What Are People's Actual Utility Functions in Budget Aggregation?

Ayelet Amster
The Open University
Israel
ayelet.amster@gmail.com

Lioz Akirav
Ariel Univesity
Israel
lioz.akirav@gmail.com

Rica Gonen
The Open University
Israel
ricagonen@gmail.com

Erel Segal-Halevi
Ariel Univesity
Israel
erelsgl@gmail.com

ABSTRACT

Background: *Budget aggregation* is a process in which citizens declare their ideal budget allocation, and a rule aggregates these into a single outcome. Recent work proposes various aggregation rules and impossibility results, typically relying on assumptions about how voters evaluate non-ideal allocations. However, these assumptions have not been empirically validated on human subjects.

Objectives and Research Questions: Recent theoretical works have suggested various rules for budget-aggregation, as well as impossibility results for simultaneously satisfying some desirable axioms. The analysis of both aggregation rules and impossibility results typically relies on assumptions about how voters evaluate non-ideal budget allocations; the analysis breaks when the utility model is different. Despite this, these assumptions have never been validated empirically on human.

Methods: We present a framework for empirically testing hypotheses about human utility functions using pairwise comparisons. We introduce a modular open-source polling system that elicits a subject's ideal budget allocation and then presents carefully generated pairs of non-ideal alternatives. Different pair-generation algorithms enable testing of various properties of utility functions. We demonstrate the framework by conducting polls on hundreds of human subjects.

Results: The results show that standard utility models (ℓ_1, ℓ_2, Leontief) fit human behavior poorly, with few subjects consistent with them. We find stronger support for general properties such as star-shaped, multi-dimensional single-peaked, and peak-linear. Most humans show asymmetry in sign (gains vs losses) and across issues, inconsistent with any ℓ_p-based utility. These results suggest that practical budget aggregation requires more general models of human utility.

Conclusions: Our findings suggest that standard utility models do not fully capture human preferences in budget aggregation. At the same time, preferences exhibit consistent structural patterns alongside notable asymmetries. These insights point toward the need for more flexible utility models in practice.

CCS CONCEPTS

• **Computing methodologies → Algorithmic game theory and mechanism design**.

KEYWORDS

Budget aggregation, Preference elicitation, Utility measurement, GTEP

ACM Reference Format:
Ayelet Amster, Lioz Akirav, Rica Gonen, and Erel Segal-Halevi. 2026. What Are People's Actual Utility Functions in Budget Aggregation?. In *Appears at the 8th Games, Agents, and Incentives Workshop (GAIW-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., Paphos, Cyprus, May 2026*, IFAAMAS, 41 pages.

1 INTRODUCTION

This research is motivated by the growing interest in *participatory budgeting* — a process by which the citizens can participate in deciding how to divide the budget of their city or state. A common model is *budget proposal aggregation*, where voters report ideal allocations that are aggregated into a final budget. A simple aggregation rule is the arithmetic mean; it has good axiomatic properties [9, 18], but provides strong incentives for voters to report false preferences in order to manipulate the outcome.This led to *truthful* aggregation rules that resist manipulation. Such algorithms typically use sophisticated variants of the median rule [7, 8, 13, 14, 19].

Analyzing the properties of an aggregation rule requires some assumptions about the voters' preferences over non-ideal budgets.

Different papers have different assumptions on this matter. Many papers assume that voters evaluate a non-ideal budget based on its distance from their ideal budget according to some metric, such as ℓ_1 [7, 12] or ℓ_∞ [14] [1].

Recently, Brandt et al. [6] have argued in favor of utility functions called *Leontief*, that are not based on any metric: voters evaluate a non-ideal budget based on the smallest *ratio* of the amount given to any issue to their ideal amount.

[1]With three issues, the ℓ_∞ distance is always $1/2$ of the ℓ_1 distance, so the preferences induced by both metrics are the same. This is not true with four or more issues.

