Compact Representations of Impact-Based Fair-Ranking Policies

Yuki Uehara
Preferred Networks, Inc.
yukiuehara00@preferred.jp

Naoki Nishimura
Recruit Co., Ltd.
nishimura@r.recruit.co.jp

Noriyoshi Sukegawa
Hosei University
sukegawa@hosei.ac.jp

Yuichi Takano
University of Tsukuba
ytakano@sk.tsukuba.ac.jp

Abstract

On two-sided platforms such as e-commerce marketplaces, recommender systems deliver personalized item rankings to consumers and serve as a primary channel through which transactions occur between item providers and consumers. At the same time, promoting the activities of item providers requires the platform to balance consumer satisfaction with fair exposure across items. Recent work formulates this objective as impact-based fair ranking, which maximizes a Nash social welfare criterion over stochastic ranking policies. However, the resulting policies must be served as user-specific mixtures of deterministic rankings, whose support size can grow combinatorially. This imposes substantial storage and serving overhead, driving up operational cost in production deployment. Moreover, unseen users must fall back to a generic ranking rule. We therefore develop compact representations for serving impact-based fair-ranking policies. We first prove a tight bound that every optimum admits an impact-preserving mixture with support size at most the number of items. Using a dual reformulation, we then replace stored user-specific ranking mixtures with item-indexed dual vectors and compress the resulting sequence of dual vectors by weighted random subsampling. The resulting deployment representation is independent of the training-user population and can be used for unseen users. Experiments on real-world recommendation datasets show that the sampled dual mixture preserves the fair-ranking objective while reducing the storage required at deployment by up to 2,596$\times$ relative to a user-coupled mixture baseline. This end-to-end reduction combines a factor of about 64 from subsampling with a factor of about 40 from the item-indexed dual representation.

1 Introduction

On modern online platforms such as e-commerce marketplaces and video streaming services, ranking policies serve large user populations and allocate attention across large item catalogs, directly shaping clicks, purchases, and revenue for both sides of the market [1]. Because users typically examine only the top of a ranked list, attention concentrates on a small set of highly ranked items while many others receive little exposure. These ranking-induced disparities translate into differences in income and affect business sustainability for the item providers (sellers, creators, and other suppliers) who depend on the platform. Persistent disparities can therefore discourage supply-side participation, shrink the available catalog, and ultimately weaken the long-run health of the platform itself, making fair ranking a structural concern rather than just an ethical one.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Prior research has made important progress on fair ranking in two-sided markets, but has primarily focused on fairness definitions and learning algorithms rather than on how the resulting policies are represented and served [26, 27]. Earlier work formalized protected-group top-k constraints [36], fairness of exposure and amortized attention [4, 29], and learning-to-rank objectives that reduce disparate exposure [35]. Later work developed policy-learning and scalable optimization methods for fair ranking and two-sided recommendation [9, 14, 22, 24, 25, 30, 31]. Moving beyond exposure alone, Saito and Joachims [28] introduced impact-based fair ranking, which evaluates fairness through realized item outcomes such as clicks, purchases, and revenue. This is the formulation we build on. Subsequent research has advanced scalable optimization for impact-based fair ranking using Sinkhorn-based algorithms and Frank–Wolfe methods [11, 32, 33]. Expohedron-based methods compactly represent distributions over rankings for a single query, amortizing exposure over repeated requests for that query [18, 19]. In contrast, two-sided platforms serve personalized rankings to many users, and item-side fairness couples these rankings across users. Compact representations of such coupled policies remain open.

In deployment, the system must return one deterministic ranking per request, so a learned stochastic policy must be realized as an executable mixture of deterministic rankings whose distribution matches the policy at each user. The space of deterministic rankings is combinatorially large: with just the top 10 of 100 items displayed, the number of possible top-10 rankings per user is $100!/(100-10)! \approx 6.28 \times 10^{19}$. In principle, the number of components could reach this scale, yet the actual number needed to represent an impact-based fair ranking policy has not been characterized in prior work.

In existing approaches, each user requires its own mixture of deterministic rankings, so deployment storage scales with the number of users times the mixture size per user. For instance, the Frank–Wolfe-based OFFR algorithm of Usunier et al. [33] accumulates one mixture component per iteration per user. At platform scale, this user-coupled representation is costly because the stored mixtures become latency-critical deployment data that must be replicated and refreshed across serving machines. It also does not cover unseen users, forcing the system to use a fallback ranking rule rather than directly applying the learned fair-ranking policy.

In this paper, we study how to learn, store, and serve stochastic impact-based fair-ranking policies while preserving objective value. We propose a dual mirror-descent algorithm that learns a compact mixture of item-indexed dual vectors, yielding a user-independent deployment representation. Our main contributions are as follows.

1. **Exact sparse structure of optimal policies.** We show that the optimal objective value can always be attained by a mixture of deterministic ranking tensors whose support is no larger than the number of items, and that this support bound is worst-case tight. A standard affine-elimination procedure compresses any optimal mixture to this support size while preserving the objective value exactly.
2. **Approximate compression by simple subsampling.** Going beyond this exact limit, we use weighted random subsampling as the simplest possible approximate-compression primitive. We prove a local preservation guarantee for the impact objective and use this simple compressor to reveal redundancy in mixtures of deterministic rankings.
3. **User-independent deployment.** We apply this subsampling to the dual formulation of impact-based fair ranking, yielding a user-independent dual mixture whose storage does not scale with the number of users and whose serving rule applies to unseen users.
4. **Empirical validation.** On real-world recommendation datasets, the sampled dual mixture reduces storage by up to $2{,}596\times$ end to end relative to user-coupled mixtures while preserving objective quality. Relative to OFFR [33], the reduction factors into about $64\times$ fewer stored components from subsampling and 3–$40\times$ smaller components from the item-indexed dual representation.

2 Setup and Policy Representation

This section recalls the impact-based fair-ranking problem of Saito and Joachims [28] and the mixture representation needed to serve stochastic policies as deterministic rankings.

2.1 Impact-Based Fair Ranking Problem

We consider a market with m users and n items, indexed by $[m] = \{1, 2, \ldots, m\}$ and $[n] = \{1, 2, \ldots, n\}$. For a single user, a deterministic ranking is represented by a permutation matrix. Let

$$\mathcal{P} := \left\{ \Pi \in \{0,1\}^{n \times n} \;\middle|\; \sum_{k=1}^{n} \Pi_{i,k} = 1 \; \forall i \in [n],\; \sum_{i=1}^{n} \Pi_{i,k} = 1 \; \forall k \in [n] \right\} \tag{2.1}$$

denote the set of deterministic rankings for one user. Each $\Pi \in \mathcal{P}$ assigns every item $i \in [n]$ to exactly one position $k \in [n]$, and every position receives exactly one item.

A stochastic ranking policy is a tensor $X \in [0,1]^{m \times n \times n}$, where $X_{u,i,k}$ is the probability that item $i \in [n]$ is placed at position $k \in [n]$ for user $u \in [m]$. For each user, a feasible stochastic ranking is a convex combination of deterministic rankings for that user:

$$\mathcal{X} := \{X \in [0,1]^{m \times n \times n} \mid X_u \in \mathrm{conv}(\mathcal{P}) \; \forall u \in [m]\}, \tag{2.2}$$

where $X_u := (X_{u,i,k})_{(i,k) \in [n] \times [n]} \in [0,1]^{n \times n}$ denotes the ranking matrix for user u and conv denotes convex hull. Equivalently, each X_u is doubly stochastic. This tensor view is standard in continuous fair-ranking formulations [e.g., 28, 29]. We write X^{unif} for the uniform ranking policy, defined by $X^{\mathrm{unif}}_{u,i,k} = 1/n$ for all $u \in [m]$, $i \in [n]$, and $k \in [n]$.

We consider a displayed-prefix top-K setting, where only the first K positions are shown. Let $1 \geq e(1) \geq e(2) \geq \cdots \geq e(K) > 0$ denote the exposure probabilities of the displayed positions, set $e(k) = 0$ for all $k > K$, and write $E_{\max} := e(1)$. Let $r(u,i) \geq 0$ be the relevance score of item i for user u. Under a policy X, the expected impact received by item i is

$$\mathrm{Imp}_i(X) := \sum_{u=1}^{m} \sum_{k=1}^{n} e(k)\, r(u,i)\, X_{u,i,k}. \tag{2.3}$$

Here, impact refers to item-side outcomes induced by ranking, such as clicks, purchases, or revenue. We maximize the smoothed Nash social welfare (NSW)

$$F_\eta(X) := \sum_{i=1}^{n} \log\big(\eta + \mathrm{Imp}_i(X)\big), \qquad \eta > 0. \tag{2.4}$$

The parameter η ensures that $F_\eta(X)$ is well defined even when a feasible policy gives zero impact to some items. The objective favors policies that distribute impact more evenly across items rather than concentrating it on only a few of them. The resulting optimization problem is

$$\begin{array}{ll} \text{maximize} & F_\eta(X) \\ \text{subject to} & X \in \mathcal{X}. \end{array} \tag{2.5}$$

Maximizers of the Nash social welfare are envy-free and dominate uniform ranking when $K = 1$, and approximately so for $K > 1$ when items have close substitutes in relevance. They are Pareto optimal for every K [28]. Informally, no item prefers another item's allocation, every item receives at least its uniform-ranking impact, and no alternative policy improves some items without reducing another item's impact. These guarantees motivate the smoothed NSW objective. Our compression analysis focuses on preserving the value of F_η.

2.2 Mixture Representation for Serving

Serving requires a deterministic ranking per request, so we must realize X as a distribution over permutation matrices whose placement probabilities match $X_{u,i,k}$. By the Birkhoff–von Neumann theorem [5, 34], each doubly-stochastic slice X_u admits a convex decomposition $X_u = \sum_\ell \rho_{u,\ell} \Pi_u^{(\ell)}$ over permutation matrices $\Pi_u^{(\ell)} \in \mathcal{P}$. The system serves user u by sampling ℓ with probability $\rho_{u,\ell}$ and returning $\Pi_u^{(\ell)}$.

For ease of description in the compression analysis below, we adopt the equivalent joint formulation. Let $\mathcal{P}^m := \{P = [P_u]_{u \in [m]} \in \{0,1\}^{m \times n \times n} \mid P_u \in \mathcal{P} \; \forall u \in [m]\}$ be the set of deterministic

ranking tensors. Then $\mathcal{X} = \mathrm{conv}(\mathcal{P}^m)$, and any feasible policy admits a mixture representation with T components,

$$X = \sum_{t=1}^{T} \alpha_t P^{(t)}, \qquad P^{(t)} \in \mathcal{P}^m \; \forall t \in [T], \qquad \alpha_t \geq 0 \; \forall t \in [T], \qquad \sum_{t=1}^{T} \alpha_t = 1, \tag{2.6}$$

which has the same expressive power as the per-user form above.[1]

There are $(n!/(n-K)!)^m$ joint top-K configurations, growing combinatorially in m and n, so we seek a representation with much smaller support T. Computing a minimum-support decomposition is NP-hard [12], but our target is narrower. We ask how much support is needed to preserve the impact-based welfare objective. Section 3 establishes an upper bound on the support size sufficient for this preservation.

3 Compression of Exact and Approximate Stochastic Policies

The objective in problem (2.5) depends on a policy only through the impacts received by the items. This observation turns the compression question into a finite-dimensional convex-geometric problem. We first show that an optimal policy has an exact representation with support no larger than the number of items, and then give a standard affine-elimination procedure for recovering such a representation from an optimal mixture. We finally analyze weighted subsampling as a simple approximate-compression primitive. Even this simple procedure preserves the impact objective under a local bound, highlighting redundancy in mixtures of deterministic rankings. Detailed proofs are in Appendix A.

3.1 Impact Polytope

For a policy $X \in \mathcal{X}$, define its item-impact vector and the induced impact polytope by

$$\boldsymbol{I}(X) := \big(\mathrm{Imp}_i(X)\big)_{i\in[n]} \in \mathbb{R}^n, \qquad \mathcal{K} := \mathrm{conv}\{\boldsymbol{I}(P) : P \in \mathcal{P}^m\} \subseteq \mathbb{R}^n. \tag{3.1}$$

Because each Imp_i is linear in X and $\mathcal{X} = \mathrm{conv}(\mathcal{P}^m)$, $\mathcal{K}$ is exactly the feasible set of impact vectors.

Proposition 3.1 (Impact-polytope reduction)**.** *Problem* (2.5) *is equivalent to*

$$\max_{\boldsymbol{y}\in\mathcal{K}} f(\boldsymbol{y}), \qquad f(\boldsymbol{y}) := \sum_{i=1}^{n} \log(\eta + y_i). \tag{3.2}$$

This reduction moves the compression question from the policy tensor to $\mathbb{R}^n$, where the face structure of $\mathcal{K}$ controls exact support size.

3.2 Exact Optimal Sparse Representation

The next theorem gives an exact support bound for optimal policies. It shows that every optimal stochastic policy has an equivalent convex-combination representation with support at most the number of items n, preserving its item-impact vector and hence its smoothed NSW value.

Theorem 3.2 (Exact sparse optimal representation)**.** *For every optimal policy $X^\star \in \mathcal{X}$, there exists a policy*

$$X^{\mathrm{cmp}} = \sum_{s=1}^{S} \beta_s P^{(s)}, \qquad P^{(s)} \in \mathcal{P}^m \; \forall s \in [S], \qquad \beta_s \geq 0 \; \forall s \in [S], \qquad \sum_{s=1}^{S} \beta_s = 1, \tag{3.3}$$

such that $S \leq n$ and

$$\boldsymbol{I}(X^{\mathrm{cmp}}) = \boldsymbol{I}(X^\star), \qquad F_\eta(X^{\mathrm{cmp}}) = F_\eta(X^\star). \tag{3.4}$$

[1] Direct sampling of a permutation matrix from X via methods such as dependent rounding [13] is also possible, but storing X itself requires $\Theta(mnK)$ entries and does not exploit the support-level sparsity exposed by the mixture form.

The key point is that optimality places $\boldsymbol{I}(X^\star)$ on an exposed face of $\mathcal{K}$. This face has affine dimension at most $n-1$, so Carathéodory's theorem on the face represents the same impact vector using at most n deterministic impact vectors. The policy mixture then follows by linearity. The next proposition shows that this n-support bound cannot be improved in the worst case.

Proposition 3.3 (Worst-case tightness)**.** *There are instances for which every optimal mixture representation requires at least n deterministic components.*

In the simplex instance, deterministic impact vectors are the standard basis vectors in $\mathbb{R}^n$, and the reduced objective is uniquely maximized at the uniform point. Any exact representation of that point must use all n simplex vertices. Appendix A.4 gives the formal argument.

3.3 Constructive Exact Compression

Given an optimal mixture of the form (2.6), the exact support bound can be realized by affine elimination. The procedure repeatedly finds an affine dependence among active deterministic impact vectors and moves the weights until one component vanishes. Because optimal active impact vectors lie on the exposed face used in Theorem 3.2, it preserves $\boldsymbol{I}(X^\star)$ and stops once at most n components remain. Appendix A.5 gives the full procedure and proof.

Proposition 3.4 (Correctness of affine elimination)**.** *Given an optimal mixture, affine elimination terminates with at most n components and preserves the item-impact vector exactly. Consequently, it preserves F_η exactly.*

The same elimination step also applies to nonoptimal mixtures, but then the general Carathéodory threshold is $n+1$. Exact elimination still requires component impact vectors and repeated affine-dependence computations in $\mathbb{R}^n$, and even $\mathcal{O}(n)$ support can be costly for large item catalogs. Rather than solving a minimum-support compression problem, we next study lightweight approximate compression by weighted subsampling.

3.4 Approximate Compression by Random Subsampling

The point of this subsection is not to optimize the compressor, but to use a simple rule to probe redundancy in mixtures of deterministic rankings. Weighted random subsampling draws S components according to their mixture weights and keeps the empirical mixture.

Given $X = \sum_{t=1}^{T} \alpha_t P^{(t)}$, draw indices $t_s \in [T]$ independently for $s \in [S]$ with $\Pr(t_s = t) = \alpha_t$, and form

$$\hat{X}_S := \frac{1}{S} \sum_{s=1}^{S} P^{(t_s)}. \tag{3.5}$$

Linearity immediately gives

$$\mathbb{E}[\boldsymbol{I}(\hat{X}_S)] = \boldsymbol{I}(X). \tag{3.6}$$

To state a high-probability guarantee, fix a target failure probability $\delta \in (0,1)$ and let $\varepsilon_i := \max_{t:\alpha_t>0} \mathrm{Imp}_i(P^{(t)}) \sqrt{\log(4n/\delta)/(2S)}$ and $\rho_X := \max_{i\in[n]} \varepsilon_i/(\eta + \mathrm{Imp}_i(X))$. Define the range of first-order welfare scores by

$$H_X := \max_{t:\alpha_t>0} \sum_{i=1}^{n} \frac{\mathrm{Imp}_i(P^{(t)})}{\eta + \mathrm{Imp}_i(X)} - \min_{t:\alpha_t>0} \sum_{i=1}^{n} \frac{\mathrm{Imp}_i(P^{(t)})}{\eta + \mathrm{Imp}_i(X)}. \tag{3.7}$$

The next proposition shows that subsampling approximately preserves item impacts and controls the log-welfare change through a first-order fluctuation term and a second-order local error term.

