LISTEN to Your Preferences:
An LLM Framework for Multi-Objective Selection

Adam S. Jovine[*1]   **Tinghan Ye**[*2]   **Francis Bahk**[1]   **Jingjing Wang**[1]
David B. Shmoys[1]   **Peter I. Frazier**[1]
[1]Cornell University   [2]Georgia Institute of Technology

Abstract

Human experts often struggle to select the best option from a large set of items with multiple competing objectives, a process bottlenecked by the difficulty of formalizing complex, implicit preferences. To address this, we introduce **LISTEN** (**L**LM-based **I**terative **S**election with **T**rade-off **E**valuation from **N**atural-language), a framework that leverages a Large Language Model (LLM) as a zero-shot preference oracle, guided only by an expert's high-level priorities in natural language. To operate within LLM constraints like context windows and inference costs, we propose two iterative algorithms: **LISTEN-U**, which uses the LLM to refine a parametric utility function, and **LISTEN-T**, a non-parametric method that performs tournament-style selections over small batches of solutions. Evaluated on diverse tasks including flight booking, shopping, and exam scheduling, our results show LISTEN-U excels when preferences are parametrically aligned (a property we measure with a novel concordance metric), while LISTEN-T offers more robust performance. This work explores a promising direction for steering complex multi-objective decisions directly with natural language, reducing the cognitive burden of traditional preference elicitation.

1   Introduction

We consider a human decision maker choosing among a large set of available items, such as airline flight itineraries or schedules assigning employees to shifts.

The decision maker is aided by metrics describing the items (e.g., number of connections, price, total duration, departure time). Unfortunately, the decision maker lacks a precisely-defined utility function over the metrics that would support selecting the best item automatically. Instead, the decision maker must manually inspect many items to select the one they most prefer. This can lead to decision fatigue and poor decisions (Schwartz, 2015).

This problem is common. Surveys across a variety of engineering and science domains (Farzane and Alireza B, 2022; Sharma and Kumar, 2022; Gunantara, 2018) argue that most real-world engineering design problems have multiple competing objectives. Individuals also face many available options in day-to-day life, e.g., in online shopping, especially when combinatorics causes bundles of interacting products to explode in number (flight legs arranged into itineraries, computer components arranged into a server).

Traditional solutions are often inadequate. Manually comparing all options is time-consuming and error-prone. Pareto plots showing one metric versus another do not scale beyond a few metrics. Multi-objective optimization algorithms (see, e.g., Knowles 2006; Branke 2008) allow screening solutions off the Pareto frontier, but when there are many metrics the number of Pareto-optimal solutions can be large. Utility elicitation, preference learning and other algorithmic search methods that require the decision-maker to express pairwise preferences between solutions (Obayashi et al., 2007; Wang et al., 2022; Huber et al., 2025) or human-directed faceted search (Ozaki et al., 2024) can help, but still require significant human effort. The core difficulty is that *humans lack a time-efficient way to accurately articulate their preferences.*

Large language models (LLMs) offer a new paradigm for tackling this challenge. With their ability to interpret text, LLMs present an opportunity for zero-shot preference modeling, where a decision-maker's

---

[*]Equal contribution

goals can be understood directly from a verbal description without the need for expressing pairwise preferences. While recent research has begun integrating LLMs into preference learning, they are typically used as components within larger systems—for instance, to guide questioning (Lawless et al., 2023; Austin et al., 2024), extract preferences from reviews (Bang and Song, 2025), or simulate user behavior (Okeukwu-Ogbonnaya et al., 2025; Zhang et al., 2025). However, the question is understudied of how to best leverage the power of LLMs to accelerate human selections over large collection of items.

1.1 Contributions

To address this gap, we introduce **LISTEN**: **LLM**-based **I**terative **S**election with **T**rade-off **E**valuation from **N**atural-language, a framework that uses an LLM as a preference oracle for selection of a human expert's most preferred item from a list that is too long for the human to examine directly. In our framework, a human expert first describes their potentially complex priorities in natural language. Using iterative refinement the algorithm selects its estimate of the best item. Our approach must contend with limits on the LLM's context window and the LLM's ability to reason directly over large item lists, which prevent the LLM from directly selecting the best item in a single call. Our approach must also limit the number of calls to the LLM to save computational and financial costs.

We introduce two novel algorithms in the LISTEN framework: **LISTEN-U**, which chooses according to parametric utility functions adaptively generated by the LLM; and **LISTEN-T**, which uses the LLM to compare solutions in a tournament-style selection mechanism.

We find that both methods perform as well or better than baselines. **LISTEN-U** significantly outperforms other methods when the held-out human rankings are in concordance with the parametric assumption of the utility function, as measured by a novel concordance metric developed in this work. When the human rankings are not in concordance, this method may underperform unless concordance is improved by modifying the prompts that better guide through the implicit utility or modifying the form of the utility function through adaptive refinements. **LISTEN-T** is not bound by parametric utility assumptions and provides robust performance across a range of problems.

1.2 Related Work

Our work contributes to the rapidly expanding application of LLMs in optimization, yet we address a distinct and often overlooked challenge. Much of the existing research focuses on the *pre-solving phase*, from automatically formulating problems from natural language (Ramamonjison et al., 2023; AhmadiTeshnizi et al., 2023; Xiao et al., 2023; Huang et al., 2025), to configuring solver algorithms (Lawless et al., 2025) and even positioning the LLM as a direct, formulation-free optimizer (Yang et al., 2023). Complementing these efforts, other work evaluates the foundational knowledge of LLMs on core principles, such as primal-dual theory, which is crucial for reliable modeling and education (Klamkin et al., 2025). In contrast to these approaches, our work addresses the *post-solution* challenge by providing a method for users to efficiently filter and navigate the large set of trade-off solutions from a multi-objective optimization, thus complementing the existing pipeline.

Our work is also related to research on how LLMs express preferences in pairwise comparisons, a concept foundational to Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and the "LLM-as-a-judge" paradigm (Zheng et al., 2023). Li et al. (2025) demonstrated that models differ substantially in their logical preference consistency, across dimensions such as transitivity, commutativity, and negation invariance, and that improving this consistency, for example through their REPAIR method, leads to more stable, reliable, and higher-performing systems in downstream ranking and evaluation tasks. At the same time, more recent work warns us of potential pitfalls when treating LLMs as evaluators. Gao et al. (2025) show that when used to simulate or replicate human behavior. Their results suggest that LLMs' internal preference structures and responses are sensitive to extraneous artifacts (such as prompt framing, role assignment, or safety filters) and diverge from human behavior in unpredictable ways. We extend these results to new multi-objective setting, investigating whether these preference instabilities are amplified when navigating the complex trade-offs between competing objectives.

2 Problem Description

We consider the problem of selecting a single preferred item from a large collection of candidates. In many applications, this collection represents the set of Pareto-optimal solutions generated by a multi-objective optimization algorithm. Formally, let $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ be a set of N items. Each item $s_i \in \mathcal{S}$ is described by a tuple of d attributes, $s_i = (a_{i1}, \ldots, a_{id})$. The
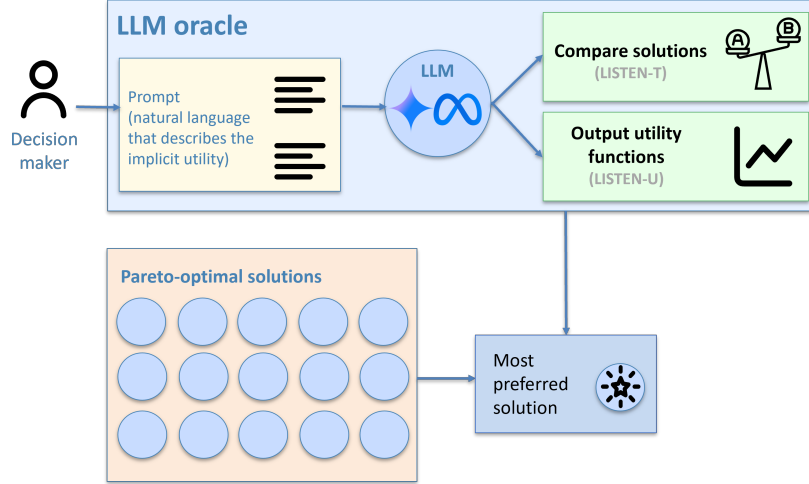
Figure 1: A schematic overview of the LISTEN framework. A human decision maker provides preferences in natural language. An iterative algorithm, either LISTEN-U or LISTEN-T, then uses an LLM as a preference oracle to progressively filter a large set of candidate items (e.g., the Pareto frontier) and identify the single most preferred solution.

attributes can be of mixed types; each attribute a_{ij} belongs to a domain $\mathcal{D}_j$ that can be **numerical** (e.g., $\mathbb{R}$ for price), **categorical** (e.g., a set of airline names), or **textual** (e.g., a free-form product description). We are also given a natural language utterance, U, that articulates a human decision-maker's preferences over these attributes.

Our goal is to identify the item $s^* \in \mathcal{S}$ that best satisfies the preferences described in U. We assume access to a pre-trained LLM, which acts as a preference oracle. The core challenge is to design an algorithm that finds s^* while adhering to a limited budget of calls to the LLM. This setting is motivated by scenarios where N and/or d are large, making it impractical for a human to exhaustively inspect all items and for an LLM to process the entire set in a single context window.

3 Methodology

To solve the selection problem defined in Section 2, we introduce **LISTEN** (**L**LM-based **I**terative **S**election with **T**rade-off **E**valuation from **N**atural-language). LISTEN is an iterative framework that employs an LLM as a zero-shot preference oracle to identify a decision-maker's most preferred item from a large candidate set. The framework begins with a natural language utterance describing the user's priorities and then progressively refines the set of candidates to find the single best option, as illustrated in Figure 1. This iterative design is crucial for navigating large solution spaces while respecting the LLM's inherent context window and query budget limitations.

3.1 Prompting the Preference Oracle

At the core of our framework is the use of an LLM to simulate the decision-maker's preferences in a zero-shot manner. We do not fine-tune the LLM; instead, all queries follow a structured prompt that encapsulates the decision context. Each prompt sent to the LLM consists of five key components:

1. **Persona Context**: Assigns a role to the LLM (e.g., "You are a University registrar").

