m\}, \quad (1)$$

where $q_m \cdot k$ is the desired count of group a_m in the *top-K* of the slate and the second term is the actual count. At each step, we prioritize the groups with the largest deficit $U_k[a_m]$ (i.e., underrepresented groups) and, within that subset, choose the highest-ranked remaining item to preserve relevance. If no group is underrepresented ($\max_{a_m \in \mathcal{A}} U_k[a_m] \leq 0$), we select the next highest-ranked remaining item. Intuitively, $U_k[a_m] > 0$ means a_m is underrepresented, $U_k[a_m] = 0$ on target, and $U_k[a_m] < 0$ overrepresented. We formally present this in Algorithm 1. Proof is available in Appendix E.

3.2 Residual Slate Bias Estimator (RSBE)

Even after fair reranking (Algorithm. 1), some imbalance can remain in the fair slate due to retrieval noise like duplicate entries, missing metadata, or asymmetric content. **Residual Slate Bias Estimator (RSBE)** converts the slate's observed group distribution into a single, prefix-sensitive scalar that summarizes *how far* (and in which *direction*) the slate deviates from a target distribution $\mathbf{q}$.

Prefix shares. For each *top-K* prefix of the slate, let $p_k(a_m)$ be the fraction of those k items that belong to group a_m (i.e., $p_k(a_m)$ is the share of group a_m among the first k positions).

Prefix deviation via NDKL. We adopt the *normalized discounted cumulative KL divergence (NDKL)* of Geyik et al. (2019) to measure prefix sensitive deviation. Given a target distribution $\mathbf{q}$ over $\mathcal{A}$ (with $q_m \geq 0$ and $\sum_{a_m \in \mathcal{A}} q_m = 1$) and a small smoothing $\varepsilon > 0$, the KL divergence at prefix k is

$$\mathrm{KL}(\mathbf{p}_k \,\|\, \mathbf{q}) = \sum_{a \in \mathcal{A}} \big(p_k(a_m) + \varepsilon\big) \log \frac{p_k(a_m) + \varepsilon}{q_m + \varepsilon}. \quad (2)$$

Higher ranks matter more, therefore Discounted Cumulative Gain (DCG) weights is used:$w_k = \frac{1}{\log_2(k+1)}$ and $B_K = \sum_{k=1}^{K} w_k$. The magnitude-only score over the *top-K* is

$$\mathrm{NDKL}(K) = \frac{1}{B_K} \sum_{k=1}^{K} w_k \, \mathrm{KL}(\mathbf{p}_k \,\|\, \mathbf{q}) \;\geq 0, \quad (3)$$

which grows when early prefixes deviate strongly from $\mathbf{q}$.

Assigning direction (per-group signed NDKL). NDKL is unsigned. In order to retain directionality for each group, we define a signed score per group

$$S_K[a_m] = \big(p_K(a_m) - q_m\big) \cdot \mathrm{NDKL}(K), a_m \in \mathcal{A}. \quad (4)$$

Positive $S_K[a_m]$ indicates over-exposure of group a_m in the top-K slate relative to the target distribution q_m, while negative indicates under-exposure; the magnitude is modulated by the prefix-sensitive deviation $\mathrm{NDKL}(K)$.

3.3 Confidence-Gated Logit Calibration (CGLC)

The RSBE signal from §3.2 summarizes how the displayed slate deviates from the target distribution across groups (via the per-group scores $S_K[a_m]$). We now convert this slate-level signal into a small, principled adjustment of the *primary* prediction. Because different classes can exhibit different group skew, we first normalize the residual per class before using it.

Standardization across classes. Let $\mathbf{R} = \big(S_K[a_m]\big)_{a_m \in \mathcal{A}}$ be the per-group signed RSBE vector. For each class $o \in \mathcal{O}$, let $\boldsymbol{\mu}_o = \big(\mu_o[a_m]\big)_{a_m \in \mathcal{A}}$ and $\boldsymbol{\sigma}_o = \big(\sigma_o[a_m]\big)_{a_m \in \mathcal{A}}$ denote training-set means and standard deviations of the RSBE component for group a_m computed over all slates with class o. We standardize component-wise:

$$z_o[a_m] \;=\; \frac{S_K[a_m] - \mu_o[a_m]}{\sigma_o[a_m] + \varepsilon}, \qquad a_m \in \mathcal{A}. \quad (5)$$

Equivalently, in vector form,

$$\mathbf{Z}_o = \mathrm{zscore}\big(\mathbf{R};\, \boldsymbol{\mu}_o, \boldsymbol{\sigma}_o\big). \quad (6)$$

Confidence-Gated logit shift. The generator produces a decision $\hat{y} = f(x, S_K) = \arg\max_{o \in \mathcal{O}} \ell_o$ from a base logit $\ell_o \in \mathbb{R}$. Rather than retraining the model, we apply a small learned *logit shift* Δ driven by (i) the standardized residual $\mathbf{Z}_o$ based on S_K, (ii) the subject x's group g, and (iii) global and class o-specific biases. The logit shift Δ consists of a bias term (with a global bias δ and a class-specific bias δ_o) and a residual term.

$$\Delta = \underbrace{(\delta + \delta_o)}_{\text{bias term}} + \underbrace{\beta z_o[g(x)]}_{\text{residual term}}, \quad (7)$$

where β depends on the sign of $z_o[g(x)]$.

The calibrated decision is based on a confidence-gated logit shift:

$$\begin{aligned} \ell_o &= \ell_o - \eta(\ell_o)\Delta, \\ \hat{y}_{\text{cal}} &= \arg\max_{o \in \mathcal{O}} \ell_o. \end{aligned} \quad (8)$$

To focus corrections on low-confidence decisions, we define a confidence gate based on the relative separation of class logits. Let $\bar{\ell} = \frac{1}{|\mathcal{O}|}\sum_{j \in \mathcal{O}} \ell_j$ denote the mean logit for a given input. We gate the logit shift via

$$\eta(\ell_o) \;=\; \exp\big(-\gamma\,|\ell_o - \bar{\ell}|\big), \qquad \gamma \geq 0, \quad (9)$$

so that classes whose logits lie close to the overall score level (i.e., low separation and low confidence) receive larger adjustments, while well-separated, high-confidence predictions are minimally affected.