Proposition 3.5 (Local welfare preservation under subsampling)**.** *If $\rho_X < 1$, then with probability at least $1-\delta$,*

$$|\,\mathrm{Imp}_i(\hat{X}_S) - \mathrm{Imp}_i(X)| \le \varepsilon_i \qquad \forall i \in [n], \tag{3.8}$$

and

$$|F_\eta(\hat{X}_S) - F_\eta(X)| \le H_X \sqrt{\frac{\log(4/\delta)}{2S}} + \frac{1}{2(1-\rho_X)^2} \sum_{i=1}^{n} \left(\frac{\varepsilon_i}{\eta + \mathrm{Imp}_i(X)} \right)^2. \tag{3.9}$$

The bound follows from Hoeffding's inequality and a second-order Taylor expansion. If $X = X^\star$ is optimal and the representation uses components on the exposed face from Theorem 3.2, then all active components have the same first-order welfare score, so $H_{X^\star} = 0$ and subsampling has only second-order local welfare error.

4 Deployment via Sparse Mixtures of Dual Variables

Section 3 established that an optimal policy admits an exact impact-preserving representation with at most n deterministic ranking tensors. However, each deterministic component $P^{(t)} \in \mathcal{P}^m$ still contains one ranking per training user. The required deployment storage therefore grows with the number of users m, and these components are defined only on the training users. We address both issues by changing what is stored. Instead of storing user-coupled deterministic components $P^{(t)}$, we store item-indexed dual vectors obtained from the dual formulation of the impact-based ranking problem.

4.1 Dual Problem and Subgradient Structure

We derive a dual form of problem (2.5) whose variables are indexed only by items. Introduce auxiliary variables $\boldsymbol{z} \in \mathbb{R}^n_{++}$ with constraints $z_i \le \eta + \mathrm{Imp}_i(X)$, and let $\lambda_i \ge 0$ be the Lagrange multiplier for the i-th constraint. Taking the Lagrangian dual and using Slater's condition gives the following equivalent dual problem. With $I_{\max} := \max_{X\in\mathcal{X}} \max_{i\in[n]} \mathrm{Imp}_i(X)$,

$$\max_{X\in\mathcal{X}} F_\eta(X) = \min_{\boldsymbol{\lambda}\in\left[\frac{1}{\eta+I_{\max}},\frac{1}{\eta}\right]^n} \left\{ G(\boldsymbol{\lambda}) := \eta \sum_{i=1}^n \lambda_i - n - \sum_{i=1}^n \log\lambda_i + \max_{Y\in\mathcal{X}} \sum_{i=1}^n \lambda_i \,\mathrm{Imp}_i(Y) \right\}. \tag{4.1}$$

The interval follows from the KKT condition $\lambda_i^\star = 1/(\eta + \mathrm{Imp}_i(X^\star))$ at a primal optimum. Appendix B.1 gives the full derivation. The key deployment point is that $\boldsymbol{\lambda}$ is item-indexed and user-independent.

Given $\boldsymbol{\lambda}$, define the inner maximizer

$$X(\boldsymbol{\lambda}) \in \arg\max_{Y\in\mathcal{X}} \sum_{i=1}^n \lambda_i \,\mathrm{Imp}_i(Y). \tag{4.2}$$

Since G is a smooth term plus a pointwise maximum, one subgradient has coordinates

$$g_i(\boldsymbol{\lambda}) := \eta + \mathrm{Imp}_i(X(\boldsymbol{\lambda})) - \frac{1}{\lambda_i}. \tag{4.3}$$

Computing $X(\boldsymbol{\lambda})$ is a user-wise ranking problem. Expanding the inner objective gives

$$\sum_{i=1}^n \lambda_i \,\mathrm{Imp}_i(Y) = \sum_{u=1}^m \sum_{i=1}^n \sum_{k=1}^n e(k)\lambda_i r(u,i) Y_{u,i,k}, \tag{4.4}$$

and the feasible set separates across users. For user u, the relevant score is $\sigma_{u,i}(\boldsymbol{\lambda}) := \lambda_i r(u,i)$, and an optimizer places the top-K scores in the displayed positions in decreasing order. Thus one dual vector $\boldsymbol{\lambda}$ defines a deterministic ranking rule for any user. At the dual optimum, $\lambda_i^\star = 1/(\eta + \mathrm{Imp}_i(X^\star))$ is large for items with low impact, so this rule moves such items up while $r(u,i)$ keeps the ranking personalized.

4.2 Mirror Descent and Sampled Dual Deployment

We propose a log-space mirror-descent method for solving the dual problem (4.1). Algorithm 1 gives the full training and subsampling procedure. At each iteration, the user-wise sorting oracle computes $X(\boldsymbol{\lambda}^{(t)})$ and hence the subgradient $\boldsymbol{g}^{(t)}$. Selecting and ordering the top-K scores takes $\mathcal{O}(n \log K)$ time per user, so one full subgradient step costs $\mathcal{O}(mn \log K)$, or $\mathcal{O}(bn \log K)$ with a user mini-batch of size $b \le m$.

Writing $\lambda_i = \exp(\nu_i)$ keeps dual variables positive and gives the log-space mirror-descent update [3] used in Algorithm 1. After T iterations, we weight each iterate by its normalized step size $w_t :=$

Algorithm 1 Dual mirror descent with trajectory subsampling

Require: Initial $\boldsymbol{\nu}^{(1)} \in \mathbb{R}^n$, horizon T, step sizes $\gamma_t > 0$ for all $t \in [T]$, sample size S

for $t \in [T]$ **do**
  $\boldsymbol{\lambda}^{(t)} \leftarrow \exp(\boldsymbol{\nu}^{(t)})$
  $X^{(t)} \leftarrow X(\boldsymbol{\lambda}^{(t)})$ via user-wise sorting
  $g_i^{(t)} \leftarrow \eta + \mathrm{Imp}_i(X^{(t)}) - \exp(-\nu_i^{(t)})$ for all $i \in [n]$
  if $t < T$ **then**
    $\boldsymbol{\nu}^{(t+1)} \leftarrow \boldsymbol{\nu}^{(t)} - \gamma_t \boldsymbol{g}^{(t)}$
  end if
end for
$w_t \leftarrow \gamma_t / \sum_{\tau=1}^{T} \gamma_\tau$ for all $t \in [T]$
for $s \in [S]$ **do**
  Sample $t_s \in [T]$ with $\Pr(t_s = t) = w_t$
end for
Output $\Gamma_S := \{(\boldsymbol{\lambda}^{(t_s)}, 1/S)\}_{s \in [S]}$

$\gamma_t / \sum_{\tau=1}^{T} \gamma_\tau$ and collect the weighted iterates into the dual trajectory

$$\mathcal{T}_T := \{(\boldsymbol{\lambda}^{(t)}, w_t)\}_{t=1}^{T}. \tag{4.5}$$

Convergence applies to the weighted ergodic mean $\bar{X}_T := \sum_{t=1}^{T} w_t X(\boldsymbol{\lambda}^{(t)})$, rather than to a single iterate. Appendix B.2 shows that weighted dual regret controls $F_\eta(X^\star) - F_\eta(\bar{X}_T)$, giving an $\mathcal{O}(1/\sqrt{T})$ rate for a constant step size $\gamma_t = \Theta(1/\sqrt{T})$ when the iterates stay in a bounded box.

The full trajectory $\mathcal{T}_T$ is useful for analysis. For serving, we retain only a sampled dual mixture. After forming $\mathcal{T}_T$, the procedure draws S dual vectors according to the weights w_t and stores

$$\Gamma_S := \{(\boldsymbol{\lambda}^{(t_s)}, 1/S)\}_{s \in [S]}. \tag{4.6}$$

Repeated indices may be merged by adding their weights. The induced policy is $\hat{X}_S := S^{-1} \sum_{s=1}^{S} X(\boldsymbol{\lambda}^{(t_s)})$. Proposition 3.5 applies to this weighted trajectory with components $X(\boldsymbol{\lambda}^{(t)})$, giving a local preservation bound for the induced impact objective. The full trajectory and affine-elimination compression are used as diagnostic variants in Section 5.

4.3 Deployment Cost and Unseen-User Generalization

The sampled dual mixture Γ_S stores at most S item-indexed dual vectors. More generally, let $\Gamma = \{(\boldsymbol{\lambda}^{(s)}, \omega_s)\}_{s=1}^{S'}$ be any weighted dual mixture with $\omega_s > 0$ and $\sum_s \omega_s = 1$. To serve either a training user or an unseen user, the system uses only Γ and that user's relevance vector $\boldsymbol{r}_u := (r(u,i))_{i \in [n]}$. It samples a component s and applies the same user-wise sorting rule as in the inner maximization, returning the top-K items under $\sigma_{u,i}(\boldsymbol{\lambda}^{(s)}) = \lambda_i^{(s)} r(u,i)$. Thus the fair-ranking layer stores no user-specific rankings, and applying the learned dual mixture to an unseen user requires neither retraining nor rerunning the dual optimizer.

Proposition 4.1 (Storage and serving cost)**.** *If Γ has S' components, it requires $\mathcal{O}(S'n)$ deployment storage and serves any user in $\mathcal{O}(n \log K)$ time. The storage bound is independent of m and T.*

We next ask whether a dual mixture learned on training users remains competitive on users not seen during optimization. Let U_{tr} denote the m training users and let U_{new} be a fresh i.i.d. batch from the same user distribution. Appendix C.2 shows that, both on U_{new} and on the enlarged platform $U_{\mathrm{all}} = U_{\mathrm{tr}} \sqcup U_{\mathrm{new}}$, the learned Γ falls short of the best dual mixture by at most its suboptimality on the training users plus statistical deviations. These deviations vanish as the populations grow, but the bound is conservative at the scale of our experiments, so Section 5 also evaluates this transfer empirically.

5 Experiments

Our experiments evaluate the deployment consequences of the compact policy representations developed above. We test whether compact dual mixtures preserve F_η while reducing deployment storage and serving users not seen during optimization. Compression ablations then diagnose how much of each learned trajectory is needed.

Setup. We evaluate on MovieLens-100K, MovieLens-1M [15], and Last.fm-15k from Last.fm-360K [7, 8]. Following Usunier et al. [33], we fit iALS [17] once and use the resulting dense scores as fixed relevance inputs. MovieLens-100K uses $K = 5$ to keep full doubly stochastic matrix optimization computationally tractable, while MovieLens-1M and Last.fm-15k use $K = 40$. We set the position exposure probabilities to $e(k) = 1/\log_2(k+1)$ and set $\eta = 10^{-8}$. We report F_η, retained support size, deployment storage, and optimization time. Storage counts only deployment data for the fair-ranking layer after relevance scores are available. The upstream scorer is treated as part of the base recommendation system. Values are averaged over five market seeds. For randomized subsampling methods, we use twenty repeats per seed. Full preprocessing, hyperparameters, implementation details, and hardware are in Appendix E.

Methods. Uniform serves the user-wise uniform ranking, and MaxRel serves the deterministic top-K relevance ranking. On MovieLens-100K, we also include two dense-matrix optimization references. These methods first optimize a full doubly stochastic ranking matrix for each user and then convert the resulting stochastic policy to deterministic rankings by Birkhoff–von Neumann (BvN) decomposition [5, 34]. MOSEK+BvN solves Problem (2.5) with the conic optimizer MOSEK [23], while Sinkhorn+BvN uses the Sinkhorn-based algorithm of Uehara et al. [32]. We restrict these BvN-based references to MovieLens-100K because storing dense policies and decomposing them user by user are costly at larger scale.

The large-scale iterative methods are OFFR [33], a Frank–Wolfe method for Problem (2.5), and our sampled dual deployment method in Algorithm 1. We report the latter as *Ours*. For our method, Dual denotes the full weighted trajectory $\mathcal{T}_T$, Dual-Sample(S) denotes the sampled dual mixture Γ_S, and Dual-Exact denotes affine-elimination compression of $\mathcal{T}_T$. The main deployment results use Dual-Sample(32). Our implementation uses an AdamW-style stabilized update in the same log-dual parameterization as Algorithm 1 for numerical robustness [20]. This changes only how the trajectory is generated, not the form of the returned dual mixture. For OFFR, the suffix *-Sample(S)* likewise denotes weighted size-S subsampling of its trajectory. Unless otherwise stated, full trajectories use $T = 2048$.

Deployment storage and quality. Panel A of Table 1 reports the main deployment comparison. On MovieLens-1M and Last.fm-15k, where OFFR is the scalable reference, Ours (Dual-Sample) has a rounded F_η shortfall of 0.00% while reducing storage by $2{,}160\times$ and $2{,}596\times$, respectively. On MovieLens-100K, where the conic reference is tractable, Ours (Dual-Sample) is within 0.16% of MOSEK+BvN in F_η and uses $1{,}044\times$ less storage. At the same budget $S = 32$, the item-indexed representation alone reduces storage by 3.0–$40.2\times$ relative to OFFR-Sample with nearly identical F_η. Appendix E.7 reports the full results, including MaxRel, Sinkhorn+BvN, OFFR-Sample, and fairness diagnostics.

Panel B isolates exact trajectory compression on MovieLens-100K for both OFFR and Dual. Affine elimination preserves the reported F_η while reducing support and storage, but it adds a non-negligible compression step. For deployment, the main comparison therefore uses randomized subsampling.

The storage reduction comes from changing what is stored. OFFR stores T top-K rankings for each of the m users, requiring TmK ranking entries. Ours (Dual-Sample) stores only S item-indexed dual vectors, requiring Sn entries.[2] To leading order, the storage used by OFFR relative to Ours (Dual-Sample) is $(T/S)(mK/n)$. The first factor, $T/S = 64$, comes from subsampling and the second from the item-indexed representation. The second factor matches the equal-support column of Table 1 and is large when $mK \gg n$, as on MovieLens-1M and Last.fm-15k. This vector representation reduces storage at the cost of an $\mathcal{O}(n \log K)$ top-K selection over the current user's relevance scores at serving time. In our benchmark, the 95th-percentile latency of Ours (Dual-Sample) is below 0.04 ms (Appendix E.7).

Subsample-size ablation. Figure 1(a) shows that randomized subsampling preserves the value of trained trajectories at small support. On MovieLens-100K, with retention normalized between Uniform (0%) and MOSEK+BvN (100%), both Dual-Sample and OFFR-Sample retain about 97% of the improvement at $S = 16$ and more than 99% at $S = 32$. The curves are nearly indistinguishable once $S \geq 32$, suggesting substantial redundancy in the impact-space behavior of the trained trajectories.

[2] For a fixed user set, the same dual mixture can instead be materialized as induced top-K lists using $\mathcal{O}(SmK)$ entries. We report the vector form because it serves unseen users without materializing user-specific rankings.

Table 1: Deployment storage and quality. Panel A compares Ours (Dual-Sample, $S = 32$) with the reference in each row, against which the end-to-end reduction and the F_η shortfall are measured. The equal-support reduction is measured against OFFR-Sample with the same $S = 32$ budget and numerical precision, and isolates the gain from storing item-indexed dual vectors instead of user-coupled rankings. Entries with $\pm$ give the mean and SD across five market seeds, and brackets give the mean within-seed SD over twenty subsamples. The across-seed SD of F_η mainly reflects differences between the sampled markets, which affect all methods alike. Panel B reports exact-compression diagnostics for OFFR and Dual on MovieLens-100K, including optimization (Opt.) and compression (Comp.) times. Storage counts only the fair-ranking layer's deployment storage. Arrows indicate preferred directions.