2. **Metric Definitions**: Defines the attributes of the items (e.g., "Conflicts: students with two exams at the same time").

3. **User Priorities**: The natural language utterance U describing the user's goals (e.g., "prioritize minimizing back-to-back exams").

4. **Solutions**: The candidate item(s) to be evaluated in the current step.

5. **Format Instructions**: Specifies the desired output format (e.g., a JSON object).

This consistent structure provides the LLM with all the necessary context to act as the preference oracle for a given algorithmic step, whether it is refining a utility function or selecting a champion from a batch.

3.2 LISTEN-U: Iterative Utility Refinement

Our first algorithm, LISTEN-Utility (**LISTEN-U**), is a parametric approach. It assumes a linear utility function, $u(s) = \mathbf{w}^\top s^{\mathrm{num}}$, defined exclusively over

the **numerical** attributes of an item (denoted s^{num}). The non-numerical attributes provide context for the LLM's reasoning but are not part of this scoring function. The algorithm iteratively refines the weight vector $\mathbf{w}$ to identify the best item, as detailed in Algorithm 1.

Algorithm 1 LISTEN-U: Iterative Utility Refinement

Require: Solution set $\mathcal{S}$, max iterations T, LLM oracle $\mathcal{L}$, preference utterance U
1: **Initialization:**
2: Construct initial prompt p_1 with U and definitions of all attributes
3: Elicit initial weight vector $\mathbf{w}_1$ for numerical attributes from $\mathcal{L}(p_1)$
4: For each $s_i \in \mathcal{S}$, let s_i^{num} be the vector of its numerical attributes
5: Let $\tilde{s}_i^{\text{num}}$ be the version of s_i^{num} with attributes normalized to $[0, 1]$
6: $s^\star \leftarrow \arg\max_{s_i \in \mathcal{S}} \mathbf{w}_1^\top \tilde{s}_i^{\text{num}}$
7: **Iterative Refinement:**
8: **for** $t \leftarrow 2$ to T **do**
9:     Construct refinement prompt p_t using U, $\mathbf{w}_{t-1}$, and all attributes (numerical and non-numerical) of the *unnormalized* solution $s^\star$
10:     Elicit refined weight vector $\mathbf{w}_t$ for the numerical attributes from $\mathcal{L}(p_t)$
11:     $s^\star \leftarrow \arg\max_{s_i \in \mathcal{S}} \mathbf{w}_t^\top \tilde{s}_i^{\text{num}}$
12: **end for**
13: **return** final weight vector $\mathbf{w}_T$ and solution $s^\star$

The algorithm operates in two phases. In the **initialization phase** (lines 2-6), the LLM is prompted to define an initial weight vector, $\mathbf{w}_1$, for the numerical attributes based on the user's utterance. To score the items, the numerical attributes of every solution in $\mathcal{S}$ are first normalized to a common $[0, 1]$ scale. The algorithm then computes the utility of each item by taking the dot product of the weights and the normalized numerical attribute vector, selecting the highest-scoring item as the initial best solution, $s^\star$.

The **iterative refinement phase** (lines 8-12) then begins. In each iteration, the prompt for the LLM is constructed using the current weight vector $\mathbf{w}_{t-1}$ alongside the complete, **unnormalized** description of the current best solution, $s^\star$, including both its numerical and non-numerical attributes. Presenting the true values and full context is crucial for the LLM to reason about real-world trade-offs (e.g., "the price is too high *for this particular brand*"). The LLM is asked to critique this solution and propose a refined weight vector, $\mathbf{w}_t$. After the LLM returns the updated weights, the algorithm once again scores the entire solution set using the consistently **normalized** numerical attributes to select the new best solution.

3.3 LISTEN-T: Tournament-Based Selection

Our second algorithm, LISTEN-Tournament (**LISTEN-T**), is a non-parametric method that emulates a tournament to efficiently search the solution space. It is designed for a budget of $T \geq 3$ calls to the LLM, as detailed in Algorithm 2.

Algorithm 2 LISTEN-T: Tournament-Based Selection

Require: Solution set $\mathcal{S}$, batch size B, max iterations $T \geq 3$, LLM oracle $\mathcal{L}$, utterance U
1: **Preliminary Rounds:**
2: $m \leftarrow T - 1$ ▷ Number of preliminary rounds
3: $\mathcal{K} \leftarrow \emptyset$ ▷ Initialize set of batch champions
4: **for** $j \leftarrow 1$ to m **do**
5:     $\mathcal{C}_j \leftarrow \text{Sample}(\mathcal{S}, B)$ ▷ Sample a batch of size B
6:     $c_j \leftarrow \text{LLM-Choose}(\mathcal{C}_j, U)$ ▷ LLM selects batch champion
7:     $\mathcal{K} \leftarrow \mathcal{K} \cup \{c_j\}$
8: **end for**
9: **Final Playoff:**
10: $s^\star \leftarrow \text{LLM-Choose}(\mathcal{K}, U)$ ▷ Select winner from champions
11: **return** solution $s^\star$

The algorithm proceeds in two stages using a total of T calls to the LLM. First, in the **preliminary rounds** (lines 2-6), the algorithm conducts $m = T - 1$ rounds. In each round, it samples a batch of B items uniformly at random from the full solution set $\mathcal{S}$ and prompts the LLM to select the single most preferred item from that batch. This winning item, or "batch champion," is added to a set of champions, $\mathcal{K}$.

Second, in the **final playoff** (line 8), the LLM is presented with the set of all m champions collected from the preliminary rounds and is asked to select the single overall winner. By requiring $T \geq 3$, this structure ensures the final playoff is a meaningful comparison between at least two distinct batch champions, allowing the LLM to make a final, decisive trade-off.

4 Experiments

We conduct a series of experiments to evaluate the effectiveness of our LISTEN framework in identifying a user's most preferred item from a large set of multi-attribute options. We ground our evaluation in three realistic decision-making domains: (i) selecting a flight itinerary from a commercial flight search engine, (ii) choosing a product from an e-commerce website, and (iii) scheduling university final exams from a set of Pareto-optimal solutions generated by an optimization solver.

In the following sections, we detail our methodology for collecting real human preference data (Section 4.1), describe the experimental setup (Section 4.2), outline the baseline algorithms for comparison (Section 4.3), and define our evaluation metrics (Section 4.4), before presenting the main results in Section 4.5.

4.1 Human Preference Data Collection

To establish a ground truth for evaluating our algorithms, we created a human preference dataset for each of our three domains. The data was generated by an expert with significant experience in each respective task, following a two-step protocol. First, the expert articulated their decision-making criteria in natural language, framed as a detailed directive to an assistant. This text serves as the preference utterance, U, for each problem. Second, the expert meticulously ranked a subset of items from the full candidate set according to these same preferences. This process yields a paired dataset comprising a high-level natural language goal (U) and a corresponding ground-truth ranked list of items. This ranking is designed to capture the nuanced, implicit trade-offs an expert would make in a real-world scenario, providing a realistic benchmark for evaluating an algorithm's alignment with complex human preferences. The specifics of each dataset are detailed below.

Flights Dataset. We generated a dataset of realistic flight itineraries using the `fast_flights` Python package. For a given origin, destination, and travel date, the package returns a collection of available flights. Each itinerary is described by primary attributes such as airline, departure/arrival times, duration, number of layovers, and price. We augmented these with engineered features, such as the ground travel distance from the airport to a user's specified final location. For our parametric algorithm (LISTEN-U) and relevant baselines requiring numerical inputs, categorical attributes like airline were one-hot encoded; for all LLM prompts, these attributes were retained as their original text to provide full context.

We designed three distinct experimental scenarios to test a range of preferences: **Flights00**: A student flying from Chicago, IL to New York, NY; **Flights01**: A student flying from New York, NY to Chicago, IL; **Flights02**: A professor flying from Ithaca, NY to Reston, VA.

For each scenario, we crafted a unique preference utterance (U) written in conversational English to emulate a real-world directive. For instance, a preference might be articulated as: *"If I fly in on Friday, I prefer to fly in early enough to get a good night of sleep... I prefer a direct flight. I have no preference for airline."*

The full set of preference utterances is provided in the appendix.

Shopping Dataset. Our second domain involves an e-commerce scenario where a college student is shopping for headphones. The preference utterance (U) for this task specifies a desire for a reliable pair of daily-use headphones that are over-ear, wireless, have active noise cancellation, a built-in microphone, and long battery life.

To generate the dataset, we collected public product information for a range of headphones from Amazon.com using the `Scrapingdog` API. Each product is an item in our set $\mathcal{S}$, characterized by a mix of attribute types. **Numerical** attributes include price, average rating, review count, battery life (hours), and weight. **Categorical** attributes include the headphone type (e.g., "Over-ear"), connectivity, noise cancellation mode, and the presence of a microphone. Additionally, each item contains **textual** fields such as the product name, brand, and full description, which provide rich context for the LLM's evaluation.

Final Exam Scheduling Dataset. Our third domain addresses a large-scale university final exam scheduling problem. We generated a set $\mathcal{S}$ of 5,000 diverse, Pareto-optimal schedules using the ParEGO multi-objective optimization algorithm (Knowles, 2006), based on the mixed-integer programming methods in Ye et al. (2024). The quality of each schedule is evaluated against a comprehensive set of metrics to be minimized, as detailed in Appendix. The primary metric is **Direct Conflicts**, counting students with overlapping exams. A suite of metrics addresses **Exam Overload** by penalizing high-density patterns, such as five (*Quints*), four (*Quads*), or three (*Triples*) consecutive exams, as well as near-consecutive patterns (e.g., *Four in Five Slots*). General student fatigue is measured by the total number of **Back-to-Back** exams, which are separated into *Evening-to-Morning* and all *Other* cases. Finally, the **Schedule Spread** is measured by the *Average Time of Last Exam*. Together, these metrics capture the nuanced trade-offs between student workload, fairness, and logistical constraints.

4.2 Experimental Setup

Across all experiments, we use two llms as our preference oracle: `Llama-3.3-70B-Versatile` (Meta AI, 2024) and `Gemini 2.5 Flash-Lite` (Comanici et al., 2025). To ensure the robustness of our results, all algorithmic runs are replicated 50 times with different random seeds for generation while keeping the prompt consistent for each replication.