Why margin-level updates help fairness. Group selection gaps typically arise near decision boundaries: when groups exhibit different score densities in regions of low class separation, a fixed argmax decision rule can induce unequal selection rates. By gating updates using a confidence measure derived from the relative separation of class logits, CGLC primarily affects low-confidence, borderline cases where class scores are clustered while leaving well-separated, high-confidence predictions largely unchanged. It parallels classic post-processing approaches for demographic parity and equality of opportunity, which focus corrective interventions on ambiguous decisions rather than confident ones (Hardt et al., 2016).

Parameters and Optimization. We parameterize CGLC with parameters

$$\theta \;=\; \{\delta,\ \gamma,\ \beta,\ \{\delta_o\}_{o \in \mathcal{O}}\}. \quad (10)$$

We use a small training set X to learn these parameters. We use a grid search strategy to learn the optimal values heuristically, details shown in Appendix A.6. To improve optimization performance, we also introduce a intercept term b_o to Δ to bridge the optimization gap. For example, for binary groups we target demographic parity by minimizing the mean risk difference (RD):

$$\begin{aligned} \mathrm{RD}(o) &= \big|P(\hat{y}_{\text{cal}}{=}1 \mid g{=}0) - P(\hat{y}_{\text{cal}}{=}1 \mid g{=}1)\big|, \\ \mathrm{RD}_{\text{mean}} &= \frac{1}{|\mathcal{O}|}\sum_{o \in \mathcal{O}} \mathrm{RD}(o), \\ \theta^{\star} &= \arg\min_{\theta}\ \mathrm{RD}_{\text{mean}}(\theta). \end{aligned} \quad (11)$$

This objective targets demographic parity directly.

In settings where a different fairness notion is desired (e.g., equality of opportunity), the same calibration form can be trained with the corresponding gap (TPR gap) substituted for $\mathrm{RD}(o)$, leaving the rest of the pipeline unchanged.

4 Experiment Setup

A typical RAG pipeline comprises three elements: a generator (LLM), a retrieval corpus, and a retriever. To study fairness comprehensively in this

Table 1: Utility and fairness metrics comparison across multiple baselines for dense and sparse retrievers.

Retriever	Re-Ranker	Utility			Fairness		
		Hit Rate	MRR	NDCG	Min Skew	Max Skew	NDKL
FAISS +BGE (Dense)	Vanilla	0.438	0.281	0.310	-3.441	0.525	0.488
	Det const sort	0.461	0.263	0.301	-0.375	**0.156**	0.147
	FairFilter	0.463	0.282	0.314	**-0.334**	0.520	0.468
	FairFT	0.473	0.283	0.320	-0.613	0.489	0.447
	In Context	0.438	0.281	0.310	-3.441	0.525	0.488
	FGR (Ours)	0.462	0.263	0.301	-0.381	0.161	**0.146**
BM25 (Sparse)	Vanilla	0.474	0.284	0.318	-3.781	0.515	0.488
	Det const sort	0.480	0.241	0.289	-1.515	**0.263**	**0.227**
	FairFilter	0.666	0.290	0.353	-2.482	0.497	0.397
	In Context	0.452	0.222	0.289	-2.408	0.417	0.354
	FGR (Ours)	0.479	0.238	0.288	**-1.510**	0.268	**0.227**

setting, we (i) specify a concrete task, (ii) select an external corpus as the retriever's source, and (iii) construct an evaluation set for model evaluations.

4.1 Task Definition

We choose the task as occupation classification. Each inference example prompt (query) x (a short biography) in the evaluation set is annotated with a protected attribute $a_m \in \mathcal{A}$ (gender) and a ground truth occupation $o \in \mathcal{O}$. The generator's task is to predict the occupation for x. To assist the generator, a retriever searches against an external corpus and returns a context slate $S_K = \{d_1, \ldots, d_K\}$, which is provided to the generator alongside x. The end-to-end system thus maps (x, S_K) to $\hat{y} = f(x, S_K)$. Our objective is to assess the proposed fair RAG pipeline across both stages of retrieval and generation, measuring utility (classification and retrieval quality) and fairness (group disparities) jointly.

4.2 Data

Inference-time test set. We evaluate on *Bias in Bios* (De-Arteaga et al., 2019), a dataset of short professional biographies; each biography x, annotated with a protected attribute g (gender) and a label of occupation. We first filter professions to those with sufficient male/female coverage and retain $|\mathcal{O}| = 22$ occupations (others are dropped due to missing splits). For every occupation $o \in \mathcal{O}$, we treat it as binary classification task like one vs. rest. We build a balanced set of 400 biographies: 100 *positive* male and 100 *positive* female (ground truth occupation label $= o$), plus 100 *negative* male and 100 *negative* female (ground truth occupation label $\neq o$). Thus each occupation has 200 positives and 200 negatives, evenly split by gender.

External corpus & retriever setup. Our retriever indexes *Humans of Wikipedia*[1], a large collection of Wikipedia-derived biographical pages. We use it strictly as the external retrieval corpus $\mathcal{D}$. Concretely, we sample 126,609 person pages spanning 24 occupations (including the 22 inference occupations) and segment each page into short passages (each passage having around 150 tokens). Each passage, along with its metadata (person's name, occupation, gender) is a retrieval unit. We report the gender distribution per occupation) for $\mathcal{D}$ in Fig. 2.

At inference, the query is the *Bias in Bios* biography x; the retriever searches $\mathcal{D}$ and returns a top-K slate of passages as context for the generator.

4.3 Fair RAG Pipeline

For each biography x:

1. **Retrieve:** encode x and retrieve top-T passages $S_T \subset \mathcal{D}$ from the *external corpus*. We report results for both dense and sparse retrievers.
2. **Rerank (fairness-aware):** apply the *Fair Greedy Reranker (FGR)* to produce the finaL *top-K* fair slate S_K of length $K \leq T$.
3. **Generate/Classify:** pass (x, S_K) to the LLM classifier to obtain per-class logits and the predicted occupation $\hat{y}$.
4. **Calibrate (post-hoc):** use the *RSBE* and *CGLC* to debias near the decision margin.

4.4 Evaluation Metrics

- **Retriever-side utility:** hit rate, MRR, and NDCG on occupation-relevant retrieval.