Appears at the 8th Games, Agents, and Incentives Workshop (GAIW-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., Armstrong, Curry, Hosseini, Mattei, Tsang, Wąs (Chairs), May 2026, Paphos, Cyprus.

Yet another way to compare distributions is the *Kullback-Leibler (KL)* divergence, which measures how different one allocation is from another in terms of information loss. It is commonly used to compare probability distributions, but has been recently used in a social choice context [3].

Different utility models imply different trade-offs between desirable properties. As an example, Brandt et al. [6] prove that, for three or more issues and three or more voters, if voters' preferences are based on ℓ_1 or ℓ_∞ metrics, then *no* aggregation rule is truthful, Pareto-efficient and satisfies a weak fairness notion called *proportionality*. In contrast, if voters have Leontief utilities, then the algorithm maximizing the Nash welfare (the product of utilities) is group-strategyproof (stronger than truthful), and satisfies core fair share (stronger than both Pareto-efficiency and proportionality).

These vastly different results invoke the question which is at the heart of the present research:

> *What utility functions are actually used by real people when comparing different budgets?*

As different people may have different utility functions, our aim is to construct a generic polling framework, that allows to check various properties of individual users' utility functions.

1.1 Our contribution

We present a framework for conducting opinion polls based on *pairwise comparisons*, where participants choose between two non-ideal budget allocations after specifying their ideal allocation. Pairwise comparisons reduce cognitive load, but designing informative pairs is challenging (sample screenshots of the poll interface are shown in Appendix A). We present various pair-generation algorithms, and report the outcomes of running the resulting polls on a representative sample of the voter population in Israel. The code for our polling framework is open-source and can easily be used by researchers elsewhere.

Our first algorithm accepts as input two utility models (e.g. ℓ_1 and Leontief), and generates pairs that test whether the user consistently adheres to one of these models over the other one. Using this algorithm, we generated six polls, corresponding to all pairwise comparisons among four common utility models (see Section 3 for the formal definitions). Surprisingly, in all six polls, a majority of the users did not answer consistently with any single model.

This indicates that none of these utility models accurately reflects humans' preferences (see subsection 5.2 for complete results).

Following these negative results, we developed pair-generation algorithms for checking consistency with more general properties. Specifically, we checked whether subjects' utility functions are single-peaked, star-shaped, or peak-linear (see Section 5.3). In these polls the results were more positive: almost 90% of the users replied consistently with star-shaped or single-peaked utilities, and almost 80% replied consistently with peak-linear utilities (a stronger condition than star-shaped).

Next, we aimed to check whether humans' utility functions are consistent with any ℓ_p metric. All ℓ_p metrics possess two types of symmetry: (1) *Sign Symmetry* — adding x and subtracting x from the ideal allocation contribute the same amount to the distance; (2) *Issue Symmetry* — adding x to different issues contributes the same amount to the distance. For each symmetry type, we developed a pair-generation algorithm that tests whether subjects' utility functions exhibit this type of symmetry. Our results here were, again, negative: less than 10% of the subjects showed at least 90% symmetry in both respects (see Section 5.4 for more details).

Further analyses using generalized ℓ_p metrics with issue-specific weights or sign-specific weights revealed very limited consistency: fewer than 20% of participants were fully consistent with issue-specific weights, and none with sign-specific weights. We also checked a satisfaction-based model, recently introduced by Gourvès et al. [17], by which agents' utility is determined by the number of issues funded by at least their ideal amount. We found only limited support for this model, as about half the responses contradict it. Overall, these findings suggest that simple symmetric or weighted asymmetric ℓ_p metrics, as well as the satisfaction-based model, are insufficient to fully capture human preference patterns, highlighting the need for more flexible utility models. Detailed results for all of these properties and their analyses can be found in Section 5.5.

