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

Adam S. Jovine[1] , **Tinghan Ye**[2] , **Francis Bahk**[1] , **Jingjing Wang**[1]
Matthew Ford[1] , **David B. Shmoys**[1] , **Peter I. Frazier**[1]
[1]Cornell University
[2]Georgia Institute of Technology
asj53@cornell.edu, joe.ye@gatech.edu, {feb47,jw2446,mtf62,dbs10,pf98}@cornell.edu

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 (LLM-based Iterative Selection with Trade-off Evaluation from Natural-language), an agentic LLM-based framework that treats the LLM as a decision-making agent capable of iteratively refining its internal preference model and taking actions (e.g., proposing utilities or selecting candidates) to maximize alignment with a user's implicit goals. 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 overall. This work explores a promising direction for steering complex multi-objective decisions directly with natural language, reducing the cognitive burden of traditional preference elicitation. Code is available at https://github.com/AdamJovine/LISTEN.

1 Introduction

We consider a human decision maker choosing among a large set of available items, such as airline flight itineraries or final exam schedules. The decision maker is aided by attributes 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 attributes that would support the automatic selection of the best item. Instead, the decision maker must manually inspect many items to select the one they prefer most. This can lead to decision fatigue and poor decisions [Schwartz, 2015].

This problem is common. Surveys across 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 options in day-to-day life, e.g., in online shopping, where combinatorial bundles of interacting products (flight legs arranged into itineraries, computer components arranged into a server) can explode in number.

Traditional solutions are often inadequate. Manual comparison of all options is time-consuming and prone to errors. Pareto plots displaying one attribute against another do not scale beyond a few attributes. Multi-objective optimization algorithms (see, e.g., [Knowles, 2006; Branke, 2008]) allow screening solutions off the Pareto frontier; however, when there are many attributes, 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.*, 2023; 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) open a new path here. With their ability to interpret text, LLMs enable zero-shot preference modeling, where a decision-maker's goals can be understood directly from a verbal description without expressing pairwise preferences. While recent research has begun integrating LLMs into preference learning, they are typically used as components within larger systems—to guide questioning [Lawless *et al.*, 2024; 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, how to best use LLMs to accelerate human selections over a large collection of items remains understudied.

1.1 Contributions

To address this gap, we introduce **LISTEN**: **L**LM-based **I**terative **S**election with **T**rade-off **E**valuation from **N**atural-language, an agentic framework in which an LLM acts as a

[†]An extended version with full appendix is available at https://arxiv.org/abs/2510.25799.

decision-making agent that iteratively observes candidate solutions, reasons about trade-offs, updates an internal preference representation, and takes actions to refine the candidate set.

In our framework, a human expert first describes their priorities in natural language. Then, 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 its ability to reason directly over large item lists, which can prevent the LLM from 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 algorithms in the LISTEN framework: **LISTEN-U**, which makes decisions 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 at the level of the baselines or better. Our baselines span uniform random selection, an equal-weighted average over numerical attributes, single-call LLM ranking of all candidates, and human re-ranking. **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 adjusting the prompt or the utility form. **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.*, 2024; Xiao *et al.*, 2024; 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.*, 2024]. 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]. [Liu *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 through methods such as 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] shows some common pitfalls when using LLMs 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 settings, 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 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 $s^\star \in \mathcal{S}$ that best satisfies the preferences described in U. We assume access to a pre-trained LLM, which acts as a preference-driven decision agent, maintaining an evolving internal preference state and producing actions (utility updates or selections) conditioned on its observation history. The core challenge is to design an algorithm that finds $s^\star$ 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 unrealistic for a human to exhaustively inspect all items and unlikely for an LLM to process the entire set in a single context window. We treat U as one decision-maker's articulation of their own preferences, and target alignment with that individual user.