[1] Humans of Wikipedia: Explore the lives and legacies of over 1.4M individuals documented in English Wikipedia's biographical articles with detailed metadata.

- **Retriever-side fairness:** prefix-sensitive skew (e.g., signed NDKL) and group exposure statistics on the slate S_K (Geyik et al., 2019).
- **Generator-side utility:** accuracy, precision, recall, and F1 for occupation classification.
- **Generator-side fairness:** demographic parity via risk difference across groups, reported per class and macro-averaged.

4.5 Baselines

We evaluate retrieval using both sparse (BM25 (Robertson and Zaragoza, 2009)) and dense (FAISS + BGE (Douze et al., 2024)) retrievers. On the generation side, we pair dense retrievers with multiple open-source LLMs including Flan-T5 (3B version) (Chung et al., 2022), Llama2 (7B version) (Touvron et al., 2023), Gemma-2B (Team, 2024), DeepSeek (6.7B version) (DeepSeek-AI et al., 2025), and Falcon (7B version) (Almazrouei et al., 2023). Configurations that apply no fairness constraints regardless of the retriever or LLM are grouped under the label VANILLA. Building on these vanilla pipelines, we additionally consider reranking and fairness interventions as baselines: DET-CONST-SORT and FAIRRILTER are applied as rerankers, while FAIRFT and IN CONTEXT are fairness mechanisms applied at the LLM stage.

**Due to space constraints, full details of the data, prompt, evaluation metrics and baselines are provided in Appendix A, B, C and D respectively.

4.6 Research Questions

On the task defined in §4.1, we aim to answer the following questions with our experiments:

1. **RQ1:** How do different retrievers perform on utility and retriever-side fairness, with and without fair reranking?

2. **RQ2:** Does post-hoc calibration reduce generator-side disparity while preserving task performance?

3. **RQ3:** Which reranker, when paired with calibration, yields the best fairness–utility trade-off end-to-end?

4. **RQ4:** Can calibration be tuned to meet alternative fairness criteria (e.g., equal opportunity) without notable utility loss?

5 Result Analysis

5.1 Retriever Performance (RQ1)

For both of sparse and dense retrievers, enforcing fairness at the re-ranking time sharply reduces exposure skew with only modest utility trade-offs. From Table 1 we can see:

For FAISS + BGE, the dense retriever, the vanila model (VANILLA) shows large skew (min -3.44, NDKL 0.488). FGR and DET-CONST-SORT both curb the skew substantially; FGR achieves the best prefix fairness (NDKL 0.146; max-skew 0.161), at a small MRR drop (from 0.281 to 0.263). Utility-first baselines (FAIRFILTER/FAIRFT) slightly raise NDCG (0.314–0.320) but leave sizable skew (NDKL 0.447–0.468). IN-CONTEXT does not affect retrieval metrics, as expected.

For BM25, the sparse retriever, a similar pattern holds: baseline skew is high (min -3.78, NDKL 0.488). FGR and DET-CONST-SORT deliver the lowest NDKL (0.227), with FGR slightly better on min-skew (-1.510 vs. -1.515). Utility drops modestly (MRR ≈ 0.238, NDCG ≈ 0.288). FAIRFILTER boosts utility (hit 0.666, NDCG 0.353) but retains substantial skew (NDKL 0.397).

We have the following observations: (i) FGR consistently attains the strongest prefix fairness (lowest NDKL) for both types of retrievers with small utility cost; (ii) heuristic sorting reduces exposure but lags FGR on NDKL or hurts MRR more; (iii) utility-oriented tweaks alone do *not* control exposure; (iv) prompting (IN-CONTEXT) cannot fix retriever-side bias, fairness must be enforced in the re-ranker.

5.2 Generator Performance (RQ2)

Overall, our post-hoc CGLC substantially reduces demographic disparity (RD) across LLMs with limited impact on utility and in several cases improves F1 via recall gains. From Table 2 we can see:

Gemma-2B. RD halves ($0.088 \rightarrow \mathbf{0.047}$), matching IN-CONTEXT. This comes with a small utility cost (F1 $0.730 \rightarrow 0.700$, acc $0.736 \rightarrow 0.713$), illustrating the fairness–utility trade-off when the base is already strong.

Llama2-7B. Our method achieves the lowest RD 0.037 while sacrificing small accuracy ($0.694 \rightarrow \mathbf{0.659}$) and *raising* F1 ($0.426 \rightarrow \mathbf{0.585}$). Utility-oriented baselines (FAIRFILTER) can lift F1 (0.654) but increase RD (0.105).

DeepSeek 6.7B. No retriever exhibits very low (0.026) RD, but this result is not meaningful be-

Table 2: Utility and fairness metrics comparison across multiple baselines for generator side.

LLM	Model	Utility				Fairness
		Acc	Prec	Rec	F1	RD
Gemma-2B	No Retriever	0.614	0.796	0.383	0.428	0.061
	Vanilla	0.736	0.778	0.716	0.730	0.088
	Det const sort	0.714	0.777	0.726	0.694	0.071
	FairFilter	0.683	0.742	0.731	0.650	0.073
	FairFT	0.737	0.763	0.810	0.750	0.075
	In Context	0.722	0.763	0.777	0.717	0.077
	Our	0.713	0.764	0.742	0.700	**0.047**
Llama2 7B	No Retriever	0.620	0.782	0.345	0.426	0.213
	Vanilla	0.694	0.743	0.323	0.426	0.065
	Det const sort	0.657	0.657	0.511	0.573	0.051
	FairFilter	0.697	0.749	0.704	0.654	0.105
	FairFT	0.594	0.743	0.323	0.426	0.065
	In Context	0.648	0.752	0.492	0.566	0.080
	Our	0.659	0.758	0.526	0.585	**0.037**
Deepseek 6.7B	No Retriever	0.547	0.894	0.162	0.194	**0.026**
	Vanilla	0.642	0.811	0.432	0.509	0.049
	Det const sort	0.642	0.811	0.432	0.509	0.049
	FairFilter	0.710	0.826	0.613	0.617	0.066
	FairFT	0.606	0.823	0.325	0.415	0.035
	In Context	0.646	0.808	0.443	0.525	0.059
	Our	0.645	0.808	0.443	0.519	0.048
Falcon-7B	No Retriever	0.619	0.693	0.470	0.503	0.085
	Vanilla	0.740	0.787	0.649	0.651	0.063
	Det const sort	0.740	0.787	0.649	0.651	0.063
	FairFilter	0.722	0.782	0.724	0.684	0.094
	FairFT	0.732	0.787	0.678	0.680	0.047
	In Context	0.732	0.787	0.678	0.680	0.047
	Our	0.717	0.761	0.664	0.660	**0.041**
Flan-T5	No Retriever	0.782	0.710	0.823	0.755	0.130
	Vanilla	0.742	0.684	0.768	0.715	0.108
	Det const sort	0.713	0.660	0.732	0.683	0.099
	FairFilter	0.742	0.684	0.768	0.715	0.109
	FairFT	0.741	0.682	0.771	0.714	0.097
	In Context	0.743	0.727	0.706	0.702	0.075
	Our	0.721	0.712	0.811	0.739	**0.044**