The ground truth for evaluation is derived from human

Table 1: Dataset statistics and the Concordance metric. Concordance measures the alignment of human preferences with random linear utilities (a lower value indicates a more difficult problem for linear methods). Values for the Concordance column are reported as mean $\pm$ 2SE (approximating a 95% CI).

Dataset	Concordance	Total Items (N)	Ranked Items (m)	Ranked Prop. (m/N)
Exam Scheduling	0.001 ± 0.002	4938	100	0.020
Flights00	0.003 ± 0.004	903	20	0.022
Flights01	0.005 ± 0.005	800	17	0.021
Headphones	0.232 ± 0.027	77	15	0.195
Flights02	0.326 ± 0.030	216	12	0.056

expert rankings. For each dataset, as it is impractical to rank the entire set of N items, an expert manually ranked a subset of the most relevant items ($m \ll N$). This ranked list serves as the ground truth for measuring how well an algorithm's selected item aligns with nuanced human preferences.

4.3 Baseline Methods

We compare the performance of our LISTEN algorithms against the following two baseline methods.

Uniform Random Selection (`baseline/random`). This non-informative baseline selects an item uniformly at random from the full candidate set $\mathcal{S}$. It is used to establish a lower bound on performance and quantify the improvement gained by any guided selection strategy.

Normalized Average Score (`baseline/z-score-avg`). This method considers only the numerical attributes of the items. For each numerical attribute, it first standardizes the values across the entire set $\mathcal{S}$ to have zero mean and unit variance (i.e., calculates the z-score). The scores for attributes that are to be minimized (e.g., price) are then negated. The algorithm selects the item with the highest average standardized score across all numerical attributes. This baseline represents a simple, non-learned linear utility function that weights all normalized metrics equally and ignores all categorical or textual information.

4.4 Evaluation Metrics

We evaluate algorithm performance primarily using the rank of the selected item within the human-generated ground-truth ranking. We also introduce a novel metric to characterize the intrinsic difficulty of each dataset for our utility-based method.

Normalized Average Rank. For a given dataset with N total items, let the ground-truth list contain

m items ranked by a human expert $(1, 2, \ldots, m)$. For an item $s^\star$ selected by an algorithm, its rank is determined as follows: If $s^\star$ is within this top-m list, its rank is its position in that list. If $s^\star$ is not in the list, it is considered unranked. All $N - m$ unranked items are assigned a shared average rank of $(m + 1 + N)/2$, representing the mean of the available ranks from $m + 1$ to N. To ensure a consistent scale across datasets of different sizes, we normalize this rank by dividing by the total number of items, N. The final metric is a value in $(0, 1]$, where a lower score indicates better performance.

Concordance Metric. To measure how well a problem's preference structure aligns with the linear utility assumption of LISTEN-U, we introduce a *concordance* metric. This metric, computed for each dataset independently of any algorithm, quantifies the "difficulty" for a parametric approach. To compute it, we generate 1,000 random linear utility functions, $u(s) = \mathbf{w}^\top s^{\text{num}}$, by sampling each weight w_i independently from $\mathcal{U}[-1, 1]$. The concordance is the fraction of these random functions for which the optimal item, $\arg\max_s u(s)$, is identical to the human expert's top-ranked item. A low concordance value, as seen for the exam scheduling dataset in Table 1, suggests that the human preference is complex and not easily approximated by a simple linear model. Table 1 shows that our datasets span a wide range of concordance values, indicating varying difficulty for linear preference models. This provides a robust testbed for comparing our parametric (LISTEN-U) and non-parametric (LISTEN-T) algorithms.

4.5 Results & Discussions

Our experimental results, presented in Figure 2, highlight the effectiveness of both proposed algorithms, particularly the surprising power of LISTEN-U's iterative refinement process even on complex, low-concordance problems.

(a) Exam Scheduling Dataset.

(b) Headphones Dataset.

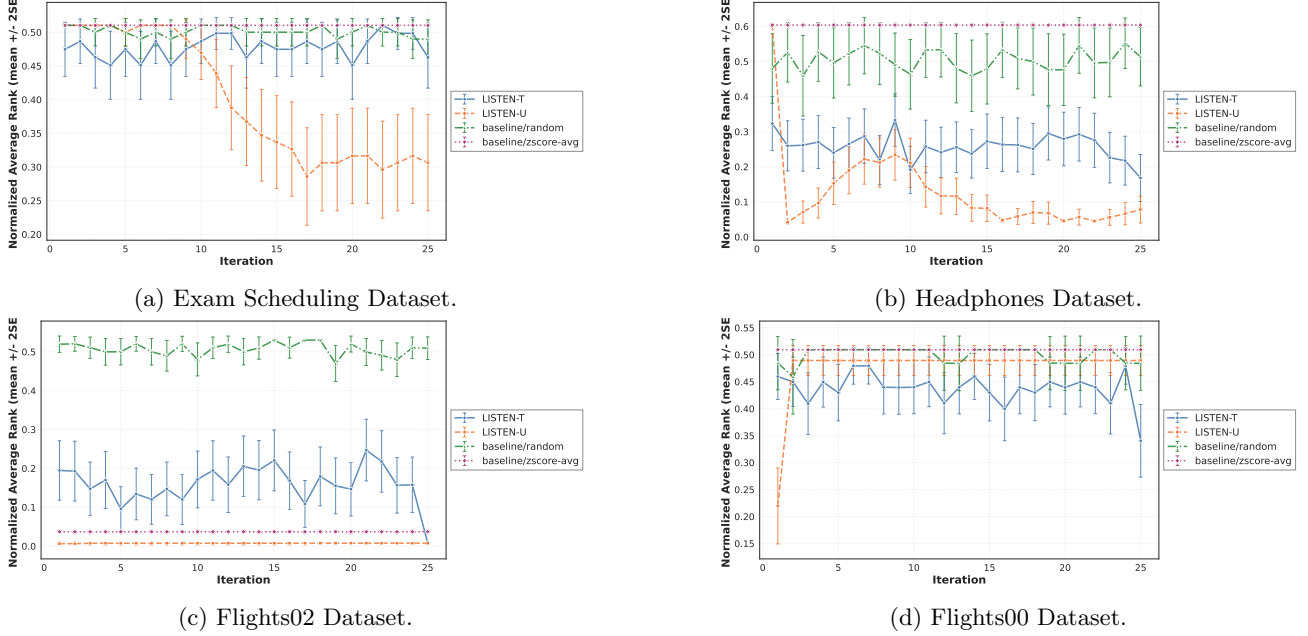(c) Flights02 Dataset.

(d) Flights00 Dataset.

Figure 2: Performance of LISTEN algorithms and baselines on four datasets, showing the Normalized Average Rank (lower is better) over 25 iterations. The plots show results using the Llama model; similar results for the Gemini model are provided in the appendix. The results for the Flights01 dataset, which are similar to Flights00, are also in the appendix.

LISTEN-U's Refinement Excels on Complex Problems. The power of iterative refinement is most evident in the Exam Scheduling results (Figure 2a). Despite this dataset having the lowest concordance score, LISTEN-U starts with performance comparable to the random baseline but progressively refines its utility function to find dramatically better solutions, ultimately achieving the best performance of any method. It also performs exceptionally well on the high-concordance Headphones and Flights02 datasets. The algorithm's only failure is on the Flights00 dataset, indicating its performance can be context-dependent.

LISTEN-T is a Consistently Strong Performer. Our non-parametric algorithm, LISTEN-T, is a robust method that consistently outperforms the random and z-score baselines across all tested scenarios. While it may not always exhibit the same dramatic learning curve as LISTEN-U, it reliably and quickly identifies high-quality solutions, making it a safe and effective choice regardless of the problem's underlying preference structure.

Discussion on Guided Utility Generation. The surprising success of LISTEN-U on the low-concordance Exam Scheduling task suggests that iterative refinement can overcome a poor initial linear approximation. Even if a human's preference is glob-

ally non-linear, the LLM's ability to critique a specific solution and suggest local improvements to the utility weights can effectively guide the search toward a high-quality outcome. This implies that the success of a parametric approach like LISTEN-U may depend not just on the problem's concordance, but also on the ability of the natural language utterance to provide actionable, iterative feedback to the LLM.

4.6 Validating the Concordance Metric

To directly test the relationship between preference complexity and our Concordance metric, we conducted a targeted experiment on the Headphones dataset. We created a more challenging version of the utterance, `Headphones-Strict`, by adding several hard constraints (e.g., a strict budget and required categorical features) to the original `Headphones-General` prompt. This change intentionally introduced non-linearities into the preference structure, causing the Concordance score to drop sharply from **0.232** to **0.053**. This drop is expected, as linear utility functions struggle to model threshold-based requirements.

As predicted by this change in concordance, Figure 3 shows a significant degradation in performance for LISTEN-U on the `Headphones-Strict` version. In contrast, the non-parametric LISTEN-T maintained its robust performance across both prompts. This result validates that our Concordance metric is an ef-

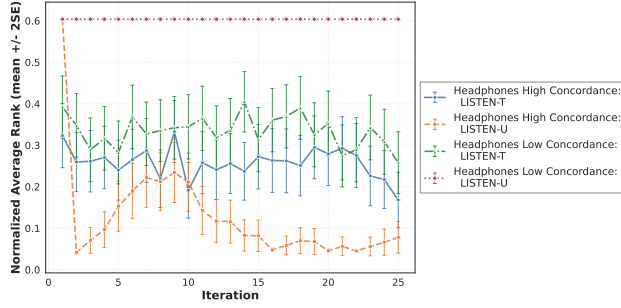fective predictor for the performance of a parametric approach like LISTEN-U.

