Appendix A. Appendix / Related Works

To recap Section 1, the two core distinctions that we present from existing work are in:

- Formulating a definition of multi-label fairness that is grounded in the utility model that underlies the final serving layer.
- Connecting the two key mechanisms for industry-scale recommendation systems - filtration and ranking, and demonstrating how biases in either step can lead to downstream fairness gaps in user utility.

We organize our literature review as they relate to these two points separately. In terms of connecting the notions of industrial ML systems with algorithmic frameworks for fairness and utility, Ekstrand et al. (2022) represents the closest related work through a comprehensive overview of industry-scale modeling systems. The authors recognize the distinction between short-term proxies and user utility, as well as various fairness definitions in ranking. While they identify fairness issues beyond the ML modeling stage—such as in the candidate retrieval layer and behavioral distribution shifts across groups—they fall back on traditional individual model fairness and do not address the practical compositional nature of the problem.

A.1. Recap of Fairness in Multi-Label Settings

Previous works have recognized that industrial systems often comprise multiple models, each predicting different which are proxies of downstream outcomes (e.g., "clicks" as a proxy of engagement, "likes" as a proxy of preference). In these settings, the authors typically develop a multi-label analogs of existing fairness definitions such as equalized odds (Atwood et al. (2023)). Specifically, this entails defining a "composite label" Y_C as a function of the individual labels $Y_1, \ldots, Y_K$, each of which represent a different aspect of the goodness of an item (and similarly for the composite prediction $\hat{Y}_C$), and then applying a standard single fairness definition on the composite label and prediction. A key motive behind our research was our opinion that the proposed "composite" labels do not adequately reflect how predictions are served. To give some examples, Dwork and Ilvento (2018) and Wang et al. (2021b) suggest having the composite label Y_C as the product over the individual labels $Y_C = \prod_k Y_k$ (and similarly for the composite prediction). On the other hand, Atwood et al. (2023) suggests using the maxima such that $Y_C = \max_k Y_k$. From a user utility perspective, the interpretation of the product means that the item must be qualified in *all* aspects to be useful for the user, while the interpretation of the maxima means that a singular good aspect makes the item useful for the user. Our definition of the composite label presented in Section 3.1 as a weighted sum $Y_C = \sum_k \alpha_k^* Y_k$ therefore takes a natural middle ground, where users have (potentially heterogeneous) preferences over different aspects of an item. As mentioned in Figure 1, this is indeed the mechanism used across multiple industrial recommendation systems. DiCiccio et al. (2023) studies a closely related setting by ensuring that the weighted sum of predictions is calibrated with respect to the weighted sum of labels. This is the same setup that we analyze, with the key difference being that DiCiccio et al. (2023) addresses fairness for the items being ranked, rather than that of the viewer, which is our focus in this paper.

A.2. Recap of Fairness in Pipelines

Several fairness works have been motivated by the use of pipeline-style (also termed "sequential") systems in the real world to make decisions. For instance, resume filtering[2] and promotion candidacy are commonly framed as a multi-step filter process, where a decision-maker reduces the candidate pool at each step until they end up with a group of desired size. Bower et al. (2017) is an early work in this category and exactly studies fairness in pipeline systems, where a pipeline is fair if the final outcome obeys equal opportunity. In it, the authors illustrate that a pipeline constructed of models that are stand-alone fair may not be fair with respect to the final outcome. Dwork et al. (2020) analyzes a similar setting, but instead assesses notions of individual fairness rather than group fairness and makes similar conclusions. Blum et al. (2022) studies the same setting, but focuses on formulating a constrained optimization approach to optimize for both performance and fairness.

In our work, we connect the retrieval/selection pipeline with the serving layer as an extension of the above ideas. This paradigm introduces the perspective that pipeline systems can be a bottleneck for fairness specifically because they can be disconnected from the objectives of the serving layer. Namely, whereas the serving layer may optimize heavily for one of the labels Y_k, the retrieval pipeline may have been optimized for an entirely different label. In other words, the pipeline step could be largely disconnected from the actual downstream proxies and the users' heterogeneous preferences for those proxies. If there is only one single aspect of relevance (i.e. single-label case), then the prior research of fairness in pipelines directly applies in our setting. However, when items have multiple aspects of qualification (multi-label case), the quality of a retrieval mechanism becomes more nuanced, as we describe in Section 3.2.