Finally, we wanted to check whether humans' utility functions are consistent with any norm-based metric, and particularly, whether they satisfy the triangle inequality. This turned out to be the most challenging check, as the triangle inequality involves a sum of two distances. To cope with this challenge, we asked the subjects to compare *biennial budgets*. We conducted a preliminary poll, in which we found out that about 60% of the subjects compare biennial budgets in a way that is consistent with adding utilities (Section 5.6). Among these subjects, we conducted another poll which checked whether their replies are consistent with the triangle inequality. A large majority of the subjects' replies were *contrary* to the triangle inequality, i.e., they preferred the sum of distances $\|x\| + \|y\|$, to the sum $\|x + y\| + 0$.

Taken together, our results indicate that utility functions based on metrics, particularly metrics that are symmetric with respect to sign and issue (such as ℓ_p), are not very good for modeling human budget preferences. However, most humans' utility functions do belong to more general classes such as star-shaped or single-peaked or peak-linear. Future work could focus on these more general classes, and try to detect within them, the sub-classes that better fit actual utility functions.

2 RELATED WORK

Participatory budgeting (PB) studies how individual preferences over public spending can be aggregated into collective decisions.

2.1 Utility models in Participatory Budgeting

2.1.1 Discrete participatory budgeting. Most practical PB instances are based on *project selection*: each project has a fixed cost, and voters simply indicate which projects they support, effectively casting binary yes/no votes. Participants do not control the exact level of funding; instead, aggregation rules determine which subset of projects is implemented. See Rey et al. [20] for a recent survey of this setting.

In this setting, too, there are various assumptions regarding the voters' utility functions (also known as *satisfaction functions*). The most common ones are: count-based utilities (a voter's utility is the number of supported projects that are funded), and cost-based utilities (a voter's utility is the total cost of supported projects that are funded). Intermediate utility models (such as the square-root of cost) are also studied [11]. We are not aware of direct experiments testing which of these utility models, if any, reflects humans' real preferences. The closest one we know of is by Rosenfeld and Talmon [21]. They presented indirect evidence in favor of the count utilities: in several scenarios, they computed the utilitarian-optimal budget-allocation (the allocation that maximizes the sum of utilities) under five different utility models, and asked the subjects to choose which of the five resulting allocations they prefer. Most subjects prefer the budget that was utilitarian according to count-utilities.

2.1.2 Continuous participatory budgeting. Continuous PB extends discrete PB by allowing approval or cardinal ballots, sometimes called *fair mixing* [1]. Extensions include donation-based budgets, where coordinating contributions efficiently and fairly depends on assumptions about agents' utility functions [4, 5]. For instance, additive utilities create impossibility results, whereas Leontief utilities allow strong fairness and efficiency guarantees. No empirical studies are known in this setting. Mechanisms such as Iterative Local Voting (ILV) elicit preferences dynamically, with convergence guarantees under structured utilities; empirical results suggest ℓ_∞ updates are particularly stable [15]. Suksompong and Teh [24] provide a comprehensive survey of continuous PB models and algorithms.

2.2 Empirical Research in Participatory Budgeting

Empirical PB research studies both preference elicitation interfaces and aggregation rules, as both affect outcomes and user behavior.

2.2.1 Data Elicitation Formats. The elicitation format strongly affects expressiveness, cognitive effort, and aggregation quality. Prior work has studied formats such as Knapsack Voting, Cumulative Voting, k-Approval, and Threshold Approval [2, 10, 16, 23], highlighting the trade-off between usability and preference detail. For instance, comparisons of Discrete Choice Experiments and Constant-Sum Paired Comparisons show that simpler formats reduce cognitive load, while paired comparisons provide richer information on relative priorities [22]. This motivates our poll format, which combines repeated paired comparisons with budget allocation tasks to capture participants' nuanced priorities (Section 4).