Figure 3: Performance of LISTEN-U and LISTEN-T on the `Headphones-General` prompt and the modified `Headphones-Strict` prompt, which has lower concordance.
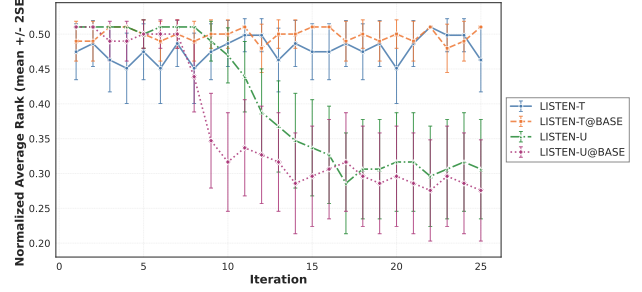
4.7 Ablation Study: Impact of the Preference Utterance

To isolate the value of the user's natural language preference utterance (U), we conducted an ablation study comparing performance with a full, preference-guided prompt against a *base prompt* containing only the persona and metric definitions. The results in Figure 4 show that the benefit of the detailed utterance depends on whether the preferences are intuitive or subjective.
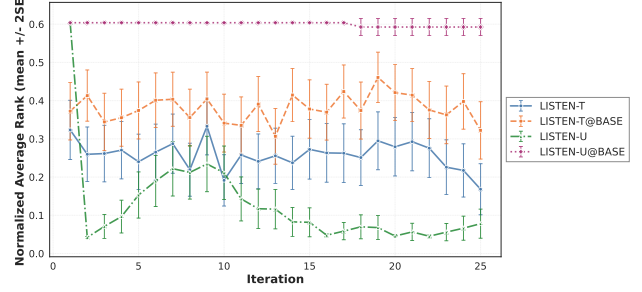
The contrast is most pronounced between the two datasets. For the Headphones task, where preferences are highly subjective, the utterance provides a substantial performance improvement, particularly for LISTEN-U. Conversely, for the Exam Scheduling task, where the goals (e.g., minimizing conflicts) are largely implied by the metrics themselves, the utterance yields negligible benefit. Crucially, we found that including the preference utterance never degraded performance. This suggests that providing more specific, user-aligned context to the LLM is consistently beneficial or, at worst, neutral.

5 Conclusion

In this work, we introduced **LISTEN**, a framework that leverages large language models as zero-shot preference oracles for multi-objective selection problems. We proposed two algorithms: a robust, non-parametric tournament method (**LISTEN-T**) and a parametric utility-refinement method (**LISTEN-U**). Our experiments demonstrate that LLMs can successfully translate high-level natural language goals into high-quality selections from large, complex sets of items. Furthermore, we introduced a novel concordance metric that effectively predicts when the para-

(a) Exam Scheduling Ablations.

(b) Headphones Ablations.

Figure 4: Ablation study evaluating the impact of preference utterance. Performance using the preference-guided prompt is compared to a base prompt containing only the persona and metric definitions. (Llama results; see Appendix for Gemini results).

metric approach is likely to succeed or fail based on the problem's preference structure.

That said, our study has several limitations that present clear avenues for future work.

Subjectivity of Human Rankings. The ground-truth rankings used in our evaluation reflect the preferences of a single expert for each domain and may not generalize broadly. Future work could explore methods for aggregating preferences from multiple users.

Generalization Across Domains. While we evaluated LISTEN on three distinct domains, its performance on a wider range of problems, such as apartment hunting, healthcare scheduling, or logistics planning, remains an open question. Expanding to more varied and larger-scale datasets would further validate our findings.

Richer Utility Representations. The LISTEN-U algorithm's reliance on a linear utility function is a key limitation for problems with highly non-linear preferences. Future work should explore more expressive utility representations (e.g., non-linear or piecewise functions) and hybrid methods that combine LLM guidance with structured optimization.

Acknowledgements

References

AhmadiTeshnizi, A., Gao, W., and Udell, M. (2023). Optimus: Optimization modeling using mip solvers and large language models. *arXiv preprint arXiv:2310.06116*.

Austin, D. E., Korikov, A., Toroghi, A., and Sanner, S. (2024). Bayesian optimization with llm-based acquisition functions for natural language preference elicitation. *arXiv preprint arXiv:2405.00981*.

Bang, S. and Song, H. (2025). Llm-based user profile management for recommender system. *arXiv preprint arXiv:2502.14541*.

Branke, J. (2008). *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., Marris, L., Petulla, S., Gaffney, C., Aharoni, A., Lintz, N., Pais, T. C., Jacobsson, H., Szpektor, I., Jiang, N.-J., Haridasan, K., and (and many authors), . (2025). Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Farzane, K. and Alireza B, D. (2022). A review and evaluation of multi and many-objective optimization: Methods and algorithms. *Global Journal of Ecology*, 7(2):104–119.

Gao, T., Xu, R., Wang, W., and Chen, D. (2025). Take caution in using llms as human surrogates: Scylla ex machina. *arXiv preprint arXiv:2410.19599*.

Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242.

Huang, C., Tang, Z., Hu, S., Jiang, R., Zheng, X., Ge, D., Wang, B., and Wang, Z. (2025). Orlm: A customizable framework in training large models for automated optimization modeling. *Operations Research*.

Huber, F., Gonzalez, S. R., and Astudillo, R. (2025). Bayesian preference elicitation for decision support in multiobjective optimization. *arXiv preprint arXiv:2507.16999*.

Klamkin, M., Deza, A., Cheng, S., Zhao, H., and Van Hentenryck, P. (2025). Dualschool: How reliable are llms for optimization education? *arXiv preprint arXiv:2505.21775*.

Knowles, J. D. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

Lawless, C., Li, Y., Wikum, A., Udell, M., and Vitercik, E. (2025). Llms for cold-start cutting plane separator configuration. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 51–69. Springer.

Lawless, C., Schoeffer, J., Le, L., Rowan, K., Sen, S., Hill, C. S., Suh, J., and Sarrafzadeh, B. (2023). "I Want It That Way": Enabling Interactive Decision Support Using Large Language Models and Constraint Programming. *arXiv preprint arXiv:2312.06908*. Submitted December 12, 2023; revised October 1, 2024.

Li, F. et al. (2025). Aligning with logic: Measuring, evaluating and improving logical preference consistency. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. Poster.

Meta AI (2024). The llama 3 herd of models.

Obayashi, S., Jeong, S., Chiba, K., and Morino, H. (2007). Multi-objective design exploration and its application to regional-jet wing design. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 50(167):1–8.

Okeukwu-Ogbonnaya, A., Amatapu, R., Bergtold, J., and Amariucai, G. (2025). Llm-based community surveys for operational decision making in interconnected utility infrastructures. *arXiv preprint arXiv:2507.13577*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ozaki, R., Ishikawa, K., Kanzaki, Y., Suzuki, S., Takeno, S., Takeuchi, I., and Karasuyama, M. (2024). Multi-objective bayesian optimization with active preference learning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 14490–14498, Vancouver, Canada. AAAI Press.

Ramamonjison, R., Yu, T., Li, R., Li, H., Carenini, G., Ghaddar, B., He, S., Mostajabdaveh, M., Banitalebi-Dehkordi, A., Zhou, Z., et al. (2023). Nl4opt competition: Formulating optimization problems based on their natural language descriptions. In *NeurIPS 2022 competition track*, pages 189–203. PMLR.

Schwartz, B. (2015). The paradox of choice. *Positive psychology in practice: Promoting human flourishing in work, health, education, and everyday life*, pages 121–138.

Sharma, S. and Kumar, V. (2022). A comprehensive review on multi-objective optimization techniques: Past, present and future. *Archives of Computational Methods in Engineering*, 29(7):5605–5633.

Wang, X., Jin, Y., Schmitt, S., and Olhofer, M. (2022). Recent advances in bayesian optimization. *arXiv preprint arXiv:2206.03301*.

Xiao, Z., Zhang, D., Wu, Y., Xu, L., Wang, Y. J., Han, X., Fu, X., Zhong, T., Zeng, J., Song, M., et al. (2023). Chain-of-experts: When llms meet complex operations research problems. In *The twelfth international conference on learning representations*.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. (2023). Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Ye, T., Jovine, A., van Osselaer, W., Zhu, Q., and Shmoys, D. B. (2024). Cornell university uses integer programming to optimize final exam scheduling. *arXiv preprint arXiv:2409.04959*.

Zhang, H., Zhu, Q., and Dou, Z. (2025). Enhancing reranking for recommendation with llms through user preference retrieval. *Proceedings of the 31st International Conference on Computational Linguistics*, pages 658–671.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A   Metrics

This section details the metrics used across the three evaluation scenarios. The specific objectives for the final exam scheduling problem, the flight dataset, and the headphones dataset are documented in Table 2, Table 3, and Table 4, respectively.

Table 2: Final Exam Scheduling Conflict Metrics

Metric	Description
Conflicts	Instances of a student having two or more exams in the same time slot.
Quints	Instances of a student having five exams in consecutive time slots.
Quads	Instances of a student having four exams in consecutive time slots.
Four in Five Slots	Instances of a student having four exams within five consecutive time slots.
Triple in 24h (no gaps)	Instances of a student having three back-to-back exams in a 24-hour period.
Triple in Same Day (no gaps)	Instances of a student having three back-to-back exams on the same day.
Triples	Triple in 24h (no gaps) + Triple in Same Day (no gaps).
Three in Four Slots	Instances of a student having three exams within four consecutive time slots.
Evening/Morning B2Bs	Instances of a student having an exam in the last slot of one day and the first slot of the next day.
Other B2Bs	All other instances of a student having exams in adjacent time slots.
Back-to-backs	Evening/Morning B2Bs + Other B2Bs.
Two in Three Slots	Instances of a student having two exams within three consecutive time slots.
Average Time of Last Exam	Average time slot in which students take their last exams.

Table 3: Flights Metrics

Metric	Description
Name	Name of airline operating the flight.
Origin	Origin airport.
Destination	Destination airport.
Departure Time	Time of departure from origin airport.
Arrival Time	Time of arrival at destination airport.
Duration	How long the flight is.
Stops	Number of layover stops.
Price	Cost of the flight.
dis_from_origin	Distance of origin airport from where the customer prefers (lower is better).
dis_from_dest	Distance of arrival airport from where the customer prfers (lower is better).
departure_seconds	Time of departure since a fixed date in seconds.
arrival_seconds	Time of arrival since a fixed date in seconds.
duration_min	Duration of total flight in minutes (lower is better).