Appendix B. Appendix / Details on Constrained EI and EHVI

A common approach to fairness in Bayesian Optimization (BO) is constrained expected improvement (EI) Gardner et al. (2014), where performance and fairness criteria are modeled as separate Gaussian Processes (GPs) over the decision variable α (preference weights). The standard EI approach Frazier (2018) selects the next candidate by optimizing:

$$EI_n(\alpha) = \mathbb{E}_n\left[[f(\alpha) - f_n^*]^+\right].$$

The "fair variant" of the algorithm introduces an indicator variable $I(\alpha)$ parameterized by a GP $c(\alpha)$ to represent if a point α satisfies the constraints or not. By assuming independence between the constraint and objective, Gardner et al. (2014); Perrone et al. (2021) presents the constrained objective EIC_n where γ is a hyperparameter that denotes the slack on the constraint $c(\alpha) - \gamma \leq 0$.

$$EIC_n(\alpha) = \mathbb{E}_n\left[[f(\alpha) - f_n^*]^+ \cdot I(\alpha)\right] = \mathbb{E}_n\left[[f(\alpha) - f_n^*]^+\right] \cdot \mathbb{E}_n\left[I(\alpha)\right] = EI_n(\alpha) \cdot \mathbb{P}(c(\alpha) \leq \gamma)$$

While this method is interpretable and computationally efficient, applying it directly to DER poses two challenges. First, DER is not independent of the overall utility objective,

2. We term this situation as "resume filtering", where individuals are being filtered, to disambiguate the situation that we are in, which is when jobs are filtered and shown to a user. In the former situation, the fairness is with respect to the candidates while in the latter, the fairness is with respect to the viewer.

invalidating the assumptions behind constrained EI. Second, setting an appropriate fairness threshold γ is nontrivial—business objectives often seek to improve both DER and overall utility simultaneously rather than enforcing a hard constraint.

Instead of constrained EI, we frame the problem as a multi-objective optimization and use *Expected Hyper-Volume Improvement* (EHVI) (Daulton et al., 2020; Yang et al., 2019). EHVI is the multi-objective analog of EI, optimizing improvements in Pareto frontier hypervolume rather than a single scalar objective. We include the definition from Daulton et al. (2020) below for completeness; see Daulton et al. (2020); Yang et al. (2019) for more background on the method.

Definition 7 *(EHVI) Given a reference point $r \in \mathbb{R}^K$, the hypervolume indicator of a Pareto set $\mathcal{P}$ is the K-dimensional Lebesgue measure λ_K of the space dominated by $\mathcal{P}$ and bounded from below by α: $HV(\mathcal{P}, r) = \lambda_K\left(\bigcup_{i=1}^{|\mathcal{P}|}[r, y_i]\right)$ where $[r, y_i]$ denotes the hyperrectangle bounded by vertices r and y_i.*

By optimizing for the global utility and DER simultaneously, we aim to find points that present the best possible tradeoffs. At each iteration, EHVI returns a set of preference weights that have the highest expectation of hypervolume improvement and we prioritize those with the highest expected DER. This allows us to find points that encourage fairness while also boosting global utility.

Appendix C. Appendix / Proofs

C.1. Proof of Lemma 3

Proof First we observe that by definition, the optimal utility is realized when we pick the best retrieved item with respect to the true expected outcomes conditional on the predictions $\mathbb{E}[Y_k|f_1(X, Z^j), \ldots, f_K(X, Z^j)]$ and true user preferences α_k^*. Hence, the best item j^* is the solution to the following, where without loss of generality we only need to consider the selected items where $I(X, Z^j) = 1$, which we denote as $1, \ldots, m^+$:

$$j^* := \text{argmax}_{1\le j\le m^+}\mathbb{E}\left[\sum_{k=1}^{K}\alpha_k^* Y_k^j \big| f_1(X, Z^j), \ldots, f_K(X, Z^j)\right]$$

Now we show that this item j^* is indeed selected whenever $\alpha = c\cdot\alpha^*$ and $c > 0$ by plugging these values into the selection function Eq. 1 (and denoting $\hat{j}(\alpha)$ as the retrieved item when using SPR coefficients α).

$$\begin{aligned}
\hat{j}(\alpha^*) &:= \text{argmax}_{1\le j\le m^+}\sum_{k=1}^{K}\alpha_k f_k(X, Z^j) \\
&= \text{argmax}_{1\le j\le m^+}\sum_{k=1}^{K} c\cdot\alpha_k^*\cdot\mathbb{E}\left[Y_k^j \mid f_1(X, Z^j), \ldots, f_K(X, Z^j)\right] \\
&= \text{argmax}_{1\le j\le m^+}\mathbb{E}\left[c\sum_{k=1}^{K}\alpha_k^* Y_k^j \big| f_1(X, Z^j), \ldots, f_K(X, Z^j)\right]
\end{aligned}$$

Where in the second line, we utilized the assumption that the model is calibrated. In the last line, we rely on the fact that if $c > 0$, the argmax of the last line is still the same as j^*.