cause the accuracy is 0.547, indicating performance close to random guessing. Baseline RD is moderate (0.049); our RD is similar (0.048) while keeping utility near baseline. FAIRFT attains the lowest RD (0.035) but substantially degrades utility (acc 0.606, F1 0.415), underscoring the appeal of post-hoc methods when retraining harms performance.

Falcon-7B. RD improves from 0.063 to 0.041 with small utility changes (F1 0.651 → 0.660).

Flan-T5. No retriever achieves high accuracy (0.782) because the prompt is minimally verbose, which is easier for a 250M parameter model to handle; however, this setting exhibits high RD. RD drops from 0.130 (NO RETRIEVER) to 0.044 (ours), the best among baselines. F1 rises from 0.715 to 0.739 driven by higher recall (0.768 → 0.811), with a modest accuracy dip (0.743 → 0.721). Prompt-only control (IN-CONTEXT) lowers RD to 0.075 but hurts F1 (0.702).

Takeaways. (i) Post-hoc CGLC reliably reduces RD across models, often achieving the best fairness without retraining; (ii) it frequently *improves* F1 via recall gains (e.g., Flan-T5, Llama2-7B); (iii) when baseline bias is already modest (DeepSeek), fairness gains are smaller, but utility remains stable; (iv) prompt- or retrieval-only tweaks do not consistently reduce generator-side disparity.

Table 3: Rerankers paired with CGLC.

LLM	Model	Utility				Fairness
		Acc	Prec	Rec	F1	RD
Gemma-2B	Det const sort + CGLC	0.704	0.748	0.743	0.692	0.061
	FairFilter + CGLC	0.672	0.712	0.769	0.655	0.062
	FGR + CGLC	0.713	0.764	0.742	0.700	**0.047**
Llama2 7B	Det const sort + CGLC	0.654	0.760	0.509	0.571	0.059
	FairFilter + CGLC	0.684	0.716	0.732	0.658	0.090
	FGR + CGLC	0.659	0.758	0.526	0.585	**0.037**
Deepseek 6.7B	Det const sort + CGLC	0.643	0.763	0.471	0.538	0.097
	FairFilter + CGLC	0.702	0.788	0.654	0.635	0.095
	FGR + CGLC	0.645	0.808	0.443	0.519	**0.048**
Falcon-7B	Det const sort + CGLC	0.719	0.760	0.675	0.666	0.053
	FairFilter + CGLC	0.709	0.752	0.752	0.686	0.081
	FGR + CGLC	0.717	0.761	0.664	0.660	**0.041**
Flan-T5	Det const sort + CGLC	0.727	0.727	0.780	0.733	0.064
	FairFilter + CGLC	0.742	0.730	0.835	0.761	0.053
	FGR + CGLC	0.721	0.712	0.811	0.739	**0.044**

5.3 Performance on Rerankers Paired with CGLC (RQ3)

Across all LLMs in Table 3, **FGR + CGLC** consistently achieves the *lowest* risk difference (RD), while maintaining competitive utility: *Flan-T5*: RD drops to **0.044** (best), with strong F1 (0.739) and recall gains over VANILLA/DET CONST SORT; FAIRFILTER + CGLC attains the highest F1 (0.761) via aggressive recall (0.835) but with higher RD (0.053). *Llama2-7B*: **0.037** RD (best) for FGR, with F1 0.585; FAIRFILTER + CGLC maximizes F1 (0.658) but worsens RD (0.090), indicating a utility–fairness trade-off. *Gemma-2B*: FGR yields **0.047** RD (best) with balanced utility (F1 0.700), outperforming DET CONST SORT/FAIRFILTER on fairness. *DeepSeek 6.7B*: FGR halves RD vs. other rerankers (0.048 vs. 0.095–0.097) at near-baseline

Table 4: CGLC performance for equality of opportunity calibration on Gemma-2B.

Model	Utility				Fairness
	Acc	Prec	Rec	F1	TPR-Gap
Vanilla	0.736	0.778	0.716	0.730	0.100
Det const sort	0.714	0.777	0.726	0.694	0.094
FairFilter	0.683	0.742	0.731	0.650	0.078
FairFT	0.737	0.763	0.810	0.750	0.081
In Context	0.722	0.763	0.777	0.717	0.083
Our	0.715	0.761	0.768	0.714	**0.052**

Table 5: CGLC ablation for demographic parity on Gemma-2B.

Model	Utility				Fairness
	Acc	Prec	Rec	F1	RD
FGR Only	0.715	0.767	0.711	0.708	0.081
Skew	0.714	0.767	0.740	0.701	0.078
No η	0.705	0.746	0.751	0.698	0.070
No δ_o	0.709	0.740	0.768	0.709	0.071
No b_o	0.707	0.744	0.756	0.702	0.069
NDKL + η + δ_o + b_o	0.713	0.764	0.742	0.700	**0.047**

utility. *Falcon-7B*: FGR attains **0.041** RD (best) with F1 0.660; FAIRFILTER pushes F1 (0.686) at a large RD cost (0.081).

Takeaways. (i) Pairing a *prefix-fair* reranker with post-hoc calibration is crucial: **FGR + CGLC** dominates RD across models, showing the most reliable end-to-end fairness; (ii) when maximum F1 is the sole goal, FAIRFILTER + CGLC can help via recall, but at the expense of fairness; (iii) DET CONST SORT + CGLC offers moderate fairness gains with smaller utility shifts, but rarely matches FGR on RD. Overall, FGR provides the best *fairness–utility* balance for end-to-end RAG.