B   Evaluations Using Estimated Utility Functions

In the main paper, we presented Normalized Average Rank, an evaluation methodology that used human rankings of items to assess the quality of LISTEN algorithms. This methodology is advantageous because it uses real human rankings and it does not make parametric assumptions about human preferences. It also has disadvantages, however. Not all items are ranked due to dataset size. Additionally, if an item at rank k is much better

Table 4: Headphones Metrics

Metric	Description
Product Name	Name of the headphone model.
Brand	Manufacturer.
Price	Cost in U.S. dollars.
Type	Headphone design (e.g., over-ear, in-ear, on-ear).
Connectivity	Connection type (wired vs. wireless).
Noise Cancellation	Noise reduction method (active vs. passive).
Battery Life	Battery duration, measured in hours.
Bluetooth Version	Bluetooth version, with higher values indicating newer technology.
Driver Size	Diameter of the audio driver, measured in millimeters.
Weight	Physical weight of the headphones, measured in ounces.
Water Resistance	Protection against dust and water, described qualitatively and via the IPXX rating system (higher values indicate stronger resistance).
Microphone	Presence of built-in microphone.
Review Rating	Average customer rating score.
Review Count	Total number of customer reviews.
Description	Marketing text describing the product.

than the one at $k+1$, the algorithm should be rewarded more for choosing it. Yet the current methodology gives equal credit regardless of how large or small the quality gap is. Also, humans may make errors and rank one item over another despite preferring it less.

To mitigate these disadvantages, we introduce a complementary methodology for evaluating the quality of a LISTEN algorithm. This methodology fits a linear utility function to the human rankings and credits each LISTEN algorithm with the utility of the item selected. Compared to Normalized Average Rank, this has the disadvantage of making parametric assumptions about the nature of the human's preferences but the advantage of ranking all items, providing credit proportional to utility rather than rank, and providing robustness to occasional ranking errors.

To explain this methodology, we first describe the algorithm for fitting the linear utility function from a human ranking (Section B.1), then explain how the utility function is used to compute a score for an algorithm called the Average Utility Score (Section B.2), and finally present results comparing the LISTEN algorithms and baselines using the Average Utility Score metric (Section B.3). In that section, we discuss similarities and differences to trends in Normalized Average Rank seen in the main paper.

B.1 Algorithm for Learning the Utility Weights

The goal of this procedure is to learn a utility function that assigns a numerical score to each item, reflecting its desirability according to human preferences. To achieve this, we fit a linear utility model $u(s) = \hat{\mathbf{w}}^\top s^{\mathrm{num}}$ to the human-provided rank data used to compute Normalized Average Rank, representing each item s with a vector of numerical attributes s^{num} and letting $\hat{\mathbf{w}}$ be a vector of weights to be fit. Non-numerical attributes (such as "brand" in the shopping dataset) are not included in s^{num}. We use maximum likelihood estimation under the assumption that, for any pair of items s_i and s_j, the probability that a human prefers s_i over s_j is given by $P(s_i \succ s_j) = \mathrm{Logistic}\big(u(s_i) - u(s_j)\big)$. This approach allows us to learn the weight vector $\hat{\mathbf{w}}$ in u such that the predicted pairwise preferences align as closely as possible with observed human rankings.

We implement the following algorithm for non-exam datasets using the `scikit-learn` Python package. We randomly generate a total of $N = 10{,}000$ pairwise comparisons (s_i, s_j), split evenly into 5,000 ranked-vs-ranked comparisons (comparisons between two items that were both ranked by the human), where the item with the better human rank is chosen with probability $q = 0.95$; and 5,000 ranked-vs-unranked comparisons, where the ranked item is always chosen.

Algorithm 3 Learning Utility Weights via Logistic Regression

Require: Ranked solution set $\mathcal{S}$, number of simulations N, probability q
 1: Initialize empty lists X, y
 2: **for** $i \leftarrow 1$ to $N/2$ **do**   ▷ Ranked-vs-ranked
 3:     Randomly select two items $s_A, s_B \in \mathcal{S}$ uniformly with replacement
 4:     Identify $s_{\text{better}} \leftarrow$ lower-ranked item, $s_{\text{worse}} \leftarrow$ higher-ranked item, breaking ties at random
 5:     $\mathbf{x}_i \leftarrow s_{\text{better}}^{\text{num}} - s_{\text{worse}}^{\text{num}}$
 6:     With probability q, set $y_i = 1$; else $y_i = 0$
 7:     Append $\mathbf{x}_i, y_i$ to X, y
 8: **end for**
 9: **for** $i \leftarrow 1$ to $N/2$ **do**   ▷ Ranked-vs-unranked
10:     Randomly select $s_A \in \mathcal{S}$ and $s_B \notin \mathcal{S}$, uniformly with replacement
11:     $\mathbf{x}_i \leftarrow s_A^{\text{num}} - s_B^{\text{num}}$
12:     $y_i = 1$ ▷ Ranked always preferred
13:     Append $\mathbf{x}_i, y_i$ to X, y
14: **end for**
15: Fit logistic regression: $P(y_i = 1) = \text{Logistic}(\alpha + \hat{\mathbf{w}}^\top \mathbf{x}_i)$
16: Return learned weights $\hat{\mathbf{w}}$

The algorithm operates in two phases. In the **pairwise simulation phase**, we first generate comparisons between items. For ranked items, two solutions are randomly selected, and the one with the better human rank is labeled as s_{better}. A binary label y_i is set to indicate whether the "better" item is chosen, and a feature difference vector $\mathbf{x}_i = s_{\text{better}}^{\text{num}} - s_{\text{worse}}^{\text{num}}$ is computed. Next, for ranked-versus-unranked comparisons, a ranked item s_A and an unranked item s_B are randomly selected. The ranked item is always treated as "better," and the corresponding $\mathbf{x}_i$ vector is constructed with $y_i = 1$. These simulations produce a large dataset of $(\mathbf{x}_i, y_i)$ pairs that reflect human preferences while introducing controlled stochasticity for ranked items.

In the **logistic regression phase**, the algorithm fits a model to the simulated data. Specifically, the probability of choosing the "better" item is modeled as $P(y_i = 1) = \text{Logistic}(\alpha + \hat{\mathbf{w}}^\top \mathbf{x}_i)$, where $\hat{\mathbf{w}}$ is the vector of learned metric weights and α is the intercept, typically close to zero. Once trained, this weight vector $\hat{\mathbf{w}}$ is used to compute the utility of each item in the dataset, including previously unranked items, as $u(s) = \hat{\mathbf{w}}^\top s^{\text{num}}$. This allows all items to be ranked according to learned human preferences, providing a continuous, utility-based ordering that reflects the observed rankings.

In all datasets except the exam dataset, the human-ranked items are strictly ordered. Thus, there are no ties in the pairwise simulation phase. In the exam dataset, however, due to its size and the effort required to rank items, the top $K = 100$ items are considered to have the same rank. Thus, in the exam dataset, the resulting utility function predicts the likelihood that a given item belongs to the top tier, rather than its relative position within the ranked set.

B.2   Average Utility Score

In the previous section, we trained a model to predict the probability that one item is preferred over another based on its features. This process yields a hidden weight vector, which we use to assign a utility score to every item. At each iteration of an algorithm, we compute the utility of the item it selects. We then average this across replications to obtain the *Average Utility Score*. This metric quantifies the quality of each algorithm, allowing us to distinguish between algorithms that select mediocre (unranked) items from those that select poor unranked items. It thus provides a continuous, utility-based ranking of algorithms.

B.3   Utility Plots (Llama)

This section discusses results presented in Figure 5 and their similarities and differences to results in the main paper's Figure 2. Here, for brevity, we refer to Normalized Average Rank as NAR and Average Utility Score as AUS.

(a) Exam Scheduling Dataset.

(b) Headphones Dataset.

(c) Flights02 Dataset.

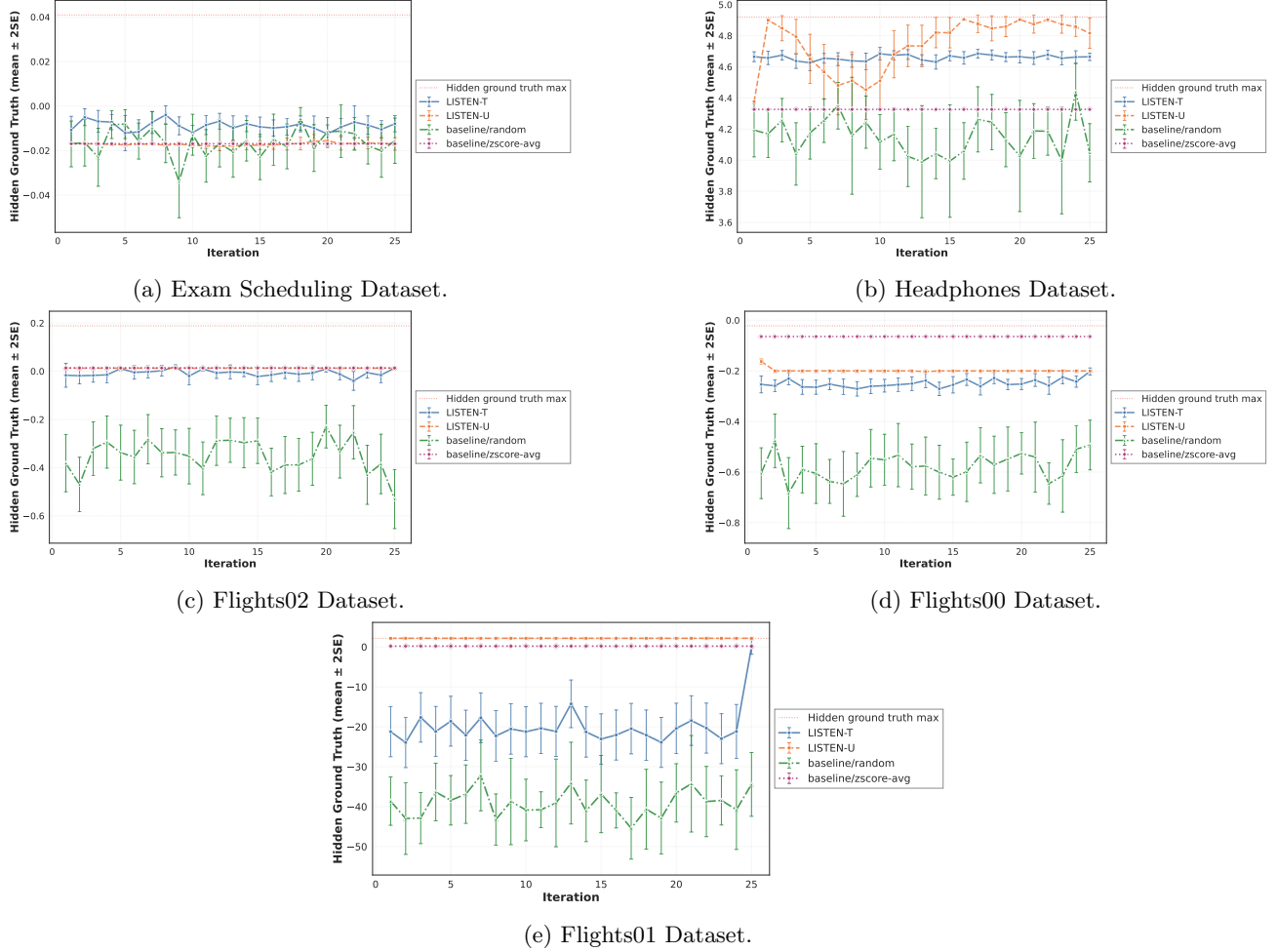(d) Flights00 Dataset.