Next, to show that if we select $\alpha' \neq c \cdot \alpha^*$ then $U(X, I, f, \alpha', \alpha^*) \leq U(X, I, f, c \cdot \alpha^*, \alpha^*)$, we use the assumption that any item can be selected a.s. and that $f_k(X, Z^j)$ can take on any value in its range to demonstrate that we can always construct cases where using α' will lead to selection of a suboptimal item. To first provide some intuition for why α' can be suboptimal, consider changing a single coordinate such that $\alpha'_k = c \cdot \alpha^*_k + \epsilon_1$. Then the selected item will be based on:

$$\hat{j}(\alpha') := \text{argmax}_{1 \leq j \leq m^+} \mathbb{E}\left[\epsilon Y_k + c \sum_{k=1}^{K} \alpha^*_k Y^j_k \,|\, f_1(X, Z^j), \ldots, f_K(X, Z^j)\right]$$

From this, we can see that the argmax now considers an extra term ϵY_k which is disconnected the true preferences of label k. The key intuition is therefore that if an item has a label with very low true preference has high expectation, then it may be selected over an item of overall higher quality. We proceed formally via proof by contradiction.

By our assumption, we have a nonzero probability of encountering the following list with two items Z^- and Z^+. We construct a list so that both items have $\mathbb{E}[Y_k|f] = b_k$ (where we use the shorthand f in the conditioning to denote the set of predictions for an item) except at indices i and j, where for scalars $e^+_j, e^-_i, e^-_j \in \mathbb{R}$, we have that:

$$E[Y^+|f] = \{b_1, b_2, \ldots, b_i, b_j + e^+_j, \ldots, b_K\}$$

$$E[Y^-|f] = \{b_1, b_2, \ldots, b_i + e^-_i, b_j + e^-_j, \ldots, b_K\}$$

We continue the construction by letting Z^+ have higher ground truth relevance. That is:

$$\begin{aligned}
&\mathbb{E}\left[\sum_k^K \alpha^*_k Y^+_k\right] > \mathbb{E}\left[\sum_k^K \alpha^*_k Y^-_k\right] \\
&\implies \sum_k^K \alpha^*_k b_k + \alpha^*_j e^+_j > \sum_k^K \alpha^*_k b_k + \alpha^*_i e^-_i + \alpha^*_j e^-_j \\
&\iff \alpha^*_j e^+_j > \alpha^*_i e^-_i + \alpha^*_j e^-_j \\
&\iff (e^+_j - e^-_j) > \frac{\alpha^*_i}{\alpha^*_j} e^-_i \qquad (8)
\end{aligned}$$

In the last line, division is permitted as α^* are positive by construction. Now suppose that we serve recommendations using preferences that diverge from $c \cdot \alpha^*$ by using $\alpha' = c \cdot \alpha^* + \beta d$ where we have $\beta d \neq c \cdot \alpha^*$ (i.e., α' is not in the same direction as α^*) and $\beta d > -c \cdot \alpha^*$ (since preferences cannot be negative). We assume that α' is an optimal solution, which implies that there does not exist a set of e^+_j, e^-_j, e^-_i both satisfying inequality 8 and leads to Z^- getting picked (therefore yielding suboptimal utility). Having Z^+ be selected requires that

$$
\begin{aligned}
&\mathbb{E}\left[\sum_k^K \alpha_k' Y_k^+\right] > \mathbb{E}\left[\sum_k^K \alpha_k' Y_k^-\right] \\
&\implies \sum_k^K (c\cdot\alpha_k^* + \beta d_k) b_k + (c\cdot\alpha_j^* + \beta d_j) e_j^+ \\
&\qquad > \sum_k^K (c\cdot\alpha_k^* + \beta d_k) b_k + (c\cdot\alpha_i^* + \beta d_i) e_i^- + (c\cdot\alpha_j^* + \beta d_j) e_j^- \\
&\iff (c\cdot\alpha_j^* + \beta d_j) e_j^+ > (c\cdot\alpha_i^* + \beta d_i) e_i^- + (c\cdot\alpha_j^* + \beta d_j) e_j^- \\
&\iff e_i^- < (e_j^+ - e_j^-)\left(\frac{c\cdot\alpha_j^* + \beta d_j}{c\cdot\alpha_i^* + \beta d_i}\right)
\end{aligned}
$$