5.4 Calibration for Different Fairness Criteria (RQ4)

When calibrated for *equality of opportunity* (minimizing TPR-Gap), CGLC achieves the strongest fairness on Gemma-2B, reducing TPR-Gap from 0.100 (Vanilla) to $\mathbf{0.052}$ (a 48% reduction), and outperforming FairFilter (0.078), FairFT (0.081), In-Context (0.083), and Det-Const-Sort (0.094); see Table 4. Utility remains competitive: our F1 is 0.714 (close to In-Context 0.717), with precision 0.761 and recall 0.768. While FairFT attains the highest utility (F1 0.750, Acc 0.737), it exhibits a larger TPR-Gap (0.081). Overall, CGLC offers the best fairness–utility balance among the baselines to achieve equality of opportunity.

5.5 CGLC Ablation

Table 5 isolates the contribution of each CGLC component.

FGR only. Fair reranking without CGLC leaves sizable disparity (RD $= 0.081$) with F1 $= 0.708$.

Skew-only signal. Driving calibration with a coarse skew proxy (instead of signed, prefix-sensitive NDKL) modestly helps over FGR only but still yields high disparity (RD $= 0.078$) and lower F1 (0.701).

No gating η. Removing the w raises disparity relative to the full model (RD $= 0.070$ vs. 0.047) with minor utility changes (F1 $= 0.698$), underscoring the value of margin-focused updates.

No δ_o. Dropping class-specific bias terms weakens parity (RD $= 0.071$) even though F1 is the highest among ablations (0.709), indicating uncorrected class-level skews.

No b_o. Removing class-specific intercept harms parity (RD $= 0.069$) and reduces F1 to 0.702.

Full (NDKL $+ \eta + \delta_o + b_o$). The complete calibrator achieves the lowest disparity (RD $= \mathbf{0.047}$) with Acc $= 0.713$, Prec $= 0.764$, Rec $= 0.742$, and F1 $= 0.700$. While utility is comparable to ablations, the fairness gain is substantial.

Overall, all components contribute: signed NDKL provides a stronger residual signal than a coarse skew proxy; the confidence gate targets near-threshold flips that most affect disparity; and per-class terms (δ_o, b_o) adapt corrections to class-specific bias patterns, delivering the best fairness–utility trade-off.

6 Conclusion

We presented an end-to-end fairness framework for RAG that jointly controls bias in retrieval and generation. Our pipeline - Fair Greedy Reranker (FGR), Residual Slate Bias Estimator (RSBE) with signed NDKL, and Confidence-Gated Logit Calibration (CGLC) reduces demographic disparity while preserving task utility. Across dense and sparse retrievers and multiple LLMs, FGR yields the lowest slate skew and CGLC achieves state-of-the-art reductions in risk difference without retraining core models. The method is post-hoc, retrieval-agnostic, and tunable to other criteria (e.g., equal opportunity). Future work includes multi-attribute fairness, robustness to noisy metadata.

7 Limitations

The pipeline introduces a calibration layer that must be trained on a small dataset, incurring additional computational overhead. As future work, we plan to investigate alternative approaches that mitigate bias effectively without requiring any additional training.

8 Ethical Considerations

We aim to mitigate bias in RAG systems through a fairness-aware pipeline. Although we rely on widely used retrieval corpora that may contain harmful or biased content, responsibility for such material lies with the original sources. Our work focuses on reducing biased associations within the RAG process to produce less biased outputs, evaluated using bias-related metrics, while maintaining overall performance. We do not examine potential extrinsic harms that may arise from deploying the debiasing methods studied.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. 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, page 610–623, New York, NY, USA. Association for Computing Machinery.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 120–128, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Omkar Dige, Jacob-Junqi Tian, David B. Emerson, and Faiza Khan Khattak. 2023. Can instruction fine-tuned language models identify social bias through prompting? *ArXiv*, abs/2307.10472.

Xiangjue Dong, Yibo Wang, Philip S. Yu, and James

Caverlee. 2024. Disclosure and mitigation of gender bias in llms. *ArXiv*, abs/2402.11190.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. 2011. Fairness through awareness.

Ismael Garrido-Muñoz , Arturo Montejo-Ráez , Fernando Martínez-Santiago , and L. Alfonso Ureña-López . 2021. A survey on bias in deep nlp. *Applied Sciences*, 11.

Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2221–2231, New York, NY, USA. Association for Computing Machinery.

Farsheed Haque, Zhe Fu, Depeng Xu, Shuhan Yuan, and Xi Niu. 2025a. Fine-tuning LLMs with cross-attention-based weight decay for bias mitigation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15785–15798, Suzhou, China. Association for Computational Linguistics.

Farsheed Haque, Depeng Xu, and Xi Niu. 2025b. A comprehensive survey on bias and fairness in large language models. In *Trends and Applications in Knowledge Discovery and Data Mining*, pages 83–101, Singapore. Springer Nature Singapore.

Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA. Curran Associates Inc.

Mengxuan Hu, Hongyi Wu, Ronghang Zhu, Zihan Guan, Dongliang Guo, Daiqing Qi, and Sheng Li. 2025. No free lunch: Retrieval-augmented generation undermines fairness in LLMs, Even for vigilant users. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 18145–18170, Suzhou, China. Association for Computational Linguistics.

Muneeswaran I, Advaith Shankar, Varun V, Saisubramaniam Gopalakrishnan, and Vishal Vaddina. 2024. Mitigating factual inconsistency and hallucination in large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 1169–1170, New York, NY, USA. Association for Computing Machinery.

Yuelyu Ji, Hang Zhang, and Yanshan Wang. 2025. Bias evaluation and mitigation in retrieval-augmented medical question-answering systems.

Taeyoun Kim, Jacob Springer, Aditi Raghunathan, and Maarten Sap. 2025. Mitigating bias in rag: Controlling the embedder. *arXiv preprint arXiv:2502.17390*.

To Eun Kim and Fernando Diaz. 2025. Towards fair rag: On the impact of fair ranking in retrieval-augmented generation. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, ICTIR '25, page 33–43, New York, NY, USA. Association for Computing Machinery.

