Orbit: A Framework for Designing and Evaluating Multi-objective Rankers

Chenyang Yang[*]
Carnegie Mellon University

Tesi Xiao[†]
Amazon

Michael Shavlovsky[†]
Amazon

Christian Kästner
Carnegie Mellon University

Tongshuang Wu
Carnegie Mellon University

Abstract

Machine learning in production needs to balance multiple objectives: This is particularly evident in ranking or recommendation models, where conflicting objectives such as user engagement, satisfaction, diversity, and novelty must be considered at the same time. However, designing multi-objective rankers is inherently a dynamic wicked problem – there is no single optimal solution, and the needs evolve over time. Effective design requires collaboration between cross-functional teams and careful analysis of a wide range of information. In this work, we introduce Orbit, a conceptual framework for Objective-centric Ranker Building and Iteration. The framework places objectives at the center of the design process, to serve as boundary objects for communication and guide practitioners for design and evaluation. We implement Orbit as an interactive system, which enables stakeholders to interact with objective space directly and supports real-time exploration and evaluation of design trade-offs. We evaluate Orbit through a user study involving twelve industry practitioners, showing that it supports efficient design space exploration, leads to more informed decision-making, and enhances awareness of the inherent trade-offs of multiple objectives. Orbit (1) opens up new opportunities of an objective-centric design process for any multi-objective ML models, as well as (2) sheds light on future designs that push practitioners to go beyond a narrow metric-centric or example-centric mindset.

CCS Concepts

• **Human-centered computing** → **Interaction design**; • **Information systems** → **Information retrieval**.

Keywords

Multi-objective ranking, Recommendation systems, Interactive interface, Design space exploration, Evaluation framework

[*]Work done at Amazon.
[†]Equal contribution.

1 Introduction

Machine learning models are ubiquitous, yet it can be hard to do them right in production [42]. To train a model, it is customary to define appropriate training *objectives*: For a language model, the training objective is to predict next tokens, with every token serving as the training target for the context before. For an image classification model, the training objective is to predict the correct label annotated. However, in many cases, a model has *more than one objective* to train for, with the most prominent example being ranking or recommendation models [54].

Take video recommendation as an example: There are many user behavioral signals that can be used as objectives, from `clicks` and `watch time` to capture user engagement, to `likes` and `ratings` to capture user satisfaction [74]. Beyond recommendation accuracy, objectives like `diversity`, `serendipity`, `novelty`, and `coverage` [30] can also be incorporated into model training, such as to avoid filter bubbles [44], encourage user exploration [62], and ultimately improve long-term user experience. These different objectives can easily conflict with each other [75] and might be prioritized differently by different stakeholders [58]. End users might care more about `likes` and `ratings` to find enjoyable content, while content creators might care more about `diversity` and `novelty` for better video visibility. Advertisers might prioritize `click` to maximize ads visibility, while content creators would favor higher `watch time` for their content. With only one single ranking order that can be produced, there is always a decision to make on what items should go above others – and there is no strict best order for that decision [54].

Designing multi-objective rankers is, therefore, inherently a wicked problem [53]: There is no definitive formulation, no stopping rules, and no single "best" solution. There are always choices on what objectives to incorporate and how to trade off different objectives. Furthermore, this wicked problem of ranker design is *dynamic* with changing stakeholder needs. For example, Youtube's video recommenders evolved from considering only watch time [13], to multiple user behavioral signals capturing user satisfaction beyond engagement [74], and more recently to incorporating diversity as part of the objectives to mitigate echo chamber effect [64]. For a ranking system in production, there are constantly new observations and

feedback from stakeholders that motivate the need for *continuous (re-)design of multi-objectives rankers.*

To thoroughly consider trade-offs for ranker design, practitioners need significant efforts both (a) analyzing and incorporating feedback from different stakeholders and (b) looking across various kinds of evidence.

Communication and collaboration. First, as is typical with many other machine learning systems [42], designing multi-objective rankers is a *cross-functional* efforts: We observed (cf. Section 2.1) that, various stakeholders (e.g., product managers) frequently provide *feedback* on where a ranker can improve, and technical stakeholders (e.g., ML engineers, scientists) will need to translate the feedback into appropriate updates to model objectives and re-design the model. However, it can be challenging for different stakeholders to effectively communicate and collaborate: Less technical stakeholders can struggle to provide actionable feedback, while technical stakeholders have to spend significant efforts to analyze their feedback and incorporate it into the ranker if plausible.

Design and evaluation. Second, designing and evaluating multi-objective rankers is an endeavor involving careful analysis of a rich set of information: We observed (cf. Section 2.2) that stakeholders need to track aggregated *metrics* to understand overall trends of each objective, inspect concrete *examples* to understand users' concrete experiences, and also inspect data *slices* [8] to analyze important subgroups and more nuanced phenomena. Tracking all the information at the same time is challenging, and makes practitioners struggle to design appropriate rankers.

In this work, we propose a conceptual framework, ORBIT, for *Objective-centric Ranker Building and Iteration.* The key idea is that **objectives should take the central role in the model design process, to guide communication, exploration, and evaluation**. We argue that objectives can act as the boundary object [57] between stakeholders, to be interpreted colloquially and connected to stakeholder feedback and concrete examples, and also to be defined precisely in mathematical terms for model training (cf. Section 3). For practitioners designing and evaluating multi-objective rankers, we argue that objectives can help navigate design space and forage information for evaluation, as they define where to explore, inform what to evaluate, and explicate the inherent trade-offs.

We implemented ORBIT as an interactive system that affords interactive ranker design: Users can directly operate on the objective space, and observe how concrete examples and aggregated metrics change in real time, allowing much more efficient and well-informed exploration of the design space. ORBIT also serves as a platform for less techncial stakeholders to better understand multi-objective rankers and potentially provide more constructive feedback in the design process. To evaluate ORBIT, we conducted a user study with twelve experienced industry practitioners. Our evaluation shows that with ORBIT, users can explore the design space more efficiently, make more informed decisions, and are more likely to communicate the inherent trade-offs to other stakeholders.

To summarize, our work makes the following contribution:

- A perspective of multi-objective ranker design as a dynamic wicked problem and its associated challenges identified in practice, which enable new design approaches.

- An objective-centered conceptual framework for multi-objective ranker design that provides a foundation for our and future system design.
- ORBIT, an interactive system supporting interactive ranker design and evaluation.
- Insights from user studies that objective-centered design supports users to explore the design space more efficiently, make more informed decisions, and be more aware of the trade-offs, shedding light on designing similar systems for other multi-objective ML problems.

2 Motivation

We embedded ourselves in a team (n=50) responsible for commercial product rankers over five months. The team members have a wide range of different roles, from applied scientists, software engineers, product managers, to machine learning engineers. The team regularly updates new models on a monthly basis. We conducted informal interviews with team members, studied their existing workflows, as well as analyzed internal documents on past cross-functional communication on the product rankers. Through this process, we identified two key challenges for ranker design in their day-to-day activities, which we summarize below.

2.1 Communication and Collaboration: Lack of a Shared Language

We first found that designing and evaluating rankings is not a one-side effort from technical stakeholders. Lots of less technical stakeholders were involved in the past history and wanted to provide feedback on the ranker to improve user experiences on different data slices and dimensions. This can be particularly helpful, as they often bring in domain expertise and provide feedback grounded in concrete observations and customer experiences. However, even though these stakeholders have the domain expertise and some insights, we found they can struggle to provide *actionable* feedback that can be incorporated in the next model iteration.

Indeed, the process of understanding and incorporating feedback is often perceived to be frustrating and sometimes not constructive – it typically takes weeks of communication efforts. This is because less technical stakeholders only have a vague notion of what objectives the current model is trained for – they communicate what they *want*, without understanding what can be achieved, especially with the constraints of balancing multiple objectives. Meanwhile, incorporating their feedback into model design requires deliberate thinking on how one can translate and express feedback in the objectives, which can take lots of effort for technical stakeholders. The lack of a shared language slows down communication and collaboration, and hinders faster iteration over model design.

This, along with our other observations in Section 2.2, motivates our first design goal:

- G1. *Objective-centric.* **Objectives should be the first class citizen.** The current design process suffers from the lack of a shared language and appropriate guidance. The system should surface objectives as the main object for stakeholders to navigate through the design space, communicate their findings, and negotiate over trade-offs.