Where in the first line, we cancel out the large sum and leave the extra terms involving e_j^+, e_j^-, e_i^- and in the second and third lines we can safely divide by $c\cdot\alpha_j^* + \beta d_j$ since by construction it is positive. Now we apply inequality 8 on the last line and let $(e_j^+ - e_j^-) = \frac{\alpha_i^*}{\alpha_j^*} e_i^- + \epsilon$ for some $\epsilon > 0$. This gives us the following inequality:

$$
\begin{aligned}
e_i^- &< \left(\frac{\alpha_i^*}{\alpha_j^*} e_i^- + \epsilon\right)\left(\frac{c\cdot\alpha_j^* + \beta d_j}{c\cdot\alpha_i^* + \beta d_i}\right) \\
&\iff 0 < e_i^-\left[\frac{\alpha_i^*}{\alpha_j^*}\cdot\frac{c\cdot\alpha_j^* + \beta d_j}{c\cdot\alpha_i^* + \beta d_i} - 1\right] + \epsilon\left(\frac{c\cdot\alpha_j^* + \beta d_j}{c\cdot\alpha_i^* + \beta d_i}\right)
\end{aligned}
$$

At this point, observe that since $\epsilon > 0$ and $c\cdot\alpha + \beta d > 0$ for both $d \in \{d_i, d_j\}$ and hence the second term on the right-hand side of the inequality is positive. However, if the term multiplied with e_i^- is nonzero, we can use our construction to drive e_i^- upwards (by widening the upper bound gap of $e_j^+ - e_j^-$) or downwards to make the inequality false. Making the term next to e_i^- zero requires that $d_i = \beta\alpha_i^*$ and $d_j = \beta\alpha_j^*$, but this is false by assumption (since we assumed that the d vector is in a different direction of the optional α^* vector) and we have thus reached a contradiction for an arbitrary choice of β, d_i, d_j (including if one of them equals zero). Putting these statements together, we have shown that α^* is a maximizer of the utility function. ■

C.2. Proof of Theorem 4

We show that the expected difference in groupwise utility over the shared distribution (under fair, calibrated models) can be upper bounded as follow:

Proof

$$
\begin{aligned}
&\mathbb{E}_{S_X}\left[U_1(X, I, f, \alpha, \alpha^*) - U_0(X, I, f, \alpha, \alpha^*)\right] \\
&\quad = \mathbb{E}_{S_X}\left[\mathbb{E}\left[\sum_k \alpha_k^*(1) Y_k^{\hat{j}} - \alpha_k^*(0) Y_k^{\hat{j}} \middle| X\right]\right] \\
&\quad = \mathbb{E}_{S_X}\Bigg[\mathbb{E}\Bigg[\sum_k \alpha_k^*(1)(Y_k^{\hat{j}} - f_k(X, Z^{\hat{j}})) - \alpha_k^*(0)(Y_k^{\hat{j}} - f_k(X, Z^{\hat{j}})) \\
&\qquad\quad + (\alpha_k - \alpha_k^*(0)) f_k(X, Z^{\hat{j}}) + (\alpha_k^*(1) - \alpha_k) f_k(X, Z^{\hat{j}}) \Big| X\Bigg]\Bigg] \\
&\quad \leq \mathbb{E}_{S_X}\Bigg[\mathbb{E}\Bigg[\sum_k (\alpha_k^*(1) - \alpha_k^*(0)(Y_k^{\hat{j}} - f_k(X, Z^{\hat{j}})) \\
&\qquad\quad + |\alpha_k - \alpha_k^*(0)| f_k(X, Z^{\hat{j}}) + |\alpha_k - \alpha_k^*(1)| f_k(X, Z^{\hat{j}}) \Big| X\Bigg]\Bigg]
\end{aligned}
$$

■

In the third line, we add and subtract terms and then collect them. In the last line, we utilize that $x \leq |x|$.