(e) Flights01 Dataset.

Figure 5: Performance of LISTEN algorithms and baselines on five datasets, showing the Average Utility Score (higher is better) over 25 iterations. The plots show results using the Llama model.

High-Concordance Datasets. We first discuss results from Figure 5 for Flights02 and the Headphones dataset, which exhibited the highest concordance. These are the datasets for which human preferences align best with the linear utility model and where we most expect AUS and NAR to agree. In line with this expectation, the order and curves of the algorithms remain consistent when ranking performance by NAR versus AUS.

Low-Concordance Datasets. In contrast to high-concordance datasets, Figure 5 focuses on the Exam, Flights00, and Flights01 datasets, which exhibit the lowest concordance. In these cases, human preferences do not align well with the linear utility assumption, and thus the AUS and NAR metrics are expected to deviate. In Figure 5, for the Exam dataset, LISTEN-U selects solutions that yield little improvement in utility, even though their ranks consistently decrease, as shown in Figure 2 (main text). Similarly, for Flights00, LISTEN-U attains higher average utility scores than LISTEN-T, but in the main paper, LISTEN-U repeatedly selects unranked solutions, only marginally outperforming the baselines and performing worse than LISTEN-T. The Flights01 dataset exhibits a similar pattern: LISTEN-U achieves the highest average utility but remains the lowest-performing algorithm according to human rankings. The differences between Figure 2 (main text) and Figure 5 arise from low concordance. When human preferences are poorly captured by linear utility models, performance metrics derived from these models may diverge from those based directly on human preferences. In this sense, the average utility score serves as a useful complementary metric for datasets with low concordance.

C   Additional Results on Flights01

This section presents additional data, in Figure 6, assessing algorithm performance using Normalized Average Rank on an example problem, Flights01, that did not fit in the main paper. This example problem is similar to an example problem that did appear in the main paper, Flights00. Both Flights00 and Flights01 are single-leg flights and they share similar prompt structures with minor differences in preference.

By comparing Figure 6 here and Figure 2d (main text), we see that the experimental results are similar. LISTEN-T outperforms the other algorithms in both problems. Also, LISTEN-T improves significantly at 25 iterations, whereas the other three algorithms hover around 0.5. One small difference between the results of Flights00 and Flights01 is that Flights00 in the main paper's Figure 2d shows a slight difference between the performance of LISTEN-U and z-score average, but Flights 01 as shown in Figure 6 does not.

Figure 6: Performance of LISTEN algorithms and baselines on Flights01 dataset, showing the Normalized Average Rank (lower is better) over 25 iterations. The plots show results using the Llama model.

D   Additional Ablation Results

We showed in Figure 4 (main text) that preference utterance never degraded performance in the headphones and exam datasets. In Figure 7 below, which shows ablations for all the flight datasets, we see that incorporating guidance through human preferences significantly improves the performance of LISTEN-T and marginally improves LISTEN-U. The difference in performance gains might be due to the fact that LISTEN-U already benefits from its built-in, iterative, utility refinement guidance. Nevertheless, Figure 7 shows that preference utterance always beneficial, or at worst, neutral, which agrees with the main paper.

E   Results from Gemini-2.5-flash-lite

As mentioned in the main paper, experimental runs were also completed using `Gemini 2.5 Flash-Lite`. This sections details the results from those runs that were not included earlier.

E.1   Human Rank Plots (Gemini)

This section presents the plots obtained from the same experimental setup as Figure 2 (main text), but run using Gemini instead.

The first plot of Figure 8 shows very similar results to the results in Figure 2a, with LISTEN-U outperforming the other methods significantly. Plot 8b also behaves similarly to Plot 2b, with LISTEN-U outperforming the rest of the methods. The other three algorithms also follow a similar trend to the ones run by Llama, with LISTEN-T outperforming both baseline algorithms. Plot 8c shows a very similar trend to Plot 2c, with the main difference being that the z-score average in Gemini outperforms LISTEN-U slightly, but LISTEN-U outperforms z-score average in the Llama runs. Plot 8d is also fairly similar to Plot 2d, with LISTEN-T outperforming the other algorithms; however, there is no difference in performance in z-score average and LISTEN-U when run by Gemini, but there is a difference when run by Llama (LISTEN-U is slightly more effective). It is worth pointing out that in the Gemini-runs, z-score average and LISTEN-U seemed to do equally poorly, whereas in the Llama-runs, LISTEN-U very slightly beat z-score average. This could be chalked up to differences in LLM,

(a) Flights02 Ablations.

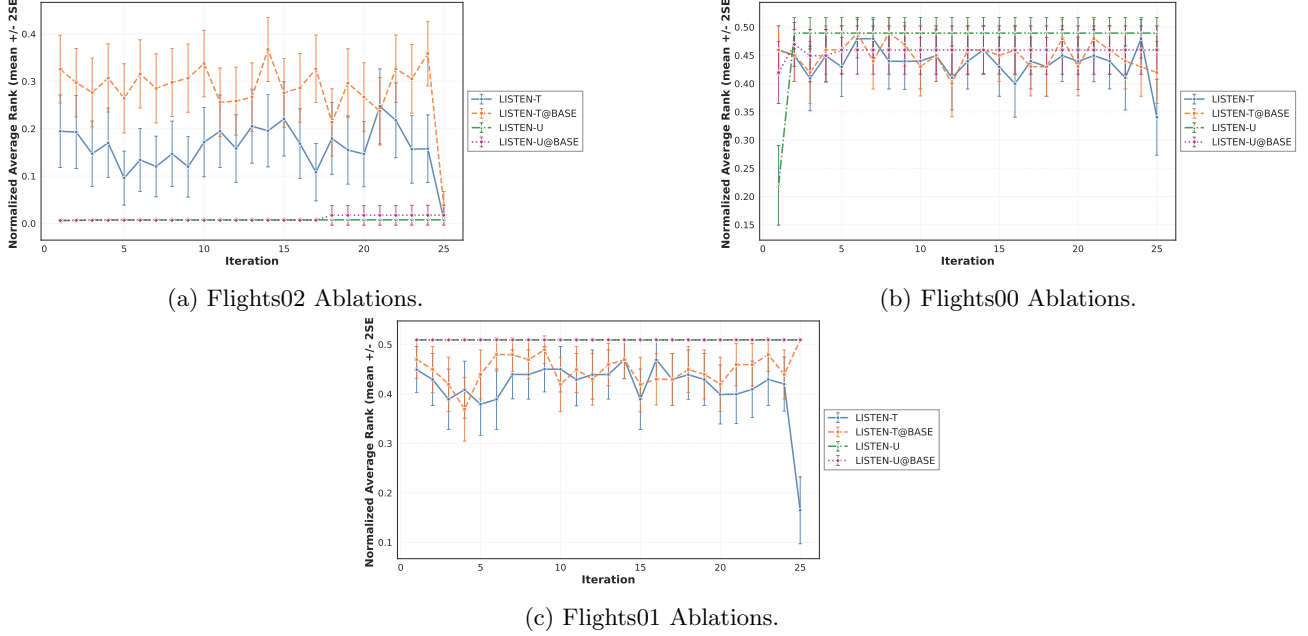(b) Flights00 Ablations.

(c) Flights01 Ablations.

Figure 7: Ablation study evaluating the impact of preference utterance. Performance using the preference-guided prompt is compared to a base prompt containing only the persona and metric definitions. The plots show the Normalized Average Rank (lower is better) over 25 iterations, using the Llama model.

but could also be a result of the Llama-based run being able to find just one ranked solution. Since the line stays flat, we know that other than one (or a small number) of ranked solutions, the algorithm was not able to find more solutions that were deemed "ideal" by the human. This suggests that Gemini and Llama result in very similar outcomes with the same prompt given. Lastly, the results in 8e are fairly similar to 8d, with LISTEN-T being the best performing algorithm.

E.2 Utility Plots (Gemini)

In this section, we describe similarities and differences between Gemini NAR in Figure 8 and Gemini AUS in Figure 9, showing that Gemini preserves the key observation (made with Llama in Section B.3) that ranking-based and utility-based performance tend to correspond only in high-concordance datasets. Then, we describe the similarities and differences between the utility plots made with Llama in Figure 5 and utility plots made with Gemini in Figure 9.

