ELGAR: EXPOSURE-LEDGER GUIDED ADAPTIVE ROUTING FOR FAIR GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation (GR) has recently attracted growing attention, with Semantic-ID (SID)-based GR emerging as a major paradigm. They formulate recommendation as autoregressive item generation, where each item is represented by a sequence of discrete semantic tokens. Despite strong preference modeling, these models tend to over-expose popular items, causing realized item exposure to deviate from the popularity-calibrated fairness target. Existing methods face three limitations: *misalignment* between token-level improvements and group-level exposure, *rigidity* in corrections that do not respond to evolving exposure deficits, and an *adaptation burden* when debiasing requires changes to SID construction and model training.

We propose **ELGAR**, an **E**xposure-**L**edger **G**uided Adaptive **R**outing framework for fair GR. Specifically, ELGAR prepends a popularity-group token to each item's existing SID, making group routing the first generation decision before semantic pruning. An Exposure Ledger records group-wise deviations from the training-interaction reference and is updated from displayed recommendations. At the routing step, ELGAR combines ledger-conditioned fairness signals with user-preference signals, allowing subsequent decisions to respond to exposure deficits while retaining each backbone's existing semantic codes. Experiments on four datasets and three backbones demonstrate improved fairness while maintaining recommendation performance. Our code is available at `https://anonymous.4open.science/r/ELGAR-Anon-FB51`.

1 INTRODUCTION

Generative recommendation (GR) has attracted growing attention, particularly approaches based on Semantic IDs (SIDs) (Rajput et al., 2023; Wang et al., 2024a; Kong et al., 2025). These methods represent each item as a sequence of discrete tokens and generate its SID autoregressively based on the user's interaction history. Unlike direct item scoring, this process progressively restricts the reachable candidate set. Each prefix identifies a subtree of items, and beam search discards prefixes with lower scores. SIDs therefore serve both as item representations and as a *candidate-access structure* that determines which items remain reachable during decoding (Rajput et al., 2023).

This mechanism creates additional pathways for popularity bias, which enters the pipeline at two stages (Yin et al., 2026; Fan et al., 2026). During training, maximum likelihood on long-tailed data provides more positive supervision to frequently observed items and tokens (Yin et al., 2026); although item popularity and token frequency are not identical, a tail item whose distinguishing tokens receive little support obtains a less competitive generation path. Besides, decoding then amplifies the disparity: pruning a prefix removes every item beneath it, so an item's exposure opportunity depends on the survival of its entire generation path. The result is a concentration of exposure that reduces long-tail discoverability and catalog coverage (Abdollahpouri et al., 2019) and is reinforced by feedback loops (Chaney et al., 2018). Figure 1 shows that, although each popularity group accounts for 10% of training interactions, the most popular group receives approximately 20%–40% of recommendation exposure across the three backbones, while tail groups remain underexposed relative to their interaction shares.

Addressing this popularity bias amounts to an item-side fairness problem: preserving relevance while correcting the over-concentration of exposure on popular items. We therefore study

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

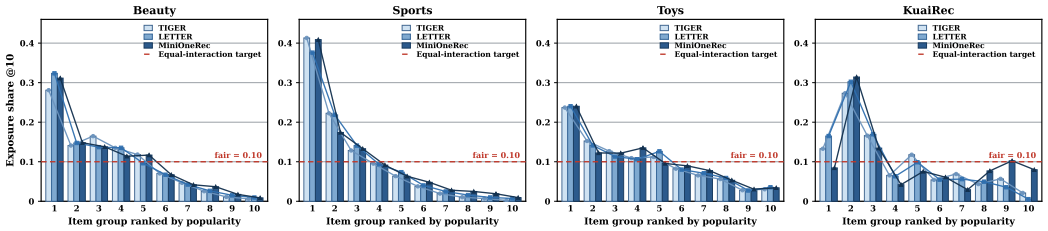

Figure 1: Exposure remains concentrated on head items across all generative recommenders. Items are sorted by popularity into $G = 10$ equal-interaction groups, each accounting for 10% of training interactions, from the most popular ($g = 1$) to the least popular ($g = 10$). The red dashed line denotes exposure proportional to interaction demand (0.10 per group).

popularity-calibrated fairness (Jiang et al., 2024; Abdollahpouri et al., 2020), where cumulative exposure across popularity groups is aligned with their training-interaction shares (Singh & Joachims, 2018). Existing methods fall short of this objective in three ways. **(i) Misalignment.** LETTER (Wang et al., 2024a) regularizes code assignment for diversity, CRAB (Fan et al., 2026) rebalances codebooks by splitting popular tokens, and CARE (Lin et al., 2026) mitigates token-level bias through cascaded reasoning. These methods operate on tokens or generation paths, but token-level balance does not directly control exposure across item groups because tokens are shared by items with different popularity levels. **(ii) Rigidity.** IFairLRS (Jiang et al., 2024) directly corrects item-group bias through reweighting and reranking, but its corrections are based on aggregate group statistics rather than the evolving exposure state during serving. Moreover, reranking can only act on candidates that survive generation. **(iii) Adaptation burden.** Ghost (Yin et al., 2026) mitigates path competition through specialized SID construction and asymmetric unlikelihood training. Because its debiasing mechanism jointly modifies item tokenization and the training objective, adapting it to a new generative backbone requires changes to both the SID pipeline and the training recipe, rather than lightweight integration with an existing model.

This calls for a mechanism that **adaptively adjusts group-level allocation to cumulative exposure as requests arrive**. We propose **ELGAR (Exposure-Ledger Guided Adaptive Routing)**, a backbone-agnostic framework that separates group-level exposure allocation from within-group SID generation. Realizing this framework raises three challenges. *Where should group allocation occur?* SID generators make token-level decisions, but the fairness objective concerns item groups. Once a group's generation paths have been pruned, later correction cannot recover its items. *How can allocation respond to changing exposure?* The relevant deficit depends on items actually displayed across requests, whereas standard training data contain no serving-time exposure stream. *How can fairness preserve relevance?* An underexposed group is not necessarily suitable for the current user, so routing cannot depend on exposure deficits alone.

ELGAR prepends a popularity-group token to each item's SID, making group routing the first generation decision. An **Exposure Ledger** records group-wise deviations from the training-interaction reference and tracks exposure progress. During training, shuffled mini-batches construct short exposure episodes that provide varying ledger states, and one-step fairness gains indicate how each routing choice would change the current discrepancy. A ledger-conditioned fairness head produces group logits that are fused with the backbone's relevance logits at the routing step; target-group supervision keeps this decision sensitive to user history. At inference, displayed recommendations update the ledger, allowing subsequent requests to respond to realized exposure. The remaining SID tokens are generated using the backbone's existing semantic codes. In summary, we make three contributions: (1) We formulate fairness-aware SID generation as stateful group-level allocation followed by within-group relevance modeling, connecting cumulative exposure objectives to generation decisions before semantic pruning. (2) We propose ELGAR, which combines an explicit group-routing token with an Exposure Ledger to connect generation decisions to cumulative exposure. Ledger-conditioned fairness signals are fused with the backbone's relevance-based routing and learned through one-step fairness gains and target-group supervision, while displayed Top-K recommendations update the ledger for subsequent requests. (3) We evaluate ELGAR on four datasets and three SID-based generative backbones spanning encoder–decoder and decoder-only architectures, demonstrating improved fairness while maintaining recommendation performance.

2 PRELIMINARIES

2.1 GENERATIVE RECOMMENDATION

Sequential recommendation predicts the next item from a user's interaction history (Kang & McAuley, 2018). SID-based GR formulates this task as autoregressive generation of discrete item identifiers (Rajput et al., 2023; Zheng et al., 2024; Wang et al., 2024a). Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets, respectively. For each user $u \in \mathcal{U}$, the chronological interaction sequence is $\mathbf{S}_u = (i_{u,1}, \ldots, i_{u,T_u})$. At position τ, the model predicts the next item $i_{u,\tau}$ from the preceding history $\mathbf{x}_{u,\tau} = (i_{u,1}, \ldots, i_{u,\tau-1})$.

SIDs are commonly constructed using vector quantization methods such as VQ-VAE (van den Oord et al., 2017), RQ VAE (Lee et al., 2022), and RQ-KMeans (Luo et al., 2024). Taking residual quantization as an example, given an item embedding $\mathbf{v}_i$, let $\boldsymbol{\epsilon}_i^{(1)} = \mathbf{v}_i$ be the initial residual and $\mathbf{e}_a^{(\ell)}$ be the a-th codeword at level ℓ. The code index and residual are updated as

$$c_i^{(\ell)} = \arg\min_a \left\| \boldsymbol{\epsilon}_i^{(\ell)} - \mathbf{e}_a^{(\ell)} \right\|_2^2, \qquad \boldsymbol{\epsilon}_i^{(\ell+1)} = \boldsymbol{\epsilon}_i^{(\ell)} - \mathbf{e}_{c_i^{(\ell)}}^{(\ell)}. \tag{1}$$

The resulting code indices form the semantic identifier $\mathrm{SID}(i) = [c_i^{(1)}, \ldots, c_i^{(L_i)}]$, where L_i is the SID length and $c_i^{(<\ell)} = [c_i^{(1)}, \ldots, c_i^{(\ell-1)}]$ denotes the prefix before level ℓ. The model learns to generate SIDs autoregressively from interaction histories using token-level negative log-likelihood and produces Top-K recommendations through beam search at inference.

2.2 POPULARITY-CALIBRATED ITEM-SIDE FAIRNESS

We formulate popularity-calibrated item-side fairness at the group level. Let f_i denote the interaction count of item i in the *held-2-out training split*, and let $\mathcal{G} = \{1, \ldots, G\}$ denote the set of G mutually exclusive popularity groups. The mapping $g : \mathcal{I} \to \mathcal{G}$ assigns each item to one group. Treating each observed training interaction as one unit of historical mass, we define the training-interaction reference distribution as

$$\pi_g = \frac{\sum_{i \in \mathcal{I}} f_i \, \mathbb{I}[g(i) = g]}{\sum_{i \in \mathcal{I}} f_i}, \qquad \boldsymbol{\pi} = (\pi_1, \ldots, \pi_G). \tag{2}$$

For a collection $\mathcal{Q}$ of recommendation requests, let $\mathcal{R}_t = (i_t^{(1)}, \ldots, i_t^{(K)})$ denote the Top-K list returned for request $t \in \mathcal{Q}$. Treating each displayed position as one unit of exposure, let $n_g = \sum_{t \in \mathcal{Q}} \sum_{k=1}^{K} \mathbb{I}[g(i_t^{(k)}) = g]$ denote the exposure count of group g, and $\mathbf{n} = (n_1, \ldots, n_G)$ the corresponding exposure-count vector. We define its normalized exposure distribution as

$$\rho_g(\mathbf{n}) = \frac{n_g}{\sum_{h=1}^{G} n_h}, \qquad \boldsymbol{\rho}(\mathbf{n}) = (\rho_1(\mathbf{n}), \ldots, \rho_G(\mathbf{n})). \tag{3}$$

For the empty exposure-count vector $\mathbf{n} = \mathbf{0}$, we set $\boldsymbol{\rho}(\mathbf{0}) = \mathbf{0}$ by convention to represent the initial exposure condition.

Following popularity-calibrated item-side fairness formulations (Jiang et al., 2024; Abdollahpouri et al., 2020), we measure the mismatch between recommendation exposure and the training-interaction reference distribution by

$$D(\mathbf{n}; \boldsymbol{\pi}) = \sum_{g=1}^{G} \left| \rho_g(\mathbf{n}) - \pi_g \right|. \tag{4}$$

Our objective is to reduce this exposure discrepancy D while preserving recommendation utility.

3 METHOD

ELGAR characterizes the group-level exposure state with an **Exposure Ledger** and realizes popularity-calibrated item-side fairness through ledger-guided adaptive routing. A prepended

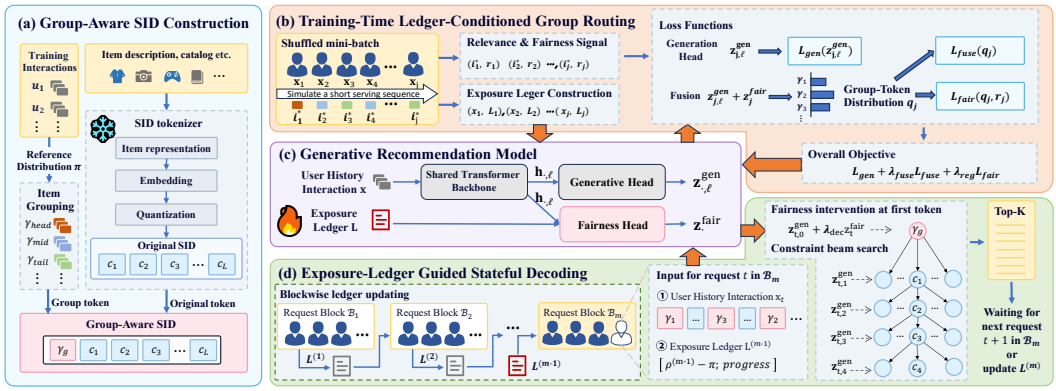

Figure 2: Overview of ELGAR: (a) group-augmented SIDs; (b) training-time exposure-state construction from shuffled mini-batches; (c) fusion of generative and ledger-conditioned signals for group routing; and (d) inference with blockwise exposure updates.

popularity-group token exposes an explicit group-routing decision before item-specific semantic decoding. During training, the ledger and one-step fairness gains provide the state-dependent signal for learning routing, while $\mathcal{L}_{\text{fair}}$ promotes exposure alignment and $\mathcal{L}_{\text{fuse}}$ preserves target-group relevance. At inference, displayed Top-K recommendations update the ledger for subsequent requests, forming a closed loop of ledger $\rightarrow$ adaptive routing $\rightarrow$ recommendation $\rightarrow$ ledger. Figure 2 summarizes the framework, while the complete serving-time decoding and training procedures are provided in Algorithms A1 and A2 in Appendix J.

3.1 Group-Aware SID Routing

Using the popularity group mapping $g(i)$ defined in Sec. 2.2, we expose group membership as an explicit routing decision in the generative process. Given the backbone-specific identifier $\text{SID}(i) = [c_i^{(1)}, \ldots, c_i^{(L_i)}]$, we introduce one dedicated token γ_g for each popularity group and denote the group-token set by $\Gamma = \{\gamma_1, \ldots, \gamma_G\}$. These tokens are added to the backbone output vocabulary, and we prepend the corresponding token to the original identifier,

$$\widetilde{\text{SID}}(i) = [\gamma_{g(i)}, c_i^{(1)}, \ldots, c_i^{(L_i)}]. \tag{5}$$

The original SID is therefore unchanged after the first position. At the first decoding step, only tokens in Γ are valid. Once γ_g is selected, every subsequent token must remain a valid SID prefix of an item assigned to group g; the decoded path cannot switch groups after the routing decision.

For recommendation context $\mathbf{x}$, generation consequently factorizes as

$$p_\theta\left(\widetilde{\text{SID}}(i) \mid \mathbf{x}\right) = p_\theta\left(\gamma_{g(i)} \mid \mathbf{x}\right) \prod_{\ell=1}^{L_i} p_\theta\left(c_i^{(\ell)} \mid \mathbf{x}, \gamma_{g(i)}, c_i^{(<\ell)}\right). \tag{6}$$

Thus, the first autoregressive step provides a group-level control point before item-specific semantic decoding. ELGAR intervenes at this routing step and leaves the backbone's within-group SID generation unchanged.