C.3. Proof of Theorem 6

Proof By definition, we have:

$$
Q_m(I(X, Z), Y) = \mathbb{E}\left[\max_{j \in \{1,\ldots,m\}} \left(I(X, Z_j) \cdot \sum_{k=1}^{K} Y_k^j\right)\right].
$$

Since we assumed that the serving model is optimal as we are using calibrated models and know the true preferences w.r.t. each label, this implies that the expected utility optimized with respect to the best of the selected items. Since $Q_m(I(X, Z), Y)$ represents the expected maximum γ-good item retrieved, without loss of generality we can say that the expected utility on the distribution $X|G = 1$ is upper bounded by $\gamma > 0$ such that:

$$
0 \leq \mathbb{E}_1\left[U_1(X, I, f, \alpha, \alpha^*)\right] \leq \gamma \cdot |\alpha_k^*(0)|_\infty
$$

By similar reasoning, the expected utility on the distribution S_X is upper bounded as:

$$
0 \leq \mathbb{E}_{S_X}\left[U_1(X, I, f, \alpha, \alpha^*)\right] \leq (\gamma - \epsilon) \cdot |\alpha_k^*(0)|_\infty
$$

Putting these bounds together to analyze the difference in the best case expected utility across the distributions, we get that:

$$
\begin{aligned}
&\sup_{f,\alpha} \mathbb{E}_1\left[U_1(X, I, f, \alpha, \alpha^*)\right] - \sup_{f,\alpha} \mathbb{E}_{S_X}\left[U_1(X, I, f, \alpha, \alpha^*)\right] \\
&\qquad = \gamma \cdot |\alpha_k^*(1)|_\infty - (\gamma - \epsilon) \cdot |\alpha_k^*(1)|_\infty \\
&\qquad = \epsilon \cdot |\alpha_k^*(1)|_\infty \\
&\qquad \geq \epsilon \cdot \min_k \alpha_k^*(1)
\end{aligned}
$$

The last inequality follows from the fact that $|\alpha_k^*(1)|_\infty \geq \min_k \alpha_k^*(1)$. Intuitively, this means that even if we optimize the model and serving coefficients, because there is a discrepancy in the quality of the candidate retrieval on at least one of the labels Y_k across distributions $X|G=1$ and S_x, this will propagate through the pipeline in the form of a utility gap across groups.

■

Appendix D. Appendix / Data and Experiment Details

In this section, we first illustrate the diversity of the datasets by showing the base performance metrics of each model as well as the utility and DER surfaces. Then, we provide details on how each dataset and model was constructed.

D.1. Model performance

As each dataset has two labels, we show the model performance on each label for the overall and for each group. We keep each dataset at a split of 80/20 across the two groups. This is to simulate the primary fairness concern of our work which is that optimizing for global utility in the presence of heterogeneous preferences and imbalanced representation will lead to disproportional benefits for one group.

Group	Prevalence	Y_1 AUC	Y_2 AUC
All	100%	0.995	0.996
0	80%	0.996	0.996
1	20%	0.995	0.996

Table 1: Synthetic Model Performance

Group	Prevalence	Y_1 AUC	Y_2 AUC
All	100%	0.782	0.929
0	80%	0.780	0.927
1	20%	0.790	0.934

Table 2: MovieLens Model Performance

Group	Prevalence	Y_1 AUC	Y_2 AUC
All	100%	0.827	0.869
0	80%	0.830	0.854
1	20%	0.826	0.872

Table 3: Modcloth Model Performance

Group	Prevalence	Y_1 AUC	Y_2 AUC
All	100%	0.861	0.830
0	80%	0.857	0.829
1	20%	0.865	0.830

Table 4: Electronics Model Performance

D.2. Utility and DER surface

Loosely speaking, the hardness of this multi-objective Bayesian Optimization problem comes down to how hard it is the find the globally optimal utility and DER separately, and how close or far those points are from each other (which makes finding the tradeoff difficult). These factors in turn depend on several factors, including the sparsity of positive labels, the noisiness of the labels, and model's predictive ability. For example, Figure 5 shows that the optimal multi-optjective solution in the clean Synthetic data case is likely any

point on the diagonal where the two α_k values are equal. However, the optimal utility in the Movielens data is a much noisier surface as shown in Figure 6. The Modcloth data in Figure 7 showcases an instance where the region with the optimal DER is in the opposite corner as the optimal utility. Lastly, the Electronics data is characterized by a relatively flat utility and DER surface, where any random point could feasibly yield relatively high utility and DER. This explains why both EI and Fair EHVI did not beat random search by a wide margin. In fact, EI had an average performance below random search in our experiments.