2.2 Design and Evaluation: Plethora of Information

To design rankers and evaluate a design, practitioners need to forage various information, from metrics, examples, to slices: They need to track aggregated *metrics* to understand the overall trends of each objective, mostly in a designated dashboard. However, as mentioned by one of the practitioners, *"...(our metric) aggregated NDCG is sparse and not always reliable,"* echoing existing concerns on common information retrieval evaluation metrics [e.g., 18–20, 40], and generally discussion on how aggregated metrics can hide lots of nuances in machine learning [e.g., 52]. Therefore, practitioners also heavily rely on inspecting concrete examples to confirm whether the aggregated metric improvements aligned with human expectations, which is supported by an internal platform to inspect and analyze individual examples. Stakeholders also conduct more customized analyses on important subsets (known as data slices [8]), for which they need to switch to computational notebooks for their expressive power.

This three-fold metric-example-slice information foraging process is much more complicated compared to an idealized machine learning setup, where stakeholders exclusively focus on optimizing models towards a well-defined objective and measure progress through aggregated metrics. In one of our observation sessions, we found the practitioner started with analyzing examples in an explanability tool, but quickly jumped to notebooks for more detailed analysis, and switched to an example analysis platform once an example was found interesting. Because it is mentally challenging to forage comprehensive information for model design, stakeholders often choose to test design hypotheses highly selectively, and only explore a few alternatives per design hypothesis. This leaves a large design space mostly unexplored, and potentially many iteration opportunities missing. Sometimes, they do not have time to conduct comprehensive evaluations, leaving their decision-making up to a few key metrics, and lots of nuances unexplored. These observations motivate our second design goal:

> G2. *Comprehensive-evals.* **Evaluations should be comprehensive and cover different types of information.** Users need evaluation results to assist their model design iteration. The evaluations should be comprehensive, with both qualitative (examples) and quantitative (metrics, slices) information, allowing users to quickly assess a design.

As mentioned above, we also observed that practitioners heavily rely on computational notebooks for any more customized and detailed analysis. Computational notebooks are particularly powerful for their expressiveness, supporting practitioners to experiment with complicated objectives and define appropriate metrics for evaluations. Such customizability is essential to ranker design, motivating our final design goal:

> G3. *Customizability.* **Objectives and metrics should be easily customizable.** Practitioners often want to define refined objectives with complex interactions and customize what to evaluate. The system should support them to customize their design and evaluation, such that they can experiment with different designs and perform in-depth analysis.
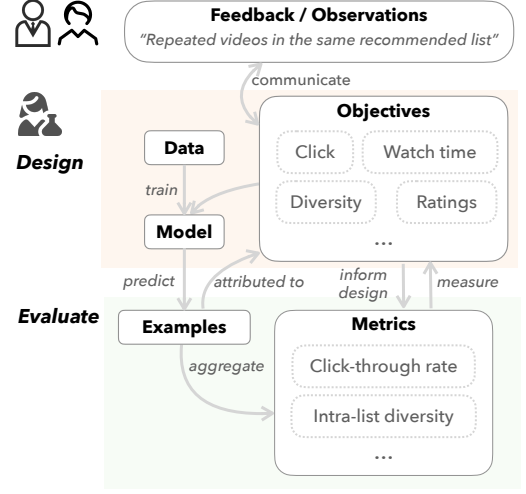

Figure 1: ORBIT's conceptual model. Objectives take the central role of ranker design: They can be translated from feedback and concrete observations, serving as the bridge between different stakeholders. They can help practitioners explore different model designs. They can help conduct evaluation, by informing what metrics to design and track and providing attribution for concrete ranking results.

3 ORBIT

We proposed ORBIT, a conceptual framework for multi-objective ranker design and evaluation, surfacing *objective* as the core concept (Figure 1).

What are objectives? Objectives can be formally defined as *"functions we want to minimize or maximize"* for machine learning [22]. For example, an objective for user click can be defined as a cross-entropy loss function:

$$\mathcal{L}_{\text{click}} = -\frac{1}{N} \sum_{i=1}^{N} \left[r_{ui} \log(\hat{r}_{ui}) + (1 - r_{ui}) \log(1 - \hat{r}_{ui}) \right]$$

where r_{ui} is whether user u clicked item i, $\hat{r}_{ui}$ is the softmax predicted click *probability* for the same user and item, and N is the total number of user-item pairs in the dataset. Model training is expected to optimize for this objective, i.e., minimizing the gaps between predictions and ground truth. In reality, there are usually multiple objectives (e.g., click, purchase, relevance, ratings) that can be defined and optimized for, which requires practitioners to actively explore different designs and trade-offs.

For evaluation, practitioners need to design appropriate metrics for each objective. This is usually a one-to-many mapping, with different metrics capturing different notions of the same objective: For example, MAP [77] and NDCG [28] can both be used for the user click objective here to measure the "goodness" of a produced ranking in terms of clicks, with NDCG capturing additional position information.

Objectives can also be interpreted colloquially and connected to stakeholder feedback and concrete examples: For example, the
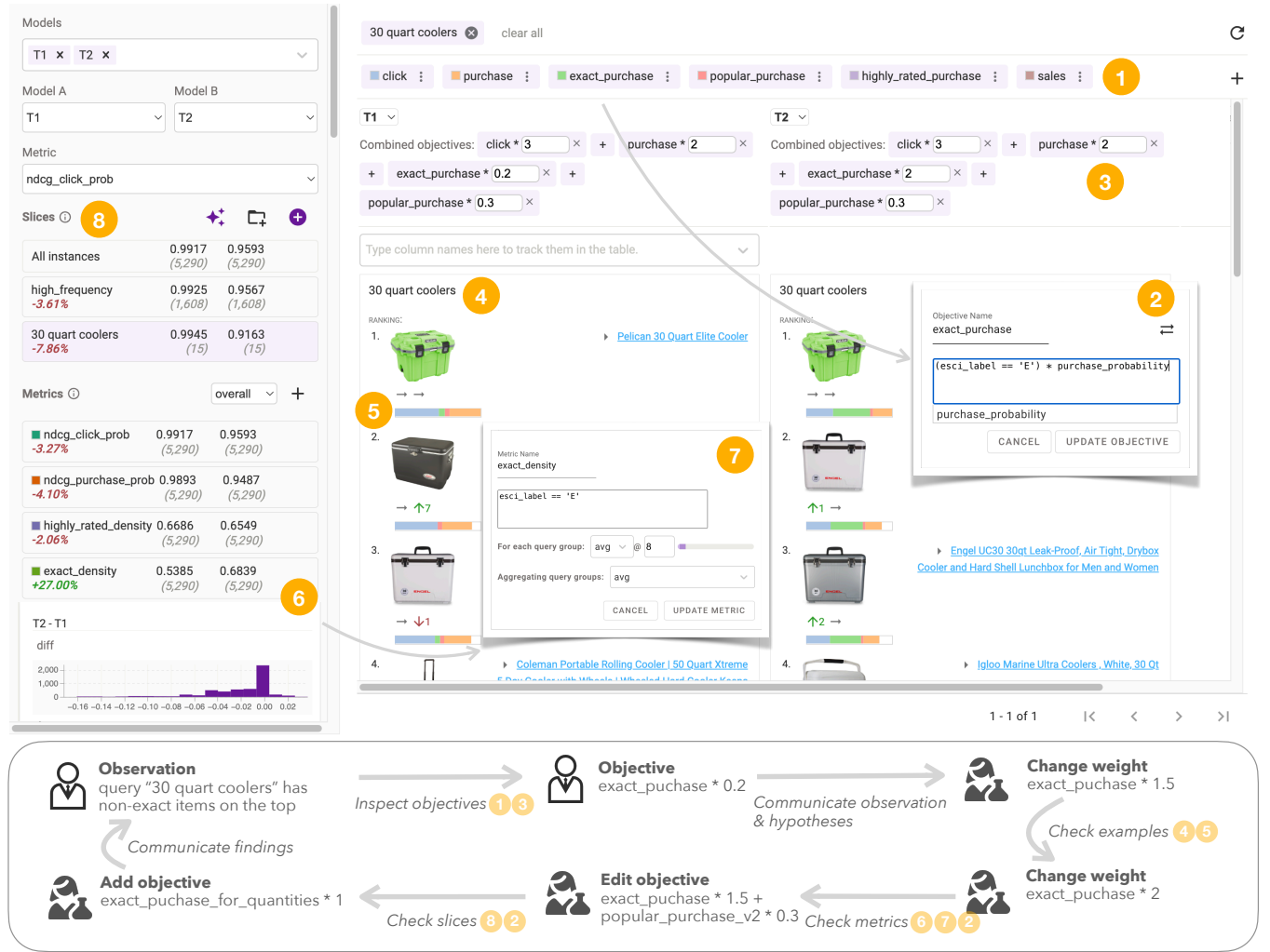
Figure 2: ORBIT's interface and example usage. ORBIT surfaces objectives as the first class citizen in ① objective overview bar, and allows users to interactively ② inspect, edit, or create objectives. Users can ③ specify how multiple objectives are combined and incorporated into a model, and observe the impact in real-time. Users can look at ④ side-by-side comparison for example-level information, and ⑤ tie rankings back to objectives for explanations when needed. Users can also look at ⑥ metrics and ⑧ slices for aggregated information, with the ability to interactively define ⑦ new metrics and new slices, and ⑥ inspect slices with larger metric differences. Below the interface, we demonstrate how in our running examples, stakeholders can use ORBIT to translate observations to actionable feedback, explore different designs, and gather evaluation information.