2.2.2 Empirical Evaluation of Aggregation Rules. On the backend, experiments evaluate how different aggregation rules perform in terms of fairness, efficiency, and robustness to strategic behavior. Studied rules include greedy algorithms, Equal Shares (MES) [10], utilitarian aggregation, and the Nash-product rule [21]. A further support is provided by recent experimental studies, which show how citizens perceive different aggregation rules in terms of fairness and legitimacy, highlighting important trade-offs for practical system design [25].

3 MODEL AND NOTATIONS

In a *budget allocation* problem, there is a set A of m *alternatives* (also called *issues* or *projects*). The total budget is denoted by B. The set of all possible budget allocations is the simplex

$$\Delta(B) := \{\mathbf{q} \in \mathbb{R}^m \mid \mathbf{q} \geq \mathbf{0} \text{ and } \sum_{j\in A} q_j = B\}.$$

In our polls, we always assume $B = 100$, meaning "100%" (in other words, the numbers in our polls are interpreted as a percentage of the total budget). Hence, we represent the set of possible budget allocations simply by Δ. We assume that each person has a preference ranking over Δ, which can be represented by a utility function $u : \Delta \to \mathbb{R}$. We further assume that u can be presented as $u(\mathbf{q}) = U(\mathbf{p}, \mathbf{q})$, where —

- $\mathbf{p}$ is an *ideal budget allocation* (also called the *peak*) — a unique vector in Δ which the person thinks is the best way to allocate the budget of B among the m issues.
- U is a *utility model function* — a function from $\Delta \times \Delta$ to $\mathbb{R}$, that represents the utility of an agent with ideal budget allocation $\mathbf{p}$ when the actual allocation is $\mathbf{q}$.

Whereas typically each person has a different utility function, we believe that different people may have similar utility *model* functions; these are the functions we study in the present research. Some common utility models are:

- *ℓ_1 disutilities*: $U(\mathbf{p}, \mathbf{q}) = -\sum_{j\in A} |p_j - q_j|$;
- *ℓ_2 disutilities*: $U(\mathbf{p}, \mathbf{q}) = -\sqrt{\sum_{j\in A} (p_j - q_j)^2}$;
- *ℓ_p disutilities*, for any $p \geq 1$: $U(\mathbf{p}, \mathbf{q}) = -(\sum_{j\in A} |p_j - q_j|^p)^{1/p}$
- *Leontief utilities*: $U(\mathbf{p}, \mathbf{q}) = \min_{j\in A}(\frac{q_j}{p_j})$;
- *Kullback-Leibler divergence*: $U(\mathbf{p}, \mathbf{q}) = -\sum_{j\in A} p_j \cdot \ln\left(\frac{\mathbf{p}_j}{\mathbf{q}_j}\right)$.

4 EXPERIMENTAL SETUP

Poll-generation framework. We constructed a modular framework that lets one generate polls by combining several components:

- *Story* — a textual description of what the budget exactly is divided. In our experiments we compared two stories: government budget vs. municipal budget.
- *Issues* — a list of m issues among which the budget should be allocated.
- *Pair-generation algorithm* — a custom algorithm that takes as input the subject's ideal budget and returns a list of pairs. Each pair-generation algorithm is carefully designed to test specific properties of utility functions. Section 5 describes the various algorithms in detail.
- *User filter* — a custom filter that decides which users are suitable for a particular poll. In most polls, the filter only required that the ideal budget assign positive amounts to at least two issues.
  Some polls needed a stronger filter — see Section 5 for details.
- *Language* — all polls are available in English, but can be easily translated to the subjects' native language.

See Appendix A for screenshots of the user interface, and Appendix J for a detailed system description and guidelines for reproducibility.

Each poll used several basic measures against behavioral biases: Pairs were randomized to avoid primacy effects. We removed participants who always chose the same position (<1Each poll included two *alertness tests*, where one option matched the subject's ideal budget. Subjects failing these checks were excluded as inattentive. Subjects were required to choose one option (no indifference) to prevent trivial indifference responses. Additionally, to reduce cognitive load, we rounded all budget-allocation vectors to multiples of 5%.