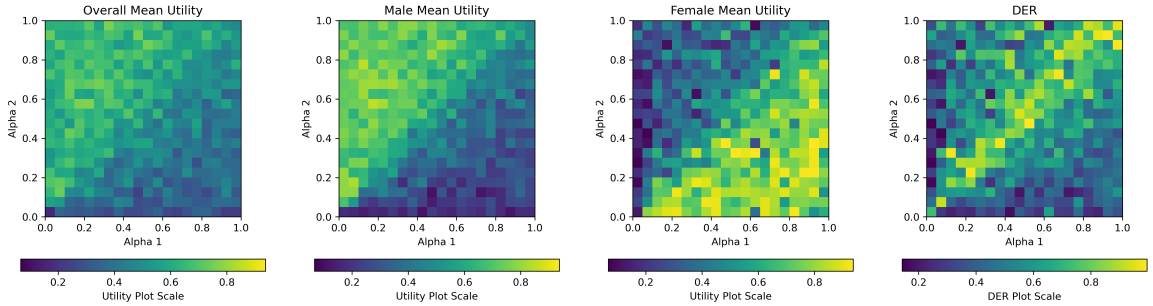

Figure 5: Utility and DER surface - Synthetic data

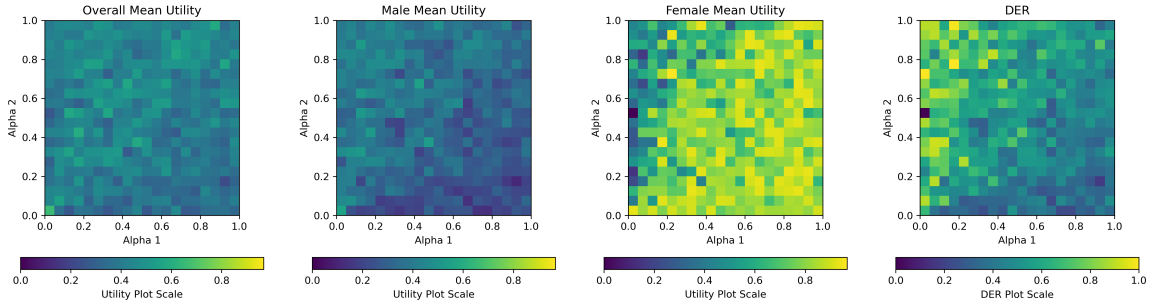

Figure 6: Utility and DER surface - Movielens data

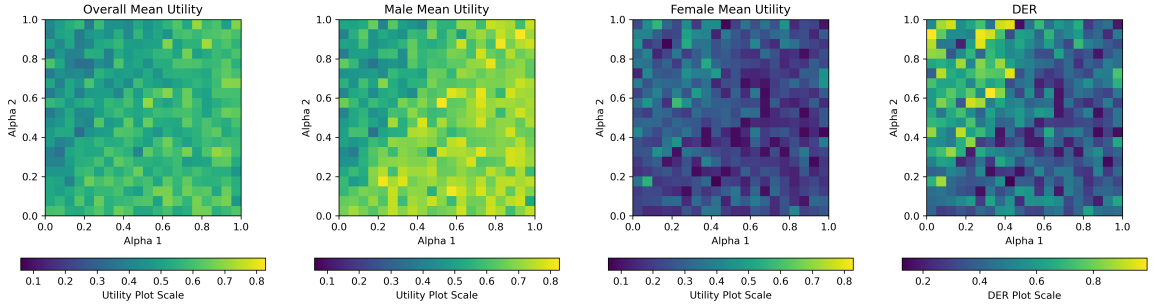

Figure 7: Utility and DER surface - Modcloth data

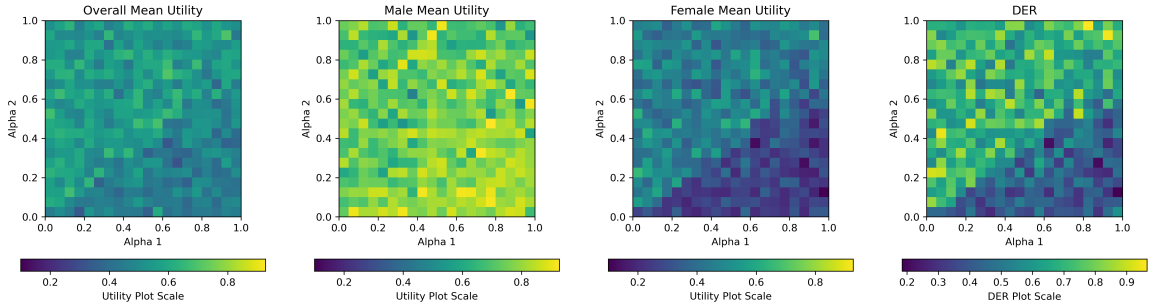

Figure 8: Utility and DER surface - Electronics data

D.3. Data Generation

We now explain how we processed each dataset, created models, and processed the scores to create labels. We start with the MovieLens dataset, being the most complex model as it required hand-crafting features. Then we describe the Modcloth and Electronics datasets, which we processed by modifying the source code from Wan et al. (2019). Finally, we end with the Synthetic dataset as it is identical in the mechanics to the above processes, but significantly simpler to implement. In particular, the general process of processing these datasets is sampling synthetic negatives, constructing a model, and placing the data into a batch serving and scoring mechanism. The last step (common to all datasets) is designed to simulate a real-world recommendation system. In it, we simulate a batch of users coming onto the platform, where we fetch 20 items for each user, rank them based on predicted preferences, and finally obtaining feedback in terms of user utility on the best item (as stated in our framework).

In terms of the models used, we use random forests for Synthetic and MovieLens datasets and neural networks for the Modcloth and Electronics datasets. We emphasize that the specific model used does not matter beyond the performance of the model, which we disclosed above. This is because our algorithmic innovation is strictly on the adaptive experiment side using BO, which effectively treats the model as a black-box system and only sees the scores and the labels.

D.3.1. MovieLens

We use the MovieLens dataset. The original task in MovieLens is to predict movie ratings given user ID, item ID, and one-hot encoded genre data about each movie that a user watched. We transform this into a multi-task, multi-stage, preference learning problem. We prepare the data generation process as follows:

1. Generate a new label y_{watch}, which represents if a user watched a movie. We generate random synthetic negatives in the data to create this label. We let the other label be y_{liked} based on if the user watched the movie and gave it a rating of 3 or higher.