3.2 Training-Time Exposure Ledger and One-Step Fairness Gain

Training does not provide a persistent serving stream. Rather than reproducing the model-induced serving trajectory, we use each shuffled mini-batch of size B to construct a short sequence of exposure states for learning a state-conditioned correction rule. Let $\mathcal{E} = \{(\mathbf{x}_j, i_j^*)\}_{j=1}^{B}$ denote one shuffled training mini-batch, treated as a short exposure episode in its loader order. Each request contributes one item-level exposure event through its ground-truth target i_j^*. Let $g_j = g(i_j^*)$, and let $\mathbf{n}_j = (n_{j,1}, \ldots, n_{j,G})$ denote the exposure counts accumulated from the preceding requests in the same batch, with $\mathbf{n}_1 = \mathbf{0}$.

Rather than feeding these unbounded counts directly to the model, we define the **Exposure Ledger** for request j as

$$\mathbf{L}_j = \left[\boldsymbol{\rho}(\mathbf{n}_j) - \boldsymbol{\pi}; \min\left(\frac{\sum_{g=1}^{G} n_{j,g}}{H}, 1\right)\right] \in \mathbb{R}^{G+1}, \tag{7}$$

where $H > 0$ is a ledger horizon measured in item-exposure units. The first G dimensions record the current group-wise exposure deviation from the training-interaction reference distribution $\boldsymbol{\pi}$, while the last dimension acts as an exposure-level warm-up indicator: it increases during the first H accumulated item exposures and remains at 1 thereafter.

To quantify how each possible group-routing decision would affect the exposure mismatch under the current ledger $\mathbf{L}_j$, we introduce a training-only **one-step fairness gain**, where "one-step" refers to the next group-token routing decision. Let $\boldsymbol{\delta}_g$ denote the one-hot vector for group g. Using the discrepancy in Eq. 4,

$$r_j(g) = D(\mathbf{n}_j; \boldsymbol{\pi}) - D(\mathbf{n}_j + \boldsymbol{\delta}_g; \boldsymbol{\pi}), \qquad g = 1, \ldots, G, \tag{8}$$

where $r_j(g) > 0$ indicates that routing the next exposure to group g would reduce the current mismatch. We collect the gains as $\mathbf{r}_j = [r_j(1), \ldots, r_j(G)]$.

After constructing $(\mathbf{L}_j, \mathbf{r}_j)$, we update the raw exposure counts with the ground-truth target group, $\mathbf{n}_{j+1} = \mathbf{n}_j + \boldsymbol{\delta}_{g_j}$, and then construct $(\mathbf{L}_{j+1}, \mathbf{r}_{j+1})$ from the updated counts $\mathbf{n}_{j+1}$ for the next request. Thus, requests within a mini-batch sequentially observe the exposure accumulated from preceding requests, while the counts are reset for each new mini-batch. Repeated resets expose the router to finite-prefix ledger states at different stages of accumulation. The resulting ledger–gain pairs provide the exposure-aware conditioning and fairness signal for adaptive group routing in Sec. 3.3.

3.3 Ledger-Conditioned Group Routing

With $(\mathbf{L}_j, \mathbf{r}_j)$, ELGAR learns a routing distribution that combines request-specific relevance with the current exposure imbalance. We retain the backbone generative head W_{gen} and add a fairness head W_f on top of the same decoder backbone. The two heads share the backbone hidden representations but use different output projections and learning signals.

Generative Head. For training sample j, let $\mathbf{h}_{j,\ell} \in \mathbb{R}^d$ denote the decoder hidden representation used to predict the ℓ-th token $\widetilde{c}_{j,\ell}$ of the group-aware SID. The backbone generative head produces $\mathbf{z}^{\text{gen}}_{j,\ell} = W_{\text{gen}}\mathbf{h}_{j,\ell}$. We denote the hidden representation used for the first group-token prediction by $\mathbf{h}^G_j \in \mathbb{R}^d$ and its generative logits restricted to Γ by $\mathbf{z}^G_j \in \mathbb{R}^G$. Let $p_{\text{gen}}(\cdot \mid \mathbf{x}_j, \widetilde{c}_{j,<\ell}) = \mathrm{softmax}(\mathbf{z}^{\text{gen}}_{j,\ell})$. For target item i^*_j, define $\widetilde{c}_{j,0} = \gamma_{g_j}$ and $\widetilde{c}_{j,\ell} = c^{(\ell)}_{i^*_j}$ for $\ell \geq 1$. The generative objective is

$$\mathcal{L}_{\text{gen}} = -\frac{1}{B}\sum\nolimits_{j=1}^{B}\sum\nolimits_{\ell=0}^{L_{i^*_j}} \log p_{\text{gen}}(\widetilde{c}_{j,\ell} \mid \mathbf{x}_j, \widetilde{c}_{j,<\ell}). \tag{9}$$

It trains the backbone's relevance-driven group preference together with subsequent SID generation.

Fairness Head and Group Fusion. At the group-routing position, the fairness head W_f combines the group-prediction representation with the current Exposure Ledger,

$$\mathbf{z}^{\text{fair}}_j = W_f\Big(\mathbf{h}^G_j + f_\phi(\mathbf{L}_j)\Big) \in \mathbb{R}^G, \qquad \mathbf{q}_j = \mathrm{softmax}\Big(\mathbf{z}^G_j + \mathbf{z}^{\text{fair}}_j\Big), \tag{10}$$

where $f_\phi : \mathbb{R}^{G+1} \to \mathbb{R}^d$ is the ledger adapter and $\mathbf{q}_j$ is the *ledger-conditioned group-routing distribution* obtained by fusing the backbone's generative group logits with fairness logits. Fairness intervention is applied only at the first routing step, whereas subsequent SID prediction retains the backbone's original generative head.

To explicitly learn this fused routing distribution, we supervise $\mathbf{q}_j$ with the observed target group,

$$\mathcal{L}_{\text{fuse}} = -\frac{1}{B}\sum\nolimits_{j=1}^{B} \log q_j(g_j). \tag{11}$$

Thus, $\mathcal{L}_{\text{fuse}}$ directly trains the relevance–fairness fusion at the group-routing step. Since the supervision is derived from the observed target group, it also anchors the fused distribution to request-level relevance and prevents fairness correction from dominating the backbone signal.

Fairness Regularization. While $\mathcal{L}_{\text{fuse}}$ learns the fused routing distribution, the one-step fairness gains $\mathbf{r}_j$ further shape this distribution according to the current exposure mismatch. We define

$$\mathcal{L}_{\text{fair}} = -\frac{1}{B}\sum\nolimits_{j=1}^{B}\sum\nolimits_{g=1}^{G} q_j(g) r_j(g). \tag{12}$$

Minimizing $\mathcal{L}_{\text{fair}}$ shifts routing probability toward groups whose next exposure would better reduce the discrepancy under the current ledger. Since $r_j(g)$ changes as the within-batch exposure counts evolve, the preferred correction direction can change from one training request to the next.

Overall Training Objective. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{gen}} + \lambda_{\text{fuse}}\mathcal{L}_{\text{fuse}} + \lambda_{\text{reg}}\mathcal{L}_{\text{fair}}, \tag{13}$$

where λ_{fuse}, λ_{reg} control relevance-preserving fused supervision and fairness regularization, respectively. Both routing objectives are defined on $\mathbf{q}_j$ at the first generation step. As both heads share the backbone, their gradients can also affect the representations used for subsequent SID generation.[1] Appendix K interprets $\mathcal{L}_{\text{fair}}$ as an exposure-alignment prior, casting ELGAR as a generalized MAP estimator relative to the backbone's MLE, and gives a conditional bound on the exposure–relevance trade-off it induces.

3.4 Exposure-Ledger Guided Stateful Decoding

At inference, the request sequence forms a single serving episode partitioned into consecutive update blocks $\{\mathcal{B}_m\}_{m=1}^{M}$. Before block m, the raw group-exposure counts $\mathbf{n}^{(m)}$, initialized with $\mathbf{n}^{(1)} = \mathbf{0}$, summarize the exposure accumulated from preceding blocks and determine a shared Exposure Ledger $\mathbf{L}^{(m)}$ through Eq. 7. All requests $t \in \mathcal{B}_m$ condition on this ledger, and the fairness head reuses Eq. 10 to obtain the corresponding ledger-conditioned logits $\mathbf{z}_t^{\text{fair},(m)}$ from the backbone group representation $\mathbf{h}_t^G$.

For each request t, the generative group logits $\mathbf{z}_t^G$ are fused with the ledger-conditioned fairness logits to produce the decoding logits

$$\mathbf{z}_t^{\text{dec},(m)} = \mathbf{z}_t^G + \lambda_{\text{dec}}\mathbf{z}_t^{\text{fair},(m)}, \tag{14}$$

where $\lambda_{\text{dec}} \geq 0$ controls the strength of the ledger-guided intervention. Constrained beam search then uses $\mathbf{z}_t^{\text{dec},(m)}$ to select the group token at the first decoding step. Once γ_g is selected, the beam search is restricted to valid SID continuations of items in group g, with no further fairness intervention at subsequent decoding steps.

The ledger remains fixed while all requests in $\mathcal{B}_m$ are decoded. Once their Top-K recommendation lists have been generated, the realized exposure is aggregated over the entire block to update the raw group-exposure counts as $n_g^{(m+1)} = n_g^{(m)} + \sum_{t\in\mathcal{B}_m}\sum_{k=1}^{K}\mathbb{I}[g(i_t^{(k)}) = g]$ for $g = 1, \ldots, G$, from which the next Exposure Ledger $\mathbf{L}^{(m+1)}$ is constructed. Serving therefore follows the closed loop

$$\mathbf{L}^{(m)} \rightarrow \text{Stateful Decoding over } \mathcal{B}_m \rightarrow \text{Aggregate Top-}K\text{ Exposure} \rightarrow \mathbf{n}^{(m+1)} \rightarrow \mathbf{L}^{(m+1)}.$$

Hence, $|\mathcal{B}_m|$ controls the refresh frequency: $|\mathcal{B}_m| = 1$ updates the ledger after every request, whereas larger blocks allow multiple requests to share the same exposure condition.

4 Experiments

4.1 Experiment Settings

Datasets. We evaluate on three Amazon 5-core datasets—Beauty, Toys and Games, and Sports and Outdoors—and on KuaiRec*, a 5-core subset derived from KuaiRec (He & McAuley, 2016; Gao et al., 2022).[2] Following prior popularity-bias research (Abdollahpouri et al., 2019), the main experiments use two routing groups ($G_{\text{route}} = 2$): Head items are the most popular items whose cumulative frequency accounts for approximately 80% of the raw training-interaction mass, and the remaining items form the Tail group. Appendix F examines alternative two-group Head–Tail split configurations and their resulting exposure redistribution across popularity regions. Fine-grained routing with $G_{\text{route}} > 2$ is examined in Section 4.5. Detailed dataset statistics, preprocessing, and split construction are provided in Appendix C.1.

Baselines. We consider three representative SID GR backbones: TIGER, LETTER, and MiniOneRec (Rajput et al., 2023; Wang et al., 2024a; Kong et al., 2025). For each backbone, we compare the vanilla model with two variants of IFairLRS (Jiang et al., 2024): **IFairLRS-RW**, which applies

[1] We also consider an optional gain-estimation variant that explicitly predicts the one-step fairness gain. Its formulation is provided in Appendix H.

[2] Amazon: cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html; KuaiRec: kuairec.com/

Table 1: Main results on four datasets. ↑/↓ indicates that higher/lower is better. Within each method block, the best and second-best available results are shown in **bold** and underlined, respectively.

Datasets	Methods	HR ↑		NDCG ↑		Tail NDCG ↑		MGU ↓	
		@5	@10	@5	@10	@5	@10	@5	@10
Beauty	TIGER	**0.0361**	**0.0559**	**0.0240**	**0.0303**	0.0009	0.0018	0.1883	0.1827
	+ IFairLRS-RW	0.0352	0.0549	0.0231	0.0295	0.0016	0.0024	0.1813	0.1736
	+ IFairLRS-RR	0.0351	0.0546	0.0230	0.0293	0.0027	0.0041	0.1199	0.1198
	+ CARE	0.0264	0.0450	0.0168	0.0228	0.0014	0.0021	0.1499	0.1520
	+ ELGAR (Ours)	0.0359	0.0556	0.0234	0.0298	**0.0037**	**0.0054**	**0.0962**	**0.0703**
	LETTER	0.0380	0.0604	0.0255	0.0327	0.0017	0.0034	0.1820	0.1753
	+ IFairLRS-RW	0.0373	0.0586	**0.0256**	0.0324	0.0038	0.0057	0.1213	0.1099
	+ IFairLRS-RR	0.0366	0.0577	0.0249	0.0317	0.0053	**0.0080**	**0.0066**	**0.0125**
	+ CARE	0.0258	0.0493	0.0156	0.0232	0.0037	0.0051	0.1294	0.1264
	+ ELGAR (Ours)	**0.0381**	**0.0611**	0.0255	**0.0329**	**0.0055**	**0.0080**	0.0656	0.0505
	Ghost	0.0360	0.0569	0.0234	0.0302	**0.0063**	0.0080	0.0295	0.0338
	MiniOneRec-SFT	0.0389	0.0593	0.0260	0.0326	0.0012	0.0018	0.1812	0.1739
	+ IFairLRS-RW	**0.0399**	**0.0608**	**0.0267**	**0.0334**	0.0021	0.0032	0.1616	0.1471
	+ IFairLRS-RR	0.0384	0.0602	0.0258	0.0329	0.0045	0.0062	0.0314	0.0519
	+ CARE	/	/	/	/	/	/	/	/
	+ ELGAR (Ours)	0.0382	0.0597	0.0249	0.0318	0.0056	**0.0084**	**0.0248**	**0.0064**
Toys	TIGER	0.0317	**0.0515**	**0.0209**	**0.0272**	0.0056	0.0082	0.1516	0.1417
	+ IFairLRS-RW	0.0283	0.0485	0.0184	0.0249	0.0035	0.0064	0.1589	0.1475
	+ IFairLRS-RR	0.0291	0.0483	0.0185	0.0247	0.0065	0.0094	0.0777	0.0844
	+ CARE	0.0258	0.0433	0.0162	0.0218	0.0034	0.0059	0.1259	0.1230
	+ ELGAR (Ours)	**0.0320**	0.0490	**0.0209**	0.0263	**0.0097**	**0.0129**	**0.0455**	**0.0300**
	LETTER	**0.0355**	**0.0564**	**0.0229**	**0.0296**	0.0078	0.0116	0.1475	0.1373
	+ IFairLRS-RW	0.0348	0.0536	0.0227	0.0287	0.0121	0.0161	0.0937	0.0842
	+ IFairLRS-RR	0.0335	0.0524	0.0217	0.0278	**0.0178**	**0.0225**	0.1299	0.0404
	+ CARE	0.0261	0.0469	0.0161	0.0228	0.0087	0.0126	**0.0381**	0.0505
	+ ELGAR (Ours)	0.0350	0.0536	0.0228	0.0288	0.0101	0.0133	0.0448	**0.0321**
	Ghost	0.0329	0.0505	0.0215	0.0271	0.0089	0.0111	0.0826	0.0851
	MiniOneRec-SFT	0.0348	**0.0567**	0.0227	**0.0297**	0.0072	0.0100	0.1474	0.1354
	+ IFairLRS-RW	0.0341	0.0541	0.0222	0.0286	0.0081	0.0108	0.1306	0.1161
	+ IFairLRS-RR	0.0333	0.0533	0.0218	0.0282	**0.0130**	**0.0160**	0.0142	0.0147
	+ CARE	/	/	/	/	/	/	/	/
	+ ELGAR (Ours)	**0.0355**	0.0551	**0.0228**	0.0292	0.0111	0.0140	**0.0119**	**0.0015**
Sports	TIGER	0.0208	0.0338	0.0128	0.0169	0.0003	0.0005	0.1938	0.1907
	+ IFairLRS-RW	**0.0212**	**0.0349**	**0.0136**	**0.0180**	0.0006	0.0011	0.1872	0.1825
	+ IFairLRS-RR	0.0201	0.0342	0.0125	0.0170	**0.0036**	**0.0038**	**0.0975**	0.1421
	+ CARE	0.0131	0.0268	0.0076	0.0120	0.0004	0.0008	0.1741	0.1690
	+ ELGAR (Ours)	0.0202	0.0324	0.0131	0.0170	0.0009	0.0016	0.1249	**0.1059**
	LETTER	0.0203	**0.0332**	**0.0130**	**0.0171**	0.0007	0.0016	0.1875	0.1831
	+ IFairLRS-RW	0.0199	0.0316	0.0125	0.0163	0.0011	0.0015	0.1911	0.1876
	+ IFairLRS-RR	0.0194	0.0317	0.0121	0.0161	0.0022	0.0027	0.1373	0.1618
	+ CARE	0.0108	0.0231	0.0064	0.0103	0.0008	0.0020	0.1368	0.1389
	+ ELGAR (Ours)	**0.0206**	**0.0332**	**0.0130**	**0.0171**	**0.0024**	**0.0031**	**0.0865**	**0.0765**
	Ghost	0.0205	0.0331	0.0133	0.0173	0.0022	0.0026	0.1079	0.1104
	MiniOneRec-SFT	**0.0260**	**0.0406**	0.0167	**0.0214**	0.0015	0.0021	0.1761	0.1716
	+ IFairLRS-RW	0.0254	0.0397	0.0163	0.0209	0.0017	0.0025	0.1719	0.1678
	+ IFairLRS-RR	0.0252	0.0400	0.0161	0.0208	**0.0034**	**0.0045**	0.1038	0.1177
	+ CARE	/	/	/	/	/	/	/	/
	+ ELGAR (Ours)	0.0258	0.0390	**0.0168**	0.0204	0.0029	0.0038	**0.0871**	**0.0852**
KuaiRec*	TIGER	0.0718	0.1119	0.0448	0.0575	0.0075	0.0113	0.1389	0.1220
	+ IFairLRS-RW	0.0588	0.1138	0.0404	0.0577	0.0008	0.0140	0.1368	0.0999
	+ IFairLRS-RR	0.0662	0.1129	0.0424	0.0575	**0.0177**	**0.0329**	0.1074	0.0478
	+ CARE	0.0634	0.1241	0.0393	0.0587	0.0138	0.0185	**0.0339**	**0.0098**
	+ ELGAR (Ours)	**0.0746**	**0.1278**	**0.0485**	**0.0655**	0.0126	0.0240	0.0532	0.0157
	LETTER	0.0858	0.1287	0.0546	0.0685	0.0008	0.0033	0.1782	0.1589
	+ IFairLRS-RW	0.0793	0.1241	0.0513	0.0655	0.0019	0.0069	0.1543	0.1353
	+ IFairLRS-RR	0.0821	0.1250	0.0516	0.0653	0.0072	0.0137	**0.0469**	0.0381
	+ CARE	0.0382	0.1138	0.0224	0.0462	0.0114	**0.0236**	0.0583	**0.0033**
	+ ELGAR (Ours)	**0.0877**	**0.1325**	**0.0578**	**0.0721**	**0.0124**	0.0233	0.0849	0.0206
	Ghost	0.0718	0.1147	0.0503	0.0643	0.0150	0.0164	0.0460	0.0443
	MiniOneRec-SFT	0.1026	**0.1735**	0.0666	0.0900	0.0164	0.0357	0.0415	0.0176
	+ IFairLRS-RW	0.1073	0.1651	0.0660	0.0851	0.0156	0.0310	**0.0232**	**0.0049**
	+ IFairLRS-RR	0.1026	0.1651	0.0589	0.0788	**0.0297**	0.0446	0.1608	0.1531
	+ CARE	/	/	/	/	/	/	/	/
	+ ELGAR (Ours)	**0.1147**	0.1707	**0.0763**	**0.0945**	0.0291	**0.0449**	0.0478	0.0050