Panel A. Main deployment comparison

Dataset (Reference)	Storage (MB) ↓		Storage reduction ↑		F_η ↑		F_η shortfall ↓
	Ref.	Ours	End-to-end	Equal support	Ref.	Ours	
MovieLens-100K (MOSEK+BvN)	207.700	0.199 $\pm$0.001 [0.004]	1,044×	3.01× $\pm$0.01	2,416.9	2,413.1 $\pm$118.7 [0.85]	0.16%
MovieLens-1M (OFFR)	983.056	0.455 $\pm$0.003 [0.008]	2,160×	33.51× $\pm$0.14	10,829.3	10,829.3 $\pm$153.0 [0.008]	0.00%
Last.fm-15k (OFFR)	4,915.216	1.893 $\pm$0.011 [0.034]	2,596×	40.23× $\pm$0.17	49,213.1	49,213.3 $\pm$556.5 [0.014]	0.00%

Panel B. Exact-compression diagnostic on MovieLens-100K

Method	Support size	Storage (MB) ↓	F_η ↑	Opt. time (s) ↓	Comp. time (s) ↓
OFFR	2048	38.633	2,417.8	0.61	–
OFFR-Exact	1575	29.711	2,417.8	0.61	11.45
Ours (Dual)	2048	12.902	2,417.8	1.56	–
Ours (Dual-Exact)	1575	9.922	2,417.8	1.56	11.81

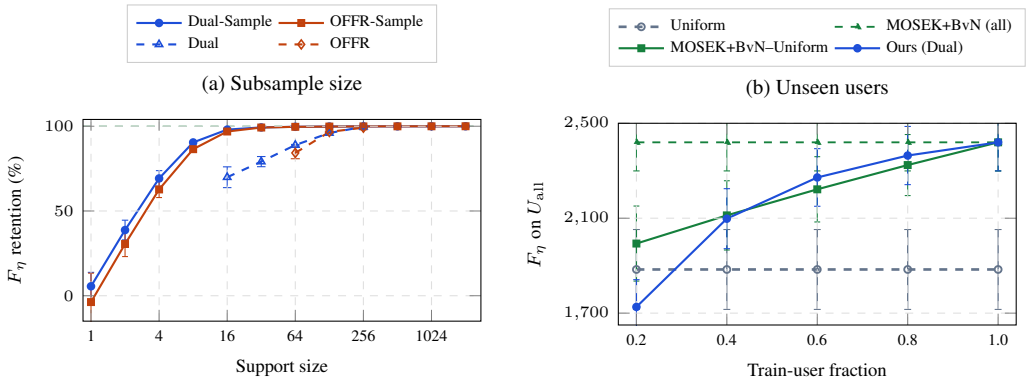


Figure 1: MovieLens-100K sweeps with $K = 5$. Panel (a) reports normalized F_η retention. Solid curves subsample trained trajectories, while dashed curves stop training at the same support size. Panel (b) evaluates policies learned on training subsets on the full user set. Error bars show $\pm 1\sigma$ over five market seeds.

At the same deployed support size, training a full trajectory and then subsampling it is more effective than stopping the optimizer after the same number of steps. The early-stopped curves lag behind the corresponding subsampled trajectories, especially at small support. Thus, the small-support loss appears to come mainly from optimization error in short trajectories rather than from compression of a trained mixture. This pattern suggests substantial impact-space redundancy in both Dual and OFFR trajectories. After optimization has explored the trajectory, a small subset of the optimized trajectory can recover most of the learned impact behavior.

Unseen-user generalization. We evaluate whether a dual mixture learned on a training subset can serve unseen users without rerunning optimization. On MovieLens-100K with $K = 5$, we fix U_{all}, train on nested subsets U_{tr} containing $20\%, 40\%, 60\%, 80\%$, or 100% of the users, and evaluate every served policy on U_{all}. A BvN mixture learned only on U_{tr} stores rankings only for those users, so we define MOSEK+BvN–Uniform as the executable transductive baseline that serves MOSEK+BvN on U_{tr} and Uniform on the remaining users. Ours (Dual) applies the same learned dual mixture to every user in U_{all}.

Figure 1(b) shows that the learned dual mixture remains useful on the enlarged platform once the training population is moderately large. Ours (Dual) is above Uniform from the 40% split onward and exceeds MOSEK+BvN–Uniform at the 60% and 80% splits. At the 20% split, the learned dual prices do not yet transfer reliably. The operational distinction is that MOSEK+BvN–Uniform remains tied to stored rankings for the training users, whereas the same dual mixture serves every user in U_{all}.

Appendix E.8 extends this evaluation to all three datasets with a 60/40 split and also reports results on users held out from optimization. On the enlarged platform, the learned dual mixture outperforms the corresponding transductive baseline on every dataset. On the held-out users alone, it retains most of the improvement on MovieLens-1M and Last.fm-15k but falls below Uniform on MovieLens-100K, whose held-out population is small.

6 Conclusion

We studied how to store and serve stochastic policies in impact-based fair ranking. The main structural message is that a policy need not be stored at the same scale at which it is optimized. Every optimum of the impact-based fair-ranking problem admits a mixture representation, with support no larger than the item set, that preserves the objective value, and affine elimination recovers such a representation from an optimal mixture. Yet this exact sparse representation still consists of user-coupled deterministic ranking tensors, so it does not by itself resolve deployment storage or unseen-user serving. Our dual-subsampling approach addresses this gap by replacing user-specific rankings with item-indexed dual vectors, yielding a compact deployment representation that applies to users not seen during optimization. Experiments on real-world recommendation datasets show that this representation preserves objective quality while reducing deployment storage by up to $2{,}596\times$ end to end. Subsampling contributes a factor of about 64 by removing redundancy in learned trajectories, and the item-indexed representation contributes a further factor of 3 to 40. These results suggest that fair-ranking algorithms should be designed together with their deployment representations. Future work includes developing optimization-based compressors that further reduce redundancy in learned trajectories and extending compact deployment representations to other objectives such as exposure-based fairness.

Limitations. Our analysis assumes a fixed item catalog and fixed relevance scores. The dual mixture must therefore be refreshed when the relevance model changes, and our guarantees do not extend to newly added items or to online adaptation. The gain from the item-indexed representation requires mK to be large relative to n, and transfer to new users depends on the dataset: with about 1.2 displayed positions per item for the held-out users of MovieLens-100K, it can fall below uniform ranking. Small-support sampling nearly preserves F_η but not necessarily other fairness diagnostics such as envy. Finally, serving a dual mixture adds a top-K selection over the relevance scores to each request.

Acknowledgments and Disclosure of Funding

This work was partially supported by JSPS KAKENHI Grant Number JP25K01447.

References

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction*, 30(1):127–158, 2020. doi: 10.1007/s11257-019-09256-1.

[2] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018. doi: 10.1080/23307706.2017.1397554.

[3] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. doi: 10.1016/S0167-6377(02)00231-6.

[4] 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 and Development in Information Retrieval*, pages 405–414, 2018. doi: 10.1145/3209978.3210063.

[5] Garrett Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumán, Rev. Ser. A*, 5: 147–151, 1946.

[6] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018. URL `https://github.com/jax-ml/jax`. Version 0.3.13.

[7] Òscar Celma. Last.fm music recommendation dataset, 2010. URL `https://doi.org/10.5281/zenodo.6090214`. Version 1.2.

[8] Òscar Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, Berlin, Heidelberg, 2010. doi: 10.1007/978-3-642-13287-2.

[9] Qinyi Chen, Jason Cheuk Nam Liang, Negin Golrezaei, and Djallel Bouneffouf. Interpolating item and user fairness in multi-sided recommendations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1589. URL `https://proceedings.neurips.cc/paper_files/paper/2024/hash/59d4e18a60490b9ed9913f3be2b14839-Abstract-Conference.html`.

[10] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016. URL `https://www.jmlr.org/papers/v17/15-408.html`.

[11] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. Two-sided fairness in rankings via Lorenz dominance. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/48259990138bc03361556fb3f94c5d45-Abstract.html`.

[12] Fanny Dufossé and Bora Uçar. Notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108–115, 2016. doi: 10.1016/j.laa.2016.02.023.

[13] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006. doi: 10.1145/1147954.1147956.

[14] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to LinkedIn talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2221–2231, 2019. doi: 10.1145/3292500.3330691.

[15] F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015. doi: 10.1145/2827872.

[16] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant,

Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[17] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008. doi: 10.1109/ICDM.2008.22.

[18] Till Kletti, Jean-Michel Renders, and Patrick Loiseau. Pareto-optimal fairness-utility amortizations in rankings with a DBN exposure model. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 748–758, 2022. doi: 10.1145/3477495.3532036.

[19] Till Kletti, Jean-Michel Renders, and Patrick Loiseau. Introducing the Expohedron for efficient Pareto-optimal fairness-utility amortizations in repeated rankings. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 498–507, 2022. doi: 10.1145/3488560.3498490.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[21] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, Cambridge, MA, second edition, 2018. doi: 10.7551/mitpress/10290.001.0001.

[22] 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*, pages 429–438, 2020. doi: 10.1145/3397271.3401100.

[23] MOSEK ApS. *The MOSEK Optimizer API for Python Manual, Version 11.0.30*, 2025. URL `https://docs.mosek.com/11.0/pythonapi/index.html`.

[24] Mohammadmehdi Naghiaei, Hossein A. Rahmani, and Yashar Deldjoo. CPFair: Personalized consumer and producer fairness re-ranking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 770–779, 2022. doi: 10.1145/3477495.3531959.

[25] 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*, pages 1194–1204, 2020. doi: 10.1145/3366423.3380196.

[26] Gourab K. Patro, Lorenzo Porcaro, Laura Mitchell, Qiuyue Zhang, Meike Zehlike, and Nikhil Garg. Fair ranking: A critical review, challenges, and future directions. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1929–1942, 2022. doi: 10.1145/3531146.3533238.

[27] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: An overview. *The VLDB Journal*, 31(3):431–458, 2022. doi: 10.1007/s00778-021-00697-y.

[28] Yuta Saito and Thorsten Joachims. Fair ranking as fair division: Impact-based individual fairness in ranking. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1514–1524, 2022. doi: 10.1145/3534678.3539353.

[29] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2219–2228, 2018. doi: 10.1145/3219819.3220088.

[30] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. In *Advances in Neural Information Processing Systems*, volume 32, pages 5427–5437, 2019.

[31] Riku Togashi, Kenshi Abe, and Yuta Saito. Scalable and provably fair exposure control for large-scale recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3307–3318, 2024. doi: 10.1145/3589334.3645390.

[32] Yuki Uehara, Shunnosuke Ikeda, Naoki Nishimura, Koya Ohashi, Yilin Li, Jie Yang, Deddy Jobson, Xingxia Zha, Takeshi Matsumoto, Noriyoshi Sukegawa, and Yuichi Takano. Fast solution to the fair ranking problem using the Sinkhorn algorithm. In *PRICAI 2024: Trends in Artificial Intelligence*, volume 15285 of *Lecture Notes in Computer Science*, pages 207–215. Springer Nature Singapore, 2025. doi: 10.1007/978-981-96-0128-8_18.

[33] Nicolas Usunier, Virginie Do, and Elvis Dohmatob. Fast online ranking with fairness of exposure. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2157–2167, 2022. doi: 10.1145/3531146.3534633.

[34] John von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games (AM-28), Volume II*, pages 5–12. Princeton University Press, Princeton, 1953. doi: 10.1515/9781400881970-002.

[35] Meike Zehlike and Carlos Castillo. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*, pages 2849–2855, 2020. doi: 10.1145/3366424.3380048.

[36] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. FA*IR: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578, 2017. doi: 10.1145/3132847.3132938.

A Proofs for Stochastic Policy Compression

This appendix proves the structural and approximation results from Section 3. The arguments use only the linearity of item impacts and elementary convex geometry.

A.1 Impact-polytope reduction

Proof of Proposition 3.1. For every $X \in \mathcal{X}$, the mixture representation (2.6) gives $X = \sum_t \alpha_t P^{(t)}$ with $P^{(t)} \in \mathcal{P}^m$. Since each Imp_i is linear in X,

$$\boldsymbol{I}(X) = \sum_t \alpha_t \boldsymbol{I}(P^{(t)}) \in \mathcal{K}. \tag{A.1}$$

Conversely, any point $\boldsymbol{y} \in \mathcal{K}$ can be written as $\boldsymbol{y} = \sum_t \alpha_t \boldsymbol{I}(P^{(t)})$. The policy $X = \sum_t \alpha_t P^{(t)}$ belongs to $\mathcal{X}$ and satisfies $\boldsymbol{I}(X) = \boldsymbol{y}$. Therefore

$$\{\boldsymbol{I}(X) : X \in \mathcal{X}\} = \mathcal{K}. \tag{A.2}$$

Because $F_\eta(X) = f(\boldsymbol{I}(X))$, the two optimization problems have the same feasible objective values and the same optimum. □

A.2 Exact sparse optimal representation

Proof of Theorem 3.2. Let $X^\star$ be an optimal policy and set $\boldsymbol{y}^\star := \boldsymbol{I}(X^\star)$. By Proposition 3.1, $\boldsymbol{y}^\star$ maximizes

$$f(\boldsymbol{y}) = \sum_{i=1}^{n} \log(\eta + y_i) \tag{A.3}$$

over $\mathcal{K}$.

The function f is differentiable and concave on a neighborhood of $\mathcal{K}$. Hence the first-order optimality condition gives

$$\nabla f(\boldsymbol{y}^\star)^\top (\boldsymbol{y} - \boldsymbol{y}^\star) \le 0 \qquad \forall \boldsymbol{y} \in \mathcal{K}. \tag{A.4}$$

Write

$$\boldsymbol{w}^\star := \nabla f(\boldsymbol{y}^\star), \qquad w_i^\star = \frac{1}{\eta + y_i^\star}. \tag{A.5}$$

Since $\eta > 0$ and $y_i^\star \geq 0$, every $w_i^\star$ is positive. In particular, $\boldsymbol{w}^\star \neq \mathbf{0}$. Equation (A.4) implies

$$(\boldsymbol{w}^\star)^\top \boldsymbol{y} \leq (\boldsymbol{w}^\star)^\top \boldsymbol{y}^\star \qquad \forall \boldsymbol{y} \in \mathcal{K}. \tag{A.6}$$

Thus $\boldsymbol{y}^\star$ lies on the exposed face

$$\mathcal{F} := \left\{ \boldsymbol{y} \in \mathcal{K} \,\middle|\, (\boldsymbol{w}^\star)^\top \boldsymbol{y} = (\boldsymbol{w}^\star)^\top \boldsymbol{y}^\star \right\}. \tag{A.7}$$

This face is contained in a proper hyperplane of $\mathbb{R}^n$, so

$$\dim \operatorname{aff}(\mathcal{F}) \leq n - 1. \tag{A.8}$$

The face $\mathcal{F}$ is the convex hull of the deterministic impact vectors that lie on it. To see this, take any $\boldsymbol{z} \in \mathcal{F}$ and write

$$\boldsymbol{z} = \sum_t \alpha_t \boldsymbol{I}(Q^{(t)}), \qquad Q^{(t)} \in \mathcal{P}^m, \quad \alpha_t \geq 0, \quad \sum_t \alpha_t = 1. \tag{A.9}$$

By (A.6), each deterministic impact vector satisfies

$$(\boldsymbol{w}^\star)^\top \boldsymbol{I}(Q^{(t)}) \leq (\boldsymbol{w}^\star)^\top \boldsymbol{y}^\star. \tag{A.10}$$

The convex combination has equality because $\boldsymbol{z} \in \mathcal{F}$. Therefore every term with positive weight must also have equality and lies in $\mathcal{F}$.

Applying Carathéodory's theorem inside $\operatorname{aff}(\mathcal{F})$ and using (A.8), there exist $P^{(1)}, \ldots, P^{(S)} \in \mathcal{P}^m$ and weights $\beta_1, \ldots, \beta_S$ such that

$$S \leq n, \qquad \beta_s \geq 0, \qquad \sum_{s=1}^{S} \beta_s = 1, \qquad \boldsymbol{y}^\star = \sum_{s=1}^{S} \beta_s \boldsymbol{I}(P^{(s)}). \tag{A.11}$$

Define

$$X^{\mathrm{cmp}} := \sum_{s=1}^{S} \beta_s P^{(s)}. \tag{A.12}$$

Then $X^{\mathrm{cmp}} \in \mathcal{X}$ and $\boldsymbol{I}(X^{\mathrm{cmp}}) = \boldsymbol{y}^\star$ by linearity. Hence $F_\eta(X^{\mathrm{cmp}}) = f(\boldsymbol{y}^\star)$, so X^{cmp} is optimal and has the claimed support size. □

A.3 Why the support bound is n rather than $n + 1$

Carathéodory's theorem in $\mathbb{R}^n$ would give $n + 1$ points for a generic point in $\mathcal{K}$. The improvement to n comes from optimality. The gradient $\nabla f(\boldsymbol{y}^\star)$ exposes a face containing the optimum, and that face lies in a hyperplane of dimension at most $n - 1$. Applying Carathéodory's theorem in this lower-dimensional affine hull yields $(n - 1) + 1 = n$ points.