click objective captures user click information and can be held accountable when stakeholders identify examples where clickbait issues [63] promote attractive-looking yet low-quality items. Effectively, objectives can serve as a boundary object [57] between stakeholders.

Objective-centered ranker design. We argue that multi-objective ranker design should be objective-centered, because objectives can serve as a boundary object for communication, help design space navigation (cf. Section 3.2), and help information foraging for evaluation (cf. Section 3.3).

We implemented ORBIT as an interactive system (Figure 2) based on Zeno [8], a framework that supports interactive behavioral

evaluation. The system was implemented and iterated over three months, with weekly feedback from internal stakeholders. To walk through how ORBIT implements the conceptual framework, we use the following running example:

> Cynthia is an ML engineer responsible for an E-commerce ranking model, which takes in a query and a list of items and returns a ranked list. Currently, the model is optimized for four objectives: `click` for estimated click probability, `purchase` for estimated purchase probability, `exact_purchase` for textual relevance (i.e., whether an item is an exact match) weighted by purchase, and `popular_purchase` for popularity (i.e., units sold) weighted by purchase. Cynthia is working with a product

manager, Eric, to decide whether to incorporate his feedback into the next model iteration.

3.1 Objective as Boundary Object

First, ORBIT explicitly surfaces objectives in the ① **objective overview bar** (**G1**) so that stakeholders can have a shared understanding of what objectives are currently used.

> Eric observes that in a query "30 quart coolers", the current model promotes too many items that are not exact matches in terms of textual relevance (i.e., different sizes), even though users are actively seeking items with a specific size. Eric notices that the model is currently optimized for four objectives: `click`, `purchase`, `exact_purchase`, and `popular_purchase`.

Furthermore, ORBIT supports users to inspect the exact definition of each objective. This helps stakeholders locate objectives relevant to their observations.

> Eric finds the objective most relevant to his observations is `exact_purchase`, defined as `esci_label == 'E'`, where `esci_label` is a feature annotating whether an item is an exact match, substitute, complement, or irrelevant to the current query.

ORBIT is also explicit about how objectives are incorporated into the model with ③ **model definitions** (**G1**), helping stakeholders provide constructive feedback grounded in existing model designs.

> Eric finds the current model is trained using a linear combination of objectives: `click` * 3 + `purchase` * 2 + `exact_purchase` *0.2 + `popular_purchase` *0.3. Eric notices that `exact_purchase` has a weight of 0.2 – he suspects that the current weight is too small to promote all items that are exact matches. Eric communicates this, along with his concrete observations, to Cynthia.

Our example here shows one way to incorporate multiple objectives through pre-training linear aggregation [16]. There are many other ways objectives can be incorporated into the model, from post-training aggregation [74] to heuristic reranking over existing models [11] (cf. Section 6.1 for a more detailed discussion), but the core design of ORBIT stays the same regardless of the training methods.

3.2 Objectives to Support Design Space Navigation

ORBIT not only helps stakeholders provide more constructive feedback on ranker design but also makes the design process itself much easier by helping them navigate the design space. This is supported both by ① **objective overview bar**, which provides the design ingredients, and by ③ **model definitions**, which shows how multiple objectives are combined and further allows users to update the aggregation methods with both smaller changes (weight-tuning) and bigger changes (adding or removing objectives).

> Receiving Eric's feedback, Cynthia decides to update the weight of `exact_purchase` to 1.5 to see if this could help and what the trade-off is. She also tests out a few different designs, including trying different weights of `exact_purchase` and changing thresholds for `popular_purchase`.

ORBIT encourages users to think about design space exploration in terms of objectives, and helps them track what they have explored and brainstorm what to explore next. Furthermore, ORBIT supports ② **interactive objective edits and definitions**, such that users can easily manipulate the objective space (**G3**). This is particularly helpful when they want to experiment with new objectives.

> Motivated by the evaluation results, Cynthia decides relevance is particularly important for queries with queries with quantity information, where users likely target exact items only. She defines and adds an additional objective `exact_purchase_for_quantities` that selectively applies `exact_purchase` to a data slice.

3.3 Objective to Support Information Foraging in Evaluation

Each model design users explore needs to be evaluated. ORBIT highlights a comprehensive set of evaluation information (**G2**) and supports effective information foraging [46] with objectives (**G1**).

Most noticeably, ORBIT helps users forage useful information at example-level with a ④ **side-by-side comparison** view of examples, where users can visually inspect concrete ranking results with rank differences highlighted. Users can also find additional information by tracking additional (objective-related) columns, or expanding a specific item to inspect details (**G2**).

> After updating the objectives, Cynthia checks the examples associated with "30 quart coolers." She finds that this change successfully demotes two non-exact items on the top.

ORBIT explicitly tie example rankings back to objectives, with an ⑤ **objective attribution chart** under each item (**G1**). The attribution chart visualizes how much each objective contributes to a specific item's ranking and can be computed using model explanability techniques [e.g., 5, 51]. Together with objective-related features, objective attribution charts help users identify information that is useful for model iteration. In other words, ORBIT helps enhance "scent" [46] of relevant information for more efficient evaluation.

> Cynthia looks at the example and finds the two non-exact items are demoted specifically because the objective of `exact_purchase` (green in the charts) takes a much bigger portion in the combined objective. She also checks the promoted items, noticing they have medium-level review counts.

Beyond examples, ORBIT also features a ⑥ **metric panel**, which provides an overview over all metrics under tracking. The metrics can usually be derived from objective, though there is not necessarily a one-to-one mapping. In ORBIT, users can easily glance over the metric changes at the dataset, slice, or example level, as well as zoom into specific slices with larger metric differences (**G2**).

> In the current iteration, there are four metrics under tracking: `ndcg_click_prob` and `ndcg_purchase_prob` measure how well highly clicked (purchased) items are placed on the top, with a position decay [28], while `exact_density` and `highly_rated_density` measure how many exact match or highly rated items are present in top-8.

> Cynthia notices that, even though boosting `exact_purchase` indeed promotes textually relevant items to the top (hence higher `exact_density`), other key metrics like `ndcg_pur-chase_prob` and `highly_rated_density` can drop a lot over the entire dataset. Cynthia tests out a few different designs to find one that balances `ndcg_purchase_prob` and `exact_den-sity` relatively well.

More importantly, users can customize any metrics they find useful with ⑦ **metric definitions**, and interactively track them. If users have specific hypotheses, they can even define data slices and track their changes with ⑧ **interactive slicing (G3)** .

> Cynthia further investigates into data slices where the metric `ndcg_purchase_prob` drops most. She notices that for some exploratory user queries that are broad, ambiguous, or even misleading, there is a strong conflict between textual relevance and user purchases, In these cases, users tend to buy a lot of supplementary items, but the new objectives can downrank these items a lot despite frequent user purchases.
>
> Motivated by this observation, Cynthia defines a slice on queries with quantities, where users likely target only for exact items – she finds the model performs well on this slice with new objectives. She further iterates the model design and tests out different designs until she finds a few satisfactory. Cynthia decides to move on with the identified designs, communicates her exploration and analysis back to Eric, and starts a few model training sessions.