training-stage sample reweighting, and **IFairLRS-RR**, which further applies post-learning reranking on top of **IFairLRS-RW**. CARE (Lin et al., 2026) is evaluated with TIGER and LETTER but is not directly compatible with the MiniOneRec pipeline, for which the corresponding entries are marked "/". We additionally include Ghost (Yin et al., 2026) in the MiniOneRec block as a generative fairness baseline. Adaptation details are given in Appendix C.2. We also compare ELGAR with a rule-based controller that samples groups from $\boldsymbol{\pi}$ at the first decoding step, using TIGER trained with the group token. Results support the value of **request-aware routing** beyond exposure matching, with detailed analysis in Appendix I.

Metrics. We report HR and NDCG for overall recommendation utility, Tail NDCG for ranking quality conditional on Tail-target interactions, and MGU for group-level exposure misalignment, where lower MGU indicates better alignment with the training-interaction reference (He et al., 2017; Jiang et al., 2024). All metrics are evaluated at $K \in \{5, 10\}$. The main setting uses the same two-group partition for routing and evaluation ($G_{\text{route}} = G_{\text{eval}} = 2$). Formal metric definitions and the stateful evaluation protocol are provided in Appendices C.3 and C.4.

4.2 Main Results

Table 1 compares ELGAR with the vanilla backbones and fairness baselines on four datasets. For TIGER and LETTER, ELGAR consistently reduces MGU and improves Tail NDCG while largely preserving HR and NDCG. On Beauty with TIGER, for example, MGU@10 falls from 0.1827 to 0.0703 and Tail NDCG@10 rises from 0.0018 to 0.0054, while NDCG changes only from 0.0303 to 0.0298. At $K = 10$, ELGAR also improves Tail NDCG and exposure alignment over MiniOneRec on all four datasets, and improves NDCG over MiniOneRec-SFT on KuaiRec*. Compared with IFairLRS-RW, ELGAR typically achieves lower MGU and higher Tail NDCG with comparable utility. IFairLRS-RR attains higher Tail NDCG in several settings, while ELGAR achieves lower MGU in most comparisons. CARE reaches lower MGU in some KuaiRec* settings at a larger utility cost. Against Ghost in the MiniOneRec block, ELGAR achieves lower MGU@10 and higher Tail NDCG@10 on all four datasets. An additional analysis on Amazon Beauty finds that ELGAR's output popularity distribution is generally slightly closer to the test-target distribution than Ghost's (Appendix G).

Table 2: Ablation study of ELGAR with TIGER on Amazon Beauty, Sports, and Toys. ↑/↓ indicates that higher/lower is better.

Dataset	Setting	$K = 5$			$K = 10$		
		NDCG ↑	MGU ↓	Tail NDCG ↑	NDCG ↑	MGU ↓	Tail NDCG ↑
Beauty	ELGAR (Full)	0.0234	0.0962	0.0037	0.0298	0.0703	0.0054
	w/o $\mathcal{L}_{\text{fuse}}$ ($\lambda_{\text{fuse}} = 0$)	0.0228	0.0418	0.0044	0.0288	0.0019	0.0064
	w/o $\mathcal{L}_{\text{fair}}$ ($\lambda_{\text{reg}} = 0$)	0.0228	0.1084	0.0024	0.0290	0.0948	0.0033
	w/o Stateful Decoding ($\lambda_{\text{dec}} = 0$)	0.0236	0.1377	0.0028	0.0299	0.1207	0.0042
Sports	ELGAR (Full)	0.0131	0.1249	0.0009	0.0170	0.1059	0.0016
	w/o $\mathcal{L}_{\text{fuse}}$ ($\lambda_{\text{fuse}} = 0$)	0.0123	0.0408	0.0017	0.0165	0.0026	0.0032
	w/o $\mathcal{L}_{\text{fair}}$ ($\lambda_{\text{reg}} = 0$)	0.0128	0.1664	0.0007	0.0172	0.1592	0.0013
	w/o Stateful Decoding ($\lambda_{\text{dec}} = 0$)	0.0136	0.1766	0.0007	0.0181	0.1689	0.0013
Toys	ELGAR (Full)	0.0209	0.0455	0.0097	0.0263	0.0300	0.0129
	w/o $\mathcal{L}_{\text{fuse}}$ ($\lambda_{\text{fuse}} = 0$)	0.0191	0.0305	0.0061	0.0247	0.0031	0.0088
	w/o $\mathcal{L}_{\text{fair}}$ ($\lambda_{\text{reg}} = 0$)	0.0200	0.1025	0.0067	0.0255	0.0950	0.0090
	w/o Stateful Decoding ($\lambda_{\text{dec}} = 0$)	0.0212	0.1113	0.0079	0.0266	0.1009	0.0105

4.3 Ablation Study

We study the three key components of ELGAR: fused-gate supervision $\mathcal{L}_{\text{fuse}}$, fairness regularization $\mathcal{L}_{\text{fair}}$, and inference-time Stateful Decoding, evaluated under the optimally tuned parameter configuration adopted by the main experiments. Table 2 shows that removing either $\mathcal{L}_{\text{fair}}$ or Stateful Decoding substantially increases MGU and reduces Tail NDCG, confirming that they provide the main training- and inference-time fairness signals. Removing $\mathcal{L}_{\text{fuse}}$ further reduces MGU@10 on all three datasets, but consistently decreases NDCG. We retain it to **limit utility degradation**, while omitting it yields **stronger fairness**. The three components therefore play complementary roles: $\mathcal{L}_{\text{fair}}$ **learns when correction is beneficial, Stateful Decoding applies the learned correction at inference, and** $\mathcal{L}_{\text{fuse}}$ **prevents the correction from overwhelming relevance**.

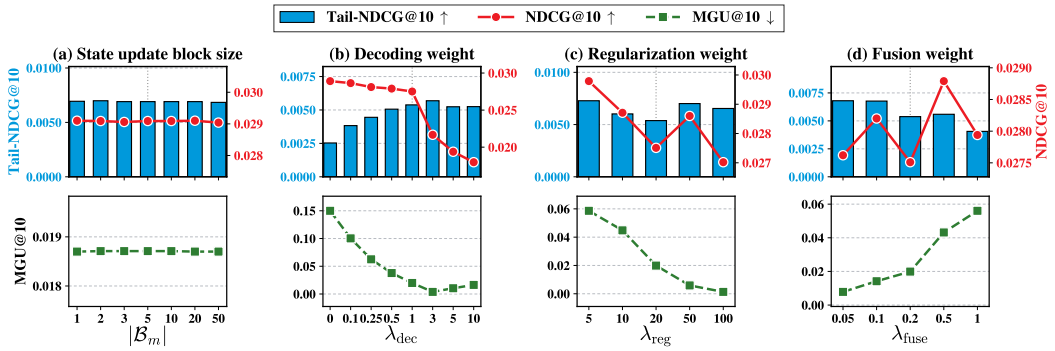

Figure 3: Parameter sensitivity of ELGAR on TIGER with Amazon Beauty under $G_{\text{route}} = G_{\text{eval}} = 2$. Panels (a)–(d) vary the ledger update block size $|\mathcal{B}_m|$, decoding weight λ_{dec}, fairness-regularization weight λ_{reg}, and fusion-supervision weight λ_{fuse}, respectively. Bars report Tail-NDCG@10, while the red and green curves report NDCG@10 and MGU@10.

4.4 Parameter Sensitivity and Robustness

We analyze the ledger update block size $|\mathcal{B}_m|$ and the three method-specific weights λ_{dec}, λ_{reg}, and λ_{fuse} on TIGER with Amazon Beauty, where one parameter is varied at a time based on the fixed parameter setting in in C.4 without tuning. Figure 3 reports Tail-NDCG@10, NDCG@10, and MGU@10 under the main two-group setting. ELGAR is largely insensitive to the ledger update block size over the tested range: Tail NDCG, NDCG, and MGU remain nearly unchanged from $|\mathcal{B}_m| = 1$ to 50, indicating that the ledger need not be refreshed after every request. Increasing λ_{dec} sharply reduces MGU and improves Tail NDCG up to a moderate intervention strength, while overly large values cause a clear drop in overall NDCG and slightly weaken the best alignment point. Increasing λ_{reg} steadily strengthens exposure alignment, whereas its effect on utility is comparatively

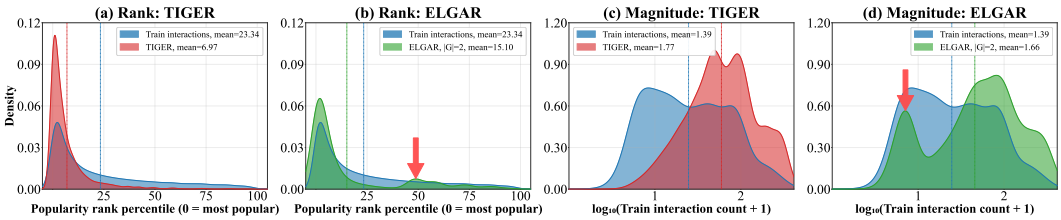

Figure 4: Exposure-distribution alignment of ELGAR with TIGER on Amazon Beauty under $G_{\text{route}} = 2$. Panels (a)–(b) show popularity-rank space, where 0 denotes the most popular items; Panels (c)–(d) show popularity-magnitude space using $\log_{10}(\text{train interaction count} + 1)$.

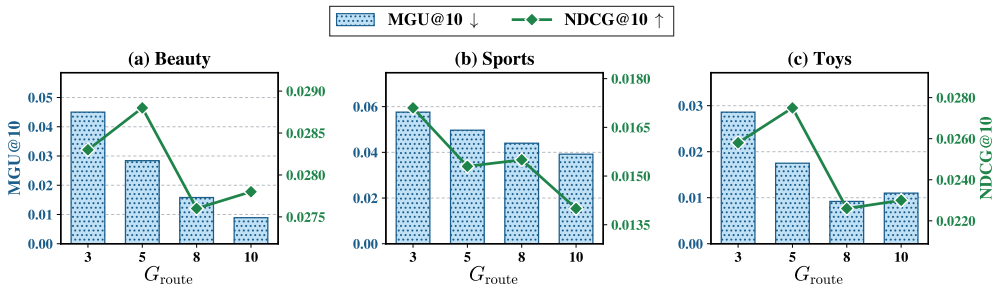

Figure 5: Effect of routing granularity $G_{\text{route}} \in \{3, 5, 8, 10\}$ on exposure-distribution alignment and recommendation utility for ELGAR with TIGER. Bars show MGU@10 and lines show NDCG@10.

moderate and non-monotonic. In contrast, larger λ_{fuse} increases MGU while NDCG stays within a relatively narrow range, consistent with fused supervision acting as a counterweight to fairness correction rather than as a direct alignment objective. Together, the results show that λ_{dec} and λ_{reg} primarily control correction strength, whereas λ_{fuse} regulates the utility–fairness balance.

4.5 Exposure Distribution Alignment Analysis

Figure 4 compares Vanilla TIGER and ELGAR with the training-interaction reference on Amazon Beauty in both popularity-rank and popularity-magnitude spaces. ELGAR reduces the concentration of exposure in the most popular region and shifts more exposure toward less-popular items, bringing the resulting distribution closer to the training-interaction reference in both views. We further vary the routing granularity as $G_{\text{route}} \in \{3, 5, 8, 10\}$ and evaluate all settings on a fixed ten-group partition ($G_{\text{eval}} = 10$) rather than the binary partition in the main experiments to capture finer-grained exposure differences that would otherwise be obscured by head–tail aggregation. As shown in Figure 5, finer routing generally reduces MGU@10, while the utility response becomes less favorable as granularity increases: NDCG@10 degrades progressively under the finest partitions. Thus, increasing G_{route} provides more granular control over exposure alignment, but the additional fairness gain is not uniformly cost-free. The group-construction rule and additional distribution visualizations are provided in Appendices C.4 and E.