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 agentic, iterative decision framework in which an LLM acts as a closed-loop preference agent. As illustrated in Figure 1, at each step, the agent observes a subset of candidate solutions, reasons about trade-offs using the user's natural-language priorities, updates an internal preference representation, and takes an action, either proposing a refined utility function (LISTEN-U) or selecting a batch winner (LISTEN-T), to guide the next iteration. 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

The core of our framework is using an LLM to simulate the decision-maker's preferences in a zero-shot manner. We do

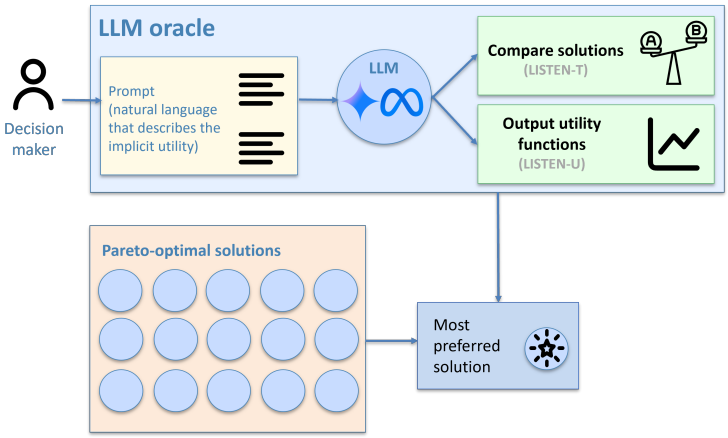


Figure 1: LISTEN as an agentic decision loop: an LLM agent iteratively refines preferences or selects champions to identify the most preferred solution.

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. **Attribute 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 in any given algorithmic step, whether it is refining a utility function or selecting a champion from a batch.

From an agentic perspective, each prompt instantiates the agent's current belief state (encoded via weights or champions), while the LLM's output corresponds to an action that updates this state or advances the selection policy. This explicit observe–reason–act loop situates LISTEN as a lightweight, tool-free decision agent.

3.2 LISTEN-U: Iterative Utility Refinement

Our first algorithm, LISTEN-Utility (**LISTEN-U**), is a parametric approach. It assumes a linear utility function, $u(s) = \mathbf{w}^\top s^{\text{num}}$, defined over the encoded feature vector s^{num}: genuinely numerical fields (e.g., price, battery life) and, when an integer-to-category mapping is supplied to the LLM in context, label-coded categorical fields. Free-text descriptions and other non-scored attributes inform the LLM's reasoning but are not part of the scoring function. The agent iteratively refines its internal utility representation $\mathbf{w}$ to identify the best item, which serves as a compact belief state over the user's latent preferences, 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, then selecting the highest-scoring item as the initial best solution, $s^\star$.

Next, the **iterative refinement phase** (lines 8-12) begins. In each iteration, the prompt for the LLM is constructed using the current weight vector $\mathbf{w}_{t-1}$ alongside the complete,

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: $R \leftarrow T - 1$ ▷ Number of preliminary rounds
3: $\mathcal{K} \leftarrow \emptyset$ ▷ Initialize set of batch champions
4: **for** $j \leftarrow 1$ to R **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 $s^\star$

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, $\mathbf{w}_t$. Once the LLM returns the updated weights, the algorithm 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. In agentic terms, LISTEN-T implements a selection policy in which the agent adaptively samples, evaluates, and promotes candidate actions (solutions) through a stochastic elimination process.

The algorithm proceeds in two stages using a total of T calls to the LLM. First, in the **preliminary rounds** (lines 2-8), the algorithm conducts $R = T - 1$ rounds. In each round, it samples a batch of B items uniformly at random without replacement from $\mathcal{S}$ 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 10), the LLM is presented with the set of all R 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 a decision-maker (DM) with significant experience in each respective task, following a two-step protocol. First, the DM 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 DM meticulously ranked a subset of items from the full candidate set according to these same preferences. This process yields a dataset comprising a high-level natural language goal (U) paired with 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 the flight datasets, categorical attributes such as airline were one-hot encoded for LISTEN-U and numerical baselines; for all LLM prompts, these attributes were retained as their original text to provide full context.

We designed two distinct experimental scenarios to test a range of preferences: **Flights CHI $\rightarrow$ NYC**: A student flying from Chicago, IL to New York, NY, and **Flights Ithaca $\rightarrow$ Reston**: 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 extended arXiv version.

Headphones 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, water resistance, 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. These categorical fields are label-encoded in the released feature table, with qualitative mappings supplied in the prompt header.