First, we evaluate algorithm performance on high-concordance datasets, namely Flights02 and Headphones. In Figure 9, the order of algorithms based on their final iteration utility is LISTEN-U, LISTEN-T, followed by the two baselines—random and zscore-avg. This same ordering appears in Figure 8. As expected, high-concordance datasets preserve the ranking of algorithms, and the performance curves align closely across both figures for both datasets. Conversely, the ordering is not preserved for low-concordance datasets such as Exam, Flights00, and Flights01. For example, in Figure 8 for Exam, LISTEN-U outperforms LISTEN-T, whereas in Figure 9 it performs worse. For both Flights00 and Flights01, LISTEN-T ranks highest according to human-preference evaluations but performs worse than the zscore-avg baseline when measured by utility. Moreover, the performance curves of the algorithms align less closely than in the high-concordance case.
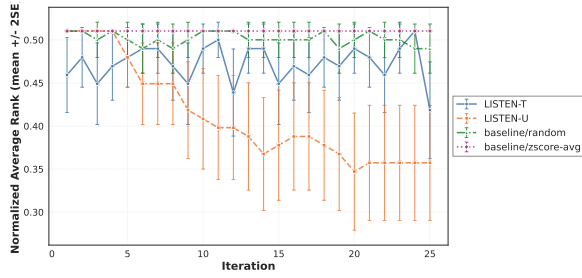
The utility plots generated with Llama in Figure 5 and with Gemini in Figure 9 are largely similar, with only two minor differences. First, for Flights01, LISTEN-U performs noticeably better with Llama than with Gemini. Second, in Flights00, LISTEN-U outperforms LISTEN-T when using Llama, but underperforms relative to LISTEN-T with Gemini.
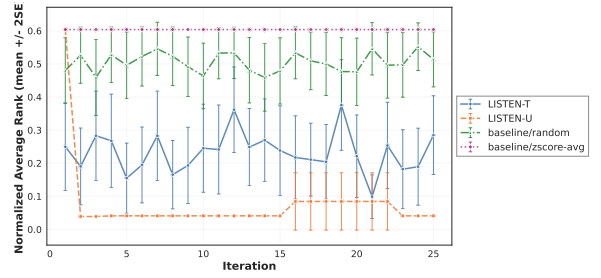
E.3 Ablation Plots (Gemini)

This section includes ablation plots for the experiments run with Gemini.

It is interesting to highlight that when using Gemini, despite the relative non-subjectivity of the exam scheduling metrics, LISTEN-U (with preferences) does perform better than both LISTEN-Ts and LISTEN-U@BASE (Figure 10a). When run with Llama, the LISTEN-U@BASE performed similarly to LISTEN-U. In this case, LISTEN-U@BASE does not perform as well. This could be chalked up to the differences in the LLMs; however, this goes to show that our previous claim that preferences only help improve the results holds true. On the other hand, as seen in Figure 10b, the headphones data set performs quite similarly whether run with Gemini or Llama. Similar to the Llama results, LISTEN-U@BASE is the worst by far, and LISTEN-U slightly outperforms the other two algorithms. The prompts with subjective preferences do better with these preferences laid out in natural language. This further supports the idea that including preferences can only help improve the results and will not degrade them.
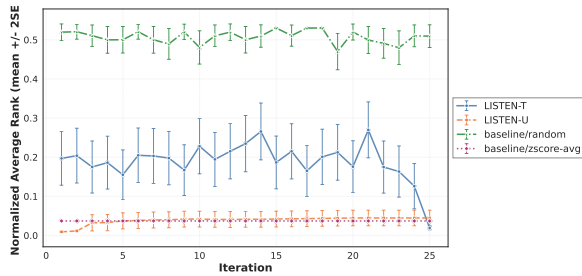
Now, Figure 10c demonstrates a similar trend as Figure 7a, which was run on Llama. Figure 10d also shows a relatively similar trend to Figure 7b (run on Llama); however, the Gemini-based LISTEN-T (both BASE and non-BASE) performs slightly better than in Llama. From this run and previous results, we can see that there is often a slight difference between the effectiveness of different LLMs. Even when given the same prompt, Gemini seemed to performing slightly better than its Llama counterpart. Lastly, Figure 10e closely resembles Figure 7c as well, with LISTEN-T following the same trend as observed in Llama.
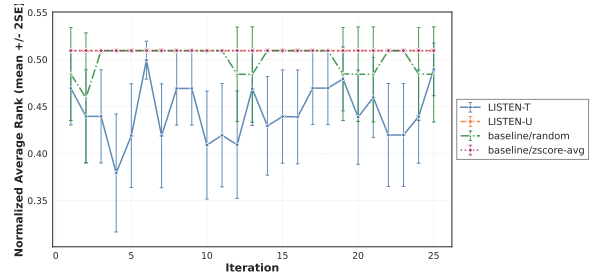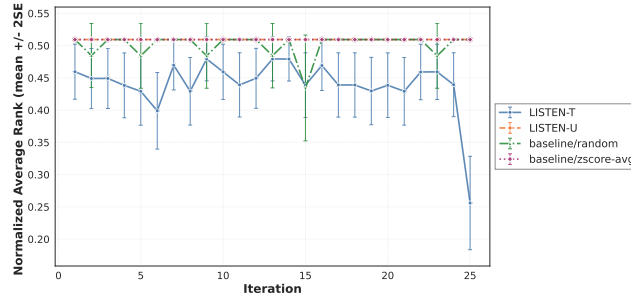

(a) Exam Scheduling Dataset (Gemini).

(b) Headphones Dataset (Gemini).

(c) Flights02 Dataset (Gemini).

(d) Flights00 Dataset (Gemini).

(e) Flight01 Dataset (Gemini).

Figure 8: Performance of LISTEN algorithms and baselines on five datasets, showing the Normalized Average Rank (lower is better) over 25 iterations. The plots show results for the Gemini model.

(a) Exam Scheduling Dataset (Gemini).

(b) Headphones Dataset (Gemini).

(c) Flights02 Dataset (Gemini).

(d) Flights00 Dataset (Gemini).
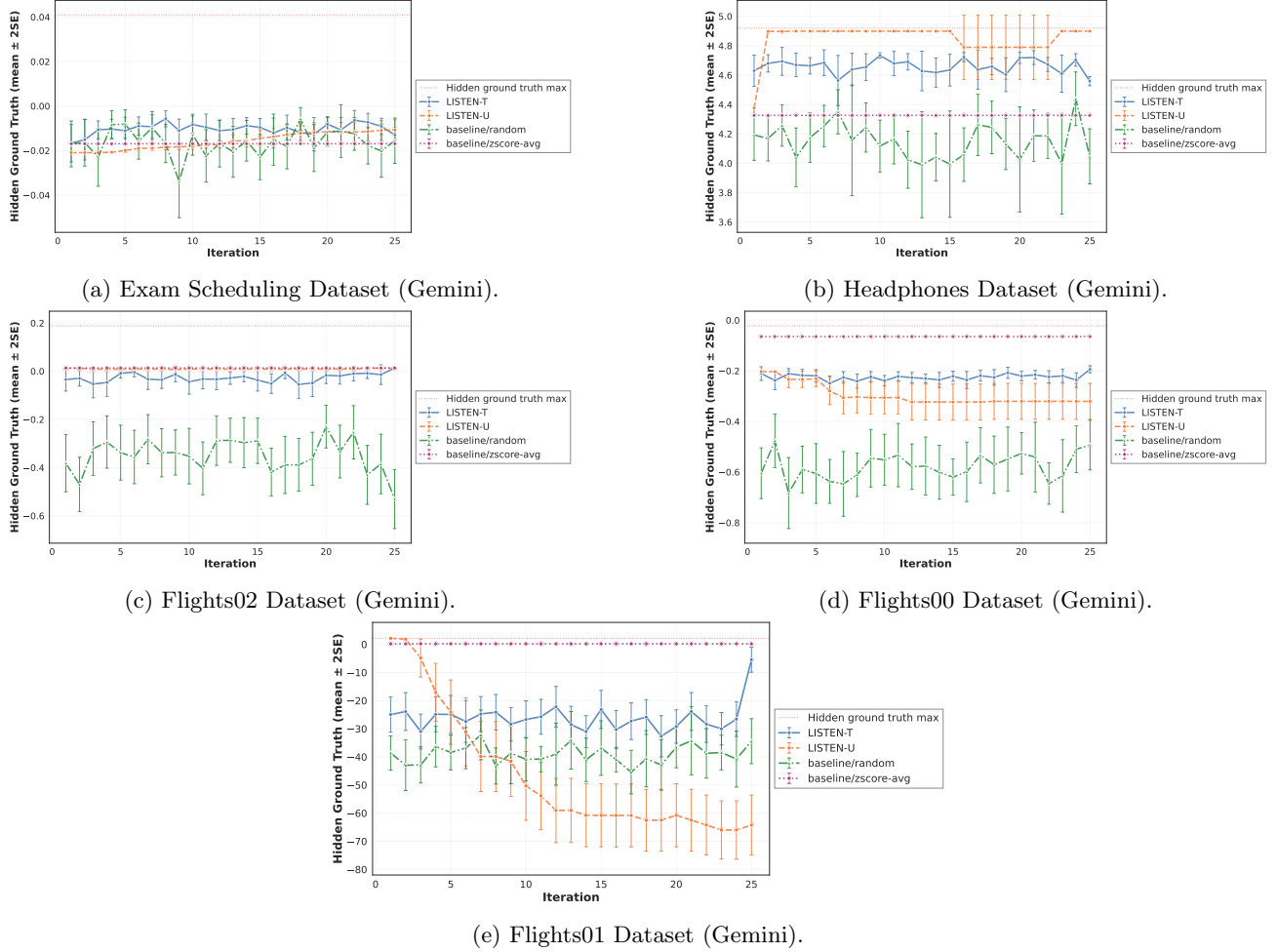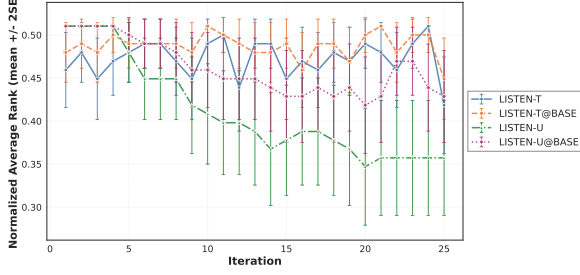
(e) Flights01 Dataset (Gemini).