5 Conclusion

We presented ELGAR, a model-agnostic framework for improving popularity-calibrated item-side fairness in SID-based GR. ELGAR reformulates generation as exposure-aware group routing followed by within-group semantic decoding. By combining a popularity-group token, a ledger-conditioned fairness head, and an Exposure Ledger, ELGAR adapts routing decisions to cumulative exposure disparities before candidate pruning. Through extensive experiments on four datasets and three representative GR backbones, we demonstrate that ELGAR effectively improves exposure fairness and tail-item recommendation while maintaining competitive overall recommendation utility. Future work will extend ELGAR beyond predefined popularity groups and fixed target distributions to support personalized, multi-attribute, and dynamically evolving fairness objectives in real-world recommendation environments.

REFERENCES

Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Managing popularity bias in recommender systems with personalized re-ranking. *arXiv preprint arXiv:1901.07555*, 2019.

Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. Addressing the multistakeholder impact of popularity bias in recommendation through calibration. *arXiv preprint arXiv:2007.12230*, 2020.

Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*, pp. 405–414, 2018.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78 (5):1103–1130, 2016.

Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 224–232, 2018.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

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

Zezhong Fan, Ziheng Chen, Luyi Ma, Jin Huang, Lalitesh Morishetti, Kaushiki Nag, Sushant Kumar, and Kannan Achan. Crab: Codebook rebalancing for bias mitigation in generative recommendation. *arXiv preprint arXiv:2604.05113*, 2026.

Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 540–550, 2022. doi: 10.1145/3511808.3557220.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pp. 299–315, 2022.

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*, pp. 507–517, 2016.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017.

Meng Jiang, Keqin Bao, Jizhi Zhang, Wenjie Wang, Zhengyi Yang, Fuli Feng, and Xiangnan He. Item-side fairness of large language model-based recommendation system. In *Proceedings of the ACM web conference 2024*, pp. 4717–4726, 2024.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, 2018. doi: 10.1109/ICDM.2018.00035.

Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Minionerec: An open-source framework for scaling generative recommendation, 2025.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11523–11532, 2022. doi: 10.1109/CVPR52688.2022.01123.

Xinyu Lin, Pengyuan Liu, Wenjie Wang, Yicheng Hu, Chen Xu, Fuli Feng, Qifan Wang, and Tat-Seng Chua. Bringing reasoning to generative recommendation through the lens of cascaded ranking. In *Proceedings of the ACM Web Conference 2026*, pp. 8939–8949, 2026.

Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, Changqing Qiu, Jiaqi Zhang, Xu Zhang, Zhiheng Yan, Jingming Zhang, Simin Zhang, Mingxing Wen, Zhaojie Liu, Kun Gai, and Guorui Zhou. QARM: Quantitative alignment multi-modal recommendation at kuaishou. *arXiv preprint arXiv:2411.11739*, 2024.

Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 429–438, 2020.

Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of the web conference 2020*, pp. 1194–1204, 2020.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. volume 36, pp. 10299–10315, 2023.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2219–2228, 2018.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6306–6315, 2017.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024a.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, et al. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3245–3254, 2024b.

Haolun Wu, Bhaskar Mitra, Chen Ma, Fernando Diaz, and Xue Liu. Joint multisided exposure fairness for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, pp. 703–714, 2022.

Chen Xu, Sirui Chen, Jun Xu, Weiran Shen, Xiao Zhang, Gang Wang, and Zhenhua Dong. P-mmf: Provider max-min fairness re-ranking in recommender system. In *Proceedings of the ACM Web Conference 2023*, pp. 3701–3711, 2023.

Chen Xu, Xiaopeng Ye, Wenjie Wang, Liang Pang, Jun Xu, and Tat-Seng Chua. A taxation perspective for fair re-ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1494–1503, 2024.

Jun Yin, Bangguo Zhu, Peng Huo, Ruochen Liu, Hao Chen, Senzhang Wang, Shirui Pan, and Chengqi Zhang. Echoes in filter bubble: Diagnosing and curing popularity bias in generative recommenders. *arXiv preprint arXiv:2605.16825*, 2026.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1435–1448. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00118.

Jieming Zhu, Mengqun Jin, Qijiong Liu, Zexuan Qiu, Zhenhua Dong, and Xiu Li. Cost: Contrastive quantization based semantic tokenization for generative recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 969–974, 2024.

A NOTATION

This appendix summarizes the notation used throughout the paper. We distinguish training-time request indices j, serving-time request indices t, and inference update-block indices m to make the exposure-update process explicit.

Recommendation and SIDs.

Symbol	Definition	Introduced in
$\mathcal{U}, \mathcal{I}$	Sets of users and items.	Sec. 2.1
$\mathbf{S}_u$	Chronologically ordered interaction sequence of user u.	Sec. 2.1
T_u	Number of observed interactions of user u.	Sec. 2.1
$\mathbf{x}_{u,\tau}$	Recommendation context preceding interaction position τ; $\mathbf{x}_j$ and $\mathbf{x}_t$ denote the corresponding contexts for training sample j and serving request t.	Sec. 2.1
$\mathcal{R}_t$	Ordered Top-K recommendation list returned for request t.	Sec. 2.2
K	Recommendation-list cutoff used for exposure aggregation or evaluation.	Sec. 2.1
$\mathbf{v}_i$	Continuous representation of item i before SID discretization.	Sec. 2.1
$c_i^{(\ell)}$	Discrete code index of item i at SID level ℓ.	Eq. 1
L_i	Length of the SID of item i.	Sec. 2.1
$\mathrm{SID}(i)$	Original SID $[c_i^{(1)}, \ldots, c_i^{(L_i)}]$ of item i.	Sec. 2.1
$\widetilde{\mathrm{SID}}(i)$	Group-aware SID $[\gamma_{g(i)}, c_i^{(1)}, \ldots, c_i^{(L_i)}]$ used by ELGAR.	Eq. 5
θ	Parameters of the GR backbone.	Sec. 3.1
$\mathcal{D}$	Training set of context–target pairs $(\mathbf{x}, i^*)$.	Sec. 2.1

Popularity Groups and Exposure Ledger.

Symbol	Definition	Introduced in
$\mathcal{G} = \{1, \ldots, G\}$	Set of G mutually exclusive popularity groups used for routing.	Sec. 2.2
$g(i)$	Popularity-group assignment of item i.	Sec. 2.2
f_i	Interaction count of item i in the held-2-out training split.	Sec. 2.2
π_g	Reference interaction proportion of group g computed from the training interactions.	Eq. 2
$\boldsymbol{\pi}$	Reference distribution $(\pi_1, \ldots, \pi_G)$.	Eq. 2
$\mathbf{n}$	Raw group-exposure count vector.	Sec. 2.2
$\rho_g(\mathbf{n})$	Normalized exposure proportion of group g induced by count vector $\mathbf{n}$.	Eq. 3
$\boldsymbol{\rho}(\mathbf{n})$	Normalized exposure distribution $(\rho_1(\mathbf{n}), \ldots, \rho_G(\mathbf{n}))$.	Eq. 3
$D(\mathbf{n}; \boldsymbol{\pi})$	ℓ_1 discrepancy between the realized exposure distribution and reference distribution $\boldsymbol{\pi}$.	Eq. 4
γ_g	Dedicated routing token associated with popularity group g.	Sec. 3.1
Γ	Group-token set $\{\gamma_1, \ldots, \gamma_G\}$.	Sec. 3.1
$\boldsymbol{\delta}_g$	G-dimensional one-hot vector corresponding to group g.	Eq. 8
H	Exposure horizon, measured in item-exposure units, used to normalize the progress component of the Exposure Ledger;	Eq. 7

Training-Time Routing.

Symbol	Definition	Introduced in
$\mathcal{E}$	Shuffled mini-batch treated as a short training exposure episode.	Sec. 3.2
B	Number of training instances in one mini-batch episode.	Sec. 3.2
j	Training-request index within a mini-batch episode.	Sec. 3.2
i_j^*	Ground-truth target item for training request j.	Sec. 3.2
$g_j = g(i_j^*)$	Ground-truth popularity group of training target i_j^*.	Sec. 3.2
$\mathbf{n}_j$	Raw exposure counts accumulated from requests preceding j in the current training episode.	Sec. 3.2
$\mathbf{L}_j$	Exposure Ledger for request j, containing group-wise exposure deviations and normalized exposure progress.	Eq. 7
$r_j(g)$	One-step fairness gain obtained by hypothetically assigning the next exposure to group g.	Eq. 8
$\mathbf{r}_j$	One-step fairness-gain vector $[r_j(1), \ldots, r_j(G)]$.	Sec. 3.2
$\mathbf{h}_j^G$	Decoder hidden representation used to predict the first, group-routing token.	Sec. 3.3
$\mathbf{z}_j^G$	Backbone generative logits over the G group tokens.	Sec. 3.3
W_{gen}	Original generative output head of the backbone.	Sec. 3.3
W_f	Fairness head producing ledger-conditioned group-level logits.	Sec. 3.3
$f_\phi(\cdot)$	Adapter mapping the Exposure Ledger from $\mathbb{R}^{G+1}$ to the decoder hidden space.	Eq. 10
$\mathbf{z}_j^{\text{fair}}$	Ledger-conditioned fairness logits produced for training request j.	Eq. 10
$\mathbf{q}_j$	Fused ledger-conditioned group-routing distribution for request j.	Eq. 10
$\mathcal{L}_{\text{gen}}$	Autoregressive generation loss over the group-aware SID.	Eq. 9
$\mathcal{L}_{\text{fuse}}$	Group-level supervision preserving target-group relevance after relevance–fairness fusion.	Sec. 3.3
$\mathcal{L}_{\text{fair}}$	Fairness regularization based on expected one-step fairness gain.	Eq. 12
λ_{fuse}	Weight of the fused-routing supervision objective.	Eq. 13
λ_{reg}	Weight of the fairness-regularization objective.	Eq. 13

Serving-Time Stateful Decoding.

Symbol	Definition	Introduced in
t	Serving-request index.	Sec. 2.2
m	Index of an inference-time ledger update block.	Sec. 3.4
$\mathcal{B}_m$	Set of consecutive serving requests decoded using the same Exposure Ledger in block m.	Sec. 3.4
$\lvert\mathcal{B}_m\rvert$	Ledger update-block size, controlling how frequently serving-time exposure is refreshed.	Sec. 3.4
$\mathbf{n}^{(m)}$	Cumulative realized Top-K group-exposure counts before decoding block m.	Sec. 3.4
$\mathbf{L}^{(m)}$	Exposure Ledger constructed from $\mathbf{n}^{(m)}$ and shared by all requests in $\mathcal{B}_m$.	Sec. 3.4
$\mathbf{z}_t^G$	Backbone group-token logits for serving request t.	Sec. 3.4
$\mathbf{z}_t^{\text{fair},(m)}$	Fairness logits for request t conditioned on ledger $\mathbf{L}^{(m)}$.	Sec. 3.4
$\mathbf{z}_t^{\text{dec},(m)}$	Final group-routing logits used by Stateful Decoding in block m.	Eq. 14
λ_{dec}	Inference-time weight controlling the strength of the ledger-guided fairness intervention.	Eq. 14

Continued on next page

Notation (continued)

Symbol	Definition	Introduced in
G_{route}	Number of popularity groups used by ELGAR for group routing.	Sec. 4.1
G_{eval}	Number of groups used to evaluate exposure alignment; it may differ from G_{route} in fine-grained analyses.	Sec. 4.1
K_{ledger}	Top-K cutoff whose displayed recommendations are accumulated to update the serving-time Exposure Ledger.	Sec. C.4

B Related Work

B.1 GR.

GR shifts item prediction from candidate scoring to identifier generation. Early work such as P5 (Geng et al., 2022) explores a unified text-to-text formulation. TIGER (Rajput et al., 2023) develops a semantic-token approach, using residual quantization to represent items as SIDs. Subsequent studies strengthen the connection between semantic representations and user behavior. LETTER (Wang et al., 2024a) and CoST (Zhu et al., 2024) incorporate collaborative and relational signals into tokenization, while EAGER (Wang et al., 2024b) combines behavioral and semantic generation. More recent work expands the scope and scale of this paradigm. OneRec (Deng et al., 2025) unifies retrieval and ranking with preference alignment, while MiniOneRec (Kong et al., 2025) supports LLM-based recommendation through supervised fine-tuning and reinforcement learning. Alongside these advances in preference modeling, ELGAR introduces exposure-adaptive group allocation within SID generation.

B.2 Fairness in Recommendation.

Item-side fairness concerns how recommendation opportunities are distributed across items. Popularity bias can concentrate these opportunities on head items, while feedback loops further reinforce this imbalance (Abdollahpouri et al., 2019; Chaney et al., 2018). In conventional recommendation, fairness is commonly addressed through ranking and exposure allocation. Singh and Joachims (Singh & Joachims, 2018) impose exposure constraints, while Biega et al. (Biega et al., 2018) consider fairness accumulated across rankings. FairRec (Patro et al., 2020) and Wu et al. (Wu et al., 2022) further consider both consumers and producers. Moving toward online allocation, FairCo (Morik et al., 2020) and P-MMF (Xu et al., 2023) dynamically regulate exposure across requests. Tax-rank (Xu et al., 2024) improves control over the utility–fairness trade-off. GR introduces additional intervention points in tokenization and decoding. IFairLRS (Jiang et al., 2024) adapts reweighting and reranking to LLM-based recommendation. CARE (Lin et al., 2026) modifies history encoding and reasoning during generation, while Ghost (Yin et al., 2026) and CRAB (Fan et al., 2026) address bias through tokenization and training objectives. ELGAR brings dynamic exposure control into this generative process. Its Exposure Ledger conditions group routing on cumulative exposure before item-specific semantic pruning.

C Experimental Details

C.1 Datasets and Preprocessing

We use the Amazon Beauty, Toys and Games, and Sports and Outdoors 5-core datasets (He & McAuley, 2016), together with KuaiRec* derived from the KuaiRec short-video interaction logs (Gao et al., 2022). Amazon records product-review interactions, whereas KuaiRec records user–video interactions. All interaction sequences are ordered chronologically, and popularity statistics are computed from the raw training split only, before rolling-prefix expansion into context–target instances.

Data construction. The sequential recommendation split follows the held-2-out setting used in the main formulation: the final interactions are reserved for validation and test, while preceding interactions form the training history. Training context–target pairs are then constructed by rolling-prefix expansion from the training portion. Popularity groups, item frequencies, and the reference distribution $\boldsymbol{\pi}$ are computed before this expansion.

C.2 Baselines and Adaptation Details

Backbones. TIGER (Rajput et al., 2023), LETTER (Wang et al., 2024a), and MiniOneRec (Kong et al., 2025) are used as the three base SID generative recommenders. The vanilla rows use the corresponding back-

Table 7: Effect of adapting IFairLRS reweighting to TIGER on four datasets. Higher HR, NDCG, and Tail NDCG are better; lower MGU is better. Bold values indicate the lowest MGU within each dataset.