Final Exam Scheduling Dataset. Our third domain addresses a large-scale university final exam scheduling problem. We generated a set $\mathcal{S}$ of 4,938 diverse, Pareto-optimal schedules using the ParEGO multi-objective optimization algorithm [Knowles, 2006], based on the mixed-integer programming methods of [Ye *et al.*, 2026]. The quality of each schedule is evaluated against a comprehensive set of attributes to be minimized, listed in the extended arXiv version. The primary attribute is **Direct Conflicts**, counting students with overlapping exams. A suite of attributes 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 attributes capture the nuanced trade-offs between student workload, fairness, and logistical constraints.

A Dataset Diagnostic: Concordance. To analyze why our parametric (LISTEN-U) and non-parametric (LISTEN-T) algorithms perform differently across datasets, we introduce a metric to quantify the complexity of the underlying preference structures. This metric, computed for each dataset independently of any algorithm, characterizes the intrinsic "difficulty" of reaching the human's top picks using a linear utility function. To compute it, we generate 10,000 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 score is the fraction of these random functions for which the optimal item, $\arg\max_s u(s)$, lies within the m-item human-ranked set (see m in Table 1). This measures how broadly the human's top picks are reachable by linear utilities, not just whether the single #1 pick is. A low concordance value, as seen for the exam scheduling dataset in Table 1, suggests that the human preference is complex and not easily selected by a linear model. A high concordance value, as in Flights Ithaca $\rightarrow$ Reston, suggests a much more linearly separable problem. 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.

4.2 Experimental Setup

Across all experiments, we use two LLMs as our preference oracle: `Llama-3.3-70B-Versatile` [Grattafiori *et al.*, 2024] and `Gemini 2.5 Flash-Lite` [Comanici *et al.*, 2025]. To ensure the robustness of our results, all algorithmic runs are replicated 40 times with different random seeds for generation, while keeping the prompt consistent for each replication. Code, prompts, and the human-preference datasets are available at https://github.com/AdamJovine/LISTEN-IJCAI.

The ground truth for evaluation is derived from human expert rankings. For each dataset, it is impractical to rank the entire set of N items, so 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 four 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/zscore-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.

Full Batch (`baseline/full_batch`). This method works like a single preliminary iteration of LISTEN-T, with the entire dataset passed into the prompt in this single iteration. The LLM is asked to rank the items. We randomize the seed used for sampling in the LLM and the order the items are presented in each replication of the method. The top-ranked item from each replication is reported.

Human Rankers (`baseline/human_rerank`). To compare against a human baseline with access to textual attributes in addition to numerical, we tasked human rankers different from the decision-maker creating the ground truth labels with assigning rankings. These humans were given a description of the attributes of items in each dataset, the decision-maker's preference utterance, and a system allowing them to filter and sort by multiple attributes at a time. We tasked them with ranking the top 20 items they believed best matched the preference utterance, while ensuring they explored many high-quality items. However, like the other baselines and methods, we only report their final top-ranked item.

Dataset	Concordance	Total Items (N)	Ranked Items (m)	Ranked Prop. (m/N)
Exam Scheduling	0.0015 ± 0.0008	4938	11	0.002
Flights CHI $\rightarrow$ NYC	0.0025 ± 0.0011	903	16	0.018
Headphones	0.0552 ± 0.0047	77	15	0.195
Flights Ithaca $\rightarrow$ Reston	0.3357 ± 0.0095	216	12	0.056

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).

4.4 Evaluation Metric: Normalized Average Rank

We evaluate algorithm performance primarily using the rank of the selected item within the human-generated ground-truth ranking. 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.

4.5 Results & Discussions

We present experimental evaluation of the LISTEN framework in Figure 2. We then analyze the comparative performance of our algorithms and conclude with ablation studies.

Comparisons to Baselines

The LISTEN methods tend to outperform the non-human baselines. We now focus on the full-batch and human-rerank baselines, as they are the most competitive and also give insights into the complexity of our problem setting.