Svetlana Kiritchenko and Saif M Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics (* SEM)*, pages 43–53.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large language models: A survey.

Daisuke Oba, Masahiro Kaneko, and Danushka Bollegala. 2024. In-contextual gender bias suppression for large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1722–1742, St. Julian's, Malta. Association for Computational Linguistics.

Jahna Otterbacher. 2018. Addressing social bias in information retrieval. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 121–127. Springer.

Navid Rekabsaz and Markus Schedl. 2020. Do neural ranking models intensify gender bias? In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 2065–2068, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Shirin Seyedsalehi, Amin Bigdeli, Negar Arabzadeh, Batool AlMousawi, Zack Marshall, Morteza Zihayat, Ebrahim Bagheri, et al. 2025. Understanding and mitigating gender bias in information retrieval systems. *Foundations and Trends® in Information Retrieval*, 19(3):191–364.

Gemma Team. 2024. Gemma.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xuyang Wu, Shuowei Li, Hsin-Tai Wu, Zhiqiang Tao, and Yi Fang. 2025. Does RAG introduce unfairness in LLMs? evaluating fairness in retrieval-augmented generation systems. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10021–10036, Abu Dhabi, UAE. Association for Computational Linguistics.

Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, SSDBM '17, New York, NY, USA. Association for Computing Machinery.

Zheng Yao, Shuai Wang, and Guido Zuccon. 2025. Pretraining vs. fine-tuning: A reproducibility study on dense retrieval knowledge acquisition. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3276–3285, New York, NY, USA. Association for Computing Machinery.

George Zerveas, Navid Rekabsaz, Daniel Cohen, and Carsten Eickhoff. 2022. Mitigating bias in search results through contextual document reranking and neutrality regularization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2532–2538.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 335–340. ACM.

Tianhui Zhang, Yi Zhou, and Danushka Bollegala. 2025a. Evaluating the effect of retrieval augmentation on social biases.

Zheng Zhang, Ning Li, Qi Liu, Rui Li, Weibo Gao, Qingyang Mao, Zhenya Huang, Baosheng Yu, and Dacheng Tao. 2025b. The other side of the coin: Exploring fairness in retrieval-augmented generation. *ArXiv*, abs/2504.12323.

A Implementation Details

A.1 Environment and Stack.

All components are implemented in Python. We use `Haystack` for indexing/retrieval, `FAISS` for dense ANN search, `sentence-transformers` for embeddings, and HuggingFace `transformers` for LLM inference. Computation seeds are fixed (`SEED=42`); NumPy/PyTorch are used for numerical routines. Experiments run on a single GPU for LLM/encoder inference and CPU for reranking, RSBE, and calibration.

A.2 Corpora and Preprocessing

External corpus (Humans of Wikipedia). We ingest a CSV (`wiki_people_subset_24.csv`) with columns {`text`, `gender`, `title`, `occupation_level_3`}. Each page is split into fixed-size passages of at most `MAX_TOKENS=150` words. The passages retain metadata (`gender`, `title`, `occupation_level_3`).

RAG train/test sets (Bias in Bios). Each example provides a biography, a binary protected attribute (gender) and an occupation label. The train split is loaded from `dev.csv`. For test-time reporting, we iterate over per-occupation CSV files in `bios_test/` (one file per target occupation).

CGLC optimization time train set construction (Bias in Bios). From the `LabHC/bias_in_bios` *test* split, we retain professions with $\geq$25 male and $\geq$25 female bios. For each retained profession o, we create a 100-example balanced mini-set: 25 male + 25 female positives (original profession $= o$, y=1) and 25 male + 25 female negatives (drawn from professions $\neq o$, y=0), without row reuse. We preserve `source_profession` for auditing, concatenate all mini-sets, shuffle, verify exact $50/50$ per profession and uniqueness, and save as `train.csv`.

Inference time test set construction (Bias in Bios). From the `LabHC/bias_in_bios` train split, we first retain only occupations with at least 100 biographies per gender. For each eligible occupation o (22 in total), we build a per-occupation CSV with exactly 400 rows: 100 male and 100 female *relevant* biographies (profession $= o$), and 100 male and 100 female *irrelevant* biographies (profession $\neq o$). This yields a perfectly balanced file (200 relevant / 200 irrelevant; 200 male / 200 female). Aggregating across 22 occupations produces $22 \times 400 = 8{,}800$ test examples. Files are saved under `bios_test/` and named by occupation (e.g., `surgeon.csv`); sampling uses a fixed seed for reproducibility.

A.3 Dense Index: FAISS + BGE

We build a dense index with `FAISSDocumentStore` configured as:

- `faiss_index_factory_str`: `HNSW32`;
- similarity: `dot_product`; embedding dim: 1024;
- SQL backing: `sqlite:///faiss_wiki.db`; index name: `wiki_people.faiss`;