4 Evaluation

To evaluate ORBIT, we want to understand how it supports ranker design. As discussed before, ranker design is a wicked problem that requires thinking about and trading off multiple objectives – there is no single success criteria or quality measure, hence, assessing the speed of task completion would be an inadequate measure because it is easy to create a poorly thought out solution quickly with or without tool support. Instead, we want to evaluate how much users explore and evaluate trade-offs, as we consider the depth of engagement as a proxy for their efforts put into creating a well-thought-out and balanced solution. While we do not have any ground truth to evaluate the quality of a solution (which is hardly ever possible for a wicked problem), we can measure how deeply users engage with reasoning about the problem in a given time. We expected ORBIT can support users to explore design space more broadly, conduct more comprehensive evaluations for decision-making, and derive better justifications considering trade-offs.

More specifically, we conducted a user study to evaluate:

- **RQ1 (more efficient navigation)**: To what degree does ORBIT help users explore the design space more easily and efficiently?
- **RQ2 (more informed decision-making)**: How well does ORBIT help users make more informed decision?
- **RQ3 (more trade-off thinking)**: How well does ORBIT encourage users to think about and communicate trade-offs?

4.1 Study Design

4.1.1 Experimental conditions. We design our user study as a within-subject controlled experiment, where participants complete two tasks in two conditions: *treatment* and *control*. In the treatment condition, participants use ORBIT, while in the control condition, participants use Jupyter notebooks, as commonly used in their existing workflow. To make it a fair comparison, we also provide additional utility functions for designing objectives and computing metrics in the control condition, such that the control group is better supported than an average practitioner doing this task.

4.1.2 Procedure. We conducted the study one-on-one with all participants. Each session lasted for 90 minutes and had a structure as follows: The participants first filled out a pre-study survey for demographics and expertise information. Next, the participants went through an interactive tutorial in Jupyter Notebook, where they were introduced to (1) the dataset used in the study, (2) provided notebook utilities for the control condition, and (3) key functionalities of ORBIT. The tutorial serves to equip participants with background information for the study tasks. After the tutorial, the participants were asked to try a demo task with ORBIT, to make sure the participant understood the task and how to use ORBIT. The introductory part took up to 30 minutes.

Next, participants worked on the two tasks for 25 minutes each, one in the treatment condition and one in the control condition. The participants were asked to work on the tasks think-aloud [26], such that we could better understand their thought processes and decision points. To mitigate learning effects, we use a Latin square design [7] with four groups, counterbalancing (1) which condition a participant encounters first, and (2) which task a participant works on first. In the end, participants filled out a post-study survey for their feedback (details in Appendix A).

4.1.3 Tasks. We designed the tasks in the following structure: The participant was first shown feedback on specific model outputs from other (hypothetical) stakeholders. They then had 20 minutes to explore, evaluate, and analyze different model designs think-aloud to understand (1) whether and how the feedback can be incorporated and (2) what are the potential trade-offs. Finally, they had 5 minutes to draft a response to the stakeholder feedback based on their exploration and analysis.

For our user study, we designed two task scenarios with similar difficulty, using the public ESCI dataset [48].[1]

(1) A stakeholder found the query "30 quart coolers" has lots of products with different sizes (e.g., 54 quarts) on the top. These products are not exact matches and might be irrelevant to customers looking for coolers with a specific size.
(2) A stakeholder found the query "uconn hoodie" has lots of products with bad ratings on the top. These products are poorly sold and rated, and can negatively impact customers' shopping experience.

4.1.4 Measurements and analysis.

---

[1]We annotated the dataset with additional synthetic features, ending up with having *text relevance, click-through probability, purchase probability, review ratings, review counts, units sold* as objective-relevant columns.

User activity. We characterize user activities into two categories: design and evaluation. For design, we distinguish between small-step exploration (weight-tuning) and big-step exploration (others), to understand how ORBIT impacts users' design exploration in more nuances. For evaluation, we further break it down into example-based and metric-based evaluations, and distinguish between standard evaluations (dataset-level metrics, provided anecdotes) and additional evaluations (others), to understand how ORBIT impact users' information-seeking behaviors. We re-construct participants' activity sequences from the telemetry data (for treatment) and execution history (for control) collected during their interactions, and one author went through all screen recordings to validate the activity sequences. The final user activity sequences produced are visualized in Figure 4.

For RQ1 (more efficient navigation), we measure how much design space users explore, with **distinct trade-offs (M1)** users explore, which we define as the number of different objectives users design and evaluate. We also measure how many design dimensions users explore, with **distinct big-step trade-offs (M2)**, where users change objective constituents. To understand the complexity of objectives users explore, we additionally measure degree of feature interaction for each new or edited objective.

For RQ2 (more informed decision-making), we measure how comprehensive evaluations users conduct for each trade-off, with **distinct evaluations per trade-off (M3)**. More comprehensive evaluations imply more informed decision-making. We consider each evaluation result that gives new information as distinct – for example, if users look at the same example multiple times for one trade-off, we would consider them as one distinct evaluation.

We further measure the comprehensiveness of users' *overall* evaluation, with **distinct additional evaluations (M4)**, to understand how much users go beyond standard setups of dataset metrics and provided anecdotes. We also measure how balanced users' evaluations are, with **metric-example balance (M5)**, to understand how much users rely on one-sided information. We define this metric as KL-divergence from the uniform distribution:

$$D_{\mathrm{KL}}(Q \parallel P) = Q(e) \log\left(\frac{Q(e)}{P(e)}\right) + Q(m) \log\left(\frac{Q(m)}{P(m)}\right)$$

where $P(e) = P(m) = \frac{1}{2}$, and $Q(e)$ ($Q(m)$) measures the proportion of example-based (metric-based) evaluations. The smaller the metric, the more balanced users' evaluations.

We summarize all our measurements collected from user activity in Table 1. For all these measurements, we also analyzed with a repeated measures ANOVA analysis, testing how much the condition (whether participants use ORBIT) impacts the measurements, while considering the potential impact from other independent variables: In our analysis, we test to what degree our tool, the task, the order (tool first or tool last), and the participant's past experiences explain variance in the outcome variables (cf. Table 2).

User responses. For RQ3 (more trade-off thinking), we annotated users' responses to stakeholders on whether they directly mention trade-offs. We classified any responses mentioning tensions between different objectives (metrics) as directly mentions (e.g.,
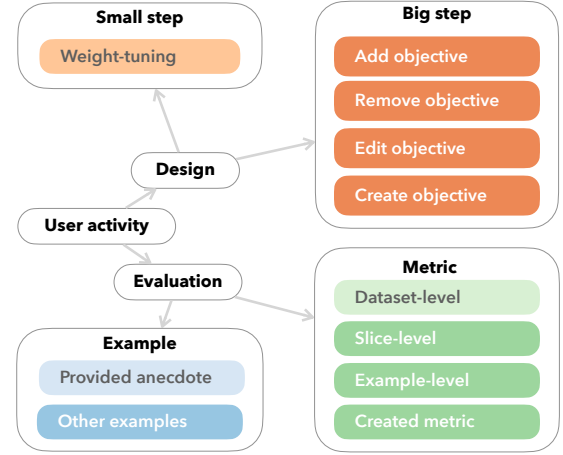

Figure 3: Taxonomy of user activities: We characterize user activities into two categories: design and evaluation. For design, we distinguish between small-step exploration (weight-tuning) and big-step exploration (others), to understand how ORBIT impacts users' design exploration in more nuances. For evaluation, we further break it down into example-based and metric-based evaluations, and distinguish between standard evaluations (dataset-level metrics, provided anecdotes) and additional evaluations (others), to understand how ORBIT impact users' information-seeking behaviors.
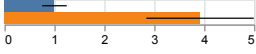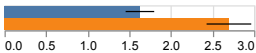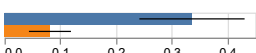
"achieve better [metricA] without compromising [metricB]"). We measure the proportion of responses mentioning trade-offs. In addition, we went through user study transcripts, and identified and counted all verbal mentions of trade-offs.

For our annotations, we have two authors independently annotate a same subset of responses, achieving substantial agreement (0.67 kappa score). One author proceeded with the rest of annotations.

Survey. For all our findings, we also triangulate the findings using users' survey responses, where they rate different aspects of each study condition (trade-off understanding, usability, usefulness, etc.) and the importance of different ORBIT features (e.g., side-by-side visualization, metric tracking). We also quote their written feedback for the relevant findings.