Conducting the polls. We recruited over 2000 subjects for all polls combined.Participants were recruited via Panel4All, a large online panel provider. At our request, we prioritized re-engaging previous participants. Samples were designed to be demographically representative. Overall, 1068 participants passed the alertness checks.

5 INDIVIDUAL POLLS: ALGORITHMS AND RESULTS

We describe the pair-generation algorithms and summarize the main results; full details appear in the appendix.

5.1 Distribution of peak allocations

Among three-category budgets, the most frequent peak allocation is $[40, 30, 30]$ (321 responses), followed by $[50, 25, 25]$ (199) and $[60, 20, 20]$ (104). These results indicate a clear concentration around moderately unequal yet structured splits. See Table 1 in appendix B for a full breakdown.

5.2 Comparing specific utility models

In the first set of polls, we assumed, based on many theoretical works in participatory budgeting, that agents' utility models are one of ℓ_1, ℓ_2, Leontief or KL (see Section 3 for the formal definitions). We conducted all $\binom{4}{2} = 6$ pairwise comparisons between these four models.

To mitigate this problem, we developed an improved pair-generation algorithm, that generates the pairs with the highest difference in utilities. The algorithm works as follows (see Algorithm 1 in Appendix C for the pseudo-code).

First, the algorithm constructs a set V of all budget-allocation vectors in which all components are multiples of 5%. To avoid zero-bias effects, the algorithm only constructs vectors with strictly positive components (at least 5%).

Next, for each budget-allocation $\mathbf{q}$ in V, the algorithm computes the utilities under both models, $u_i(\mathbf{q}) := U_i(\mathbf{p}, \mathbf{q})$ for all $i \in \{1, 2\}$. To enable meaningful comparison between utilities of different models, the raw utilities are converted into values in $[0, 1]$. We tried two normalization methods: in *linear normalization*, the normalized value is computed as: (raw value - min value) / (max value - min value). In *ordinal normalization*, all vectors are ordered in increasing order of utility. Suppose there are d distinct utility values, $U_1 < \cdots < U_d$; then, all vectors with raw utility U_i receive normalized utility $(i-1)/(d-1)$. In preliminary experiments we did not find substantial differences in results between the two normalization methods, so we decided to use only the ordinal normalization.

Next, the algorithm examines all unordered pairs $\mathbf{q}_A, \mathbf{q}_B$ in V and identifies pairs for which the two utility models induce opposite preference orderings. For each such pair, the *difference score* is defined as the smaller of the two normalized utility differences, that is $\text{score}(\mathbf{q}_A, \mathbf{q}_B) := \min(|u_1^{norm}(\mathbf{q}_A) - u_1^{norm}(\mathbf{q}_B)|, |u_2^{norm}(\mathbf{q}_A) - u_2^{norm}(\mathbf{q}_B)|)$, where u_i^{norm} denotes the normalized utility under model $i \in \{1, 2\}$.

Finally, the algorithm selects the k pairs with the highest scores. An example pair is shown in Appendix C. To find these k top pairs, we simply generated all pairs, sorted them by decreasing score, and picked the first k. This approach is efficient for up to $m = 5$ issues; a faster method for larger m is given in Appendix C.2.

We applied this algorithm six times with all of the possible combinations, with $k = 10$ pairs. About 30 subjects participated in each individual poll.

5.2.1 *Results.*

Our original plan was to partition the participants into four groups: the "ℓ_1 people" (those whose preferences are based on ℓ_1 metric), the "ℓ_2 people", the "Leontief people" and the "KL people". To our surprise and dismay, most participants did not belong to any of these groups!