A.4 Tightness of the support bound

Proof of Proposition 3.3. Take $m = 1$, $K = 1$, $e(1) = 1$, and $e(k) = 0$ for $k \geq 2$. Let $r(1, i) = 1$ for all items. A deterministic ranking is identified by the item placed in the first position. If item i is placed first, the resulting impact vector is the standard basis vector $\boldsymbol{e}_i \in \mathbb{R}^n$. Hence

$$\mathcal{K} = \operatorname{conv}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\} = \left\{ \boldsymbol{y} \in \mathbb{R}^n_+ \,\middle|\, \sum_{i=1}^{n} y_i = 1 \right\}. \tag{A.13}$$

The objective f in (3.2) is strictly concave and symmetric on this simplex. Its unique maximizer is

$$\boldsymbol{y}^\star = \frac{1}{n}\mathbf{1}. \tag{A.14}$$

The only convex representation of $\boldsymbol{y}^\star$ using the simplex vertices assigns weight $1/n$ to every vertex. Therefore every optimal mixture representation requires all n deterministic components in this instance. □

Algorithm 2 Affine elimination for impact-preserving compression

Require: Optimal mixture $X^\star = \sum_{t=1}^{T} \alpha_t P^{(t)}$ with $P^{(t)} \in \mathcal{P}^m$, $\alpha_t \geq 0$, and $\sum_{t=1}^{T} \alpha_t = 1$
Require: Impact vectors $\boldsymbol{v}^{(t)} := \boldsymbol{I}(P^{(t)})$ for $t \in [T]$
$\mathcal{S} \leftarrow \{t : \alpha_t > 0\}$ and $\beta_t \leftarrow \alpha_t$ for $t \in \mathcal{S}$
while $|\mathcal{S}| > n$ **do**
  Find nonzero $\{a_t\}_{t\in\mathcal{S}}$ with $\sum_{t\in\mathcal{S}} a_t = 0$ and $\sum_{t\in\mathcal{S}} a_t \boldsymbol{v}^{(t)} = \mathbf{0}$
  $\theta \leftarrow \min_{t\in\mathcal{S}:\, a_t>0} \beta_t / a_t$
  $\beta_t \leftarrow \beta_t - \theta a_t$ for all $t \in \mathcal{S}$
  $\mathcal{S} \leftarrow \{t \in \mathcal{S} : \beta_t > 0\}$
end while
Output $\mathcal{S}$ and $\{\beta_t\}_{t\in\mathcal{S}}$

A.5 Correctness of affine elimination

We first state the one-step invariant used by Algorithm 2.

Lemma A.1 (One-point elimination)**.** *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J$ lie in an affine space of dimension D. Suppose positive weights $\beta_1, \ldots, \beta_J$ satisfy*

$$\sum_{r=1}^{J} \beta_r = 1, \qquad \boldsymbol{v} = \sum_{r=1}^{J} \beta_r \boldsymbol{v}_r. \tag{A.15}$$

If $J > D + 1$, then $\boldsymbol{v}$ has a convex representation using at most $J - 1$ of these points.

Proof. Since $J > D + 1$, the points are affinely dependent. There exists a nonzero vector $\boldsymbol{a} = (a_1, \ldots, a_J)$ such that

$$\sum_{r=1}^{J} a_r = 0, \qquad \sum_{r=1}^{J} a_r \boldsymbol{v}_r = \mathbf{0}. \tag{A.16}$$

The set $I_+ := \{r : a_r > 0\}$ is nonempty. Otherwise the first equality would force $\boldsymbol{a} = \mathbf{0}$. Define

$$\theta := \min_{r\in I_+} \frac{\beta_r}{a_r}, \qquad \beta'_r := \beta_r - \theta a_r. \tag{A.17}$$

For $r \in I_+$, $\beta'_r \geq 0$ by the definition of θ. For $r \notin I_+$, $\beta'_r \geq \beta_r > 0$. At least one coefficient becomes zero. Moreover,

$$\sum_{r=1}^{J} \beta'_r = \sum_{r=1}^{J} \beta_r - \theta \sum_{r=1}^{J} a_r = 1, \tag{A.18}$$

and

$$\sum_{r=1}^{J} \beta'_r \boldsymbol{v}_r = \sum_{r=1}^{J} \beta_r \boldsymbol{v}_r - \theta \sum_{r=1}^{J} a_r \boldsymbol{v}_r = \boldsymbol{v}. \tag{A.19}$$

Removing a zero-weight point gives the claim. □

Proof of Proposition 3.4. Let $\boldsymbol{y}^\star = \boldsymbol{I}(X^\star)$. By the proof of Theorem 3.2, every active impact vector of an optimal mixture lies in the exposed face $\mathcal{F}$ containing $\boldsymbol{y}^\star$, whose affine dimension is at most $n - 1$. Therefore, whenever Algorithm 2 has more than n active components, those impact vectors are affinely dependent and the requested coefficients $\{a_t\}$ exist.

Each update preserves total weight and the represented impact vector:

$$\sum_{t\in\mathcal{S}} (\beta_t - \theta a_t) = \sum_{t\in\mathcal{S}} \beta_t, \tag{A.20}$$

$$\sum_{t\in\mathcal{S}} (\beta_t - \theta a_t) \boldsymbol{v}^{(t)} = \sum_{t\in\mathcal{S}} \beta_t \boldsymbol{v}^{(t)}. \tag{A.21}$$

The choice of θ keeps all weights nonnegative and makes at least one active weight zero. The support therefore decreases at each iteration, and the algorithm terminates with at most n active points. Since the final mixture has the same item-impact vector as the input optimal mixture, it has the same objective value. □

A.6 Weighted random subsampling

Proof of Proposition 3.5. Write

$$\mu_i := \mathrm{Imp}_i(X), \qquad \hat{\mu}_i := \mathrm{Imp}_i(\hat{X}_S), \qquad a_i := \eta + \mu_i. \tag{A.22}$$

For each active component, let $v_i^{(t)} := \mathrm{Imp}_i(P^{(t)})$. Then $\hat{\mu}_i = S^{-1}\sum_{s=1}^{S} v_i^{(t_s)}$ and $\mathbb{E}[\hat{\mu}_i] = \mu_i$. We first prove the itemwise concentration event. For a fixed item i, let

$$B_i := \max_{t:\alpha_t>0} v_i^{(t)}. \tag{A.23}$$

If $B_i = 0$, then $\hat{\mu}_i = \mu_i = 0$. Otherwise, Hoeffding's inequality gives

$$\Pr[|\hat{\mu}_i - \mu_i| > \varepsilon_i] \le 2\exp\left(-\frac{2S\varepsilon_i^2}{B_i^2}\right) = \frac{\delta}{2n}. \tag{A.24}$$

A union bound over items shows that $|\hat{\mu}_i - \mu_i| \le \varepsilon_i$ for all $i \in [n]$ with probability at least $1 - \delta/2$. We next control the linearized welfare term. Define

$$Y_s := \sum_{i=1}^{n} \frac{v_i^{(t_s)}}{a_i}. \tag{A.25}$$

By the definition of H_X, the random variables Y_s lie in an interval of length H_X. Moreover,

$$\frac{1}{S}\sum_{s=1}^{S} Y_s - \mathbb{E}[Y_s] = \sum_{i=1}^{n} \frac{\hat{\mu}_i - \mu_i}{a_i}. \tag{A.26}$$

If $H_X = 0$, the left-hand side below is zero. Otherwise, Hoeffding's inequality gives

$$\left|\sum_{i=1}^{n} \frac{\hat{\mu}_i - \mu_i}{a_i}\right| \le H_X\sqrt{\frac{\log(4/\delta)}{2S}} \tag{A.27}$$

with probability at least $1 - \delta/2$. Combining the two events gives probability at least $1 - \delta$. On this event, set

$$u_i := \frac{\hat{\mu}_i - \mu_i}{a_i}. \tag{A.28}$$

The itemwise concentration bound and the assumption $\rho_X < 1$ imply $|u_i| \le \rho_X < 1$. Taylor's theorem gives

$$\log(1 + u_i) = u_i + R_i, \qquad |R_i| \le \frac{u_i^2}{2(1-\rho_X)^2}. \tag{A.29}$$

Since

$$F_\eta(\hat{X}_S) - F_\eta(X) = \sum_{i=1}^{n} \log(1 + u_i), \tag{A.30}$$

we obtain

$$|F_\eta(\hat{X}_S) - F_\eta(X)| \le \left|\sum_{i=1}^{n} u_i\right| + \frac{1}{2(1-\rho_X)^2}\sum_{i=1}^{n} u_i^2 \tag{A.31}$$

$$\le H_X\sqrt{\frac{\log(4/\delta)}{2S}} + \frac{1}{2(1-\rho_X)^2}\sum_{i=1}^{n}\left(\frac{\varepsilon_i}{\eta + \mathrm{Imp}_i(X)}\right)^2. \tag{A.32}$$

This proves the proposition. □

A.7 Learned approximate mixtures

The subsampling guarantee does not require the input mixture to be optimal. This subsection records the form used when an optimizer returns an approximate mixture.

Let

$$\widetilde{X}_T = \sum_{t=1}^{T} \alpha_t P^{(t)} \tag{A.33}$$

be any feasible mixture, where $P^{(t)} \in \mathcal{P}^m$, $\alpha_t \geq 0$ for all $t \in [T]$, and $\sum_{t=1}^{T} \alpha_t = 1$. Draw indices $t_s \in [T]$ independently for $s \in [S]$ with $\Pr(t_s = t) = \alpha_t$, and set $\hat{X}_S = S^{-1} \sum_{s=1}^{S} P^{(t_s)}$. Let $\varepsilon_i(\widetilde{X}_T)$, $\rho_{\widetilde{X}_T}$, and $H_{\widetilde{X}_T}$ be the quantities from Section 3.4 with X replaced by $\widetilde{X}_T$. When $\rho_{\widetilde{X}_T} < 1$, define

$$\mathcal{E}_{\mathrm{loc}}(\delta, \widetilde{X}_T) := H_{\widetilde{X}_T} \sqrt{\frac{\log(4/\delta)}{2S}} + \frac{1}{2(1-\rho_{\widetilde{X}_T})^2} \sum_{i=1}^{n} \left(\frac{\varepsilon_i(\widetilde{X}_T)}{\eta + \mathrm{Imp}_i(\widetilde{X}_T)} \right)^2. \tag{A.34}$$

Corollary A.2 (Compression of learned approximate mixtures)**.** *If $X^\star$ is optimal and*

$$F_\eta(X^\star) - F_\eta(\widetilde{X}_T) \leq \epsilon_T, \tag{A.35}$$

and if $\rho_{\widetilde{X}_T} < 1$, then with probability at least $1 - \delta$ over the subsampling,

$$F_\eta(X^\star) - F_\eta(\hat{X}_S) \leq \epsilon_T + \mathcal{E}_{\mathrm{loc}}(\delta, \widetilde{X}_T). \tag{A.36}$$

Proof. Proposition 3.5 gives, with probability at least $1 - \delta$ over the subsampling,

$$|F_\eta(\hat{X}_S) - F_\eta(\widetilde{X}_T)| \leq \mathcal{E}_{\mathrm{loc}}(\delta, \widetilde{X}_T) \tag{A.37}$$

where the quantities in the proposition are evaluated at $\widetilde{X}_T$. On this event,

$$F_\eta(X^\star) - F_\eta(\hat{X}_S) \leq F_\eta(X^\star) - F_\eta(\widetilde{X}_T) + |F_\eta(\widetilde{X}_T) - F_\eta(\hat{X}_S)| \tag{A.38}$$

$$\leq \epsilon_T + \mathcal{E}_{\mathrm{loc}}(\delta, \widetilde{X}_T). \tag{A.39}$$

□

If $\widetilde{X}_T$ is random and its optimization gap bound holds with probability at least $1 - \delta_T$, combining that event with the subsampling event gives the same conclusion with probability at least $1 - \delta_T - \delta$.

B Proofs for Dual Deployment

This appendix gives the derivations behind the dual deployment method in Section 4. The first subsection derives the dual objective, its effective domain, and the subgradient used by Algorithm 1. The second subsection proves the optimality conditions and the ergodic averaging statement. The final subsection verifies the compressed deployment representation and its serving cost.

We use the following notation throughout this appendix. Write

$$\boldsymbol{I}(X) := (\mathrm{Imp}_1(X), \ldots, \mathrm{Imp}_n(X)) \in \mathbb{R}^n, \qquad h(\boldsymbol{\lambda}) := \eta \sum_{i=1}^{n} \lambda_i - n - \sum_{i=1}^{n} \log \lambda_i, \tag{B.1}$$

and define

$$\Psi(X, \boldsymbol{\lambda}) := h(\boldsymbol{\lambda}) + \sum_{i=1}^{n} \lambda_i \, \mathrm{Imp}_i(X), \tag{B.2}$$

$$\mathcal{V}(\boldsymbol{\lambda}) := \max_{X \in \mathcal{X}} \sum_{i=1}^{n} \lambda_i \, \mathrm{Imp}_i(X). \tag{B.3}$$

B.1 Dual objective, domain, and subgradient

The derivation exposes two deployment-relevant facts: the optimizer has an item-indexed dual vector, and one subgradient evaluation reduces to a user-wise ranking problem.

Derivation of the dual objective and subgradient. Introduce auxiliary variables $\boldsymbol{z} = (z_1, \dots, z_n)$ and consider

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \log z_i \\ \text{subject to} \quad & z_i \le \eta + \mathrm{Imp}_i(X) \ \ \forall i \in [n], \\ & X \in \mathcal{X},\ \boldsymbol{z} \in \mathbb{R}^n_{++}. \end{aligned} \tag{B.4}$$

This is equivalent to the original smoothed NSW problem. For a fixed feasible X, each constraint is tight at optimum because $\log z_i$ is strictly increasing in z_i. Therefore the optimal value of (B.4) is exactly

$$\max_{X \in \mathcal{X}} F_\eta(X). \tag{B.5}$$

Now form the Lagrangian

$$L(X, \boldsymbol{z}, \boldsymbol{\lambda}) := \sum_{i=1}^{n} \log z_i - \sum_{i=1}^{n} \lambda_i \big(z_i - \eta - \mathrm{Imp}_i(X)\big), \qquad \boldsymbol{\lambda} \in \mathbb{R}^n_+. \tag{B.6}$$

The problem satisfies Slater's condition. Indeed, for any $X \in \mathcal{X}$, choosing $z_i = \eta/2$ gives

$$z_i - \eta - \mathrm{Imp}_i(X) \le -\eta/2 < 0, \tag{B.7}$$

where the inequality uses $\mathrm{Imp}_i(X) \ge 0$. Hence strong duality holds for (B.4).

For a fixed $\boldsymbol{\lambda} \in \mathbb{R}^n_+$, the dual function is

$$q(\boldsymbol{\lambda}) := \max_{X \in \mathcal{X},\ \boldsymbol{z} \in \mathbb{R}^n_{++}} L(X, \boldsymbol{z}, \boldsymbol{\lambda}) \tag{B.8}$$

$$= \eta \sum_{i=1}^{n} \lambda_i + \max_{X \in \mathcal{X}} \sum_{i=1}^{n} \lambda_i \,\mathrm{Imp}_i(X) + \sum_{i=1}^{n} \max_{z_i > 0} \big(\log z_i - \lambda_i z_i\big), \tag{B.9}$$

because the maximization separates between X and the coordinates of $\boldsymbol{z}$. For each coordinate,

$$\frac{d}{dz_i}\big(\log z_i - \lambda_i z_i\big) = \frac{1}{z_i} - \lambda_i. \tag{B.10}$$

Thus, when $\lambda_i > 0$, the maximizer is $z_i^\star(\boldsymbol{\lambda}) = 1/\lambda_i$. Therefore

$$\max_{z_i > 0} \big(\log z_i - \lambda_i z_i\big) = -1 - \log \lambda_i \qquad \text{for } \lambda_i > 0, \tag{B.11}$$

while the supremum is $+\infty$ if $\lambda_i = 0$. Hence the effective dual domain is $\mathbb{R}^n_{++}$, and

$$G(\boldsymbol{\lambda}) := q(\boldsymbol{\lambda}) = \eta \sum_{i=1}^{n} \lambda_i - n - \sum_{i=1}^{n} \log \lambda_i + \max_{X \in \mathcal{X}} \sum_{i=1}^{n} \lambda_i \,\mathrm{Imp}_i(X) \qquad (\boldsymbol{\lambda} \in \mathbb{R}^n_{++}). \tag{B.12}$$

Strong duality now gives

$$\max_{X \in \mathcal{X}} F_\eta(X) = \min_{\boldsymbol{\lambda} \in \mathbb{R}^n_{++}} G(\boldsymbol{\lambda}). \tag{B.13}$$

With the notation above, $G(\boldsymbol{\lambda}) = \max_{X \in \mathcal{X}} \Psi(X, \boldsymbol{\lambda})$. Proposition B.2 below shows that every dual optimum satisfies

$$\lambda_i^\star = \frac{1}{\eta + \mathrm{Imp}_i(X^\star)} \qquad \text{for all } i \in [n]. \tag{B.14}$$

Because $\mathrm{Imp}_i(X^\star) \in [0, I_{\max}]$, every dual optimum lies in $\Lambda = [(\eta + I_{\max})^{-1}, \eta^{-1}]^n$. Hence the dual problem can be written over Λ exactly as in Section 4.