4.1.5 Participants. We recruited 12 participants from a large tech company. All participants have prior ML experience and currently work on product ranking, with 83% of them "extremely familiar" or "very familiar" with Jupyter notebooks in self-report. Their daily work areas span across feature extraction, model training, anomaly detection, offline model evaluation, online A/B testing, etc. As is the standard practice, we pilot-tested the evaluation with 4 participants from the same company, which are not included in the final results.

4.1.6 Limitations. Our user study is designed as a controlled experiment. Controlled experiments give us the power to ensure high confidence in the reliability of the findings in the given context with statistical techniques, but the results might not generalize easily to other tasks, settings, or ML practitioners beyond our participant

Metric	Definition	Hypothesis	Result
distinct trade-offs (M1)	The number of different objectives users design and evaluate	Increased trade-off exploration with ORBIT	
distinct big-step trade-offs (M2)	The number of different objectives users design and evaluate, where objectives are added, removed, or edited	Increased big-step trade-off exploration with ORBIT	
distinct evaluations per trade-offs (M3)	The number of different evaluations users conduct for each trade-off	Increased evaluation per exploration with ORBIT	
distinct additional evaluations (M4)	The number of different additional metric-based or example-based evaluations users conduct	Increased additional evaluation with ORBIT	
metric-example balance (M5)	KL-divergence from the uniform distribution between metric-based and example-based evaluations	More balanced evaluation (**M5** ↓) with ORBIT	

■ treatment ■ control

Table 1: User study metrics and results for RQ1 and RQ2. With ORBIT, participants explored more distinct trade-offs (M1) in bigger steps (M2). They also conducted more distinct evaluation (M3) beyond standard setups (M4) in a more balanced way (M5).

	distinct trade-offs (M1)	distinct big-step trade-offs (M2)	distinct evaluations per trade-offs (M3)	distinct additional evaluations (M4)	metric-example balance (M5)
Interv.: Used ORBIT?	23.81***	8.18*	11.03**	12.16**	6.50*
Task number	7.14*	6.42*	2.71	0.12	3.14
Tool Order	0.34	0.54	0.43	0.01	0.02
Notebook experience	4.97*	0.42	0.19	0.04	0.79

$^{***}p < 0.001, \quad ^{**}p < 0.01, \quad ^{*}p < 0.05$

Table 2: User study ANOVA results: We report the F-value and p-value, which quantify the extent to which each variable accounts for the observed variances. Our analysis reveals that the use of ORBIT significantly explains the differences for all measurements with the biggest impact, while all other variables can not significantly explain the observed variances, except for Task number for M1 and M2, and notebook experience for M1, with smaller F-values.

population. This is a common trade-off when designing evaluations [56] – readers should be careful when generalizing findings beyond our study setting. In addition, the controlled experiments have a short study duration where ORBIT is introduced to participants for the first time, leading to potential novelty effects [33].

Our analysis uses a series of metrics as proxies to measure how users explore design space, conduct evaluation, and think about trade-offs. We do not have a single metric to measure the "goodness" of the derived solutions, as this would require a fixed view of how to prioritize different objectives. Our analysis also relies on human annotations for some metrics, which can be inherently subjective and unreliable – we mitigated the problem with multiple raters to establish annotation reliability.

4.2 Finding: ORBIT Helps Users Explore Design Space More Efficiently (RQ1)

4.2.1 Users explored 183% more distinct trade-offs with ORBIT. We found that on average, users can explore much more *distinct* trade-offs (**M1**) with ORBIT (10.8 vs. 3.8 in control, Table 1). This demonstrates that, given the same amount of time, users can explore the design space more efficiently, correlating with user perception (Table 3) that ORBIT is easier to use (83% vs. 41% for notebooks), and helps them accomplish the tasks more easily (91% vs. 33% for notebooks).

A closer inspection of screen recordings reveals that, in the control condition, users spend much more time foraging example-level information, while in the treatment condition, the side-by-side visualization makes it much faster to gather the same information. This echoes their own perception that side-by-side comparison is the most important feature in ORBIT: 100% participants rate it as "extremely important" or "very important" (Figure 5), as side-by-side comparison *"provides a more visual way to experiment with tradeoffs"* (P4).
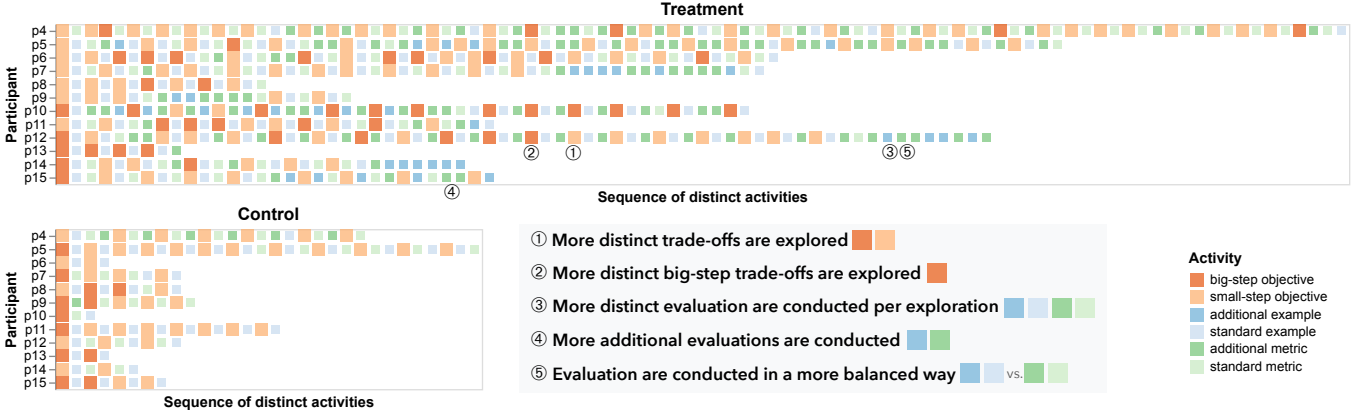
Figure 4: Participant's sequence of distinct activities. With Orbit, participants explored more *distinct* trade-offs ▮▮, in *bigger* steps ▮, and conducted more *distinct* evaluation ▮▮▮ beyond standard setups (▮▮vs. ▮) in a more *balanced* way (▮vs. ▮). Overall participants also explored big-step changes throughout the session with Orbit (vs. mostly only did big-step changes in the beginning followed by small weight-tuning when using notebooks).

4.2.2 Users explored 292% more trade-offs in larger steps with Orbit. Further breaking down the trade-offs users explored, we found that users not only explored more distinct trade-offs but also explored the design space in larger steps: Indeed, they explored 291.7% more *big-step* changes (**M2**, 3.9 vs. 1.0 in control, Table 1) and created or edited 142.9% more objectives (1.4 vs. 0.6 in control). This is well illustrated by the activities of P11 (shown in Figure 4): In the control condition, P11 only defined and added a new objective once, with the remaining time exclusively focusing on weight-tuning, while in the treatment condition, P11 defined, added and edited objectives throughout the process.

Comparing the objectives users defined, Orbit also enabled users to explore more complex objectives that they would not have considered before: Treatment group users are observed to explicitly explore feature interactions (e.g., (esci_label == 'E') · purchase_probability · (review_rating > 4), P10), while control group users exclusively create simple objectives (e.g., esci_label == 'E'). Overall, treatment groups defined or edited 19 objectives with 1.6 interactions on average, while control groups only defined or edited 7 objectives with 1.1 interactions on average.

Combining the results, we found that Orbit helps users explore design space more efficiently, and in the way that they also explore bigger changes and more complex interactions. That is, given the same amount of time, users are able to test out more *divergent* design ideas with Orbit.

4.3 Finding: Orbit Leads to More Informed Decision Making (RQ2)

Beyond more efficient exploration of the design space, we also observed that users on average conducted 65.7% more distinct evaluations per exploration with Orbit (**M3**, Table 1). Participants felt that Orbit helps them *"easy to process (information) and estimate the effects of changes quickly"* (P15) and mentally *"makes it easier to navigate through tradeoffs"* (P12).