Dataset	Method	HR ↑		NDCG ↑		Tail NDCG ↑		MGU ↓	
		@5	@10	@5	@10	@5	@10	@5	@10
Beauty	TIGER	0.0361	0.0559	0.0240	0.0303	0.0009	0.0018	0.1883	0.1827
	+ IFairLRS-RW (Native)	0.0370	0.0557	0.0246	0.0306	0.0007	0.0012	0.1944	0.1909
	+ IFairLRS-RW (Adapt)	0.0352	0.0549	0.0231	0.0295	0.0016	0.0024	0.1813	0.1736
Sports	TIGER	0.0208	0.0338	0.0128	0.0169	0.0003	0.0005	0.1938	0.1907
	+ IFairLRS-RW (Native)	0.0208	0.0341	0.0132	0.0175	0.0001	0.0003	0.1967	0.1947
	+ IFairLRS-RW (Adapt)	0.0212	0.0349	0.0136	0.0180	0.0006	0.0011	0.1872	0.1825
Toys	TIGER	0.0317	0.0515	0.0209	0.0272	0.0056	0.0082	0.1516	0.1417
	+ IFairLRS-RW (Native)	0.0324	0.0516	0.0212	0.0270	0.0025	0.0048	0.1673	0.1548
	+ IFairLRS-RW (Adapt)	0.0283	0.0485	0.0184	0.0249	0.0035	0.0064	0.1589	0.1475
KuaiRec*	TIGER	0.0718	0.1119	0.0448	0.0575	0.0075	0.0113	0.1389	0.1220
	+ IFairLRS-RW (Native)	0.0569	0.0886	0.0341	0.0446	0.0000	0.0000	0.1997	0.1998
	+ IFairLRS-RW (Adapt)	0.0588	0.1138	0.0404	0.0577	0.0008	0.0140	0.1368	0.0999

bone without any intervention for popularity-calibrated item-side fairness; MiniOneRec-SFT denotes the supervised fine-tuning stage used as the vanilla MiniOneRec reference.

IFairLRS-RW. IFairLRS-RW follows the reweighting principle of IFairLRS (Jiang et al., 2024), i.e., down-weighting samples from over-exposed groups and upweighting those from under-exposed groups. A direct application of the original weighting statistics, however, is unsuitable for GR with rolling-prefix training. In the original setting, each user contributes one training instance, whereas GR models expand each interaction sequence into multiple prefix–target pairs. This expansion changes the effective group distribution of the training instances and can substantially amplify the proportion of popular-item interactions. Consequently, directly applying the original ratio can produce weights inconsistent with the desired correction direction.

We therefore retain the original reweighting principle while computing the group-wise correction from the rolling-prefix-expanded training data. Specifically, let p_g^{tar} denote the group proportion among target items and p_g^{train} the corresponding effective group proportion induced by the expanded training instances. We use

$$w_g = \frac{p_g^{\mathrm{tar}}}{p_g^{\mathrm{train}}},$$

such that groups over-represented after rolling-prefix expansion receive $w_g < 1$, whereas under-represented groups receive $w_g > 1$. The weight associated with the target group is then applied to the training loss of each instance. This adaptation preserves the correction direction of IFairLRS while accounting for the training-instance distribution induced by GR. We additionally compare this adaptation with the direct original definition in Table 7. The results indicate that the native IFairLRS deteriorates Tail NDCG and MGU, thereby deviating from its originally intended objective. Our adaptation, however, retains the original goal of IFairLRS.

IFairLRS-RR. IFairLRS-RR is implemented as the post-learning reranking stage of IFairLRS on top of the corresponding IFairLRS-RW checkpoint. No additional model training is performed during this stage. Following the original reranking formulation, group-level ranking penalties are computed with the predefined reference distribution $\boldsymbol{\pi}$ as the calibration target and are applied only to the generated candidates at inference time. Thus, IFairLRS-RR uses a fixed post-hoc correction, whereas ELGAR updates its intervention according to accumulated realized exposure during serving.

Following the original formulation, group unfairness at different cutoffs is aggregated into a normalized punishment term, and the reranking score is adjusted as $\widetilde{D}_i = D_i/(1-\widehat{U}_i)^{\alpha}$, where α controls the strength of the fairness-aware punishment. We tune $\alpha \in \{0, 0.01, \ldots, 0.10\}$, consistent with the search range used in the original IFairLRS paper. For model selection, we first retain checkpoints whose validation NDCG is at least 95% of that of the corresponding vanilla backbone and select the checkpoint with the lowest validation MGU among them. If no checkpoint satisfies this utility constraint, we select the checkpoint with the highest validation NDCG. This criterion prevents fairness improvements from being obtained at the cost of excessive relevance degradation.

CARE. CARE (Lin et al., 2026) is evaluated with TIGER and LETTER using the corresponding generative backbones. The current CARE formulation/implementation is not directly compatible with the MiniOneRec training and generation pipeline; therefore, MiniOneRec+CARE is retained as an explicit "/" placeholder in Table 1 rather than being treated as a failed run.

Ghost. Ghost (Yin et al., 2026) is included in the MiniOneRec block as an additional generative fairness baseline. It is reported as a standalone method because its fairness intervention includes its own tokenization/optimization design rather than an ELGAR-style plug-in attached to MiniOneRec-SFT.

ELGAR. ELGAR is attached to each backbone through the group-routing interface described in Section 3.3; the backbone-specific SID representation and within-group autoregressive decoding are otherwise preserved. Unless stated otherwise, all comparisons within a dataset–backbone block use matched data splits and evaluation cutoffs.

C.3 Evaluation Metrics

Let $\mathcal{Q}$ be the test-request set and let rank_t denote the rank of the ground-truth target item for request t. With one relevant target per request,

$$\mathrm{HR@}K = \frac{1}{|\mathcal{Q}|} \sum_{t \in \mathcal{Q}} \mathbb{I}[\mathrm{rank}_t \le K], \tag{15}$$

and

$$\mathrm{NDCG@}K = \frac{1}{|\mathcal{Q}|} \sum_{t \in \mathcal{Q}} \frac{\mathbb{I}[\mathrm{rank}_t \le K]}{\log_2(\mathrm{rank}_t + 1)}. \tag{16}$$

Let

$$\mathcal{T}_{\mathrm{tail}} = \{t \in \mathcal{Q} : \, g(i_t^*) = g_{\mathrm{tail}}\}. \tag{17}$$

Tail NDCG is computed conditionally on Tail-target requests:

$$\text{Tail-NDCG@}K = \frac{1}{|\mathcal{T}_{\mathrm{tail}}|} \sum_{t \in \mathcal{T}_{\mathrm{tail}}} \frac{\mathbb{I}[\mathrm{rank}_t \le K]}{\log_2(\mathrm{rank}_t + 1)}. \tag{18}$$

Thus, Tail NDCG measures ranking quality conditional on Tail-target interactions rather than averaging Tail contributions over all test requests.

For exposure alignment, let G_{eval} be the cardinality of the evaluation partition and let ρ_h^{eval} and π_h^{eval} denote the realized recommendation-exposure and reference interaction proportions for evaluation group h. We report

$$\mathrm{MGU@}K = \frac{1}{G_{\mathrm{eval}}} \sum_{h=1}^{G_{\mathrm{eval}}} \left| \rho_h^{\mathrm{eval}} - \pi_h^{\mathrm{eval}} \right|, \tag{19}$$

where lower values indicate better exposure alignment. In the main experiments $G_{\mathrm{eval}} = G_{\mathrm{route}} = 2$; the fine-grained analysis fixes $G_{\mathrm{eval}} = 10$ while varying G_{route}.

C.4 Implementation Details

Backbones and optimization. TIGER and LETTER adopt the same T5 encoder–decoder backbone, with 4 encoder layers and 4 decoder layers, hidden dimension $d_{\mathrm{model}} = 128$, feed-forward dimension $d_{\mathrm{ff}} = 1024$, 6 attention heads, ReLU activation, and dropout rate 0.1. Both models are trained from scratch with Adam at a learning rate of 1×10^{-4} and an effective batch size of 256 for at most 200 epochs. MiniOneRec instead uses the pretrained Qwen2-1.5B decoder-only backbone (28 layers, hidden size 1536) and is fully fine-tuned in BF16. Its training consists of 4 epochs of semantic-alignment SFT followed by 8 epochs of recommendation SFT, both with AdamW and a learning rate of 1×10^{-4}. We use a per-device batch size of 16 with gradient accumulation of 4 under the standard two-GPU configuration, corresponding to an effective batch size of $16 \times 2 \times 4 = 128$. The maximum input length is 512 tokens, and each MiniOneRec training stage uses 20 warm-up optimization steps.

SIDs. TIGER uses the original RQ-VAE-based SIDs, consisting of three quantization levels with 256 codes per level followed by an additional collision-disambiguation code, resulting in a four-token SID. LETTER uses its separately trained four-level RQ-VAE codebook with 256 codes per level and code dimension 32, incorporating collaborative information and diversity regularization during codebook construction. We use the resulting frozen LETTER codebook in all recommendation experiments. MiniOneRec reuses a frozen base SID mapping and registers the SID tokens into the language-model tokenizer. It first performs semantic alignment between item SIDs and item titles and then conducts supervised recommendation fine-tuning from historical SID sequences to the next-item SID. Codebook-training hyperparameters are treated separately from the recommendation-model training configuration.

Checkpoint selection. For TIGER and LETTER, validation is performed every two epochs using beam-search decoding with the fairness intervention disabled, and the checkpoint with the highest validation NDCG@20 is selected. Training is stopped after 10 consecutive validation rounds without improvement, corresponding to approximately 20 epochs without progress. MiniOneRec follows a different selection protocol: the semantic-alignment stage uses its final checkpoint, whereas the recommendation SFT stage selects the checkpoint with the lowest validation `eval_loss`. Validation during MiniOneRec SFT is based on teacher-forced next-token prediction rather than autoregressive beam-search recommendation, and no additional early stopping is applied.

Inference and fairness-weight selection. Final recommendation for all backbones uses constrained beam search with beam size 20, restricting generated sequences to valid catalogue SIDs. We report ranking metrics at $K \in \{5, 10\}$. Standard inference uses batch size 256 for TIGER/LETTER and 16 for MiniOneRec, whereas stateful ELGAR decoding with a non-zero fairness intervention at Top-10 is executed with $|\mathcal{B}_m| = 1$ in main experiment so that the exposure ledger can be updated sequentially. We distinguish backbone checkpoint selection from the selection of the ELGAR decoding weight $\lambda_{\mathrm{dec}} \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$. In the main experiments, we fix $\lambda_{\mathrm{fuse}} = 0.2$ and $\lambda_{\mathrm{reg}} = 20$, while λ_{dec} is selected on the validation set by first retaining candidates satisfying

$$\mathrm{NDCG@10}(\lambda_{\mathrm{dec}}) \geq 0.95\ \mathrm{NDCG@10}(0),$$

and then choosing the candidate with the lowest MGU@10. The ablation study adopts the same tuned parameters as the main experiments and differs from the full model only in the removal of the corresponding component. In contrast, the hyperparameter analysis and the fine-grained distribution alignment study use a fixed configuration with $\lambda_{\mathrm{dec}} = 1$ without tuning.

Exposure-ledger configuration. We use a ledger horizon of $H = 20$ item exposures for all backbones and datasets. During training, the local exposure counts are initialized to zero for every shuffled mini-batch and advanced by one ground-truth target item at a time. During inference, the counts persist over the full serving episode and are updated from all displayed Top-K recommendations. The same ledger normalization in Eq. 7 is used in both cases. Because the progress coordinate is defined in item-exposure units, a Top-K serving update contributes K units rather than one request-level unit; the coordinate is clipped at 1 after H accumulated exposures.

Popularity-group construction. All popularity groups are constructed from the *raw training interactions* only, before rolling-prefix expansion into context–target training instances, and are kept fixed for validation and test. The item frequencies f_i, all group boundaries, and the reference distribution $\boldsymbol{\pi}$ are therefore unaffected by the multiplicity induced by rolling-prefix training. For the main and routing-granularity experiments, we set the cumulative interaction-mass boundary of the original Head region to $\eta = 0.8$. Alternative values of η are examined separately in Appendix F. For routing cardinality G_{route}, we allocate the subgroup budget as

$$G_T = \max(1, \mathrm{round}((1-\eta)G_{\mathrm{route}}))\,, \qquad G_H = G_{\mathrm{route}} - G_T, \tag{20}$$

where G_H and G_T are the numbers of routing groups assigned to the Head and Tail regions. Items are sorted in descending order of training frequency. The first $\eta = 80\%$ of cumulative interaction mass is partitioned into G_H contiguous strata of approximately equal interaction mass, and the remaining 20% into G_T analogous strata. The nominal interaction-mass proportion of each stratum is

$$\omega_g^{\mathrm{nom}} = \begin{cases} \eta/G_H, & g \text{ belongs to the Head region,} \\ (1-\eta)/G_T, & g \text{ belongs to the Tail region.} \end{cases} \tag{21}$$

For $G_{\mathrm{route}} \in \{2, 3, 5, 8, 10\}$, this gives $(G_H, G_T) \in \{(1,1), (2,1), (4,1), (6,2), (8,2)\}$. Because an item cannot be split across groups, each nominal boundary is placed at the item cutoff whose cumulative mass is closest to the target boundary, while enforcing at least one item per group. All downstream fairness quantities use the resulting *realized* group proportions. For cross-granularity comparison, the evaluation partition is fixed to $G_{\mathrm{eval}} = 10$ and constructed by the same rule.

D Supplementary Parameter Analysis

The hyperparameter study is conducted by varying one hyperparameter at a time based on a *preset default configuration* in C.4 with no tuning, whereas the main experiments and ablation study adopt a *tuned parameter combination* obtained via validation. Consequently, the results reported in the hyperparameter analysis are not directly comparable with those in the main experiments and ablation study, and numerical inconsistencies between the two settings are therefore expected.

The main paper reports parameter sensitivity on TIGER with Amazon Beauty. Figure 6 repeats those sweeps on the remaining datasets, using the same metrics and parameter ranges. The loss weights λ_{reg} and λ_{fuse} were swept on Beauty only and are not repeated here.

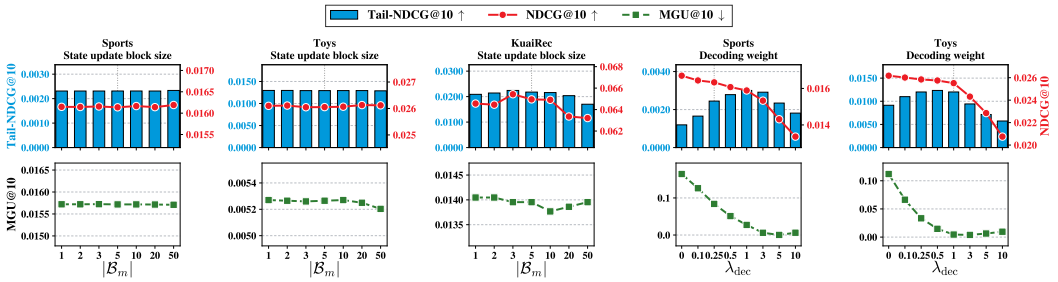

Figure 6: Parameter sensitivity on the datasets not covered by the main paper. The first three columns sweep the state update block size $|\mathcal{B}_m|$ on Sports, Toys and KuaiRec; the last two sweep the decoding weight λ_{dec} on Sports and Toys. Bars (left axis) are Tail-NDCG@10, the red line (right axis) is NDCG@10, and the bottom row is MGU@10; the dotted vertical line marks the default setting.

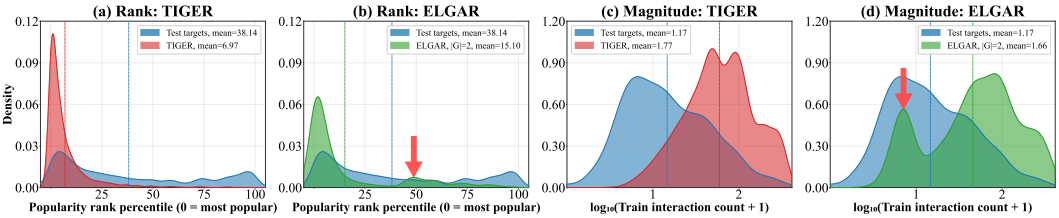

Figure 7: Exposure-distribution alignment with held-out test targets on Amazon Beauty under $G_{\text{route}} = 2$. The left panel shows popularity-rank alignment and the right panel shows popularity-magnitude alignment measured by $\log_{10}(\text{train interaction count} + 1)$. ELGAR shifts exposure toward the popularity profile of the held-out targets in both views.