`baseline/full_batch`: The LISTEN methodology offers an improvement in ranking performance relative to the naive approach of asking the LLM to rank all items at once, as demonstrated in Figure 2. The smaller ranking tasks performed in each iteration of LISTEN approaches reduce the complexity of the task. In addition, running multiple iterations where the most promising items are compared against each other gives the LLM another chance to focus on the tradeoffs present in high-quality options and make a refined top choice. We also note that for some datasets and LLM combinations, full-batch ranking would be practically infeasible due to not fitting in the LLM's context window.

`baseline/human_rerank`: In comparison with the human-rerank baseline, we must first acknowledge the large variability in its performance. One reason for this is that humans had a higher dispersion of their top-choice being ranked vs unranked compared to LISTEN methods. Recall from Section 4.4 that unranked items receive a normalized average rank of ≈ 0.5. The best human ranks tended to be very strong; however, many humans' top choices were not in the ranked set from the decision-makers' ground truth rankings. Another reason is we only had 4 samples for humans. This is the reason for the large variance in normalized average ranks.

However, this highlights the ambiguity in the distribution of preferences that humans associate with a prompt. We see that LISTEN methods more consistently chose items that were ranked than humans, but perhaps may be less likely to optimize the very fine tradeoffs the decision-maker prefers.

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. The default `Headphones` prompt (Table 1) contains hard constraints such as a strict budget and required categorical features. We created a softer variant, `Headphones-Soft`, by dropping these constraints. Removing these threshold-based requirements makes the preference more linearly representable, causing the Concordance score to rise from **0.055** to **0.244**. This rise is expected, as linear utility functions struggle to model concrete threshold-based requirements.

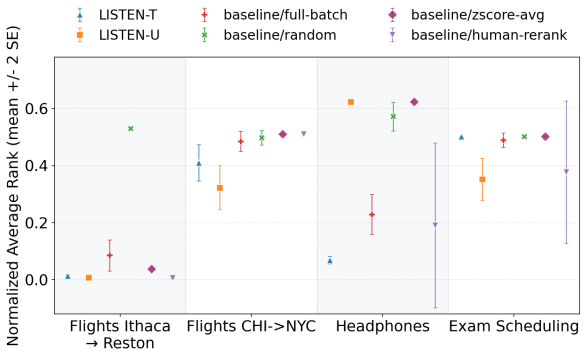
As predicted by this change in concordance, Figure 3 shows a significant increase in performance for LISTEN-U on the `Headphones-Soft` version. In contrast, the non-parametric LISTEN-T maintained its robust performance across both prompts. This result validates that our concordance metric is an effective predictor for the performance of a parametric approach like LISTEN-U.

4.6 Observed Robustness to LLM Choice and Batch Size

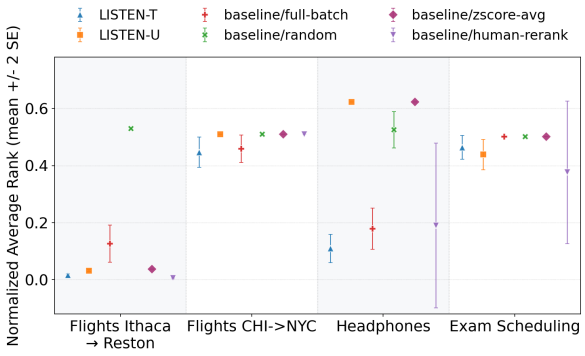
We observe a surprisingly high level of consistency in performance across batch sizes for each (LLM, dataset) pair in our experiments, as seen in Figure 4. Our full experimental results across all batch sizes (LLaMA and Gemini, batch sizes 2–32) are included in the extended arXiv version. There would be value in further analysis to see how robust this finding is across even larger batch sizes.

5 Conclusion

In this work, we introduced **LISTEN**, an agentic decision-support framework that uses large language models as preference-driven agents for multi-objective selection. By maintaining an evolving internal preference representation and acting iteratively on candidate sets, LISTEN turns natural-language goals into concrete decisions. Our experiments show that LLMs can successfully translate high-level natural language goals into high-quality selections from



(a) LLaMA



(b) Gemini

Figure 2: Performance of LISTEN algorithms and baselines on four datasets, showing the Normalized Average Rank (lower is better) of the top ranked item by each method. The LISTEN methods used 25 iterations and batch size 32. All algorithms have $n = 40$ runs except baseline/human-rerank ($n = 4$) and baseline/zscore-avg ($n = 1$).