Breaking down the evaluations, we found that with Orbit, users are especially encouraged to explore additional evaluations (**M4**,

12.4 vs. 0.6, Table 1), while they almost exclusively focused on standard setups (except for P4) in the control conditions. For example, P12 only checked evaluation results on overall metrics and the provided task example when using notebooks, but conducted a much more extensive evaluation including additional slice-level metrics and additional examples with Orbit (Figure 4). This shows that Orbit not only encourages more frequent evaluation, but also encourages evaluation beyond overall aggregated metrics and fosters generalizability thinking (additional examples).

Meanwhile, we also observed that users are conducting evaluations in a more balanced (**M5**, 75.8% closer to uniform, Table 1) way: In the control conditions, some participants almost exclusively focus on one kind of evaluation – metric-based (e.g., P9) or example-based (e.g., P15). In contrast, with Orbit, participants are more likely to think about both metrics and examples at the same time, because they can easily check *"if the solution works by looking at the (3) metrics block and (4) side by side block, both updated automatically"* (P4), with *"less mental load for human"* (P14).

Additionally, we observed that users are also more likely (+200%) to define new metrics with Orbit. For example, P4 added new metrics to capture different definitions of popularity with different thresholds for review_rating to decide what counts as a popular item, and ultimately found one that is most aligned with his observations. Overall, we found that Orbit leads to more informed decision-making as users forage more, as well as more diverse information when considering trade-offs.

4.4 Finding: Orbit Fosters More Thorough Thinking over Trade-offs (RQ3)

Finally, we found that Orbit leads participants to more easily understand trade-offs (91% vs. 50%, Q2, Table 3) and think more about them when they explore the design space. Participants in the treatment condition mentioned 28 times of trade-offs in total (vs. 15 times in control) when they explored think-aloud. This is expected, as trade-offs are particularly highlighted in Orbit through metrics, hinting users that any design changes are not single-dimensional

	Statement	Distribution	
Q1	Easy to accomplish the task with ORBIT	91%	8%
	Easy to accomplish the task with notebooks	33%	41%
Q2	Easy to understand trade-offs with ORBIT	91%	8%
	Easy to understand trade-offs with notebooks	50%	33%
Q3	Easy to use ORBIT	83%	0%
	Easy to use notebooks	41%	33%
Q4	Enjoyable to use ORBIT	91%	8%
	Enjoyable to use notebooks	33%	41%
Q5	Mentally demanding to use ORBIT	8%	41%
	Mentally demanding to use notebooks	58%	16%
Q6	Will use ORBIT in the future	83%	8%
	Will use notebooks in the future	41%	25%

■ Strongly agree ■ Agree ▢ Neutral ▮ Disagree ▮ Strongly disagree
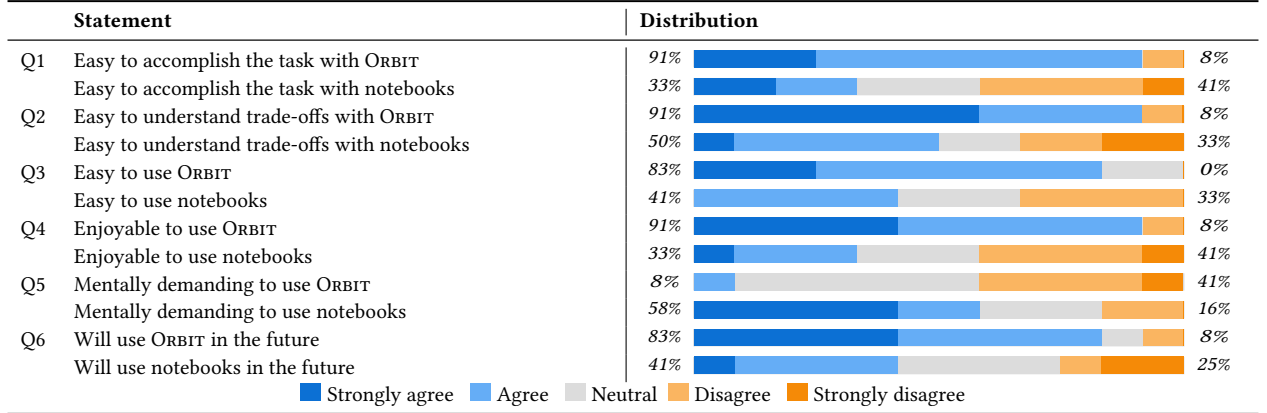
Table 3: Participants found ORBIT help them accomplish the task better (91% vs. 33%), understand trade-offs easier (91% vs. 50%), is easier to use (83% vs. 41%), more enjoyable to use (91% vs. 33%), and less mentally demanding (8% vs. 58%). There are 83% participants who want to use ORBIT in the future for similar tasks (vs. only 41% for notebooks).

optimization efforts, but rather require careful balance. This is well-demonstrated in P11's thought process: *"So even for queries with quantities. We've lost something on exact... given how much purchase_ndcg this cost me on queries with quantities, that doesn't sound super appealing to me" (P11).*

Such trade-off thinking persists from participants' design exploration to communication: We found that participants are 28.6% more likely to explicitly *mention* and *explain* the trade-offs to other stakeholders in their responses with ORBIT, as exemplified in P5's response to the first task: *"There exists trade-off between objectives such as popularity and exact... the weights/objectives suits for keywords with quantities may not perform good on the overall instances" (P5).*
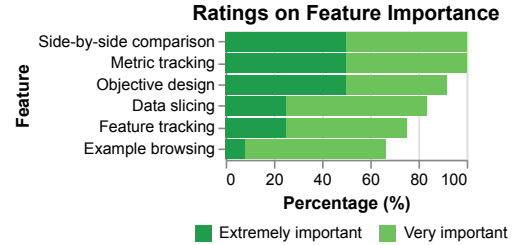

Figure 5: Participants found side-by-side comparison and metric tracking the most important features of ORBIT, followed by objective design and data slicing.

5 Discussion

5.1 (Co-)designing Models with Objectives

In the user study, participants found objectives an important construct for them to navigate through design space. ORBIT supports such objective-guided design space navigation very well, with objectives explicitly surfaced and tied to metrics and item rankings. This, as pointed out, helps participants not only explore design space more efficiently but also explore bigger changes and define more complex objectives.

We found this observation particularly interesting, since users have the same amount of, if not more, freedom to manipulate the objective space in notebooks. This is only possible because ORBIT provides an *"easy-to-use interface for customized objectives" (P9)* and more fundamentally, *"help defines "playing blocks" in the problem" (P4).* We hypothesize that because ORBIT surfaces objectives as a first-class citizen, it prompts users to think and explore in terms of objectives and effectively encourages them to engage with the objective space beyond small changes.

We believe such an objective-centered design approach can generalize beyond ranker or recommender systems, to any ML models considering multiple objectives. Recent research on multi-objective fine-tuning [e.g., 47, 76] is a good example: To properly optimize

LLMs for multiple objectives, such as safety, coherence, and verbosity, at the same time, it is also important to understand what objectives to consider and how to trade off objectives when there are conflicts. Regardless of the exact scenario, ORBIT serves as a foundation for stakeholders to actively engage in objective design, and comprehensively evaluate different design decisions.

Co-designing models across stakeholders. As discussed in Section 2.1, model design is also a collaborative effort requiring the participation of different stakeholders. Our study focuses on a workflow where other stakeholders provide concrete feedback and observations (possibly using ORBIT) and technical stakeholders take most responsibility for design and evaluation, while alternative workflows where different stakeholders collaborate in an more iterative and interactive way are not evaluated.

We envision that ORBIT can be used to support participatory design of ML models [6], as it can empower less technical stakeholders to understand model design and conduct "what-if" analysis through objectives. By shifting the focus from anecdotal problems to objectives, ORBIT encourages different stakeholders to talk in the same language and grounds their communication and collaboration in

concrete shareable analysis. To better support the participatory design settings, future work could further explore how ORBIT can enable greater agency among less technical stakeholders. This includes understanding what additional scaffolding or support might be needed to enhance their contribution to the design process and how these contributions could influence model development iteratively.

5.2 Bridging Metrics-centric and Example-centric Mindsets

In the study, we observed that many participants have a fairly metric-centered mindset – they agonize over metrics drop and are delighted when metrics increase. Such mindsets are prevalent among ML practitioners, but are pointed out to be problematic [61]. In the case of ranking or recommendation systems, it is always important to see what users *see* concretely, beyond aggregated metrics improvement. At the other extreme, less technical stakeholders are often too example-focused, without understanding how the anecdotes can generalize and the greater impact of a fix.