All metrics are essentially invariant to the state update block size: over $|\mathcal{B}_m| \in \{1, \ldots, 50\}$, NDCG@10 and MGU@10 move by at most 3.4% of their own mean, so a larger block reduces the update frequency at no measurable cost. The decoding weight reproduces the behaviour observed on Beauty: MGU@10 falls by more than two orders of magnitude up to $\lambda_{\text{dec}} = 5$ (Sports) and 3 (Toys) and rises afterwards, Tail-NDCG@10 peaks at an intermediate weight, and NDCG@10 decreases monotonically by about 20% across the full range.

E Supplementary Fine-Grained Distribution Alignment

Fine-grained distribution alignment is evaluated based on a *preset default configuration* in C.4 with no tuning. Figures 8 and 9 compare Amazon Beauty Top-10 exposures with training interactions and held-out test targets, respectively. In each five-column grid, the first column shows TIGER and the remaining columns show ELGAR with $G_{\text{route}} \in \{3, 5, 8, 10\}$. Figure 7 provides the corresponding test-target comparison for the main two-group setting. Across the finer routing settings, ELGAR attenuates TIGER's exposure peak at the most popular ranks and shifts mass toward items with lower training popularity. The test-target views show the same directional improvement in alignment, while the detailed density shape varies with routing granularity. These plots complement the fixed-partition MGU@10 results in Figure 5 by locating the redistribution along the popularity spectrum.

F Alignment under Alternative Head–Tail Reference Configurations

The main experiments adopt the 80/20 interaction-mass split used in prior popularity-bias research (Abdollahpouri et al., 2019). This choice provides a common reference setting, while ELGAR's exposure ledger and routing objective are defined for a given group partition and its training-derived reference distribution. We therefore examine alternative two-group configurations on Amazon Beauty to study how the selected boundary affects exposure redistribution along the popularity spectrum.

Reference construction and experimental protocol. We keep $G_{\text{route}} = G_{\text{eval}} = 2$ and vary the nominal Head interaction-mass boundary over $\eta \in \{0.5, 0.6, 0.7, 0.9\}$, corresponding to 50/50, 60/40, 70/30,

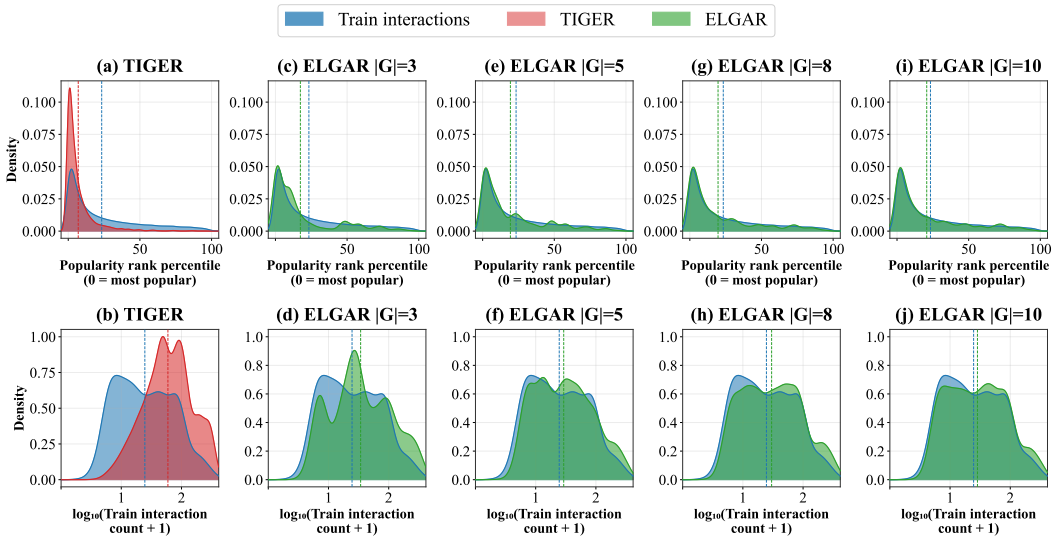

Figure 8: Top-10 exposure distributions on Amazon Beauty with training interactions as the reference. The first column compares TIGER with the reference; the remaining columns compare ELGAR at $G_{\text{route}} \in \{3, 5, 8, 10\}$. The top row shows popularity-rank percentile (0 denotes the most popular items), and the bottom row shows $\log_{10}(\text{train interaction count} + 1)$. Across the four routing granularities, ELGAR reduces TIGER's concentration at the most popular ranks.

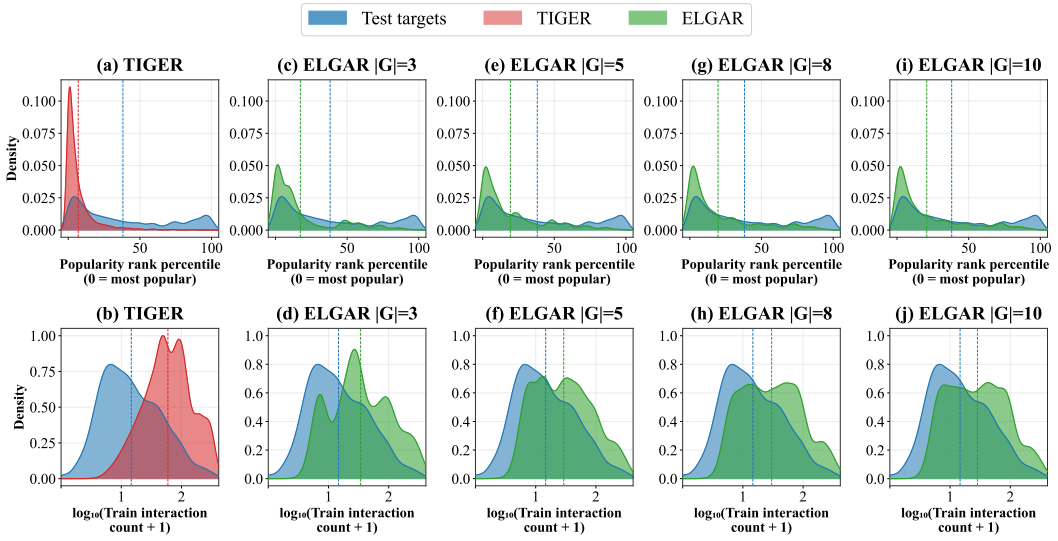

Figure 9: Top-10 exposure distributions on Amazon Beauty with held-out test targets as the reference. The first column shows TIGER; the remaining columns show ELGAR at $G_{\text{route}} \in \{3, 5, 8, 10\}$. The top row uses popularity-rank percentile and the bottom row uses $\log_{10}(\text{train interaction count} + 1)$. ELGAR weakens the extreme-head exposure peak and shifts exposure toward the test-target popularity profile across all four settings.

and 90/10 configurations. Items are sorted by their raw training-interaction counts. For each η, the Head group is the popularity-ranked prefix whose cumulative interaction mass is closest to η, and the remaining items form the Tail group. Because items are indivisible, the reference distribution $\boldsymbol{\pi}$ uses the realized interaction proportions, as defined in Eq. 2. For each configuration, we regenerate the item-group assignments and group-prefixed semantic IDs and independently retrain ELGAR. The four ELGAR runs use identical model and

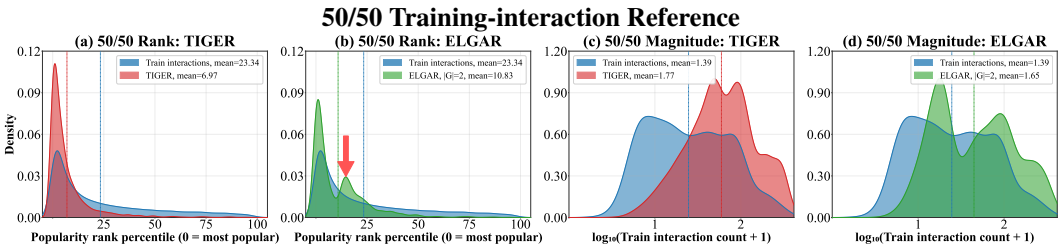

Figure 10: Top-10 exposure distributions on Amazon Beauty under the nominal 50/50 Head–Tail configuration with $G_{\text{route}} = 2$. In each row, panels (a)–(b) compare TIGER and ELGAR with the reference in popularity-rank space; panels (c)–(d) show the corresponding popularity-magnitude comparisons using $\log_{10}(\text{train interaction count} + 1)$. Blue denotes the reference, red TIGER, and green ELGAR. Dashed vertical lines mark distribution means.

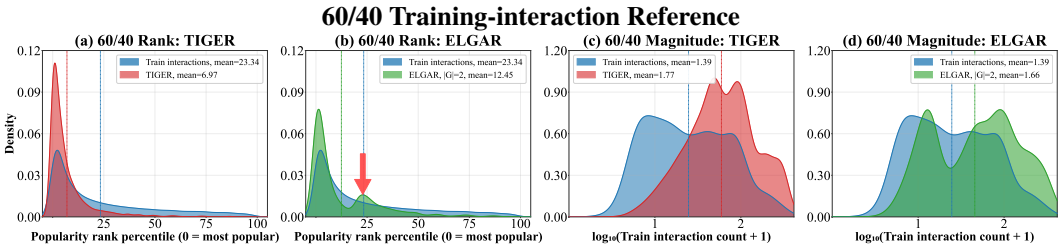

Figure 11: Top-10 exposure distributions on Amazon Beauty under the nominal 60/40 Head–Tail configuration with $G_{\text{route}} = 2$.

training configurations apart from the Head boundary. We select checkpoints by validation NDCG@20 and fix the decoding weight to $\lambda_{\text{dec}} = 0.2$, without a validation search over decoding weights.

Distribution views and group-level alignment. Figures 10–13 show Top-10 exposure distributions in popularity-rank and popularity-magnitude spaces, both computed from raw training-interaction counts. The reference and TIGER KDEs remain fixed across configurations; only the independently trained ELGAR curves change. As η increases, the onset of visible exposure redistribution generally shifts toward deeper long-tail ranks. This pattern illustrates how the selected reference configuration changes the popularity region in which alignment is applied.

Table 8 shows that ELGAR achieves lower MGU@10 than TIGER under all four corresponding partitions, demonstrating alignment across different reference configurations. Increasing η expands the Head group, raises NDCG@10, and yields an overall decrease in ELGAR's MGU@10. These cross-configuration differences reflect changes in both the achieved exposure and the fairness target itself. A lower MGU under a larger η therefore does not establish that this reference configuration is intrinsically fairer.

Choosing a reference configuration. The supplementary results demonstrate ELGAR's applicability to alternative reference configurations and we recommend choosing the Head–Tail boundary according to the dataset's empirical long-tail structure and the popularity region whose exposure requires correction. In practice, the cumulative interaction-mass curve can be inspected to identify whether a broader Tail containing moderately popular items or a narrower deep-Tail region is more relevant to the application. Candidate boundaries should be assessed using validation data for both recommendation utility and exposure alignment, and then fixed before test evaluation. The 80/20 setting provides a prior-work reference for the main experiments; it is not a universal prescription for every dataset. Once a boundary is chosen, ELGAR continues to use the realized historical interaction proportions as its exposure reference.

G Output Popularity Distribution: ELGAR vs. Ghost

MiniOneRec concentrates its recommendations on popular items, whereas both Ghost and ELGAR shift exposure toward less popular items (Figure 14). Across the two popularity measures, ELGAR's density is generally

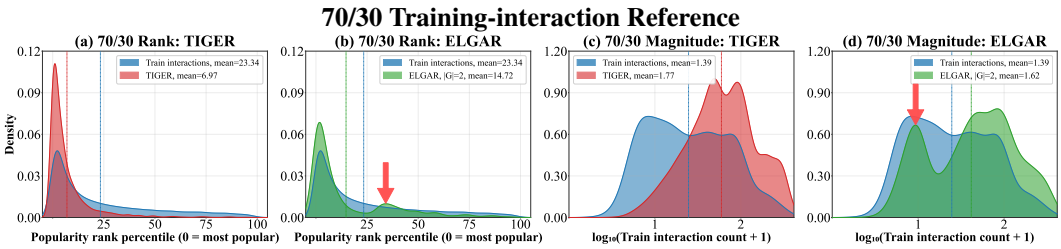

Figure 12: Top-10 exposure distributions on Amazon Beauty under the nominal 70/30 Head–Tail configuration with $G_{\text{route}} = 2$.

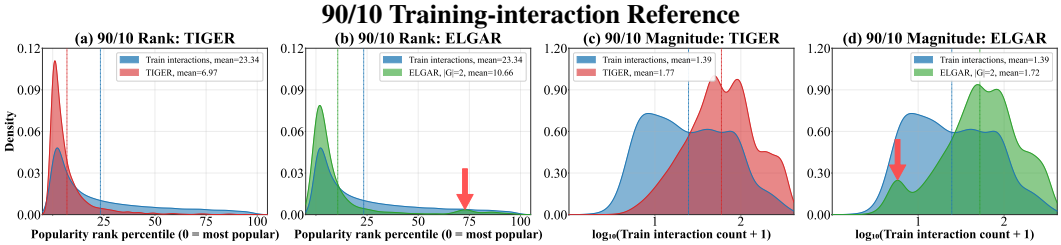

Figure 13: Top-10 exposure distributions on Amazon Beauty under the nominal 90/10 Head–Tail configuration with $G_{\text{route}} = 2$. This configuration isolates a deeper Tail region than the smaller interaction-mass boundaries.

slightly closer to the test-target distribution than Ghost's in the regions containing most recommendations. Its mean rank percentile is also closer to the test targets (17.94 versus 16.81 for Ghost; target: 38.15). The two methods remain similar overall, however, and neither reproduces the target distribution, particularly in the tail. ELGAR also retains slightly more mass among the most popular items on the interaction-count axis. The comparison with training interactions shows a similar reduction in MiniOneRec's head concentration (Figure 15).

H Regularization Variants

As an optional alternative to the fairness regularization in Eq. 12, we also consider explicitly estimating the training-time one-step fairness gains with the fairness-head outputs. Given $\mathbf{r}_j = [r_j(1), \ldots, r_j(G)]$, the estimation objective is

$$\mathcal{L}_{\text{est}} = \frac{1}{BG} \sum_{j=1}^{B} \sum_{g=1}^{G} \left(z_{j,g}^{\text{fair}} - r_j(g) \right)^2 . \tag{22}$$

This variant retains the same group-aware SID, Exposure Ledger, shared backbone, fairness head, and serving-time fusion; it changes only how the fairness head is trained. The main paper uses fairness regularization, while the gain-estimation variant is reserved for supplementary analysis.

I Rule-Based Group Routing

This experiment examines whether a simple allocation rule is sufficient to achieve exposure alignment while preserving recommendation utility. We compare ELGAR with a controller that assigns groups according to the reference distribution $\boldsymbol{\pi}$ without using the current user history for group allocation. This baseline directly targets aggregate distribution matching; relevance is used only to rank items within each assigned group.