As a pointwise maximum of affine functions, $\mathcal{V}$ is convex. If

$$X(\boldsymbol{\lambda}) \in \arg\max_{X \in \mathcal{X}} \sum_{i=1}^{n} \lambda_i \operatorname{Imp}_i(X), \tag{B.15}$$

then for any $\boldsymbol{\lambda}' \in \mathbb{R}^n_{++}$,

$$\mathcal{V}(\boldsymbol{\lambda}') \geq \sum_{i=1}^{n} \lambda'_i \operatorname{Imp}_i(X(\boldsymbol{\lambda})) \tag{B.16}$$

$$= \mathcal{V}(\boldsymbol{\lambda}) + \sum_{i=1}^{n} (\lambda'_i - \lambda_i) \operatorname{Imp}_i(X(\boldsymbol{\lambda})). \tag{B.17}$$

The equality uses the optimality of $X(\boldsymbol{\lambda})$ in the definition of $\mathcal{V}(\boldsymbol{\lambda})$. This is the subgradient inequality for $\mathcal{V}$, so $\boldsymbol{I}(X(\boldsymbol{\lambda})) \in \partial \mathcal{V}(\boldsymbol{\lambda})$. Adding the smooth part

$$\eta \sum_i \lambda_i - n - \sum_i \log \lambda_i \tag{B.18}$$

shows that a subgradient of G is

$$g_i(\boldsymbol{\lambda}) = \eta + \operatorname{Imp}_i(X(\boldsymbol{\lambda})) - \frac{1}{\lambda_i}. \tag{B.19}$$

□

The next lemma justifies the sorting primitive used in Algorithm 1.

Lemma B.1 (User-wise form of the inner maximizer)**.** *For any $\boldsymbol{\lambda} \in \mathbb{R}^n_{++}$, an optimizer in*

$$X(\boldsymbol{\lambda}) \in \arg\max_{X \in \mathcal{X}} \sum_{i=1}^{n} \lambda_i \operatorname{Imp}_i(X) \tag{B.20}$$

is obtained independently for each user by sorting items in descending order of $\lambda_i r(u,i)$ and assigning the largest scores to the largest examination weights $e(k)$.

The proof uses separability across users and the rearrangement inequality.

Proof. Expanding the objective gives

$$\sum_{i=1}^{n} \lambda_i \operatorname{Imp}_i(X) = \sum_{u=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{n} e(k) \lambda_i r(u,i) X_{u,i,k}. \tag{B.21}$$

The feasible set factors as a product over users, so the maximization separates over the slices X_u. For a fixed user u, the objective is linear over the Birkhoff polytope $\operatorname{conv}(\mathcal{P})$, hence it has an optimizer at a permutation matrix. Since $e(1) \geq \cdots \geq e(K) > 0$ and $e(k) = 0$ for $k > K$, the rearrangement inequality assigns the items with the largest values of $\lambda_i r(u,i)$ to the displayed positions in descending order of examination weight. Ties may be broken arbitrarily. □

B.2 Optimality conditions, ergodic averaging, and log-space mirror descent

We first isolate the optimality conditions of the relaxed problem. The main consequence is that the auxiliary variables are tight at optimum, which directly recovers the dual-optimal formula used in Section 4.

Proposition B.2 (KKT characterization of primal-dual optimality)**.** *A triple $(X^\star, \boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ is primal-dual optimal for* (B.4) *if and only if it satisfies*

$$X^\star \in \mathcal{X}, \qquad \boldsymbol{z}^\star \in \mathbb{R}^n_{++}, \qquad z_i^\star \leq \eta + \operatorname{Imp}_i(X^\star) \qquad \text{for all } i \in [n], \tag{B.22}$$

$$X^\star \in \arg\max_{X \in \mathcal{X}} \sum_{i=1}^{n} \lambda_i^\star \operatorname{Imp}_i(X), \tag{B.23}$$

$$\lambda_i^\star = \frac{1}{z_i^\star} \qquad \text{for all } i \in [n], \tag{B.24}$$

$$\lambda_i^\star \big(z_i^\star - \eta - \operatorname{Imp}_i(X^\star)\big) = 0 \qquad \text{for all } i \in [n]. \tag{B.25}$$

Consequently,

$$z_i^\star = \eta + \mathrm{Imp}_i(X^\star), \qquad \lambda_i^\star = \frac{1}{\eta + \mathrm{Imp}_i(X^\star)} \qquad \textit{for all } i \in [n]. \tag{B.26}$$

The proof is a direct KKT verification for the relaxed concave program.

Proof. Because (B.4) is a concave maximization problem with affine constraints and Slater's condition holds, the KKT conditions are necessary and sufficient. The first line is primal feasibility. For fixed $(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$, the part of the Lagrangian that depends on X is $\sum_i \lambda_i^\star \mathrm{Imp}_i(X)$, which gives the maximization condition in the second line. Stationarity in each z_i gives

$$\frac{\partial}{\partial z_i} L(X^\star, \boldsymbol{z}^\star, \boldsymbol{\lambda}^\star) = \frac{1}{z_i^\star} - \lambda_i^\star = 0. \tag{B.27}$$

Hence $\lambda_i^\star = 1/z_i^\star > 0$ for every i. Complementary slackness then forces each constraint to be tight:

$$z_i^\star = \eta + \mathrm{Imp}_i(X^\star). \tag{B.28}$$

Substituting this equality into $\lambda_i^\star = 1/z_i^\star$ gives

$$\lambda_i^\star = \frac{1}{\eta + \mathrm{Imp}_i(X^\star)} \qquad \text{for all } i \in [n]. \tag{B.29}$$

□

For any sequence $\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(T)} \in \mathbb{R}^n_{++}$, set $X^{(t)} := X(\boldsymbol{\lambda}^{(t)})$ and choose the subgradient $\boldsymbol{g}^{(t)} \in \partial G(\boldsymbol{\lambda}^{(t)})$ with coordinates

$$g_i^{(t)} = \eta + \mathrm{Imp}_i(X^{(t)}) - \frac{1}{\lambda_i^{(t)}}. \tag{B.30}$$

For step sizes $\gamma_t > 0$ for all $t \in [T]$, define

$$A_T := \sum_{t=1}^T \gamma_t, \qquad \bar{X}_T := \frac{1}{A_T} \sum_{t=1}^T \gamma_t X^{(t)}. \tag{B.31}$$

The next proposition explains why $\bar{X}_T$ is the right training-time object: a dual regret bound turns directly into a saddle-gap bound for the weighted average mixture.

Proposition B.3 (Ergodic averaging produces the learned mixture)**.** *Let $\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(T)} \in \mathbb{R}^n_{++}$ be any sequence, with $X^{(t)}$ and $\boldsymbol{g}^{(t)}$ defined as above. Then for every $\boldsymbol{\lambda} \in \mathbb{R}^n_{++}$,*

$$\frac{1}{A_T} \sum_{t=1}^T \gamma_t G(\boldsymbol{\lambda}^{(t)}) - \Psi(\bar{X}_T, \boldsymbol{\lambda}) \le \frac{1}{A_T} \sum_{t=1}^T \gamma_t \langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda} \rangle. \tag{B.32}$$

In particular, for any dual optimum $\boldsymbol{\lambda}^\star$,

$$0 \le G(\boldsymbol{\lambda}^\star) - \Psi(\bar{X}_T, \boldsymbol{\lambda}^\star) \le \frac{1}{A_T} \sum_{t=1}^T \gamma_t \langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda}^\star \rangle. \tag{B.33}$$

Thus any standard mirror-descent regret bound on the right-hand side yields a vanishing saddle gap for the weighted average mixture $\bar{X}_T$.

The proof converts pointwise dual regret into a bound on the weighted average policy.

Proof. Because $\Psi(X, \boldsymbol{\lambda}) = h(\boldsymbol{\lambda}) + \sum_i \lambda_i \mathrm{Imp}_i(X)$, one has

$$\sum_{t=1}^T \gamma_t \Psi(X^{(t)}, \boldsymbol{\lambda}) = A_T h(\boldsymbol{\lambda}) + \sum_{i=1}^n \lambda_i \sum_{t=1}^T \gamma_t \mathrm{Imp}_i(X^{(t)}) \tag{B.34}$$

$$= A_T h(\boldsymbol{\lambda}) + A_T \sum_{i=1}^n \lambda_i \mathrm{Imp}_i(\bar{X}_T) \tag{B.35}$$

$$= A_T \Psi(\bar{X}_T, \boldsymbol{\lambda}). \tag{B.36}$$

The second equality uses linearity of Imp_i and the definition of $\bar{X}_T$. Therefore

$$\Psi(\bar{X}_T, \boldsymbol{\lambda}) = \frac{1}{A_T} \sum_{t=1}^{T} \gamma_t \Psi(X^{(t)}, \boldsymbol{\lambda}). \tag{B.37}$$

Next, for each t,

$$G(\boldsymbol{\lambda}^{(t)}) - \Psi(X^{(t)}, \boldsymbol{\lambda}) = h(\boldsymbol{\lambda}^{(t)}) - h(\boldsymbol{\lambda}) + \left\langle \boldsymbol{I}(X^{(t)}), \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda} \right\rangle \tag{B.38}$$

$$\leq \left\langle \nabla h(\boldsymbol{\lambda}^{(t)}), \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda} \right\rangle + \left\langle \boldsymbol{I}(X^{(t)}), \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda} \right\rangle \tag{B.39}$$

$$= \left\langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda} \right\rangle. \tag{B.40}$$

Here $G(\boldsymbol{\lambda}^{(t)}) = \Psi(X^{(t)}, \boldsymbol{\lambda}^{(t)})$ by the definition of $X^{(t)}$, the inequality follows from convexity of h, and $\boldsymbol{g}^{(t)} = \nabla h(\boldsymbol{\lambda}^{(t)}) + \boldsymbol{I}(X^{(t)})$. Multiplying by γ_t, summing over t, and using the identity above yields

$$\frac{1}{A_T} \sum_{t=1}^{T} \gamma_t G(\boldsymbol{\lambda}^{(t)}) - \Psi(\bar{X}_T, \boldsymbol{\lambda}) \leq \frac{1}{A_T} \sum_{t=1}^{T} \gamma_t \langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda} \rangle. \tag{B.41}$$

If $\boldsymbol{\lambda} = \boldsymbol{\lambda}^\star$ is dual optimal, then $G(\boldsymbol{\lambda}^\star) \leq G(\boldsymbol{\lambda}^{(t)})$ for every t, so

$$0 \leq G(\boldsymbol{\lambda}^\star) - \Psi(\bar{X}_T, \boldsymbol{\lambda}^\star) \tag{B.42}$$

$$\leq \frac{1}{A_T} \sum_{t=1}^{T} \gamma_t G(\boldsymbol{\lambda}^{(t)}) - \Psi(\bar{X}_T, \boldsymbol{\lambda}^\star), \tag{B.43}$$

and the desired bound follows. □

The saddle gap on Ψ above translates into a primal F_η-suboptimality bound by choosing the comparator dual associated with $\bar{X}_T$. Define

$$\boldsymbol{\lambda}^\star_{\bar{X}_T} := \left(\frac{1}{\eta + \mathrm{Imp}_i(\bar{X}_T)} \right)_{i \in [n]}. \tag{B.44}$$

This vector lies in $[1/(\eta + I_{\max}), 1/\eta]^n$.

Corollary B.4 (Primal gap controlled by weighted dual regret)**.** *Under the setup of Proposition B.3,*

$$F_\eta(X^\star) - F_\eta(\bar{X}_T) \leq \frac{1}{A_T} \sum_{t=1}^{T} \gamma_t \langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda}^\star_{\bar{X}_T} \rangle. \tag{B.45}$$

Consequently, whenever the weighted dual-regret term on the right is $\mathcal{O}(1/\sqrt{T})$, the same rate holds for the primal objective gap. In particular, if the iterates are kept in a box containing $[1/(\eta + I_{\max}), 1/\eta]^n$, on which the subgradients are bounded, the standard mirror-descent regret guarantee with a constant step size $\gamma_t = \Theta(1/\sqrt{T})$ gives

$$F_\eta(X^\star) - F_\eta(\bar{X}_T) = \mathcal{O}(1/\sqrt{T}). \tag{B.46}$$

The proof uses the Fenchel representation of the log objective to compare $\bar{X}_T$ with the primal optimum.

Proof. By the Fenchel inequality $\log z \leq \lambda z - 1 - \log \lambda$ for $z, \lambda > 0$ with equality at $\lambda = 1/z$, summing over i with $z_i = \eta + \mathrm{Imp}_i(X)$ gives $F_\eta(X) = \min_{\boldsymbol{\lambda} \in \mathbb{R}^n_{++}} \Psi(X, \boldsymbol{\lambda})$ for every $X \in \mathcal{X}$, with the minimizer at $\lambda_i = 1/(\eta + \mathrm{Imp}_i(X))$. In particular,

$$F_\eta(\bar{X}_T) = \Psi(\bar{X}_T, \boldsymbol{\lambda}^\star_{\bar{X}_T}). \tag{B.47}$$

Applying Proposition B.3 with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^\star_{\bar{X}_T}$ and using $G(\boldsymbol{\lambda}^{(t)}) \geq G(\boldsymbol{\lambda}^\star) = F_\eta(X^\star)$,

$$F_\eta(X^\star) - F_\eta(\bar{X}_T) = G(\boldsymbol{\lambda}^\star) - \Psi(\bar{X}_T, \boldsymbol{\lambda}^\star_{\bar{X}_T}) \tag{B.48}$$

$$\leq \frac{1}{A_T} \sum_{t=1}^{T} \gamma_t G(\boldsymbol{\lambda}^{(t)}) - \Psi(\bar{X}_T, \boldsymbol{\lambda}^\star_{\bar{X}_T}) \tag{B.49}$$

$$\leq \frac{1}{A_T} \sum_{t=1}^{T} \gamma_t \langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda}^\star_{\bar{X}_T} \rangle. \tag{B.50}$$

Since $\mathrm{Imp}_i(\bar{X}_T) \in [0, I_{\max}]$, $\boldsymbol{\lambda}^\star_{\bar{X}_T}$ lies in the bounded comparator set $[1/(\eta + I_{\max}), 1/\eta]^n$. The final rate statement follows by applying the usual mirror-descent regret bound on this bounded domain. □

Finally, to implement the dual solver, let

$$\Phi(\boldsymbol{\lambda}) := \sum_{i=1}^{n} \lambda_i (\log \lambda_i - 1) \qquad \text{for } \boldsymbol{\lambda} \in \mathbb{R}^n_{++}, \tag{B.51}$$

and let

$$D_\Phi(\boldsymbol{\lambda}, \boldsymbol{\lambda}') := \Phi(\boldsymbol{\lambda}) - \Phi(\boldsymbol{\lambda}') - \langle \nabla\Phi(\boldsymbol{\lambda}'), \boldsymbol{\lambda} - \boldsymbol{\lambda}' \rangle \tag{B.52}$$

be the associated Bregman divergence. The next proposition shows that the entropic mirror-descent step has a closed form and becomes an additive update in log space.

Proposition B.5 (Entropic mirror-descent update in log space)**.** *Given $\boldsymbol{\lambda}^{(t)} \in \mathbb{R}^n_{++}$, a subgradient $\boldsymbol{g}^{(t)} \in \partial G(\boldsymbol{\lambda}^{(t)})$, and a step size $\gamma_t > 0$, consider the entropic mirror-descent update*

$$\boldsymbol{\lambda}^{(t+1)} \in \arg\min_{\boldsymbol{\lambda} \in \mathbb{R}^n_{++}} \left\{ \gamma_t \langle \boldsymbol{g}^{(t)}, \boldsymbol{\lambda} \rangle + D_\Phi(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)}) \right\}. \tag{B.53}$$

Then

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} \exp(-\gamma_t g_i^{(t)}) \qquad \textit{for all } i \in [n]. \tag{B.54}$$

Equivalently, if $\nu_i^{(t)} := \log \lambda_i^{(t)}$, then

$$\nu_i^{(t+1)} = \nu_i^{(t)} - \gamma_t g_i^{(t)} \qquad \textit{for all } i \in [n]. \tag{B.55}$$

The proof is coordinate-wise because the entropy Bregman divergence is separable.