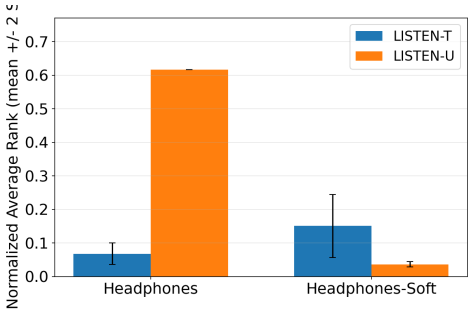


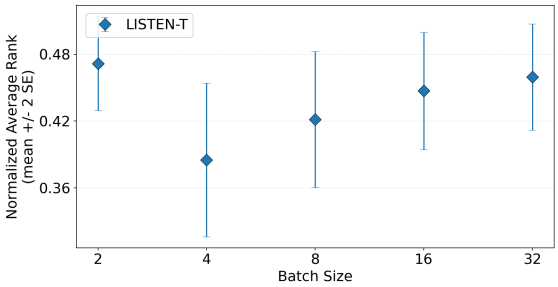
Figure 3: Performance of LISTEN-U and LISTEN-T on the default `Headphones` prompt and a softer variant `Headphones-Soft` that drops the hard constraints, which has higher concordance. All algorithms have $n = 40$ runs.

large, complex sets of items. We also introduced a concordance metric that predicts when the parametric approach is likely to succeed or fail based on the problem's preference structure.

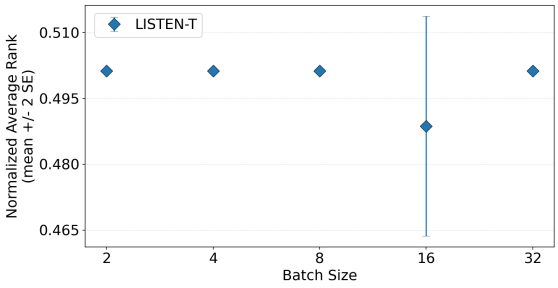
That said, our study has several limitations that suggest clear avenues for future work.

Complexity 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. We also noticed the high variability of our human re-rankers with respect to the ground-truth rankings. Future work could gather more data to understand the variability in preferences corresponding to an utterance and return single items or sets of items that satisfy the diverse potential preferences.

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, is yet to be tested at scale. Expanding to more varied and larger-scale datasets would further validate our findings.



(a) LLaMA, Flights CHI → NYC



(b) LLaMA, Exam Scheduling

Figure 4: Batch size effect on LISTEN-T ($n = 40$).

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

MF, FB, and PIF were supported by ONR N00014-22-1-2763 and N00014-25-1-2181. MF and PIF were supported in part by Moebius Solutions, Inc.

Contribution Statement

Adam S. Jovine and Tinghan Ye contributed equally to this work. All authors contributed to the conceptualization and development of the LISTEN framework. Adam S. Jovine and Tinghan Ye led the implementation, experimental design, and empirical analysis, with input from all authors. All authors contributed to interpreting the results and to writing and revising the manuscript.

References

[AhmadiTeshnizi *et al.*, 2024] Ali AhmadiTeshnizi, Wenzhi Gao, and Madeleine Udell. OptiMUS: Scalable optimization modeling with (MI)LP solvers and large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 577–596. PMLR, 2024.

[Austin *et al.*, 2024] David Eric Austin, Anton Korikov, Armin Toroghi, and Scott Sanner. Bayesian optimization with LLM-based acquisition functions for natural language preference elicitation. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys '24)*, Bari, Italy, 2024. ACM.

[Bang and Song, 2025] Seunghwan Bang and Hwanjun Song. LLM-based user profile management for recommender system. *arXiv preprint arXiv:2502.14541*, 2025.

[Branke, 2008] Jürgen Branke. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media, 2008.

[Comanici *et al.*, 2025] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, Krishna Haridasan, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

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

[Gao *et al.*, 2025] Yuan Gao, Dokyun Lee, Gordon Burtch, and Sina Fazelpour. Take caution in using LLMs as human surrogates. *Proceedings of the National Academy of Sciences*, 122(24):e2501660122, 2025.