Setup. The backbone is TIGER trained on group-augmented SIDs with $\mathcal{L}_{\text{gen}}$ only. The fairness head, ledger adapter, $\mathcal{L}_{\text{fuse}}$, and $\mathcal{L}_{\text{fair}}$ are omitted. For each request t, the controller samples the group of every displayed position independently:

$$g_t^{(k)} \sim \text{Categorical}(\boldsymbol{\pi}), \qquad k = 1, \ldots, K. \tag{23}$$

For each group g, constrained beam search fixes the first token to γ_g and generates a ranking of valid items in that group. Position k is filled with the highest-ranked item from group $g_t^{(k)}$ that has not already been

Table 8: Alternative Head–Tail reference configurations on Amazon Beauty. The Head cutoff is the percentage of catalog items assigned to Head. π_H is its realized training-interaction reference mass, and ρ_H is ELGAR's Top-10 Head-exposure proportion. TIGER MGU is recomputed from the reused baseline predictions under each corresponding partition. All values are proportions, not percentages.

Nominal split	Head cutoff (%)	π_H	ρ_H	TIGER MGU@10	ELGAR MGU@10	ELGAR NDCG@10
50/50	12.49	0.5000	0.6051	0.3424	0.1051	0.0281
60/40	19.68	0.6001	0.7087	0.3126	0.1086	0.0296
70/30	30.00	0.7000	0.7687	0.2564	0.0687	0.0298
90/10	65.94	0.9000	0.9413	0.0940	0.0413	0.0301

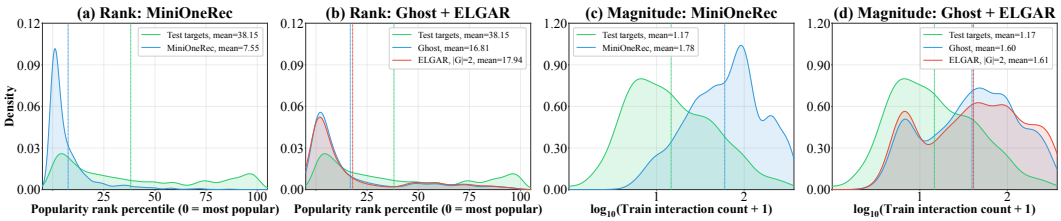

Figure 14: Popularity distribution of top-10 recommendations on Amazon Beauty, compared against the test targets (green). Panels (a)–(b) use the popularity rank percentile (0 = most popular); panels (c)–(d) use $\log_{10}$(train interaction count $+1$). Dotted vertical lines mark the distribution means. Ghost (blue) and ELGAR ($|G| = 2$, red) shift MiniOneRec's head-concentrated output toward the tail to a similar extent.

selected. Thus, within-group ranking remains conditioned on user history, but the group assigned to each position depends only on $\boldsymbol{\pi}$. The data split, group partition, and beam size follow the TIGER setting in Table 1. The checkpoint is selected by validation NDCG under the same controller.

Exposure alignment and relevance. By construction, the expected exposure share of each group equals its reference proportion. With sufficiently many requests, the empirical exposure shares approach $\boldsymbol{\pi}$ up to sampling variation, producing low MGU. This control tests whether static alignment alone suffices. Training is identical to TIGER apart from the prepended group token, and at inference we remove the entire learned group-routing score, including both the history-based relevance logits $\mathbf{z}_t^G$ and the exposure-conditioned correction $\mathbf{z}_t^{\text{fair}}$. A learned router outputs request-dependent group probabilities whose aggregate over the test stream is not guaranteed to match $\boldsymbol{\pi}$ ($\pi_{\text{Head}} \approx 0.8$ under $G_{\text{route}} = 2$). Sampling each displayed position from $\text{Categorical}(\boldsymbol{\pi})$ instead enforces the target proportions by construction, up to sampling noise.

Results. Table 8 and Figure 10 show that routing from $\boldsymbol{\pi}$ achieves near-zero MGU on the three Amazon datasets, below that of ELGAR. Its overall NDCG is nevertheless lower than both vanilla TIGER and ELGAR on all three datasets. Tail NDCG is lower than that of ELGAR on Beauty and Toys, and higher on Sports. These results illustrate the utility–exposure trade-off of a simple allocation rule: close agreement with the target distribution can coexist with weaker recommendation quality. At the reported operating points, ELGAR retains higher overall utility while reducing exposure discrepancy relative to the vanilla backbone. The comparison therefore supports considering request-level relevance together with exposure alignment when designing group routing.

J Algorithms

J.1 Stateful Decoding

Algorithm A1 summarizes the serving-time ledger-guided decoding procedure described in Section 3.4.

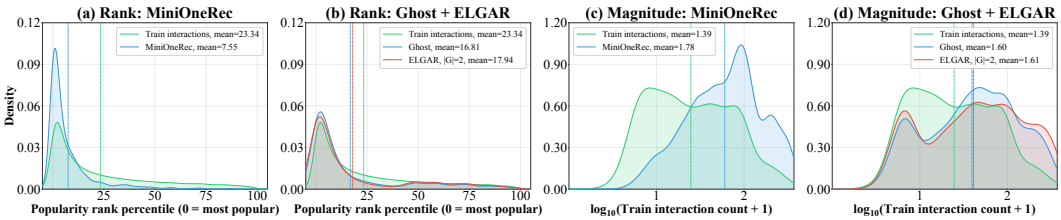

Figure 15: Same comparison as Figure 14, with the training interactions as the reference distribution (green).

Table 9: Comparison of ELGAR with rule-based random group routing on TIGER. Random routing samples the group of each displayed position from $\boldsymbol{\pi}$ at the first decoding step and uses no fairness training. ↑/↓ indicates that higher/lower is better.

Dataset	Method	HR ↑		NDCG ↑		Tail NDCG ↑		MGU ↓	
		@5	@10	@5	@10	@5	@10	@5	@10
Beauty	TIGER	0.0361	0.0559	0.0240	0.0303	0.0009	0.0018	0.1883	0.1827
	+ ELGAR	0.0359	0.0556	0.0234	0.0298	0.0037	0.0054	0.0962	0.0703
	+ Random Routing	0.0322	0.0501	0.0208	0.0265	0.0030	0.0049	0.0005	0.0003
Sports	TIGER	0.0208	0.0338	0.0128	0.0169	0.0003	0.0005	0.1938	0.1907
	+ ELGAR	0.0202	0.0324	0.0131	0.0170	0.0009	0.0016	0.1249	0.1059
	+ Random Routing	0.0177	0.0306	0.0109	0.0151	0.0013	0.0019	0.0001	0.0001
Toys	TIGER	0.0317	0.0515	0.0209	0.0272	0.0056	0.0082	0.1516	0.1417
	+ ELGAR	0.0320	0.0490	0.0209	0.0263	0.0097	0.0129	0.0455	0.0300
	+ Random Routing	0.0308	0.0484	0.0194	0.0251	0.0067	0.0088	0.0006	0.0004

Algorithm A1 ELGAR Stateful Decoding

Require: Request blocks $\{\mathcal{B}_m\}$, reference distribution $\boldsymbol{\pi}$, group-token set Γ, decoding weight λ_{dec}

Initialize raw exposure counts $\mathbf{n}^{(1)} \leftarrow \mathbf{0}$
for $m = 1, 2, \ldots$ **do**
  Construct Exposure Ledger $\mathbf{L}^{(m)}$ from $\mathbf{n}^{(m)}$
  for each request $t \in \mathcal{B}_m$ **do**
    Compute $\mathbf{z}_t^G$ and ledger-conditioned fairness logits $\mathbf{z}_t^{\text{fair},(m)}$
    $\mathbf{z}_t^{\text{dec},(m)} \leftarrow \mathbf{z}_t^G + \lambda_{\text{dec}} \mathbf{z}_t^{\text{fair},(m)}$
    Expand only group tokens $\gamma_g \in \Gamma$
    Restrict the beam to valid SIDs in group g and decode the remaining SID
    Obtain displayed Top-K list $\mathcal{R}_t$
  end for
  Update $\mathbf{n}^{(m+1)}$ from $\{\mathcal{R}_t : t \in \mathcal{B}_m\}$
end for

J.2 Training-Time Exposure-State Construction

Training data do not define a persistent serving-time request stream. We therefore use each shuffled mini-batch as a local mechanism for constructing exposure-conditioned training states, rather than as a replay or approximation of the model-induced serving trajectory. Within the batch, raw exposure counts are advanced request by request using the ground-truth target group, and the corresponding Exposure Ledger and one-step fairness gains are constructed before each update. We write the batch as $\mathcal{E} = \{(\mathbf{x}_j, i_j^*)\}_{j=1}^{B}$ and initialize $\mathbf{n}_1 = \mathbf{0}$. The counts are local to the batch and are reset before the next mini-batch.

This construction is used only to generate ledger-conditioned supervision. Model-generated groups and Top-K recommendations are therefore not required during training, and the one-step gain is not executed during Top-K inference. At serving time, the realized displayed recommendations instead determine the raw counts,

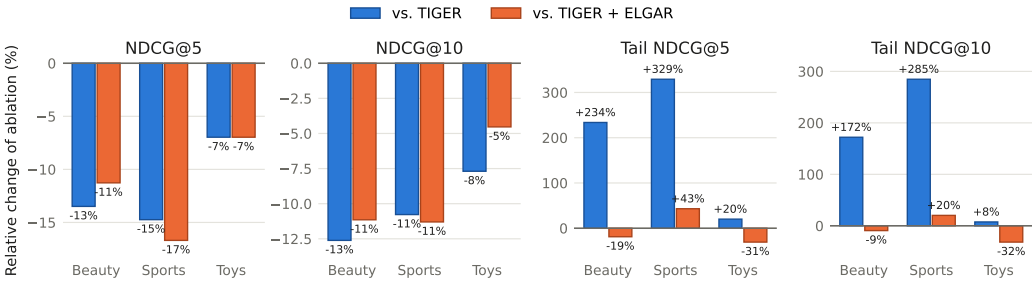

Figure 16: Relative change of random group routing with respect to TIGER (blue) and TIGER + ELGAR (orange), computed as $(\text{Random} - \text{Reference})/\text{Reference}$. Values below zero indicate that random routing underperforms the reference. Relative changes in Tail NDCG with respect to TIGER are computed over small bases.

which are projected into the same Exposure Ledger representation before routing. Algorithm A2 summarizes the training-time construction.

Algorithm A2 Training-Time Exposure-State Construction

Require: Shuffled training episode $\mathcal{E} = \{(\mathbf{x}_j, i_j^*)\}_{j=1}^{B}$, training-interaction reference distribution $\boldsymbol{\pi}$, ledger horizon H

1: Run one batched teacher-forced backbone forward pass for all samples in $\mathcal{E}$
2: Initialize raw exposure counts $\mathbf{n}_1 \leftarrow \mathbf{0}$
3: **for** $j = 1, \ldots, B$ **do**
4: Construct Exposure Ledger $\mathbf{L}_j$ from $\mathbf{n}_j$ using Eq. 7
5: Compute $\{r_j(g)\}_{g=1}^{G}$ from $\mathbf{n}_j$ and $\boldsymbol{\pi}$ using Eq. 8
6: $\mathbf{n}_{j+1} \leftarrow \mathbf{n}_j + \boldsymbol{\delta}_{g_j}$ using the ground-truth group g_j
7: **end for**
8: Compute fairness logits, fused routing distributions, and $\mathcal{L}_{\text{gen}}, \mathcal{L}_{\text{fuse}}, \mathcal{L}_{\text{fair}}$ for the episode in batch
9: Update model parameters with one batched backward pass; discard the local raw exposure counts

Only the lightweight construction of Exposure Ledgers and one-step gains is sequential; the Transformer forward pass, fairness-head computation, loss evaluation, and backward pass remain batched. Each shuffled mini-batch forms one short exposure episode; its local raw exposure counts are discarded after the batch update.

Role of the label-driven exposure episodes. The training episode is designed to provide exposure-state variation rather than to imitate the transition process encountered at serving time. For request $j > 1$, the normalized count of group g is simply the empirical frequency of that group among the preceding targets,

$$\rho_g(\mathbf{n}_j) = \frac{1}{j-1} \sum_{\ell<j} \mathbb{I}[g_\ell = g]. \tag{24}$$

Consequently, resetting $\mathbf{n}_1 = \mathbf{0}$ for every shuffled mini-batch repeatedly exposes the router to finite-prefix exposure states before the empirical proportions stabilize. This mechanism does not rely on rolling-prefix expansion changing the target-group distribution. Even in the idealized case where the shuffled target groups were independently drawn according to the reference distribution $\boldsymbol{\pi}$, we would have

$$\mathbb{E}[\rho_g(\mathbf{n}_j)] = \pi_g, \qquad \mathrm{Var}[\rho_g(\mathbf{n}_j)] = \frac{\pi_g(1-\pi_g)}{j-1}, \qquad j > 1, \tag{25}$$

so early finite prefixes naturally produce nonzero deviations around $\boldsymbol{\pi}$. Reinitializing the counts for every mini-batch prevents these early imbalance states from disappearing as training proceeds.

These deviations have a direct routing interpretation. A positive component $\rho_g(\mathbf{n}_j) - \pi_g > 0$ indicates that group g is currently over-exposed relative to the reference, whereas a negative component indicates under-exposure. The one-step gains in Eq. 8 then evaluate, under the same current counts, how assigning the next

exposure to each possible group would change the discrepancy. Training therefore learns a mapping from the current request and Exposure Ledger to a group-level correction direction,

$$(\mathbf{x}_j, \mathbf{L}_j) \longrightarrow \text{group-routing preference},$$

rather than learning to reproduce the label-driven transition sequence itself.

Relation to serving-time ledger updates. Training and serving differ in how the raw exposure counts are generated. Training advances the local counts by one ground-truth item,

$$\mathbf{n}_{j+1} = \mathbf{n}_j + \boldsymbol{\delta}_{g_j},$$

whereas serving aggregates all displayed items in a Top-K list. For a single serving request, this update can be written as

$$\mathbf{n}^+ = \mathbf{n} + \mathbf{c}_t, \qquad c_{t,g} = \sum_{k=1}^{K} \mathbb{I}[g(i_t^{(k)}) = g], \qquad \sum_{g=1}^{G} c_{t,g} = K. \tag{26}$$

ELGAR does not require these two transition mechanisms to be identical. In both cases, the resulting raw counts are compressed into the same normalized state

$$\mathbf{L} = \left[\boldsymbol{\rho}(\mathbf{n}) - \boldsymbol{\pi}; \min\left(\frac{\sum_g n_g}{H}, 1 \right) \right].$$

Thus, the fairness head conditions on the current group-level exposure state rather than on the source of the transition that produced it. A Top-K serving update may skip intermediate single-exposure states, but it does not change the interpretation of a visited ledger state: the first G coordinates identify the current over- and under-exposed groups, while the final coordinate indicates whether the episode is still in its early exposure stage.

The corresponding design assumption is therefore one of *state sufficiency*, rather than trajectory matching: once the current exposure ledger is represented by $\mathbf{L}$, the desired group-level correction is determined from that state together with the current request representation. The label-driven training episodes provide supervision for this state-conditioned correction, whereas the persistent serving ledger supplies the realized state on which the learned correction is applied.

Exposure-level progress and ledger horizon. We set $H = 20$ for all reported experiments. Importantly, H is measured in *item exposures*, not in recommendation requests. During training, each ground-truth target advances the exposure count by one, so the progress coordinate increases by $1/H$ at each step until it reaches 1. During Top-K serving, one request contributes K displayed item exposures; hence the same coordinate advances by K/H after that request. For the default Top-10 evaluation, the progress coordinate therefore reaches 1 after the first 20 displayed items, i.e., after two requests when $|\mathcal{B}_m| = 1$, and remains saturated thereafter. Its purpose is to distinguish the early warm-up regime, where normalized exposure proportions are based on few observations, from the subsequent steady regime; it is not intended to encode the absolute serving-request index.