Proof. For $\Phi(\boldsymbol{\lambda}) = \sum_i \lambda_i(\log \lambda_i - 1)$, one has

$$\nabla\Phi(\boldsymbol{\lambda}) = (\log \lambda_1, \ldots, \log \lambda_n). \tag{B.56}$$

Therefore

$$D_\Phi(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)}) = \Phi(\boldsymbol{\lambda}) - \Phi(\boldsymbol{\lambda}^{(t)}) - \langle \nabla\Phi(\boldsymbol{\lambda}^{(t)}), \boldsymbol{\lambda} - \boldsymbol{\lambda}^{(t)} \rangle \tag{B.57}$$

$$= \sum_{i=1}^{n} \left[\lambda_i \log \frac{\lambda_i}{\lambda_i^{(t)}} - \lambda_i + \lambda_i^{(t)} \right]. \tag{B.58}$$

The mirror-descent objective is separable across coordinates, so each coordinate minimizes

$$\gamma_t g_i^{(t)} \lambda_i + \lambda_i \log \frac{\lambda_i}{\lambda_i^{(t)}} - \lambda_i + \lambda_i^{(t)}. \tag{B.59}$$

Differentiating with respect to λ_i gives

$$\frac{\partial}{\partial \lambda_i} \left[\gamma_t g_i^{(t)} \lambda_i + \lambda_i \log \frac{\lambda_i}{\lambda_i^{(t)}} - \lambda_i + \lambda_i^{(t)} \right] = \gamma_t g_i^{(t)} + \log \frac{\lambda_i}{\lambda_i^{(t)}}. \tag{B.60}$$

Setting this derivative to zero yields

$$\gamma_t g_i^{(t)} + \log \frac{\lambda_i^{(t+1)}}{\lambda_i^{(t)}} = 0, \tag{B.61}$$

$$\log \frac{\lambda_i^{(t+1)}}{\lambda_i^{(t)}} = -\gamma_t g_i^{(t)}. \tag{B.62}$$

Hence

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} \exp(-\gamma_t g_i^{(t)}). \tag{B.63}$$

Taking logarithms gives the stated log-space update. □

B.3 Compressed deployment mixtures and serving cost

For a weighted dual mixture $\Gamma = \{(\boldsymbol{\lambda}^{(s)}, \omega_s)\}_{s=1}^{S'}$ with $\omega_s > 0$ and $\sum_s \omega_s = 1$, let X^Γ denote the stochastic ranking tensor obtained by sampling component s with probability ω_s and serving $X(\boldsymbol{\lambda}^{(s)})$.

The dual mixtures used in the experiments all have this form. The full trajectory is $\{(\boldsymbol{\lambda}^{(t)}, w_t)\}_{t=1}^{T}$. The sampled mixture in Algorithm 1 uses empirical weights from the subsample. The affine-elimination diagnostic runs in $\mathbb{R}^n$, and Lemma A.1 allows components to be removed until at most $n+1$ vectors remain. The same update preserves the weighted impact vector

$$\sum_{t=1}^{T} w_t \boldsymbol{I}(X^{(t)}) = \boldsymbol{I}(\bar{X}_T) \tag{B.64}$$

by the one-point elimination invariant in Lemma A.1. If the trajectory is an exact optimal mixture, the sharper n bound follows from Proposition 3.4.

The lemma verifies that the weighted dual mixture has the same impact-vector behavior as an ordinary mixture.

Lemma B.6 (Linearity of impacts for weighted dual mixtures)**.** *For any weighted dual mixture Γ defined above, the induced tensor X^Γ satisfies*

$$\boldsymbol{I}(X^\Gamma) = \sum_{s=1}^{S'} \omega_s \boldsymbol{I}(X(\boldsymbol{\lambda}^{(s)})). \tag{B.65}$$

The proof applies the law of total probability to the sampled component.

Proof. For every item i, the law of total probability gives

$$\mathrm{Imp}_i(X^\Gamma) = \sum_{u=1}^{m} \sum_{k=1}^{n} e(k) r(u,i) X^\Gamma_{u,i,k} \tag{B.66}$$

$$= \sum_{s=1}^{S'} \omega_s \sum_{u=1}^{m} \sum_{k=1}^{n} e(k) r(u,i) X(\boldsymbol{\lambda}^{(s)})_{u,i,k} \tag{B.67}$$

$$= \sum_{s=1}^{S'} \omega_s \, \mathrm{Imp}_i(X(\boldsymbol{\lambda}^{(s)})). \tag{B.68}$$

Stacking the identities over items gives the claimed vector equality. □

The storage bound follows by counting stored dual vectors, and the serving bound follows from top-K selection.

Proof of Proposition 4.1. The mixture stores S' dual-variable vectors in $\mathbb{R}^n_{++}$ and their mixture weights, requiring $\mathcal{O}(S'n)$ deployment storage. To serve a request from user u, the system samples a component s according to the stored weights, computes the reweighted scores $\lambda_i^{(s)} r(u,i)$ for all

$i \in [n]$, and extracts the top K items. Once the relevance scores are available, score formation and top-K selection take $\mathcal{O}(n \log K)$ time. The procedure uses only Γ and the current user's relevance vector, so the required fair-ranking deployment storage is independent of both the number of training users m and the optimization horizon T. □

C Additional Notes on Scaling and Unseen-User Evaluation

This appendix collects two auxiliary observations that are useful for interpretation but sit outside the main compression argument. We first show that total and per-user-average quantities differ only by scale, and then give transfer guarantees for a learned dual mixture on unseen users.

C.1 Normalized average quantities are only a rescaling

For this appendix, define the cross-impact from item j's allocation to item i by

$$C_{i,j}(X) := \sum_{u=1}^{m} \sum_{k=1}^{n} e(k)\, r(u,i)\, X_{u,j,k}. \tag{C.1}$$

Thus $\mathrm{Imp}_i(X) = C_{i,i}(X)$. We also use

$$\mathrm{EnvyGap}(X) := \frac{1}{n} \sum_{i=1}^{n} \max_{j \in [n]} \big(C_{i,j}(X) - C_{i,i}(X)\big), \tag{C.2}$$

$$\mathrm{DomGap}(X) := \max_{i} \big(\mathrm{Imp}_i(X^{\mathrm{unif}}) - \mathrm{Imp}_i(X)\big)_+. \tag{C.3}$$

To compare these total quantities with per-user averages, define

$$\bar{C}_{i,j}(X) := \frac{1}{m} C_{i,j}(X), \qquad \bar{I}_i(X) := \frac{1}{m} \mathrm{Imp}_i(X), \tag{C.4}$$

and

$$\bar{F}_{\bar{\eta}}(X) := \sum_{i=1}^{n} \log(\bar{\eta} + \bar{I}_i(X)), \qquad \overline{\mathrm{EnvyGap}}(X) := \frac{1}{n} \sum_{i=1}^{n} \max_{j \in [n]} \big(\bar{C}_{i,j}(X) - \bar{C}_{i,i}(X)\big), \tag{C.5}$$

and

$$\overline{\mathrm{DomGap}}(X) := \max_{i} \big(\bar{I}_i(X^{\mathrm{unif}}) - \bar{I}_i(X)\big)_+. \tag{C.6}$$

The next proposition shows that these normalized quantities are only a rescaling of the original totals.

Proposition C.1 (Average quantities are rescaled totals)**.**

$$\overline{\mathrm{EnvyGap}}(X) = \frac{1}{m} \mathrm{EnvyGap}(X), \qquad \overline{\mathrm{DomGap}}(X) = \frac{1}{m} \mathrm{DomGap}(X), \tag{C.7}$$

and if $\eta = m\bar{\eta}$, then

$$F_{\eta}(X) = n \log m + \bar{F}_{\bar{\eta}}(X). \tag{C.8}$$

Hence the average quantities are only a change of units.

The proof simply substitutes the normalized definitions and factors out the sample size.

Proof. By definition,

$$\bar{C}_{i,j}(X) - \bar{C}_{i,i}(X) = \frac{1}{m} \big(C_{i,j}(X) - C_{i,i}(X)\big), \tag{C.9}$$

so taking the per-item max and averaging over i yields

$$\overline{\mathrm{EnvyGap}}(X) = \frac{1}{m} \mathrm{EnvyGap}(X). \tag{C.10}$$

The same argument gives the relation for dominance. Also, with $\mathrm{Imp}_i(X) = m\bar{I}_i(X)$ and $\eta = m\bar{\eta}$,

$$F_\eta(X) = \sum_{i=1}^{n} \log(\eta + \mathrm{Imp}_i(X)) \tag{C.11}$$

$$= \sum_{i=1}^{n} \log(m\bar{\eta} + m\bar{I}_i(X)) \tag{C.12}$$

$$= \sum_{i=1}^{n} \big(\log m + \log(\bar{\eta} + \bar{I}_i(X))\big) \tag{C.13}$$

$$= n \log m + \bar{F}_{\bar{\eta}}(X). \tag{C.14}$$

□

C.2 Generalization to unseen users

We now state the welfare guarantees on enlarged platforms and on held-out users used in Section 4.3, and we bound the required uniform deviations for all finite dual mixtures. We also record the parallel average-gap statements, which are useful for evaluation but are not needed in the main deployment claim.

Let A be a finite multiset of users, and let X_A^{unif} denote the uniform-ranking policy on this same user multiset. For any stochastic policy Y on these users, define the average cross-impact and self-impact on A by

$$\bar{C}_{i,j}^{A}(Y) := \frac{1}{|A|} \sum_{u \in A} \sum_{k=1}^{n} e(k)\, r(u,i)\, Y_{u,j,k}, \tag{C.15}$$

$$\bar{I}_i^{A}(Y) := \bar{C}_{i,i}^{A}(Y). \tag{C.16}$$

The corresponding average welfare and fairness quantities are

$$\bar{F}_{\bar{\eta}}^{A}(Y) := \sum_{i=1}^{n} \log\big(\bar{\eta} + \bar{I}_i^{A}(Y)\big), \tag{C.17}$$

$$\overline{\mathrm{EnvyGap}}^{A}(Y) := \frac{1}{n} \sum_{i=1}^{n} \max_{j \in [n]} \big(\bar{C}_{i,j}^{A}(Y) - \bar{C}_{i,i}^{A}(Y)\big), \tag{C.18}$$

$$\overline{\mathrm{DomGap}}^{A}(Y) := \max_i \big(\bar{I}_i^{A}(X_A^{\mathrm{unif}}) - \bar{I}_i^{A}(Y)\big)_+. \tag{C.19}$$

A sparse dual mixture Γ induces a stochastic policy X^Γ by sampling a component and ranking items by the scores $\lambda_i r(u,i)$. We write $\bar{C}^A(\Gamma)$, $\bar{I}^A(\Gamma)$, and the corresponding objective and fairness quantities as shorthand for evaluating the induced policy X^Γ. For a user distribution $\mathcal{D}$, define $\bar{C}_{i,j}^{\mathcal{D}}$, $\bar{I}_i^{\mathcal{D}}$, $\bar{F}_{\bar{\eta}}^{\mathcal{D}}$, $\overline{\mathrm{EnvyGap}}^{\mathcal{D}}$, and $\overline{\mathrm{DomGap}}^{\mathcal{D}}$ by replacing the empirical average over A with expectation over $u \sim \mathcal{D}$ and using the corresponding uniform-ranking rule inside the dominance gap.

Let $\mathfrak{G}$ be any deployment class, such as dual mixtures with a bounded number of components. Because Γ_{tr} is selected from the training users, the statistical terms must be uniform over $\mathfrak{G}$ rather than fixed for a single dual mixture. For a sample $U_N \sim \mathcal{D}^N$, let Δ_N^F, Δ_N^E, and Δ_N^D denote uniform-deviation radii satisfying

$$\sup_{\Gamma \in \mathfrak{G}} \left| \bar{F}_{\bar{\eta}}^{U_N}(\Gamma) - \bar{F}_{\bar{\eta}}^{\mathcal{D}}(\Gamma) \right| \le \Delta_N^F, \tag{C.20}$$

$$\sup_{\Gamma \in \mathfrak{G}} \left| \overline{\mathrm{EnvyGap}}^{U_N}(\Gamma) - \overline{\mathrm{EnvyGap}}^{\mathcal{D}}(\Gamma) \right| \le \Delta_N^E, \tag{C.21}$$

$$\sup_{\Gamma \in \mathfrak{G}} \left| \overline{\mathrm{DomGap}}^{U_N}(\Gamma) - \overline{\mathrm{DomGap}}^{\mathcal{D}}(\Gamma) \right| \le \Delta_N^D. \tag{C.22}$$

These radii are the only place where the statistical capacity of the deployment class enters. Proposition C.7 below gives explicit radii for the class of all finite dual mixtures. The next theorem isolates the welfare-transfer claim used in the main text. It is deterministic conditional on the welfare deviations holding for the training and enlarged user sets.

Theorem C.2 (Transfer to an enlarged platform)**.** *Let $U_{\mathrm{tr}} \sim \mathcal{D}^m$ and $U_{\mathrm{new}} \sim \mathcal{D}^M$ be independent, and let $U_{\mathrm{all}} := U_{\mathrm{tr}} \sqcup U_{\mathrm{new}}$ with $N = m + M$. Suppose the uniform welfare-deviation bounds hold for U_{tr} and U_{all} with radii Δ_m^F and Δ_N^F. If $\Gamma_{\mathrm{tr}} \in \mathfrak{G}$ is ϵ_{alg}-optimal on the training users,*

$$\bar{F}_{\bar{\eta}}^{U_{\mathrm{tr}}}(\Gamma_{\mathrm{tr}}) \geq \sup_{\Gamma \in \mathfrak{G}} \bar{F}_{\bar{\eta}}^{U_{\mathrm{tr}}}(\Gamma) - \epsilon_{\mathrm{alg}}, \tag{C.23}$$

then

$$\bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma_{\mathrm{tr}}) \geq \sup_{\Gamma \in \mathfrak{G}} \bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma) - \epsilon_{\mathrm{alg}} - 2\Delta_m^F - 2\Delta_N^F. \tag{C.24}$$

The proof chains the two uniform-deviation events through the training optimality condition.

Proof. Fix any $\Gamma \in \mathfrak{G}$. The deviation bounds imply

$$\begin{aligned}
\bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma_{\mathrm{tr}}) &\geq \bar{F}_{\bar{\eta}}^{\mathcal{D}}(\Gamma_{\mathrm{tr}}) - \Delta_N^F && \text{(C.25)}\\
&\geq \bar{F}_{\bar{\eta}}^{U_{\mathrm{tr}}}(\Gamma_{\mathrm{tr}}) - \Delta_m^F - \Delta_N^F && \text{(C.26)}\\
&\geq \bar{F}_{\bar{\eta}}^{U_{\mathrm{tr}}}(\Gamma) - \epsilon_{\mathrm{alg}} - \Delta_m^F - \Delta_N^F && \text{(C.27)}\\
&\geq \bar{F}_{\bar{\eta}}^{\mathcal{D}}(\Gamma) - \epsilon_{\mathrm{alg}} - 2\Delta_m^F - \Delta_N^F && \text{(C.28)}\\
&\geq \bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma) - \epsilon_{\mathrm{alg}} - 2\Delta_m^F - 2\Delta_N^F. && \text{(C.29)}
\end{aligned}$$

Taking the supremum over $\Gamma \in \mathfrak{G}$ gives the enlarged-platform welfare bound. □

The theorem compares the learned dual mixture with the best member of the same deployment class on the enlarged platform. The following corollary converts this class-relative statement into a comparison with any benchmark policy.

Corollary C.3 (Benchmark comparison on the enlarged platform)**.** *Under the assumptions of Theorem C.2, for any benchmark policy $X_{\mathrm{all}}^\star$ on U_{all},*

$$\begin{aligned}
&\bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(X_{\mathrm{all}}^\star) - \bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma_{\mathrm{tr}})\\
&\leq \underbrace{\bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(X_{\mathrm{all}}^\star) - \sup_{\Gamma \in \mathfrak{G}} \bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma)}_{\textit{approximation error of } \mathfrak{G} \textit{ on } U_{\mathrm{all}}} + \epsilon_{\mathrm{alg}} + 2\Delta_m^F + 2\Delta_N^F.
\end{aligned} \tag{C.30}$$

Proof. Add and subtract $\sup_{\Gamma \in \mathfrak{G}} \bar{F}_{\bar{\eta}}^{U_{\mathrm{all}}}(\Gamma)$, then apply Theorem C.2. □

The argument of Theorem C.2 also applies to users excluded from training, as in the held-out evaluation of Appendix E.8.

Corollary C.4 (Transfer to held-out users)**.** *Suppose that the assumptions of Theorem C.2 hold with U_{all} and Δ_N^F replaced by U_{new} and Δ_M^F. Then*

$$\bar{F}_{\bar{\eta}}^{U_{\mathrm{new}}}(\Gamma_{\mathrm{tr}}) \geq \sup_{\Gamma \in \mathfrak{G}} \bar{F}_{\bar{\eta}}^{U_{\mathrm{new}}}(\Gamma) - \epsilon_{\mathrm{alg}} - 2\Delta_m^F - 2\Delta_M^F. \tag{C.31}$$

Proof. The proof of Theorem C.2 applies verbatim. □

Theorem C.2 and Corollary C.4 give the welfare guarantees used in Section 4.3. Analogous deviation bounds for the average gaps give the following evaluation diagnostic for average envy and dominance gaps.