Figure 9: Performance of LISTEN algorithms and baselines on five datasets, showing the Average Utility Score (higher is better) over 25 iterations. The plots show results using the Gemini model.
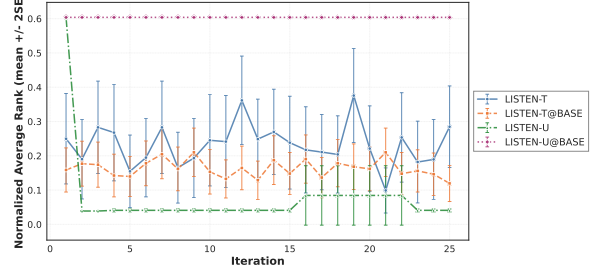
F Prompts

This section includes representative examples of the variation of prompts we utilized. As mentioned in Section 3.1, each prompt consists of five parts (persona context, metric definitions, user priorities, solutions, and format instructions). The first three parts are different in each setting and we give examples below in figures (Figure 11, Figure 12, and Figure 13). The fourth and fifth parts are relatively generic across settings. The fourth part (solutions) pulls the solution from the previous iteration, if any. The fifth part of the instructions, which covers output format, has different structures for LISTEN-U and LISTEN-T. For LISTEN-U, the task is to output the weights for numerical features in JSON format. For LISTEN-T, the task is to identify the single best solution from the options provided and return it using the exact format: `FINAL A (B, C ...)`.
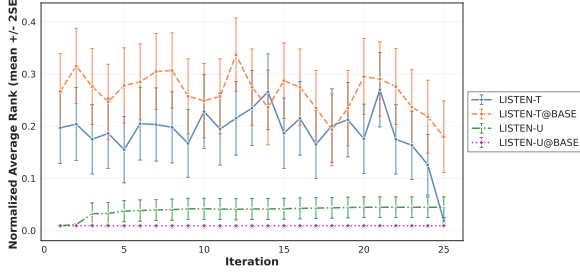
F.1 Exam Scheduling

Only the first three prompts vary significantly, so Figure 11 shows these parts of the prompt for the exam scheduling problem. The first line, "You are an expert university registrar..." provides the persona context. The next section details the metrics. Following the metric definitions in a bulleted list format, the user priorities are highlighted. In this case, the priority is to "minimize student stress and administrative burden by optimizing a specific set of metrics...".
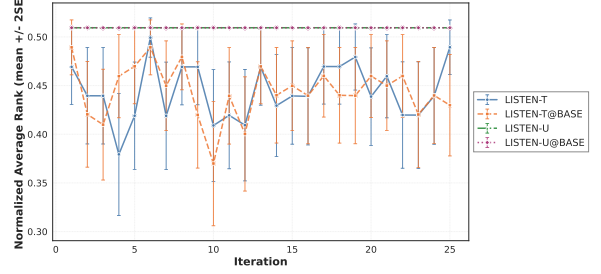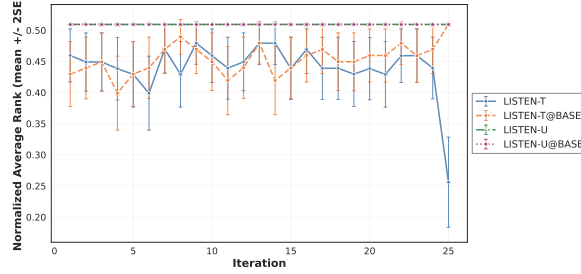
(a) Exam Scheduling Ablations (Gemini).

(b) Headphones Ablations (Gemini).

(c) Flights02 Ablations (Gemini).

(d) Flights00 Ablations (Gemini).

(e) Flights01 Ablations (Gemini).

Figure 10: Ablation study evaluating the impact of preference utterance. Performance using the preference-guided prompt is compared to a base prompt containing only the persona and metric definitions. The plots show the Normalized Average Rank (lower is better) over 25 iterations, using the Gemini model.

F.2 Flights

Similar to the exam scheduling prompt, in Figure 12, the persona context is provided in the first sentence. Following the persona context, the metrics are described. The next paragraph includes the user's preferences in natural language ("I am looking for round-trip flight...").

F.3 Shopping

Following the format of the other prompts, the first line in Section 13 details the persona context ("You are an audio equipment reviewer..."). The next section describes the metric definitions. Following these definitions, the next section details the user's preferences in natural language. These are all "soft" preferences because they do not include any strict constraints.

Exam Scheduling Prompt

```
You are an expert university registrar choosing the better final-exam schedule.
Use these definitions (lower is better unless stated otherwise):
    - conflicts: students with overlapping exams (must be minimized)
    - quints: 5 exams in a row
    - quads: 4 in a row
    - four in five slots: 4 exams within 5 slots
    - triple in 24h (no gaps): 3 exams within 24 hours (strictly consecutive
      blocks)
    - triple in same day (no gaps): 3 exams within the same calendar day
      consecutively
    - three in four slots: 3 exams within 4 slots
    - evening/morning b2b: back-to-back from an evening into next-morning slot
    - other b2b: any other back-to-back pair
    - two in three slots: 2 exams within 3 slots
    - avg max: is the average slot of the last exam for each student. Students
      tend to want to leave as early as possible.

Policy guidance: To generate an optimal final examination schedule that
minimizes student stress and administrative burden by optimizing a specific
set of metrics according to a tiered priority system.
    *Context:* There are three exam slots available per day.
    Solutions must have a conflicts value of 0 or 1.

    *Tier 1:* Minimize Mandatory Reschedules
    The highest priority is to minimize the metrics associated with intense
    exam clusters that require a mandatory reschedule. Please minimize the
    following metrics in the order listed:
        quints (five exams in 24 hours)
        quads (four exams in 24 hours)
        four in five slots
        triple in 24h (no gaps)
        triple in same day (no gaps)

    *Tier 2:* Reduce Student Exam Fatigue
    Once Tier 1 objectives have been met, the primary tie-breaker is to
    minimize the total number of back-to-back exams. This objective is
    achieved by minimizing the sum of the following two metrics:
        evening/morning b2b
        other b2b

    *Tier 3:* Enhance Overall Schedule Quality
    As a final set of tie-breakers, prioritize schedules that improve the
    general student experience by optimizing these metrics in order of
    preference:
        Lower the avg_max (the average slot of a student's final exam).
        Minimize two in three slots.
        Minimize three in four slots.
```

Figure 11: An example prompt for the exam scheduling dataset.

Flights Prompt

You are an expert travel scheduling agent that specializes in air fare.
   The data set is only for a one-way flight. In this case, it is the first leg of a
   round trip.
   Use these definitions:
   • name: name of airline operating the flight
   • origin: origin airport
   • destination: destination airport
   • departure time: time of departure from origin airport
   • arrival time: time of arrival at destination airport
   • duration: how long the flight is
   • stops: number of layover stops
   • price: cost of the flight
   • dis_from_origin: distance of origin airport from where the customer prefers
   (lower is better)
   • dis_from_dest: distance of arrival airport from where the customer prefers
   (lower is better)
   • departure_seconds: time of departure since a fixed date in seconds
   • arrival_seconds: time of arrival since a fixed date in seconds
   • duration_min: duration of total flight in minutes (lower is better)

Objective: I am looking for round-trip flight options for two adults from Chicago
to New York City for the weekend of October 11, 2025. The priority is to maximize
our time in NYC while respecting a key scheduling constraint on Friday.

Primary Requirements (Must-Haves):
   – Route: Chicago (any airport) to New York City (JFK, LGA, or EWR).
   – Passengers: 2 adults.
   – Departure Window:
   – Must depart on either Friday, October 10, 2025, or Saturday,
   October 11, 2025.
   – If departing on Friday, the flight must be after 12:30 PM.
   – Return Date: Sunday, October 12, 2025.
   – Budget: The total price per ticket must not exceed $400.

Secondary Preferences (Tie-Breakers):
   – These should be used to select the best option among flights that meet the
   above requirements.
   – Departure Time (Friday): A departure after 3:30 PM is strongly preferred.
   The ideal arrival would be early enough to get a full night's sleep or to catch
   an 11:00 PM train from the airport.
   – Departure Time (Saturday): If a Saturday departure is chosen, it should be
   the earliest possible flight to maximize time in the city.
   – Flight Type: Direct (non-stop) flights are highly preferred.
   – Cost: Among flights that meet all criteria, the cheapest option is best.
   – Destination Airport: There is a slight preference against Newark (EWR) due
   to longer ground transportation, but it is an acceptable option.
   – Airline: No airline preference.

Figure 12: An example prompt for a flight itinerary preference, which utilizes the natural language description
of the user's priorities over objectives, but includes strong constraints.

Shopping Prompt

```
You are an audio equipment reviewer choosing the best headphones.
   Use these definitions:
   - product_name: name of the product (non-metric, weight always 0).
   - brand: manufacturer reputaton (non-metric, weight always 0).
   - price: cost measured in dollars.
   - type: headphone design.
   - connectivity: wired vs wireless.
   - noise_cancellation: active vs passive.
   - battery_life: measured in hours.
   - bluetooth_version: higher versions are newer.
   - driver_size: measured in millimeters.
   - weight: measured in ounces.
   - water_resistance: measured subjectively and also through the ipXX rating system,
       which measures a device's protection against solids (first digit)
       and liquids (second digit), with higher numbers indicating stronger resistance.
   - microphone: presence/quality of mic
   - review_rating: average customer rating
   - review_count: number of reviews
   - description: qualitative marketing text (non-metric, weight always 0)

I am a college student who is looking for recommendations for new headphones. I
prefer over-ear headphones rather than in-ear or on-ear, since they feel
more comfortable for long hours and usually deliver better sound. I also prefer
wireless headphones because I don't want to deal with cables. I would like
the headphones to have active noise cancellation, since I often study in
places where there is background noise. A microphone is important to me because
I sometimes use my headphones for calls. For battery life, I'd like
headphones that can last a long time on a single charge.
I usually don't mind recharging overnight, but I don't want headphones that
run out of power quickly in the middle of the day.
In terms of build, I actually prefer headphones that feel a bit heavier, since
lightweight headphones sometimes feel cheap or less durable to me. When I look
at reviews, I prefer headphones with both a high rating and a large number of
reviews, since that gives me more confidence in the product. I would prefer
headphones which have at least 5,000 reviews and a rating of at least 4.4.
To clarify the relative importance of both rating and number of reviews, I would
rather pick headphones with a 4.8 rating and 20,000 reviews than a 4.9 rating
with 1000 reviews. Other metrics such as driver size, water resistance, and
Bluetooth version do not matter at all to me. I'm not price-sensitive at all,
so I'm willing to pay more if the headphones meet my preferences well.
```

Figure 13: An example prompt for headphone preferences.