    - Trivial solutions can be optimal if the labels have a certain dependency structure (e.g. having perfect correlation), for example if $y_{like} = 1$ is possible only if $y_{watch} = 1$. Hence even for synthetically generated negative movies that are not

watched, we impute a label by first learning the patterns of likes for the movies that are watched and then applying that like prediction (Scikit-learn (Pedregosa et al. (2011)) random forest) model to the movies that are not watched.

- When doing this, we also scramble the gender assignment in the data as we assume no $Y|X$ distribution shift across groups. This is to remove $Y|X$ distribution shifts from the data which are outside the scope of our framework.

2. Generate user features X for both tasks by taking the historical rolling average of the genres of movies that a user watched. E.g. if you watched 2 documentaries and 1 action movie in a rolling window of 3, your feature matrix for those genres would look like $[2/3, 1/3]$. These are used alongside the genre information of the movie being scored which represent the item features Z^j.

3. Split the data. We take a random 20% of the data for the models and use the other 80% for the BO process. For the models, we train one model for each label that maps $f_k : X \to [0, 1]$ to model y_k with probabilistic predictions. We train the scoring model via a Scikit-learn random forest classifier with default parameters and $n = 300$ trees. A random forest was chosen as in the authors' experience, these models work well off-the-shelf with minimal tuning.

    - We tested other off-the-shelf scikit-learn models but found that this provided the best performance in terms of ROC AUC. We did not focus on any hyperparameter tuning as default settings provided good results.

4. Create batch-sampling, batch-scoring, and batch-utility mechanisms: The point is to simulate random users coming onto the site, getting some recommendations, we score them with f_k, serve recommendations using α, and then observe rewards.

    - The user's u true preference of each item i shown is $\sum_k \alpha_k^* y_k$. Importantly, α^* *depends on the user's demographics.*
    - The predicted preference is $\sum_k \alpha_k f_k$ where α is the PM's guess of user preference
    - Compute the true preference of the highest scoring predicted item, that is the user's utility
    - Repeat the above steps for all users.