Proposition C.5 (Average-gap transfer)**.** *Suppose the uniform average-gap deviation bounds hold for U_{tr} with radii Δ_m^E, Δ_m^D and for U_{all} with radii Δ_N^E, Δ_N^D. Then any $\Gamma_{\mathrm{tr}} \in \mathfrak{G}$ satisfies*

$$\overline{\mathrm{EnvyGap}}^{U_{\mathrm{all}}}(\Gamma_{\mathrm{tr}}) \leq \overline{\mathrm{EnvyGap}}^{U_{\mathrm{tr}}}(\Gamma_{\mathrm{tr}}) + \Delta_m^E + \Delta_N^E, \tag{C.32}$$

$$\overline{\mathrm{DomGap}}^{U_{\mathrm{all}}}(\Gamma_{\mathrm{tr}}) \leq \overline{\mathrm{DomGap}}^{U_{\mathrm{tr}}}(\Gamma_{\mathrm{tr}}) + \Delta_m^D + \Delta_N^D. \tag{C.33}$$

Proof. For envy, the same deviation event gives

$$\begin{aligned}
\overline{\mathrm{EnvyGap}}^{U_{\mathrm{all}}}(\Gamma_{\mathrm{tr}}) &\leq \overline{\mathrm{EnvyGap}}^{\mathcal{D}}(\Gamma_{\mathrm{tr}}) + \Delta_N^E && \text{(C.34)}\\
&\leq \overline{\mathrm{EnvyGap}}^{U_{\mathrm{tr}}}(\Gamma_{\mathrm{tr}}) + \Delta_m^E + \Delta_N^E. && \text{(C.35)}
\end{aligned}$$

The dominance proof is identical. □

Explicit radii for finite dual mixtures. We now bound these radii for the class of all finite dual mixtures

$$\mathfrak{G}_{\mathrm{fin}} := \left\{ \Gamma = \{(\boldsymbol{\lambda}^{(s)}, \omega_s)\}_{s=1}^{S'} \;\middle|\; S' \in \mathbb{N},\ \boldsymbol{\lambda}^{(s)} \in \mathbb{R}^n_{++},\ \omega_s > 0,\ \sum_{s=1}^{S'} \omega_s = 1 \right\}, \tag{C.36}$$

which contains every class of dual mixtures with a bounded number of components. We assume that the item set and the relevance model producing $r(u,i)$ are fixed, that users are drawn i.i.d. from $\mathcal{D}$, and that $X(\boldsymbol{\lambda})$ breaks ties among equal scores by a fixed deterministic rule. We also assume $0 \le r(u,i) \le R_{\max}$ and set $B := E_{\max} R_{\max}$. For a dual mixture Γ and a user u, let $Z_i^{\Gamma}(u) := \sum_{k=1}^{K} e(k) r(u,i) X^{\Gamma}_{u,i,k} \in [0,B]$, so that $\bar{I}_i^{A}(\Gamma)$ is the average of Z_i^{Γ} over A and $\bar{I}_i^{\mathcal{D}}(\Gamma) = \mathbb{E}_{u\sim\mathcal{D}}[Z_i^{\Gamma}(u)]$. We write $Z_i^{\boldsymbol{\lambda}}$ for Z_i^{Γ} with the single-component mixture $\Gamma = \{(\boldsymbol{\lambda}, 1)\}$.

Although $\boldsymbol{\lambda}$ ranges over a continuum, it induces only finitely many joint rankings on a fixed sample of users. To count them, let $\kappa(W,p) := \sum_{j=0}^{p} 2^j \binom{W}{j}$ for integers $W, p \ge 0$. The next lemma bounds the number of sign patterns of W linear functions on $\mathbb{R}^p$, including patterns with ties, by $\kappa(W,p)$.

Lemma C.6 (Sign patterns of linear functions)**.** *Let $\varphi_1, \dots, \varphi_W$ be linear functions on $\mathbb{R}^p$. As $\boldsymbol{x}$ ranges over $\mathbb{R}^p$, the vector $\big(\operatorname{sign}\varphi_1(\boldsymbol{x}), \dots, \operatorname{sign}\varphi_W(\boldsymbol{x})\big)$ takes at most $\kappa(W,p)$ distinct values.*

Proof. Functions that vanish identically have a constant sign and can be dropped. We argue by induction on W. The claim holds for $W = 0$, and for $p = 0$ because $\mathbb{R}^0$ is a single point. Each sign pattern of $\varphi_1, \dots, \varphi_{W-1}$ corresponds to a nonempty convex cell $C \subseteq \mathbb{R}^p$. Adding φ_W splits C into $C \cap \{\varphi_W < 0\}$, $C \cap \{\varphi_W = 0\}$, and $C \cap \{\varphi_W > 0\}$, and C yields more than one new pattern only if $C \cap \{\varphi_W = 0\} \neq \emptyset$. The nonempty sets $C \cap \{\varphi_W = 0\}$ are the cells of $\varphi_1, \dots, \varphi_{W-1}$ restricted to the $(p-1)$-dimensional subspace $\{\varphi_W = 0\}$, so there are at most $\kappa(W-1, p-1)$ of them. Hence the number of patterns is at most $\kappa(W-1,p) + 2\kappa(W-1,p-1)$, which equals $\kappa(W,p)$ by Pascal's rule. □

For a sample of N users, let $W_N := N\binom{n}{2}$ be the number of pairwise score comparisons and $\kappa_N := \kappa(W_N, n)$. For $\delta \in (0,1)$, define

$$\varepsilon_N(\delta) := 2B\sqrt{\frac{2\log(2\kappa_N)}{N}} + B\sqrt{\frac{\log(2n/\delta)}{2N}}, \qquad \Delta_N^F(\delta) := n\log\Big(1 + \frac{\varepsilon_N(\delta)}{\bar{\eta}}\Big). \tag{C.37}$$

Because $\kappa_N \le (n+1)(Nn^2)^n$, the deviation $\varepsilon_N(\delta)$ is of order $B\sqrt{(n\log(Nn) + \log(n/\delta))/N}$, which vanishes as N grows. The next proposition shows that, uniformly over $\mathfrak{G}_{\mathrm{fin}}$, item impacts deviate from their population values by at most $\varepsilon_N(\delta)$ and the average-impact welfare by at most $\Delta_N^F(\delta)$, so that $\Delta_N^F(\delta)$ can serve as the welfare radius in Theorem C.2 and Corollary C.4.

Proposition C.7 (Uniform deviation for finite dual mixtures)**.** *If $U_N \sim \mathcal{D}^N$, then with probability at least $1-\delta$,*

$$\sup_{\Gamma\in\mathfrak{G}_{\mathrm{fin}}} \max_{i\in[n]} \big|\bar{I}_i^{U_N}(\Gamma) - \bar{I}_i^{\mathcal{D}}(\Gamma)\big| \le \varepsilon_N(\delta), \tag{C.38}$$

$$\sup_{\Gamma\in\mathfrak{G}_{\mathrm{fin}}} \big|\bar{F}_{\bar{\eta}}^{U_N}(\Gamma) - \bar{F}_{\bar{\eta}}^{\mathcal{D}}(\Gamma)\big| \le \Delta_N^F(\delta). \tag{C.39}$$

Proof. Fix a sample $U_N = (u_1, \dots, u_N)$. For $\boldsymbol{\lambda} \in \mathbb{R}^n_{++}$, the rankings $X(\boldsymbol{\lambda})_{u_q}$ for $q \in [N]$ are determined by the signs of the W_N linear functions $\lambda_i r(u_q,i) - \lambda_j r(u_q,j)$ with $i < j$, together with the fixed tie-breaking rule. By Lemma C.6, for each item i the vector $\big(Z_i^{\boldsymbol{\lambda}}(u_q)\big)_{q\in[N]}$ takes at most κ_N values in $[0,B]^N$ as $\boldsymbol{\lambda}$ ranges over $\mathbb{R}^n_{++}$.

Let $\xi_1, \dots, \xi_N$ be independent Rademacher signs. For fixed signs, $\sum_q \xi_q Z_i^{\Gamma}(u_q)$ is a convex combination of the corresponding single-component sums, so the supremum of its absolute value over $\mathfrak{G}_{\mathrm{fin}}$ equals that over single components. Applying Massart's finite-class lemma to the at most $2\kappa_N$ vectors $\pm\big(Z_i^{\boldsymbol{\lambda}}(u_q)\big)_{q\in[N]}$, whose Euclidean norms are at most $B\sqrt{N}$, gives

$$\mathbb{E}_{\boldsymbol{\xi}} \sup_{\Gamma\in\mathfrak{G}_{\mathrm{fin}}} \Big|\frac{1}{N}\sum_{q=1}^{N} \xi_q Z_i^{\Gamma}(u_q)\Big| \le B\sqrt{\frac{2\log(2\kappa_N)}{N}}. \tag{C.40}$$

Table 2: Benchmark markets after preprocessing.

Dataset	Source	K	Resulting size ($m \times n$)
MovieLens-100K	MovieLens-100K	5	$943 \times 1{,}574$
MovieLens-1M	MovieLens-1M	40	$3{,}000 \times 3{,}595$
Last.fm-15k	Last.fm-360K	40	$15{,}000 \times 14{,}975$

By symmetrization, $\mathbb{E}\sup_{\Gamma\in\mathfrak{G}_{\mathrm{fin}}} |\bar{I}_i^{U_N}(\Gamma) - \bar{I}_i^{\mathcal{D}}(\Gamma)|$ is at most twice this bound. Replacing one user changes the supremum by at most B/N, so McDiarmid's inequality and a union bound over $i \in [n]$ give the first inequality [see, e.g., 21]. On the same event, if $a, b \geq 0$ and $|a - b| \leq \varepsilon_N(\delta)$, then $|\log(\bar{\eta} + a) - \log(\bar{\eta} + b)| \leq \log(1 + \varepsilon_N(\delta)/\bar{\eta})$, and summing over the n items gives the second inequality. □

Applying Proposition C.7 to U_{tr} and to U_{all} or U_{new}, each with failure probability $\delta/2$, shows that Theorem C.2 and Corollary C.4 hold for $\mathfrak{G}_{\mathrm{fin}}$ with probability at least $1 - \delta$ when the radii are evaluated at $\delta/2$. When Γ_{tr} is a sampled dual mixture, ϵ_{alg} includes both the optimization error and the subsampling error. These radii vanish as the populations grow for fixed n, $\bar{\eta}$, and δ, but they increase with n and $1/\bar{\eta}$. The guarantee is therefore conservative at the population sizes in Section 5 and does not by itself ensure an improvement over uniform ranking. Applying this argument to the cross-impacts $\bar{C}_{i,j}$, with the uniform-ranking policy added as one more fixed function, yields radii of the same order for the average envy and dominance gaps in Proposition C.5.

D Broader Impacts

This work is motivated by the positive goal of making fair-ranking policies easier to deploy on two-sided platforms. Compact deployment representations may help reduce operational barriers to ranking systems that distribute item-side impact more evenly across providers.

At the same time, the societal effect of such systems depends on how impact, relevance, and provider welfare are defined in a concrete deployment. If these quantities are misspecified or learned from biased feedback, a fair-ranking objective could shift exposure in ways that disadvantage some provider groups, reduce consumer utility, or reinforce existing measurement biases. Responsible deployment should therefore monitor provider-level and user-level outcomes, validate the chosen impact definition for the application domain, and treat the compressed representation as an implementation tool rather than as a guarantee of substantive fairness.

E Experimental Details

E.1 Dataset Construction

Following Usunier et al. [33], we construct each benchmark by building implicit feedback, fitting iALS once, and using the resulting dense user–item scores as fixed relevance inputs for all fair-ranking methods. This keeps preference modeling outside the deployment comparison.

For Last.fm-15k, we use Last.fm-360K [7, 8], retain the most active users and artists, aggregate repeated user–artist interactions, and use log-transformed play counts as feedback weights. For MovieLens-1M and MovieLens-100K [15], ratings at least 3 are treated as positive feedback. MovieLens-1M retains the most active users and items. MovieLens-100K uses the same conversion as a small diagnostic benchmark with $K = 5$, making dense-matrix optimization and per-user BvN decomposition tractable.

All raw identifiers are mapped to contiguous integer indices. Table 2 reports the final market sizes and display lengths used in the experiments.

E.2 Existing Assets and Licenses

The experiments use publicly available recommendation datasets and standard optimization and numerical-computing software. MovieLens-100K and MovieLens-1M are provided by GroupLens for

research use under the conditions stated in their README files: users must acknowledge the dataset, must not imply endorsement by the University of Minnesota or GroupLens, must not redistribute the data without separate permission, and must not use the data for commercial or revenue-bearing purposes without permission. We cite Harper and Konstan [15] and do not redistribute the raw MovieLens data.

Last.fm-360K is distributed by MTG/UPF with permission of Last.fm and is made available for non-commercial use. Commercial use requires contacting Last.fm. We cite Celma [7, 8] and do not redistribute the raw Last.fm data.

The implementation uses Rust under the MIT or Apache License 2.0 terms, JAX under the Apache License 2.0, NumPy under the modified BSD license, and CVXPY under the Apache License 2.0. The MOSEK+BvN reference uses MOSEK under an academic license. We use these assets only for research evaluation and follow their stated terms.

E.3 iALS Settings

Consistent with the OFFR protocol, iALS is used only to produce a fixed preference model for the downstream fair-ranking comparison. We fit iALS with confidence weight $\alpha = 40.0$, L2 regularization 0.01, three conjugate-gradient steps per inner solve, frequency regularization disabled, and global bias and positivity shift enabled with positivity delta 10^{-6}. The embedding dimension is $d = 64$, and we run 50 outer iterations.

E.4 Evaluation Metrics

In addition to the smoothed NSW objective and deployment storage, we report the diagnostic metrics used by Uehara et al. [32]. For any evaluated policy X, define the cross-impact that item i would receive from item j's allocation by

$$C_{i,j}(X) := \sum_{u=1}^{m} \sum_{k=1}^{n} e(k)\, r(u,i)\, X_{u,j,k}. \tag{E.1}$$

The item impact in (2.3) is the self-impact $\mathrm{Imp}_i(X) = C_{i,i}(X)$. User utility is the average exposure-weighted relevance received by users,

$$\mathrm{UserUtil}(X) := \frac{1}{m} \sum_{u=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{n} e(k)\, r(u,i)\, X_{u,i,k}. \tag{E.2}$$

Mean max envy averages each item's largest gain from receiving another item's allocation,

$$\mathrm{MeanMaxEnvy}(X) := \frac{1}{n} \sum_{i=1}^{n} \max_{j \in [n]} \big(C_{i,j}(X) - C_{i,i}(X) \big). \tag{E.3}$$

Finally, items better off and worse off measure the percentage of items whose impact is respectively more than 10% above or below their impact under the uniform-ranking policy:

$$\mathrm{BetterOff}_{10}(X) := \frac{100}{n} \left| \{ i \in [n] \mid \mathrm{Imp}_i(X) > 1.1\ \mathrm{Imp}_i(X^{\mathrm{unif}}) \} \right|, \tag{E.4}$$

$$\mathrm{WorseOff}_{10}(X) := \frac{100}{n} \left| \{ i \in [n] \mid \mathrm{Imp}_i(X) < 0.9\ \mathrm{Imp}_i(X^{\mathrm{unif}}) \} \right|. \tag{E.5}$$

For all these metrics, user utility and items better off are larger-is-better, whereas mean max envy and items worse off are smaller-is-better.

E.5 Algorithm Hyperparameters

All experiments use DCG examination weights $e(k) = 1/\log_2(k+1)$ and $\eta = 10^{-8}$. Full trajectory methods use $T = 2048$, and sample variants use weighted subsamples of size $S = 32$. Reported values are means over five market seeds. For randomized subsampling methods, we average twenty independent subsamples within each seed before aggregating across seeds. For these methods, $\pm$ gives the SD of the five seed-level means and brackets give the mean within-seed SD over the twenty subsamples.

Algorithm 1 uses log-space mirror descent with AdamW [20], learning rate 0.01, weight decay 10^{-4}, gradient norm clipping at 1.0, and per-user-average gradient scaling. The log-space dual variables are box-constrained to $[-20, 20]$. Initialization is the uniform-impact dual ($\lambda_i \propto 1/\operatorname{Imp}_i(X^{\mathrm{unif}})$). All training users are processed per step ($b = m$). OFFR [33] runs Frank–Wolfe iterations on the same training set, and OFFR-Sample subsamples from this trajectory. Sinkhorn+BvN uses regularization $\epsilon = 0.1$ and runs for 2048 iterations. BvN decomposition uses smoothing 10^{-4} and tolerance 10^{-9}.