[Grattafiori *et al.*, 2024] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

[Huang *et al.*, 2025] Chenyu Huang, Zhengyang Tang, Shixi Hu, Ruoqing Jiang, Xin Zheng, Dongdong Ge, Benyou Wang, and Zizhuo Wang. ORLM: A customizable framework in training large models for automated optimization modeling. *Operations Research*, 73(6):2986–3009, 2025.

[Huber *et al.*, 2025] Felix Huber, Sebastian Rojas Gonzalez, and Raul Astudillo. Bayesian preference elicitation for decision support in multi-objective optimization. *Journal of Multi-Criteria Decision Analysis*, 2025.

[Klamkin *et al.*, 2025] Michael Klamkin, Arnaud Deza, Sikai Cheng, Haoruo Zhao, and Pascal Van Hentenryck. DualSchool: How reliable are LLMs for optimization education? *arXiv preprint arXiv:2505.21775*, 2025.

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

[Lawless *et al.*, 2024] Connor Lawless, Jakob Schoeffer, Lindy Le, Kael Rowan, Shilad Sen, Cristina St. Hill, Jina Suh, and Bahareh Sarrafzadeh. "I want it that way": Enabling interactive decision support using large language models and constraint programming. *ACM Transactions on Interactive Intelligent Systems*, 14(3), 2024.

[Lawless *et al.*, 2025] Connor Lawless, Yingxi Li, Anders Wikum, Madeleine Udell, and Ellen Vitercik. LLMs for cold-start cutting plane separator configuration. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2025)*, volume 15763 of *Lecture Notes in Computer Science*, pages 51–69. Springer, 2025.

[Liu *et al.*, 2025] Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. Aligning with logic: Measuring, evaluating and improving logical preference consistency in large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025.

[Obayashi *et al.*, 2007] Shigeru Obayashi, Shinkyu Jeong, Kazuhisa Chiba, and Hiroyuki Morino. 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, 2007.

[Okeukwu-Ogbonnaya *et al.*, 2025] Adaeze Okeukwu-Ogbonnaya, Rahul Amatapu, Jason Bergtold, and George Amariucai. LLM-based community surveys for operational decision making in interconnected utility infrastructures. *arXiv preprint arXiv:2507.13577*, 2025.

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

[Ozaki *et al.*, 2024] Ryota Ozaki, Kazuki Ishikawa, Youhei Kanzaki, Shion Takeno, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-objective bayesian optimization with active preference learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 14490–14498. AAAI Press, 2024.

[Ramamonjison *et al.*, 2023] Rindranirina Ramamonjison, Timothy Yu, Raymond Li, Haley Li, Giuseppe Carenini, Bissan Ghaddar, Shiqi He, Mahdi Mostajabdaveh, Amin Banitalebi-Dehkordi, Zirui Zhou, et al. NL4Opt competition: Formulating optimization problems based on their natural language descriptions. In *Proceedings of the NeurIPS 2022 Competitions Track*, volume 220 of *Proceedings of Machine Learning Research*, pages 189–203. PMLR, 2023.

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

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

[Wang *et al.*, 2023] Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. Recent advances in bayesian optimization. *ACM Computing Surveys*, 55(13s):1–36, 2023.

[Xiao *et al.*, 2024] Ziyang Xiao, Dongxiang Zhang, Yangjun Wu, Lilin Xu, Yuan Jessica Wang, Xiongwei Han, Xiaojin Fu, Tao Zhong, Jia Zeng, Mingli Song, et al. Chain-of-experts: When LLMs meet complex operations research problems. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[Yang *et al.*, 2024] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[Ye *et al.*, 2026] Tinghan Ye, Adam S. Jovine, Willem van Osselaer, Qihan Zhu, and David B. Shmoys. Cornell university uses integer programming to optimize final exam scheduling. *INFORMS Journal on Applied Analytics*, 56(2):159–177, 2026.

[Zhang *et al.*, 2025] Haobo Zhang, Qiannan Zhu, and Zhicheng Dou. Enhancing reranking for recommendation with LLMs through user preference retrieval. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 658–671, Abu Dhabi, UAE, 2025. Association for Computational Linguistics.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.