D.3.2. Modcloth and Electronics

The Modcloth and Electronics datasets are both e-commerce datasets. They provide data about user demographics (which the authors thoughtfully imputed) as well as user ratings of items that they purchased (clothing and electronics, respectively). We largely rely on the source code provided in Wan et al. (2019) for our purposes with some modifications which we detail below. In it, the authors model the different characteristics such as "fit" and "rating" using a collaborative filtering neural network model. Specifically, the model involves storing embeddings for each user and item, then training the embeddings by using the dot product to make predictions. Our modifications are as follows:

1. For both datasets, we need synthetic negatives to represent samples presented to the user that they would not have purchased. The original authors provide a method of generating these samples by creating a distribution over user and item IDs such that the probability of sampling a pair (as a negative) is 0 if the user/item pair exists in the data with a rating of 4 or higher. We use this exact mechanism. Then to impute the labels, we sample from a probability distribution where the density of each rating is shifted down by 1 (e.g., the probability of a 4 for a negative item is the probability of a 3 for a non-negative item). This gives us "viewed" binary labels.

2. We translate the other labels "rating" and "fit" into binary labels as well based on thresholds. We then use these labels to train a collaborative filtering model. The main difference between our model and the original is that turn it into a multitask problems where each task shares user/item embeddings and also has a task-specific layer. Additionally, we change the loss from mean squared error to binary cross-entropy.

3. Lastly, for both datasets, we observed that the labels were positively correlated to begin with (potentially due to the built-in negative sampling mechanism). We found this to be unrealistic in that generally, an ML system would focus on predicting signals that capture different characteristics of an item. In other words, an industrial application is unlikely to increase the serving complexity by building multiple models that essentially capture the same signal. Hence, we flipped the correlation of the labels and predictions by taking the *opposite* of one of the labels ("viewed" for Modcloth and "rating" for Electronics). The intuition is that now, the labels capture different aspects of "goodness" of the item such that being "good" in one label does not necessarily mean it is "good" in another label.

Beyond the steps above, the batch-sampling, batch-scoring, and batch-utility mechanisms are identical to the functions we constructed in the MovieLens dataset.

D.3.3. Synthetic Data

For the synthetic data, we utilize the Scikit-learn Python package to generate two classification datasets independently. We randomly assign groups to the data and then again train the scoring model via a random forest with default parameters and $n = 300$ trees. As there are no user/session IDs in the synthetic data, we simply collect a batch of random samples at each iteration. The batch sampling mechanism and utility computation that we use in this dataset is identical to the last step to that of the MovieLens data.

D.4. EI and Fair EHVI Details

We use the Botorch implementations of parallel EI (`qExpectedImprovement`) and EHVI (`qExpectedHypervolumeImprovement` for the solution implementation and comparisons. Specifically, the implementation provided in Balandat et al. (2020) and Daulton et al. (2020) allow for parallel EHVI (called q-EHVI). In essence, this algorithm addresses the nonconvexity of the problem by sampling, optimizing, and yielding q points that are expected to yield higher hypervolume. We use $q = 10$ and considered three strategies for picking a single point to use.

1. Pick the point that has the highest expected DER improvement
2. Pick the point that has the highest utility improvement
3. Pick a random point (since all points are expected to improve the hypervolume of the Pareto frontier)

We found that the first strategy worked the best. For random search, we simply generate random alphas to use in each iteration.

Appendix E. Appendix / Role of Unobservable Outcomes

As mentioned in Section 2, one thing we did not explicitly consider in this framework is the role of unobservable outcomes. In practical settings, unobservable outcomes may be factors that are impossible to capture, but nonetheless affect the user's utility (e.g., the user's "agreement" with the company's mission in a job recommendation, the user's opinion of the posting description). They could also refer to things that are technically capturable, but left out for simplicity in the serving framework (e.g. transformations or products of click propensity, view propensity). From an interpretability standpoint, these omissions are reasonable. While one can easily place relative values on things (e.g. "a click means twice the utility to a user compared to a view"), doing so becomes harder when those factors become more abstract. Hence, unobservable outcomes are inevitably in the picture even if they are not directly modeled.

In the context of our framework, one cause of utility gap could be that of the excluded unobservable outcomes, one group places much higher weight on them than another group. Hence excluding them hurts one group more than the other. In the industrial setting, mitigating this requires product-based intervention in similar vein as discussed in Cai et al. (2022). That is, we would recommend the business to conduct user research and segment the analysis across groups to better understand what preferences the users actually have versus what is currently modeled and served. From this lens, our framework can be utilized to quantitatively understand the benefits of product research.