E.6 Implementation and Hardware

The experiment pipeline is implemented primarily in Python. The iterative ranking methods use JAX [6] and NumPy [16]. The iALS scorer and BvN decomposition routines are implemented in Rust and called from the experimental pipeline. The MOSEK+BvN reference uses CVXPY [2, 10] with MOSEK [23]. Runtime experiments were run on an Ubuntu 22.04.3 LTS workstation with an Intel Core i9-12900K CPU, 128 GB RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB memory. The complete set of experiments reported in the paper completed in less than 24 hours on this workstation, and method-level optimization times are reported in the result tables. The implementation is available at `https://github.com/yukiu00/Compact-Representations-of-Impact-Based-Fair-Ranking-Policies` under the MIT License.

E.7 Full Deployment Results

Table 3 reports the full storage and objective results underlying Table 1, including MaxRel, the exact-compression diagnostics on MovieLens-100K, and OFFR-Sample with the same budget $S = 32$ as Ours (Dual-Sample). Table 4 reports the corresponding fair-ranking diagnostics defined in Appendix E.4. These diagnostics and the sampling-budget sweep in Table 5 come from a separate five-seed run, so their values can differ slightly from those in Table 3. At equal support, OFFR-Sample and Ours (Dual-Sample) attain nearly identical F_η, user utility, and better-off and worse-off rates, so the storage gap between them reflects the representation alone.

On MovieLens-100K, both sampled policies have a much higher mean max envy than the full trajectories. The increase is similar for OFFR-Sample and Ours (Dual-Sample), which points to the small support rather than the representation. Table 5 shows that it shrinks as S grows, at the cost of more storage. Small-support sampling thus nearly preserves F_η but need not preserve every fairness diagnostic.

Table 6 reports serving latency, measured with a release-mode Rust implementation on the workstation of Appendix E.6 using one thread, batch size one, at least 0.2 s of warm-up, and $2{,}048$ randomized requests. Timing starts once the relevance vector is available and covers component sampling, dual reweighting, top-K selection, and output copying. It excludes relevance prediction and data access, so it measures the fair-ranking layer rather than the end-to-end system. Looking up a materialized ranking is faster, whereas Ours (Dual-Sample) stores no user-specific rankings and has a 95th-percentile latency below 0.04 ms.

E.8 Evaluation on Users Not Seen During Optimization

We split the users of each dataset into disjoint sets U_{tr} and U_{new} that contain 60% and 40% of the users. The full dual trajectory is learned on U_{tr} and then frozen, with the relevance scores, item catalog, and hyperparameters unchanged. The iALS scorer is fitted on all users (Appendix E.1), but no user in U_{new} is used to optimize, tune, or select the fair-ranking policy. We evaluate the frozen policy both on the enlarged platform $U_{\mathrm{all}} = U_{\mathrm{tr}} \sqcup U_{\mathrm{new}}$, as in Section 5, and on U_{new} alone. OFFR–Uniform, the analogue of MOSEK+BvN–Uniform, serves OFFR learned on U_{tr} to those users and Uniform to U_{new}.

For an evaluation population A, we use the average-impact welfare $\bar{F}^A_{\bar{\eta}}$ of Appendix C.2 with $\bar{\eta} = 10^{-8}/|U_{\mathrm{all}}|$, which differs from F_η with $\eta = |A|\bar{\eta}$ only by a policy-independent constant (Appendix C.1). We report

$$\mathrm{Retention}_A(\Gamma) := 100\,\frac{\bar{F}^A_{\bar{\eta}}(\Gamma) - \bar{F}^A_{\bar{\eta}}(X^{\mathrm{unif}}_A)}{\bar{F}^A_{\bar{\eta}}(\mathrm{OFFR}_A) - \bar{F}^A_{\bar{\eta}}(X^{\mathrm{unif}}_A)}, \tag{E.6}$$

Table 3: Full storage–quality tradeoff for deployable fair-ranking representations. Storage counts only the fair-ranking layer's deployment storage after relevance scores are available. Values are means over five market seeds. For the randomized subsampling methods, entries with ± give the SD across seeds and brackets give the mean within-seed SD over twenty subsamples. Support is the number of retained mixture components for OFFR and Dual variants, and OFFR-Sample uses the same budget $S = 32$ as Ours (Dual-Sample). "–" denotes a non-applicable quantity. Opt. time is the optimization time, and Comp. time counts only the additional compression step. Arrows indicate preferred directions.

Dataset	Method	Storage (MB)↓	F_η ↑	Support size	Opt. time (s)↓	Comp. time (s)↓
MovieLens-100K m=943, n=1,574, K=5	Uniform	0.000	1,880.0	–	0.00	–
	MaxRel	0.000	−16,878.3	–	0.002	–
	MOSEK+BvN	207.700	2,416.9	–	221.56	–
	Sinkhorn+BvN	207.702	2,414.6	–	67.94	–
	OFFR	38.633	2,417.8	2048	0.61	–
	OFFR-Exact	29.711	2,417.8	1575	0.61	11.45
	OFFR-Sample	0.599 ±0.003 [0.010]	2,413.0 ±118.9 [1.93]	32	0.61	0.00
	Ours (Dual)	12.902	2,417.8	2048	1.56	–
	Ours (Dual-Exact)	9.922	2,417.8	1575	1.56	11.81
	Ours (Dual-Sample)	0.199 ±0.001 [0.004]	2,413.1 ±118.7 [0.85]	32	1.56	0.00
MovieLens-1M m=3,000, n=3,595, K=40	Uniform	0.000	9,558.9	–	0.00	–
	MaxRel	0.000	−15,368.5	–	0.04	–
	OFFR	983.056	10,829.3	2048	8.48	–
	OFFR-Sample	15.231 ±0.084 [0.248]	10,829.3 ±152.9 [0.097]	32	8.48	0.00
	Ours (Dual)	29.458	10,829.4	2048	5.54	–
	Ours (Dual-Sample)	0.455 ±0.003 [0.008]	10,829.3 ±153.0 [0.008]	32	5.54	0.00
Last.fm-15k m=15,000, n=14,975, K=40	Uniform	0.000	43,366.9	–	0.00	–
	MaxRel	0.000	−99,332.3	–	0.13	–
	OFFR	4,915.216	49,213.1	2048	181.09	–
	OFFR-Sample	76.152 ±0.421 [1.238]	49,213.2 ±556.7 [1.36]	32	181.09	0.00
	Ours (Dual)	122.683	49,213.5	2048	163.37	–
	Ours (Dual-Sample)	1.893 ±0.011 [0.034]	49,213.3 ±556.5 [0.014]	32	163.37	0.00

where OFFR_A is OFFR retrained on A. Since MOSEK+BvN is impractical on the larger datasets, we use OFFR_A as an empirical reference that defines 100% retention.

Table 7 reports the results. On the enlarged platform, Ours (Dual) retains 96–97% of the improvement on MovieLens-1M and Last.fm-15k, about 32 percentage points more than OFFR–Uniform, and on both datasets Ours (Dual-Sample) stays within 0.01 percentage points of Ours (Dual). On the held-out users alone, Ours (Dual) retains 75–81% on these datasets but falls below Uniform on MovieLens-100K. There, the 377 held-out users provide only about 1.2 displayed positions per item, compared with 13.4 and 16.0 on the larger datasets, and about 23% of the items receive almost no impact (at most $10\bar{\eta}$) under Ours (Dual). Because the log welfare penalizes such items heavily, this pattern is consistent with finite-sample error in the lower tail of item impact. User utility on the held-out users nevertheless remains above that of Uniform (7.80 versus 5.55).

Table 4: Additional fair-ranking diagnostics for the methods in Table 3. Mean max envy, user utility, and the item better-off/worse-off rates are defined in Appendix E.4. Values are means over five market seeds. For the randomized subsampling methods, entries with ± give the SD across seeds and brackets give the mean within-seed SD over twenty subsamples. Arrows indicate preferred directions.

Dataset	Method	Mean max envy ↓	User utility ↑	Items better off (%) ↑	Items worse off (%) ↓
MovieLens-100K m=943, n=1,574, K=5	Uniform	0.000	5.552	0.000	0.000
	MaxRel	117.368	8.573	25.858	71.919
	MOSEK+BvN	0.000	7.825	100.000	0.000
	Sinkhorn+BvN	0.306	7.815	99.975	0.000
	OFFR	0.001	7.829	100.000	0.000
	OFFR-Exact	0.001	7.829	100.000	0.000
	OFFR-Sample	2.307 ±0.462 [1.070]	7.827 ±0.588 [0.0012]	98.675 ±0.118 [0.149]	0.668 ±0.035 [0.146]
	Ours (Dual)	0.001	7.829	100.000	0.000
	Ours (Dual-Exact)	0.001	7.829	100.000	0.000
	Ours (Dual-Sample)	2.386 ±0.516 [0.953]	7.825 ±0.590 [0.0007]	98.618 ±0.183 [0.158]	0.669 ±0.050 [0.130]
MovieLens-1M m=3,000, n=3,595, K=40	Uniform	0.000	17.295	0.000	0.000
	MaxRel	89.153	27.150	39.171	58.331
	OFFR	0.000	24.667	100.000	0.000
	OFFR-Sample	0.043 ±0.022 [0.068]	24.670 ±1.031 [0.0002]	99.875 ±0.028 [0.055]	0.000 ±0.000 [0.000]
	Ours (Dual)	0.000	24.668	100.000	0.000
	Ours (Dual-Sample)	0.034 ±0.011 [0.025]	24.668 ±1.028 [0.0002]	99.874 ±0.029 [0.034]	0.000 ±0.000 [0.000]
Last.fm-15k m=15,000, n=14,975, K=40	Uniform	0.000	18.106	0.000	0.000
	MaxRel	1,065.467	29.676	22.160	75.832
	OFFR	0.000	26.851	100.000	0.000
	OFFR-Sample	0.096 ±0.211 [0.430]	26.853 ±0.986 [0.0010]	99.998 ±0.001 [0.005]	0.000 ±0.000 [0.000]
	Ours (Dual)	0.000	26.852	100.000	0.000
	Ours (Dual-Sample)	0.006 ±0.001 [0.0011]	26.852 ±0.987 [0.00008]	100.000 ±0.000 [0.000]	0.000 ±0.000 [0.000]

Table 5: Effect of the sampling budget S on MovieLens-100K. Retention is the percentage of the F_η improvement of the full trajectory over Uniform that the sampled policy keeps. Entries with ± give the SD across five market seeds and brackets give the mean within-seed SD over twenty subsamples. Arrows indicate preferred directions.

Method	Storage (MB) ↓	Retention (%) ↑	Mean max envy ↓
Ours (Dual-Sample, $S = 32$)	0.199 ±0.001 [0.004]	99.285 ±0.087 [0.171]	2.386 ±0.516 [0.953]
Ours (Dual-Sample, $S = 64$)	0.391 ±0.002 [0.009]	99.670 ±0.053 [0.072]	1.187 ±0.311 [0.523]
Ours (Dual-Sample, $S = 128$)	0.765 ±0.002 [0.016]	99.844 ±0.019 [0.029]	0.547 ±0.174 [0.324]
Ours (Dual, $T = 2048$)	12.902	100	0.001

Table 6: Serving latency of the in-memory fair-ranking layer at $S = 32$, measured as the 95th-percentile time per request in microseconds (lower is better). Entries are means ± SD of five seed-level values.

Dataset	OFFR-Sample	Ours (Dual-Sample)
MovieLens-100K	0.028 ± 0.002	3.12 ± 0.13
MovieLens-1M	0.127 ± 0.005	9.82 ± 2.02
Last.fm-15k	0.154 ± 0.005	36.75 ± 2.46

Table 7: Retention (%, higher is better) under a user-disjoint 60/40 split. Policies are learned on U_{tr} and evaluated on the enlarged platform U_{all} and on the held-out users U_{new}. Retention is measured relative to OFFR retrained on the evaluated population. Entries with $\pm$ give the SD across five market seeds and brackets give the mean within-seed SD over twenty subsamples.

	U_{all}			U_{new}	
Dataset	Ours (Dual)	Ours (Dual-Sample)	OFFR–Uniform	Ours (Dual)	Ours (Dual-Sample)
MovieLens-100K	72.77 $\pm$3.07	71.99 $\pm$3.17 [0.20]	63.21 $\pm$0.57	$-$1,649.62 $\pm$130.14	$-$1,924.09 $\pm$169.36 [10.85]
MovieLens-1M	96.42 $\pm$0.17	96.41 $\pm$0.17 [0.005]	64.27 $\pm$0.34	74.96 $\pm$1.10	74.95 $\pm$1.10 [0.014]
Last.fm-15k	97.31 $\pm$0.14	97.30 $\pm$0.14 [0.001]	64.62 $\pm$0.18	81.48 $\pm$0.97	81.47 $\pm$0.97 [0.005]

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Introduction accurately summarize the paper's scope and contributions: exact impact-preserving compression, randomized subsampling, user-independent dual deployment, and empirical validation. They present the $2{,}596\times$ storage reduction as an end-to-end figure and separate it into the factors from subsampling and from the item-indexed representation. These claims are supported by Sections 3–5 and the corresponding appendices.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations paragraph at the end of the Conclusion states the assumptions of a fixed item catalog and fixed relevance scores, the resulting need to refresh the dual mixture, the absence of guarantees for new items and online adaptation, the requirement that mK be large relative to n for the representation gain, the dataset dependence of transfer to new users, the effect of small-support sampling on fairness diagnostics such as envy, and the serving-time cost of top-K selection. Appendix C.2 further notes that the transfer bound is conservative at the population sizes in our experiments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The theoretical results are stated with the needed definitions and assumptions in Sections 2–4, and Appendix C.2 states the additional assumptions of the transfer guarantees: a fixed item set and relevance model, i.i.d. users, bounded relevance scores, and a fixed tie-breaking rule. Complete proofs are provided in Appendices A–C, with proof sketches or explanatory text in the main paper where appropriate.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 describes the experimental protocol and main comparisons, and Appendix E provides dataset construction, model settings, hyperparameters, implementation details, hardware, and full result tables. Appendix E.6 also provides the URL of the implementation used for the experiments.

   Guidelines:

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

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The paper provides the implementation URL in Appendix E.6. The experiments use publicly available datasets cited in Appendix E.1, together with preprocessing and hyperparameter details needed to reproduce the reported results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 specifies the datasets, evaluation metrics, baselines, compression modes, and aggregation protocol. Appendix E gives the preprocessing choices, iALS settings, optimizer hyperparameters, and hardware details, and Appendices E.7 and E.8 describe the serving-latency measurement and the user-disjoint 60/40 split.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Tables 1 and 3–7 report means over five market seeds with across-seed SDs for the randomized subsampling methods, the user-disjoint evaluation, and serving latency, and brackets give the mean within-seed SD over twenty subsamples. Error bars in Figure 1 show $\pm 1\sigma$ across seeds, and Appendix E.5 describes the aggregation protocol.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix E.6 reports the workstation hardware, including CPU, memory, and GPU details, and states that all reported experiments completed in less than 24 hours in total. Method-level optimization times are reported in the result tables, and Table 6 reports serving latency measured on the same workstation.

   Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The authors have reviewed the NeurIPS Code of Ethics and are not aware of any aspect of the research process or reported results that violates it. The work uses publicly available recommendation datasets, does not involve human-subject experiments or crowdsourcing, and reports reproducibility and compute details.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The Introduction discusses the positive motivation of reducing ranking-induced provider disparities and supporting platform sustainability. Appendix D discusses potential negative impacts from misspecified impact definitions, biased feedback, exposure shifts, and reduced user utility, along with monitoring-oriented mitigations.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release high-risk models or datasets such as generative models, scraped image datasets, or pretrained models with substantial dual-use risk. The released implementation concerns algorithms for fair-ranking deployment and uses publicly available recommendation datasets.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Appendix E.2 credits the existing datasets and software assets and states the relevant usage terms or licenses, including the MovieLens research-use conditions, Last.fm-360K non-commercial terms, Rust MIT/Apache-2.0 terms, JAX and CVXPY Apache-2.0 licenses, NumPy modified BSD license, and MOSEK academic license.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper releases its implementation as the new asset, but does not introduce a new dataset or pretrained model. Appendix E documents preprocessing, hyperparameters, hardware, and asset licenses. Appendix E.6 provides the repository URL, and the code is released under the MIT License.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or new research with human subjects. The experiments use previously released recommendation datasets, as described in Appendix E.1.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not conduct new research with human subjects and does not involve crowdsourcing. It uses previously released recommendation datasets under their stated terms, so IRB approval or equivalent review was not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development, theoretical analysis, and experiments do not use LLMs as important, original, or non-standard components. Any use of writing or editing assistance, if applicable, does not affect the scientific methodology, rigor, or originality of the research.

    Guidelines:

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