K Theoretical Analysis

We interpret the fairness objective as an exposure-alignment prior and derive its generalized maximum a posteriori (MAP) objective. We then compare estimation with and without this regularizer. The analysis concerns the training-time, single-exposure surrogate; it does not assume that improving this surrogate guarantees lower prediction error or convergence of deployed Top-K exposure. Throughout, ϑ denotes all trainable parameters, preserving θ for the backbone parameters in the main text.

K.1 Exposure-Alignment Prior and MAP Estimation

Likelihood and generation loss. Let $\mathcal{D} = \{(x_i, C_i)\}_{i=1}^{N}$ denote the training data, where C_i is the complete identifier of the target item. Identifiers are uniquely decodable and have finite length, including a termination decision when necessary. The generative head defines a normalized sequence distribution and assigns positive probability to the data support. Autoregressive factorization gives

$$p_{\text{gen}}(C_i \mid x_i; \vartheta) = \prod_{\ell} p_{\text{gen}}(c_{i,\ell} \mid x_i, c_{i,<\ell}; \vartheta). \tag{27}$$

Under the usual conditional likelihood factorization across observations,

$$\begin{aligned}\widehat{\mathcal{L}}_{\mathrm{gen}}(\vartheta) &= -\frac{1}{N}\sum_{i=1}^{N}\log p_{\mathrm{gen}}(C_i \mid x_i;\vartheta)\\ &= -\frac{1}{N}\sum_{i,\ell}\log p_{\mathrm{gen}}(c_{i,\ell} \mid x_i, c_{i,<\ell};\vartheta).\end{aligned} \tag{28}$$

Thus, minimizing the token-level cross-entropy in Eq. equation 28 is maximum likelihood estimation (Shalev-Shwartz & Ben-David, 2014). For a given parameter prior p_{prior}, the MAP-form objective is (Shalev-Shwartz & Ben-David, 2014)

$$\widehat{\mathcal{L}}_{\mathrm{gen}}(\vartheta) - \frac{1}{N}\log p_{\mathrm{prior}}(\vartheta). \tag{29}$$

We next specify a prior that favors exposure alignment with π.

Alignment criterion. For a shuffled mini-batch of size B, retain the label-driven ledger construction

$$n_1 = 0, \qquad n_{j+1} = n_j + \delta_{g_j}, \qquad g_j = g(i_j^*). \tag{30}$$

Here δ_g is the g-th standard basis vector. The routing probabilities are $q_j = \mathrm{softmax}(z_j^G + z_j^{\mathrm{fair}})$, with the fairness logits conditioned on L_j. Define the expected discrepancy after one group-level exposure as

$$R_B(\vartheta;\pi) = \frac{1}{B}\sum_{j=1}^{B}\sum_{g=1}^{G} q_j(g) D(n_j + \delta_g;\pi). \tag{31}$$

Write $\widehat{R}(\vartheta;\pi)$ for its average over the specified mini-batch sampling and ordering rule on the training data. The corresponding averaged training losses use the same rule, with equal sample weighting for Eq. equation 28. No independence between mini-batches is required for the identities below. We assume $0 \le D(n;\pi) \le D_{\max} < \infty$.

Prior construction. The reference π specifies the desired group exposure proportions. We seek a parameter prior that favors small $\widehat{R}$, while changing a proper base density p_0 as little as possible. Following the minimum relative-entropy, or information-projection, principle (Csiszár, 1975), we express this preference as

$$\begin{aligned}&\min_{p} \quad D_{\mathrm{KL}}(p\|p_0)\\ &\text{subject to} \quad \int p(\vartheta)\widehat{R}(\vartheta;\pi)\,d\vartheta \le \tau, \qquad \int p(\vartheta)\,d\vartheta = 1,\end{aligned} \tag{32}$$

over densities supported on the support of p_0. For a feasible constraint admitting a finite optimal multiplier $\beta \ge 0$, the Lagrangian is

$$\begin{aligned}\mathcal{F}(p) = &\int p(\vartheta)\log\frac{p(\vartheta)}{p_0(\vartheta)}\,d\vartheta\\ &+ \beta\left[\int p(\vartheta)\widehat{R}(\vartheta;\pi)\,d\vartheta - \tau\right] + \nu\left[\int p(\vartheta)\,d\vartheta - 1\right].\end{aligned} \tag{33}$$

Stationarity with respect to p yields

$$\log\frac{p(\vartheta)}{p_0(\vartheta)} + 1 + \beta\widehat{R}(\vartheta;\pi) + \nu = 0. \tag{34}$$

Normalization therefore gives

$$p_{\mathrm{align}}(\vartheta) = \frac{1}{Z}\,p_0(\vartheta)\exp[-\beta\widehat{R}(\vartheta;\pi)], \qquad Z = \int p_0(\vartheta)e^{-\beta\widehat{R}(\vartheta;\pi)}\,d\vartheta. \tag{35}$$

In particular, $e^{-1-\nu} = Z^{-1}$. Boundedness implies $e^{-\beta D_{\max}} \le Z \le 1$. The same density is the unique minimizer, up to null sets, of $D_{\mathrm{KL}}(p\|p_0) + \beta\mathbb{E}_p[\widehat{R}]$: this expression equals $D_{\mathrm{KL}}(p\|p_{\mathrm{align}}) - \log Z$. Under the multiplier and feasibility conditions above, it solves Eq. equation 32. This is the standard exponential tilt induced by an expectation constraint; the choice of the exposure risk in Eq. equation 31 specializes it to ELGAR.

The prior assigns greater density to models with smaller expected exposure discrepancy, relative to their base density. Its preference depends on the ledger states through Eq. equation 31, even when π is fixed. It does not impose a fixed preference for tail groups irrespective of their current exposure.

Recovering the fairness objective. Substituting Eq. equation 35 into Eq. equation 29 and setting $\beta = N\lambda_{\text{reg}}$ gives

$$\widehat{\mathcal{L}}_{\text{gen}}(\vartheta) + \lambda_{\text{reg}}\widehat{R}(\vartheta;\pi) - \frac{1}{N}\log p_0(\vartheta), \tag{36}$$

up to a parameter-independent constant. To connect this objective to ELGAR, substitute $r_j(g) = D(n_j;\pi) - D(n_j + \delta_g;\pi)$ into the fairness loss:

$$\begin{aligned}\mathcal{L}_{\text{fair}}(\vartheta) &= -\frac{1}{B}\sum_{j,g} q_j(g) r_j(g) \\ &= R_B(\vartheta;\pi) - \frac{1}{B}\sum_{j=1}^{B} D(n_j;\pi).\end{aligned} \tag{37}$$

The second term is independent of ϑ, since the training counts are updated with target groups. Averaging over training mini-batches preserves this identity up to a constant.

For a uniform base density on a fixed bounded parameter domain Θ of positive finite volume, Eq. equation 36 consequently recovers $\widehat{\mathcal{L}}_{\text{gen}} + \lambda_{\text{reg}}\widehat{\mathcal{L}}_{\text{fair}}$. For the full objective, define

$$\widehat{\mathcal{L}}_{\text{rel}} = \widehat{\mathcal{L}}_{\text{gen}} + \lambda_{\text{fuse}}\widehat{\mathcal{L}}_{\text{fuse}}. \tag{38}$$

Following the loss-based updating framework of Bissiri et al. (2016), using the factor $\exp[-N\widehat{\mathcal{L}}_{\text{rel}}]$ gives the generalized posterior

$$\widetilde{p}(\vartheta \mid \mathcal{D}) \propto \exp[-N\widehat{\mathcal{L}}_{\text{rel}}(\vartheta)]p_{\text{align}}(\vartheta), \tag{39}$$

whose MAP objective is exactly, up to constants,

$$\widehat{\mathcal{L}}_{\text{gen}} + \lambda_{\text{fuse}}\widehat{\mathcal{L}}_{\text{fuse}} + \lambda_{\text{reg}}\widehat{\mathcal{L}}_{\text{fair}}. \tag{40}$$

This is a data-dependent generalized MAP interpretation: p_{align} uses training-derived ledger states, and the fused supervision is included through a loss-based factor. The prior is not independent of the observations. Equivalently, the same distribution can be written as a single Gibbs update from the fixed base density:

$$\widetilde{p}(\vartheta \mid \mathcal{D}) \propto p_0(\vartheta)\exp\left\{-N\left[\widehat{\mathcal{L}}_{\text{rel}}(\vartheta) + \lambda_{\text{reg}}\widehat{R}(\vartheta;\pi)\right]\right\}. \tag{41}$$

This representation uses the combined loss, rather than treating the training-derived alignment factor as independent prior evidence. The general updating framework supplies the Gibbs form; it does not establish a generalization guarantee for ELGAR's ledger construction. The parameter τ motivates its construction; it need not be introduced as an additional training hyperparameter.

K.2 Conditional Risk Comparison

We compare the same model class with and without the fairness regularizer. Fix λ_{fuse}, the grouping, and the reference π, and define

$$\widehat{\vartheta}_0 \in \arg\min_{\vartheta\in\Theta} \widehat{\mathcal{L}}_{\text{rel}}(\vartheta), \tag{42}$$

$$\widehat{\vartheta}_{\text{MAP}} \in \arg\min_{\vartheta\in\Theta} \left[\widehat{\mathcal{L}}_{\text{rel}}(\vartheta) + \lambda_{\text{reg}}\widehat{R}(\vartheta)\right]. \tag{43}$$

When $\lambda_{\text{fuse}} = 0$, the first estimator is pure MLE. Otherwise, it is the no-fairness-loss ablation with fused supervision retained. Let $\mathcal{L}_{\text{rel}}$ and R denote population expectations on fresh data batches with the same ordering and label-driven ledger construction. We use the standard uniform-convergence approach to empirical risk minimization (Shalev-Shwartz & Ben-David, 2014), while leaving the required estimation bounds explicit.

Assumption K.1 (Uniform risk estimation)**.** *With probability at least $1-\delta$, simultaneously for every $\vartheta \in \Theta$,*

$$\left|\widehat{\mathcal{L}}_{\text{rel}}(\vartheta) - \mathcal{L}_{\text{rel}}(\vartheta)\right| \le \epsilon_{\text{rel}}, \tag{44}$$

$$|\widehat{R}(\vartheta) - R(\vartheta)| \le \epsilon_R. \tag{45}$$

The risks are finite, and the empirical minima in Eq. equation 43 are attained.

This assumption makes the following results conditional on statistical estimation accuracy and exact empirical optimization. We do not infer independence from shuffled mini-batches or assign a sample-size rate to ϵ_R. Establishing such a rate requires control of the sampling dependence and model complexity, including any dependence introduced by estimating the grouping and π from the training data.

The following standard ERM bound adapts Lemma 4.2 of Shalev-Shwartz & Ben-David (2014) to our notation.

Theorem K.1 (Standard risk bound without fairness regularization)**.** *Under Assumption K.1, with probability at least* $1-\delta$*,*

$$\mathcal{L}_{\mathrm{rel}}(\widehat{\vartheta}_0) - \inf_{\vartheta\in\Theta} \mathcal{L}_{\mathrm{rel}}(\vartheta) \le 2\epsilon_{\mathrm{rel}}. \tag{46}$$

Proof. On the uniform estimation event, for every $\vartheta \in \Theta$,

$$\begin{aligned}\mathcal{L}_{\mathrm{rel}}(\widehat{\vartheta}_0) &\le \widehat{\mathcal{L}}_{\mathrm{rel}}(\widehat{\vartheta}_0) + \epsilon_{\mathrm{rel}} \\ &\le \widehat{\mathcal{L}}_{\mathrm{rel}}(\vartheta) + \epsilon_{\mathrm{rel}} \\ &\le \mathcal{L}_{\mathrm{rel}}(\vartheta) + 2\epsilon_{\mathrm{rel}}. \end{aligned} \tag{47}$$

Taking the infimum over ϑ proves the result. □

For pure MLE, the population generation loss has the standard entropy–KL decomposition (Shalev-Shwartz & Ben-David, 2014):

$$\mathcal{L}_{\mathrm{gen}}(\vartheta) = H_P(C \mid X) + \mathbb{E}_X D_{\mathrm{KL}}\left(P(C \mid X) \,\|\, p_{\mathrm{gen}}(C \mid X; \vartheta)\right). \tag{48}$$

Since the entropy term is parameter-independent, Theorem K.1 also bounds excess KL prediction risk relative to the best model in Θ. It provides no direct bound on the exposure risk R.

The next result combines empirical optimality with the same uniform deviation argument to compare the two ELGAR objectives. It is a conditional consequence of regularized risk minimization, not an additional sample-complexity bound.

Theorem K.2 (Exposure improvement and relevance cost)**.** *Under Assumption K.1, let* $\lambda_{\mathrm{reg}} > 0$ *and define*

$$\widehat{\Delta}_R = \widehat{R}(\widehat{\vartheta}_0) - \widehat{R}(\widehat{\vartheta}_{\mathrm{MAP}}). \tag{49}$$

Then $\widehat{\Delta}_R \ge 0$*, and with probability at least* $1-\delta$*,*

$$R(\widehat{\vartheta}_{\mathrm{MAP}}) - R(\widehat{\vartheta}_0) \le -\widehat{\Delta}_R + 2\epsilon_R, \tag{50}$$

$$\mathcal{L}_{\mathrm{rel}}(\widehat{\vartheta}_{\mathrm{MAP}}) - \mathcal{L}_{\mathrm{rel}}(\widehat{\vartheta}_0) \le \lambda_{\mathrm{reg}}\widehat{\Delta}_R + 2\epsilon_{\mathrm{rel}}. \tag{51}$$

In particular, on this event, $\widehat{\Delta}_R > 2\epsilon_R$ *implies* $R(\widehat{\vartheta}_{\mathrm{MAP}}) < R(\widehat{\vartheta}_0)$.

Proof. Optimality of the two empirical estimators gives

$$\begin{aligned} 0 &\le \widehat{\mathcal{L}}_{\mathrm{rel}}(\widehat{\vartheta}_{\mathrm{MAP}}) - \widehat{\mathcal{L}}_{\mathrm{rel}}(\widehat{\vartheta}_0) \\ &\le \lambda_{\mathrm{reg}}\left[\widehat{R}(\widehat{\vartheta}_0) - \widehat{R}(\widehat{\vartheta}_{\mathrm{MAP}})\right] = \lambda_{\mathrm{reg}}\widehat{\Delta}_R. \end{aligned} \tag{52}$$

Because $\lambda_{\mathrm{reg}} > 0$, this proves $\widehat{\Delta}_R \ge 0$. On the uniform estimation event,

$$\begin{aligned} R(\widehat{\vartheta}_{\mathrm{MAP}}) - R(\widehat{\vartheta}_0) &\le \widehat{R}(\widehat{\vartheta}_{\mathrm{MAP}}) - \widehat{R}(\widehat{\vartheta}_0) + 2\epsilon_R \\ &= -\widehat{\Delta}_R + 2\epsilon_R. \end{aligned} \tag{53}$$

Applying the relevance estimation bound to Eq. equation 52 yields Eq. equation 51. □

Interpretation and scope. The prior induces a nonnegative empirical exposure improvement under exact optimization. Theorem K.2 shows when this improvement transfers to population surrogate risk: it must exceed the uncertainty $2\epsilon_R$. The theorem does not itself establish a small value of ϵ_R. When $\lambda_{\mathrm{fuse}} = 0$, Eq. equation 48 makes the relevance-risk difference in Eq. equation 51 exactly the difference in KL prediction risks. Thus, the result controls a possible prediction cost rather than guaranteeing reduced prediction error or overfitting. With fused supervision, the bound applies to the combined relevance loss. Finally, R evaluates a single exposure under label-driven ledger states. Model-generated ledger states and Top-K decoding at deployment require additional analysis; cumulative displayed exposure is not covered by these results.