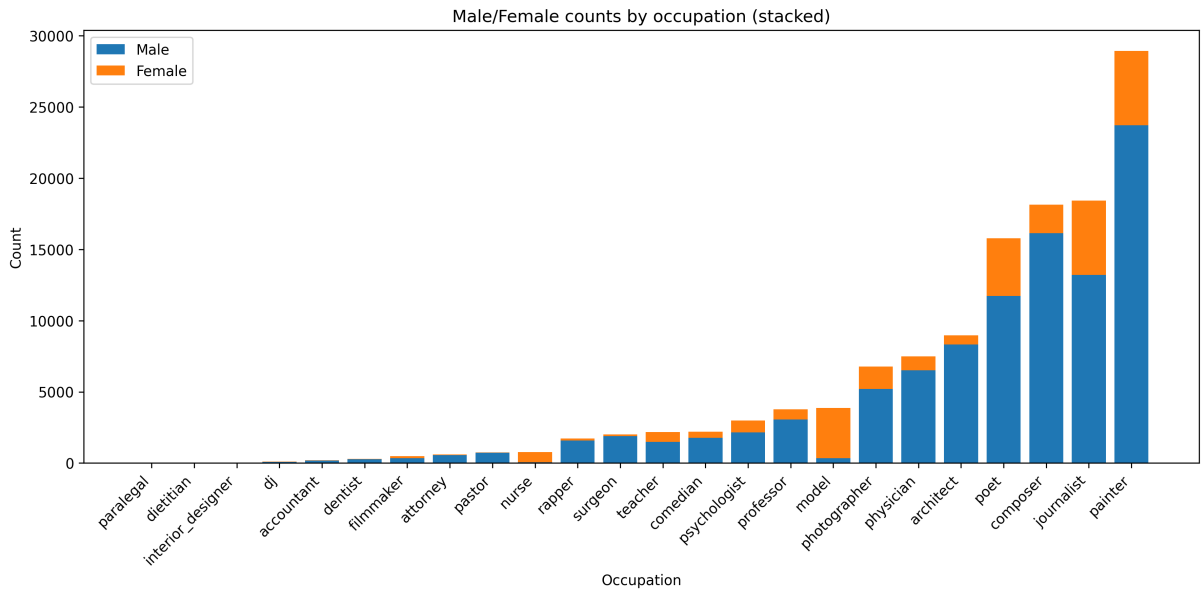

Figure 2: Male vs. Female distribution per occupation in the external corpus

- query search width: `ef_search=1024`.

Embeddings are computed with `BAAI/bge-large-en` (FP16 on GPU) in batches of 256. We persist `wiki_people.faiss` and a JSON sidecar for later loading.

A.4 Sparse Index: BM25

We also build a sparse `InMemoryDocumentStore` with `use_bm25=True` from the same passages/metadata and persist it via `save_to_disk` (or a pickle fallback for legacy versions). A `BM25Retriever` provides baseline sparse retrieval.

A.5 Retrieval and Fair Reranking

At inference, we encode queries with `BAAI/bge-large-en`, normalize vectors, and retrieve `TOP_K=100` candidates from FAISS. We then apply FGR to produce a display slate of `DISPLAY_K=15` items:

A.6 Optimizing the Confidence-Gated Logit Calibrator (CGLC)

We learn the parameter set

$$\theta \;=\; \{\, \delta,\ \gamma,\ b,\ \beta,\ (\delta_o,\ b_o)_{o\in\mathcal{O}} \,\},$$

where δ is a global logit bias, γ controls the confidence gate, b is a global group offset, β weight the positive/negative standardized residual signals, and (δ_o, b_o) are per-class corrections.

Search strategy. Stage A: Coarse random search (global). Sample candidate tuples $(\delta, \gamma, b, \beta)$ from bounded, task-appropriate ranges; evaluate on a held-out development subset; keep the best as the warm start $\theta^{(0)}$.

Stage B: Short coordinate descent (global). From $\theta^{(0)}$, sweep each of δ, γ, b, β over small centered grids; accept any change that lowers the objective. A few passes typically stabilize all six globals.

Stage C: Discrete per-class tuning. Fix $(\delta, \gamma, b, \beta)$ and refine (δ_o, b_o) per class o via discrete searches over compact symmetric sets, subject to a hard clip on $|\Delta|$. Iterate over classes until no further improvement.

Practical details. (i) *Subsampling.* Stage A uses a random subset; Stages B–C use the full development set.
(ii) *Constraints.* Enforce $\gamma \geq 0$ and hard-clip Δ for stability.
(iii) *Early stopping.* Each stage halts when a complete pass yields no improvement.
(iv) *Ablations.* Choice of residual signal and optional gating are fixed consistently between training and inference.
(v) *Freezing for inference.* The learned $\theta^{\star}$ and required summary statistics/flags are serialized and applied post-hoc at test time.

B Prompt

At test time we cast occupation prediction as a constrained multiple–choice task with three options (target class and two distractors). For each biography x with subject gender g, we provide the LLM with (i) a short instruction that it is a classifier, (ii) the three candidate occupations, (iii) the verbatim biography, and (iv) the retrieved context slate S_K (each passage truncated and annotated with its metadata gender). The model is required to output *only* the chosen occupation string.

Template (filled per instance).

```
You are a classifier.

Which of the following three professions
best matches the person described in the
biography below? Choose exactly one:
  - {OP1}
  - {TARGET}
  - {OP2}

Output only the profession name.

Biography:
"""
{BIO}
"""

Context documents:
{text: {CTX_PASSAGE_1}

{text: {CTX_PASSAGE_2}

... (up to K passages)

Answer:
```

Notes. (i) The two distractors {OP1, OP2} are sampled from other occupations to form a 3-way choice centered on the target. (ii) Context passages are drawn from the reranked slate S_K, each truncated to a fixed budget and shown with its metadata gender to make exposure explicit. (iii) The "output-only" constraint prevents explanations and enforces a single-label decision, which simplifies logit extraction and evaluation."'

C Detailed Evaluation Metrics

C.1 Retriever utility

Hit@K : fraction of queries with at least one relevant in top K.

$$\mathrm{Hit@}K \;=\; \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\Big[\max_{1\le j\le K} r_{i,j}=1\Big],$$

where $r_{i,j}\in\{0,1\}$ indicates relevance of the j-th retrieved item for query i.

MRR : mean inverse rank of the first relevant item.

$$\mathrm{MRR} \;=\; \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\min\{j\,:\,r_{i,j}=1\}} \quad \text{(0 if none).}$$

NDCG@K : DCG@K normalized by the ideal DCG@K.

$$\mathrm{DCG}_i@K \;=\; \sum_{j=1}^{K}\frac{r_{i,j}}{\log_2(j+1)},$$

$$\mathrm{NDCG@}K \;=\; \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{DCG}_i@K}{\mathrm{IDCG}_i@K}.$$

C.2 Retriever fairness

Signed NDKL (prefix) : Defined in §3.2

Min/Max skew (log-odds over prefixes) :

$$\mathrm{Skew}_i^{\min/\max} \;=\; \min/\max_{1\le t\le K}\log\frac{\hat{p}_t(a_0)+\epsilon}{\hat{p}_t(a_1)+\epsilon},$$

for a binary $\mathcal{A}=\{a_0,a_1\}$ and a small $\epsilon>0$.

C.3 Generator utility

Accuracy : $\mathrm{Acc}=\frac{1}{N}\sum_{i=1}^{N}\mathbf{1}[\hat{o}_i=o_i]$.

Precision/Recall (per class c) : $\mathrm{Prec}_c=\frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c}$, $\mathrm{Rec}_c=\frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FN}_c}$.