ORBIT encourages users from two extremes to take a more holistic view, by supporting users to forage comprehensive information, including both metrics and examples, in a unified framework. For users with example-centric mindsets, ORBIT always prompts them to think about the larger picture, with metrics and slices information readily available. For users with metric-centric mindsets, ORBIT encourages them to look at concrete observations. Furthermore, ORBIT can be particularly effective in pushing these users to explicitly rethink about metrics (cf. Section 4.3). Instead of thinking about metrics as something existing and inherently valid, users are encouraged to inspect metrics and design new metrics that *align better with their expectations*, all grounded in concrete observations.

Problems with metrics-centric and example-centric mindsets are repeatedly discussed in the literature: End users are often found example-focused and make local decisions that can hurt general performance [68] – this is also observed as a major challenge for prompt engineering [72]. Meanwhile, ML experts are found to focus too much on metrics that do not necessarily align with user-facing performance [23]. The ideas in ORBIT, from interactive metric design to support for comprehensive information foraging, can be readily applied to address similar problems in other machine learning scenarios. We acknowledge that, however, not every scenario would be as straightforward as ranking to gather and present evaluation information to users: Offline metrics can be hard to compute or unreliable for some problems, and more scaffolding might be needed to help users make sense of example results if the model outputs are hard to parse or glance over. Future work extending ORBIT might have to identify the best way to help users forage useful information for their evaluation, depending on their scenarios.

6 Related Work

6.1 Multi-objective Machine Learning

Multi-objective optimization has gained significant attention in machine learning, especially recommendation and ranking systems, due to the need to balance competing objectives such as relevance, diversity, fairness, and user satisfaction. Many techniques have been proposed to address this challenge. One approach is label aggregation, which combines multiple labels into a single supervision target for training [e.g., 9, 14]. Another widely used method is loss aggregation, where different loss functions corresponding to various objectives are merged [e.g., 27, 37, 38, 60]. In addition, post-training score aggregation is commonly employed, where outputs from different tasks in multi-task learning frameworks are combined to generate a final ranking that effectively balances these competing goals [e.g., 74]. Regardless of the exact training method, ML engineers always need to specify appropriate objective formulations and their weights, in order to control the priorities of different objectives to design the final ranking experience, and ORBIT can serve as a framework for ML engineers to explore and evaluate different ranker designs.

Beyond recommendation and ranking systems, many machine learning problems are also effectively multi-objective, as researchers have increasingly paid attention to model qualities beyond accuracy [52]: from fairness [55], robustness [21], to safety [73]. However, most research here focuses on one single (additional) objective at a time, and commonly relies on additional data augmentation [e.g., 15] or more comprehensive data curation (e.g., for LLM instruction-tuning [43]). More recently, multi-objective fine-tuning [e.g., 47, 76] has been proposed to optimize for multiple objectives, such as safety, coherence, and verbosity, at the same time, for LLM generation. We envision that ORBIT can be used as a foundational framework for trading off objectives for general multi-objective machine learning problems.

6.2 Pitfalls of Recommender Systems

Recommender systems are known to exist a series of different biases [10]: There is selection bias [39], where user behavioral signals are sparse and often missing non-randomly, leading to biased predictions. There is position bias [12], where users tend to interact with top items in the list – this can cause a self-reinforcing feedback loop where top items stay at the top with more user interactions. There is also popularity bias [32], where popular items are recommended even more frequently than their popularity – this can reduce the visibility of other items and hurt fairness. Optimizing solely for user behavioral signals can also cause undesired outcomes like filter bubbles [44] or encouraging radicalization [49].

While there are many research works on debiasing recommenders and counterfactual learning [e.g., 1, 2, 29, 36, 70], few have demonstrated the benefits in production environments [e.g., 24, 78]. Nowadays, it is common to inject extra prior knowledge, such as relevance, fairness, and diversity, into the objectives to mitigate biases [e.g., 38, 41, 65, 75]. ORBIT can be used by practitioners to identify what objectives can alleviate the biases, and also decide the trade-off between additional objectives and main objectives.

6.3 Interactive Systems for Machine Learning

Interactive machine learning [17] aims to include humans in ML model construction procedures, by helping humans understand model failures and suggest improvements [e.g., 3, 34]. Researchers have explored different forms of human feedback, from labeling [25] to feature selection [34] for model improvement, which has been found to cause local decision pitfall [68], where users over-generalize from a single observation. Instead of helping non-experts create

better models with one single objective as in interactive machine learning, ORBIT enables stakeholders to understand and explore trade-offs among multiple objectives. ORBIT also avoids local decision pitfalls by presenting users with diverse evaluation information at both the example level and aggregation level.

Another closely related area is tooling support for machine learning. For the evaluation side, existing work has explored designing tools and interfaces to support error analysis [66, 69], data slicing [8, 59], model testing [50, 52, 71], as well as LLM-powered evaluation [31]. For the design side, model sketching [35] enables practitioners to author models from high-level concepts. ConstitutionMaker [45] supports users to convert their feedback to constitutions for chatbots. There is also work on LLM chaining [4, 67] to support users to design LLM workflows. ORBIT borrows ideas from existing work on evaluation (e.g., data slicing), but focuses on the problem of effectively presenting comprehensive evaluation information to users. Compared to existing work for model design, ORBIT is the first to target multi-objective problems and explicitly surface objectives as the key design construct.

7 Conclusion

In this work, we present ORBIT, a framework that places objectives at the center of the model design process. ORBIT allows users to directly engage with objective space, enabling real-time exploration and evaluation of design trade-offs. Our evaluation shows that ORBIT helps practitioners explore the design space more efficiently, make more informed decisions, and are more aware of the inherent trade-offs. ORBIT opens new avenues for an objective-centric design process applicable to other multi-objective machine learning problems, as well as sheds light on future designs that encourage practitioners to think beyond only metrics or examples for evaluation.

Acknowledgments

References

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A general framework for counterfactual learning-to-rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.

[2] Qingyao Ai, Jiaxin Mao, Yiqun Liu, and W Bruce Croft. 2018. Unbiased learning to rank: Theory and practice. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2305–2306.

[3] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 337–346.

[4] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L Glassman. 2024. ChainForge: A Visual Toolkit for Prompt Engineering and LLM Hypothesis Testing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

[5] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research* 11 (2010), 1803–1831.

[6] Eric PS Baumer. 2017. Toward human-centered algorithm design. *Big Data & Society* 4, 2 (2017), 2053951717718854.

[7] George E. P. Box. 2009. *Statistics for experimenters: design, innovation, and discovery*. Wiley-Blackwell.

[8] Ángel Alexander Cabrera, Erica Fu, Donald Bertucci, Kenneth Holstein, Ameet Talwalkar, Jason I Hong, and Adam Perer. 2023. Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[9] David Carmel, Elad Haramaty, Arnon Lazerson, and Liane Lewin-Eytan. 2020. Multi-objective ranking optimization for product search using stochastic label aggregation. In *Proceedings of The Web Conference 2020*. 373–383.

[10] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[11] Sirui Chen, Yuan Wang, Zijing Wen, Zhiyu Li, Changshuo Zhang, Xiao Zhang, Quan Lin, Cheng Zhu, and Jun Xu. 2023. Controllable Multi-Objective Re-ranking with Policy Hypernetworks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3855–3864.

[12] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Joeran Beel. 2018. A study of position bias in digital library recommender systems. *arXiv preprint arXiv:1802.06565* (2018).

[13] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[14] Na Dai, Milad Shokouhi, and Brian D Davison. 2011. Multi-objective optimization in learning to rank. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 1241–1242.

[15] Kaustubh D. Dhole et al. 2021. NL-Augmenter: A Framework for Task-Sensitive Natural Language Augmentation.

[16] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. 2010. Towards recency ranking in web search. In *Proceedings of the third ACM international conference on Web search and data mining*. 11–20.

[17] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 39–45.

[18] Marco Ferrante, Nicola Ferro, and Norbert Fuhr. 2021. Towards meaningful statements in IR evaluation: Mapping evaluation measures to interval scales. *IEEE Access* 9 (2021), 136182–136216.

[19] Marco Ferrante, Nicola Ferro, and Eleonora Losiouk. 2020. How do interval scales help us with better understanding IR evaluation measures? *Information Retrieval Journal* 23 (2020), 289–317.

[20] Marco Ferrante, Nicola Ferro, and Silvia Pontarollo. 2017. Are IR evaluation measures on an interval scale?. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 67–74.

[21] Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness Gym: Unifying the NLP Evaluation Landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*. Association for Computational Linguistics, Online, 42–55.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[23] Mitchell L Gordon, Kaitlyn Zhou, Kayur Patel, Tatsunori Hashimoto, and Michael S Bernstein. 2021. The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[24] Shashank Gupta, Philipp Hager, Jin Huang, Ali Vardasbi, and Harrie Oosterhuis. 2024. Unbiased Learning to Rank: On Recent Advances and Practical Applications. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 1118–1121.

[25] Florian Heimerl, Steffen Koch, Harald Bosch, and Thomas Ertl. 2012. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2839–2848.

[26] Karen Holtzblatt and Hugh Beyer. 1997. *Contextual design: defining customer-centered systems*. Morgan Kaufmann.

[27] Jun Hu and Ping Li. 2018. Collaborative multi-objective ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1363–1372.

[28] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[29] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*. 781–789.

[30] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2016), 1–42.

[31] Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. 2024. Evallm: Interactive evaluation of large language model prompts on user-defined criteria. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–21.

[32] Anastasiia Klimashevskaia, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. 2024. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction* (2024), 1–58.

[33] Michael Koch, Kai von Luck, Jan Schwarzer, and Susanne Draheim. 2018. The novelty effect in large display deployments–Experiences and lessons-learned for evaluating prototypes. In *Proceedings of 16th European conference on computer-supported cooperative work-exploratory papers*. European Society for Socially Embedded Technologies (EUSSET).

[34] Josua Krause, Adam Perer, and Enrico Bertini. 2014. INFUSE: Interactive feature selection for predictive modeling of high dimensional data. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1614–1623.

[35] Michelle S Lam, Zixian Ma, Anne Li, Izequiel Freitas, Dakuo Wang, James A Landay, and Michael S Bernstein. 2023. Model sketching: centering concepts in early-stage machine learning model design. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–24.

[36] Dan Luo, Lixin Zou, Qingyao Ai, Zhiyu Chen, Dawei Yin, and Brian D Davison. 2023. Model-based unbiased learning to rank. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 895–903.

[37] Debabrata Mahapatra, Chaosheng Dong, Yetian Chen, and Michinari Momma. 2023. Multi-label learning to rank through multi-objective optimization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4605–4616.

[38] Debabrata Mahapatra, Chaosheng Dong, and Michinari Momma. 2023. Querywise fair learning to rank through multi-objective optimization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1653–1664.

[39] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267* (2012).

[40] Alistair Moffat. 2022. Batch evaluation metrics in information retrieval: Measures, scales, and meaning. *IEEE Access* 10 (2022), 105564–105577.

[41] Michinari Momma, Alireza Bagheri Garakani, Nanxun Ma, and Yi Sun. 2020. Multi-objective ranking via constrained optimization. In *Companion Proceedings of the Web Conference 2020*. 111–112.

[42] Nadia Nahar, Shurui Zhou, Grace Lewis, and Christian Kästner. 2022. Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process. In *Proceedings of the 44th International Conference on Software Engineering (ICSE)*. ACM Press, New York, NY, 413–425. https://doi.org/10.1145/3510003.3510209

[43] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[44] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.

[45] Savvas Petridis, Benjamin D Wedin, James Wexler, Mahima Pushkarna, Aaron Donsbach, Nitesh Goyal, Carrie J Cai, and Michael Terry. 2024. Constitutionmaker: Interactively critiquing large language models by converting feedback into principles. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 853–868.

[46] Peter L. T. Pirolli. 2007. *Information Foraging Theory: Adaptive Interaction with Information*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195173321.001.0001

[47] Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems* 36 (2024).

[48] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. (2022). arXiv:2206.06588

[49] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. 2020. Auditing radicalization pathways on YouTube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 131–141.

[50] Marco Tulio Ribeiro and Scott Lundberg. 2022. Adaptive testing and debugging of nlp models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3253–3267.

[51] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[52] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 4902–4912. https://doi.org/10.18653/v1/2020.acl-main.442

[53] Horst WJ Rittel and Melvin M Webber. 1973. Dilemmas in a general theory of planning. *Policy sciences* 4, 2 (1973), 155–169.

[54] Mario Rodriguez, Christian Posse, and Ethan Zhang. 2012. Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*. 11–18.

[55] Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive Biases in Natural Language Processing Models: A Conceptual Framework and Overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 5248–5264.

[56] Janet Siegmund, Norbert Siegmund, and Sven Apel. 2015. Views on internal and external validity in empirical software engineering. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 9–19.

[57] Susan Leigh Star and James R. Griesemer. 1989. Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science* 19, 3 (1989), 387–420. https://doi.org/10.1177/030631289019003001 arXiv:https://doi.org/10.1177/030631289019003001

[58] Özge Sürer, Robin Burke, and Edward C Malthouse. 2018. Multistakeholder recommendation with provider constraints. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 54–62.

[59] Harini Suresh, Divya Shanmugam, Tiffany Chen, Annie G Bryan, Alexander D'Amour, John Guttag, and Arvind Satyanarayan. 2023. Kaleidoscope: Semantically-grounded, context-specific ML model evaluation. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–13.

[60] Jie Tang, Huiji Gao, Liwei He, and Sanjeev Katariya. 2024. Multi-objective Learning to Rank by Model Distillation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5783–5792.

[61] Rachel Thomas and David Uminsky. 2020. The problem with metrics is a fundamental problem for AI. *arXiv preprint arXiv:2002.08512* (2020).

[62] Elaine G Toms et al. 2000. Serendipitous Information Retrieval.. In *DELOS*. Citeseer.

[63] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1288–1297.

[64] Yuyan Wang, Cheenar Banerjee, Samer Chucri, Fabio Soldo, Sriraj Badam, Ed H Chi, and Minmin Chen. 2024. Diversifying by Intent in Recommender Systems. *arXiv preprint arXiv:2405.12327* (2024).

[65] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. 2022. A multi-objective optimization framework for multi-stakeholder fairness-aware recommendation. *ACM Transactions on Information Systems* 41, 2 (2022), 1–29.

[66] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2019. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 747–763.

[67] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–22.

[68] Tongshuang Wu, Daniel S Weld, and Jeffrey Heer. 2019. Local decision pitfalls in interactive machine learning: An investigation into feature selection in sentiment analysis. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 4 (2019), 1–27.

[69] Tongshuang Wu, Kanit Wongsuphasawat, Donghao Ren, Kayur Patel, and Chris DuBois. 2020. Tempura: Query analysis with structural templates. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[70] Tesi Xiao, Branislav Kveton, Sumeet Katariya, Tanmay Gangwani, and Anshuka Rangi. 2023. Towards Sequential Counterfactual Learning to Rank. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 122–128.

[71] Chenyang Yang, Rishabh Rustogi, Rachel Brower-Sinning, Grace A Lewis, Christian Kästner, and Tongshuang Wu. 2023. Beyond Testers' Biases: Guiding Model Testing with Knowledge Bases using LLMs. *arXiv preprint arXiv:2310.09668* (2023).

[72] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–21.

[73] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. 2018. DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE 2018)*. Association for Computing Machinery, New York, NY, USA, 132–142.

[74] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM conference on recommender systems*. 43–51.

[75] Yong Zheng and David Xuejun Wang. 2022. A survey of recommender systems with multi-objective optimization. *Neurocomputing* 474 (2022), 141–153.

[76] Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*. 10586–10613.

[77] Mu Zhu. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo* 2, 30 (2004), 6.

[78] Lixin Zou, Haitao Mao, Xiaokai Chu, Jiliang Tang, Wenwen Ye, Shuaiqiang Wang, and Dawei Yin. 2022. A large scale search dataset for unbiased learning to rank. *Advances in Neural Information Processing Systems* 35 (2022), 1127–1139.

A   Post-study Survey Questions

- Please rate the following in terms of how much you agree or disagree with each statement. (Statements listed in Table 3)

- How important were these aspects of working with ORBIT? (Aspects listed in Figure 5)
- For similar tasks in the future, which tool do you prefer using? Why?
- What stood out to you about the experience of using ORBIT? For example, was anything good, bad, surprising, or notable?
- If you want to use ORBIT in the future, what is a scenario you want to use it for?
- If there is one thing you can change about ORBIT, what would you change? (Feel free to write more if you want)