F1 (per class, then macro) : $\mathrm{F1}_c=\frac{2\,\mathrm{Prec}_c\,\mathrm{Rec}_c}{\mathrm{Prec}_c+\mathrm{Rec}_c}$, $\mathrm{F1}_{\text{macro}}=\frac{1}{|\mathcal{O}|}\sum_{c\in\mathcal{O}}\mathrm{F1}_c$.

C.4 Generator fairness

Risk Difference (DP) per class c :

$$\mathrm{RD}_c \;=\; \big|\,P(\hat{o}=c\mid g=a_0)-P(\hat{o}=c\mid g=a_1)\,\big|,$$

$$\mathrm{RD}_{\text{mean}}=\frac{1}{|\mathcal{O}|}\sum_{c\in\mathcal{O}}\mathrm{RD}_c.$$

TPR gap (Equal Opportunity) per class c :

$$\mathrm{TPRGap}_c = \big| P(\hat{o} = c \mid o = c, g = a_0) - P(\hat{o} = c \mid o = c, g = a_1) \big|,$$

$$\mathrm{TPRGap}_{\text{mean}} = \frac{1}{|\mathcal{O}|} \sum_{c \in \mathcal{O}} \mathrm{TPRGap}_c.$$

D Baselines

For the retriever side, we evaluate both dense and sparse retrievers and for generator side, we use different pre-trained models for evaluations, including Flan-T5 (3B version) (Chung et al., 2022), Llama2 (7B version) (Touvron et al., 2023), Gemma-2B (Team, 2024), DeepSeek (6.7B version) (DeepSeek-AI et al., 2025), and Falcon (7B version) (Almazrouei et al., 2023). compare four fairness-aware baselines:

- **DetConstSort (Deterministic Constrained Sorting).** (Geyik et al., 2019) An interval-constrained reranking algorithm that enforces per-attribute minimum prefix counts while preserving score order as much as feasible via local swaps. It selects attributes whose minimum requirement just increased, inserts the next candidate, and bubble-swaps left unless doing so would violate score order or max-index feasibility. Deterministic; no retraining.
- **FairFilter (Filter-based).** (Zhang et al., 2025b) A two-step LLM prompting filter applied to retrieved documents: (i) *fairness screening* drops documents flagged as biased/stereotypical/harmful; (ii) *utility screening* re-checks relevance among the remaining items to preserve accuracy. Post-hoc and retriever-agnostic, designed to balance fairness and utility.
- **FairFT (Alignment-based Fine-Tuning).** (Zhang et al., 2025b) Aligns the retriever to the LLM's fairness preference by computing retrieval likelihoods $P_R(d \mid q)$ over top-k docs and an LLM-derived fairness distribution $Q_{\text{LM}}(d \mid q, a)$, then updating the retriever by minimizing $\mathrm{KL}(P_R \| Q_{\text{LM}})$ (LLM frozen). This steers future retrieval toward fairness-supportive evidence. Due to the nature of this method it cannot be applied in sparse retriever.
- **In Context:** (Oba et al., 2024) Prompt-only bias suppression that prepends short, templated "preambles" to the user query/contexts. Two preamble families are used: (i) *counterfactual statements* that invert occupational gender stereotypes (e.g., "Despite being a female, Alex became a software engineer"), and (ii) *gender-neutral descriptions* of stereotyped objects (e.g., occupations) built from real-world statistics. No model weights or decoding are modified; the method is plug-and-play for closed LLMs and can be applied directly in RAG by inserting the preamble before generation. Empirically shown to suppress gender bias with limited utility degradation on standard probes.

E FGR Proof

Algorithm 1 Fair Greedy (FGR)

Require: initial retrieved slate $S_T = [d_1, \ldots, d_T]$, target mix $\mathbf{q}$, final slate length K.
$picked \leftarrow [\,]$, $remaining \leftarrow S_T$
for $k = 1$ to K **do**
  compute $U_k[a_m] = q_m \cdot k - count[a_m]$ for all a_m
  $context \leftarrow$ first doc in $remaining$ whose group is in $\max_{a_m \in \mathcal{A}} U_k[a_m]$
  if $context$ not found **then**
    $context \leftarrow$ first doc in $remaining$
  end if
  append $context$ to $picked$; remove $context$ from $remaining$
end for
return $S_K = picked$

Properties of FGR The Fair Greedy (FGR) algorithm ensures that at every step k, (i) a valid group with a positive prefix deficit exists, (ii) the fairness error is bounded by prioritizing the reduction of the maximum deficit, and (iii) relevance is maximized subject to the fairness constraint.

proof Let $N_{k-1}(a)$ denote the count of group a in the partial slate S_{k-1}. The deficit is defined as $U_k(a) = q_a \cdot k - N_{k-1}(a)$.

1. Existence of Underrepresented Group. Summing the deficits over all groups $\mathcal{A}$:

$$\sum_{a \in \mathcal{A}} U_k(a) = \sum_{a \in \mathcal{A}} (q_a k - N_{k-1}(a)) =$$

$$k \sum_a q_a - \sum_a N_{k-1}(a) = k(1) - (k-1) = 1.$$

Since $\sum U_k(a) = 1$, by the pigeonhole principle, $\exists a \in \mathcal{A}$ such that $U_k(a) > 0$. Thus, as long as candidates remain, the set of underrepresented groups is non-empty.

2. Bounded Fairness Deviation. Let $a^* = \arg\max_a U_k(a)$ be the group selected at step k. The deficit for step $k+1$ becomes:

$$\begin{aligned} U_{k+1}(a^*) &= q_{a^*}(k+1) - (N_{k-1}(a^*) + 1) \\ &= (q_{a^*}k - N_{k-1}(a^*)) + q_{a^*} - 1 \\ &= U_k(a^*) - (1 - q_{a^*}). \end{aligned}$$

Since $q_{a^*} < 1$, the deficit for the selected group strictly decreases ($U_{k+1}(a^*) < U_k(a^*)$). This negative feedback loop ensures the maximum deviation does not grow monotonically.

3. Relevance Maximization. The algorithm selects document $d^* \in remaining$ such that $g(d^*) = a^*$. Since the set $remaining$ is initialized as S_T (ordered by relevance) and traversed linearly, d^* is necessarily the highest-relevance document available satisfying the group constraint a^*.