To understand why, note that a person whose preferences are based on some utility model U will always prefer the allocation that is better according to U to the allocation that is better according to some other metric (and worse according to U). However, in all six polls, over 70% of the subjects did not choose consistently according to any of the two metrics. For example, in the ℓ_1 vs. Leontief polls, 25 out of 32 subjects (78%) chose the ℓ_1-preferred option in some pairs and the Leontief-preferred option in other pairs. Even if we allow one mistake (i.e., require only 90% consistency), about 66% of the subjects are inconsistent (see Table 2, Table 3, Table 4 in Appendix C for complete results).

Thus, our first conclusion is that the preferences of over 60% of the population cannot be accurately represented by any of these four utility models.

Our second conclusion is that over 30% of the population *can* be described by one of these models — *KL utilities*. Obtaining this conclusion was not trivial, so we describe the thought process in detail.

(a) In the comparison of KL vs. Leontief, almost $1/2$ of the subjects replied with at least 90% consistency with KL (and no subject replied with even 80% consistency with Leontief). However, similar results were found in the comparison of ℓ_1 vs. Leontief and ℓ_2 vs. Leontief (over $1/3$ chose consistently with the ℓ_p utility).

(b) In the comparisons of KL vs. ℓ_1, KL vs. ℓ_2 and ℓ_1 vs. ℓ_2, the consistency was dramatically lower: at most 3 out of 30 subjects were at least 90% consistent with any of these models. Therefore, initially we thought that all four models are inapplicable, as only few subjects are consistent with any of them.

(c) The above results were obtained for budget allocation among three issues. When we ran similar polls for budget allocation among four and five issues, the consistency level was dramatically higher: at consistency level at least 90%, over $1/3$ of the subjects consistently preferred KL to both ℓ_1 and ℓ_2.

(d) To understand the difference between the 3 issues results and 4-5 issues results, we looked at the average scores of pairs in the polls. We found out that, in all 3 issues polls, the average difference-scores of the pairs were below 0.1. In the 4 issues polls the average difference-scores increased to about 0.14, and in the 5 issues polls the average difference-scores increased to about 0.17 .[2] The reason is that, when there are more issues, the simplex is larger, our algorithm has more options to choose from, and therefore, choosing the ten pairs with the highest difference-scores leads to a higher average score in the polls.

(e) Our interpretation is that, in the 3 issues polls, most pairs were in the subjects' "indifference zone" — they considered them nearly

[2] Recall that the scores are normalized to the range $[0, 1]$. Hence, a difference of 0.14 means a relative difference of about $1/7$, which is much more noticeable than a difference of less than $1/10$.

identical, and therefore did not choose consistently. However, in the 4 and 5 issues polls, the differences between the vectors were much more noticeable, and therefore over $1/3$ of the participants replied consistently. [3]

Increasing the number of issues raises cognitive load: in 5-issue settings, about 60% of participants fail the alertness test. The results reported above (and reported in detail in Appendix C) contain only subjects who passed the alertness tests. Detailed table on participants who failed the alertness tests is reported in the appendix (see Table 5).

5.3 Checking basic monotonicity properties

Besides the four models we have tested in Section 5.2, one could think of many other specific utility models to test. In order to narrow the search space, we have decided to design different kinds of polls, that check for more general properties of utility functions. We checked three basic properties.

1. Star-shaped. A utility-model function U is called *star-shaped* if the utility of an allocation strictly decreases as the allocation moves away from the agent's ideal allocation ("peak") in any direction. Formally [6], for any distribution $\mathbf{q} \neq \mathbf{p}$ and for all $\lambda \in (0, 1)$,

$$U(\mathbf{p}, \mathbf{p}) > U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}) > U(\mathbf{p}, \mathbf{q}).$$

Each of the following two properties is stronger than star-shaped.

2. Multi-dimensional single peaked (MDSP). Let $\mathbf{q}_1$ and $\mathbf{q}_2$ be two alternative distributions. We say that $\mathbf{q}_2$ is *closer to* $\mathbf{p}$ *than* $\mathbf{q}_1$ if for every issue j, either $\mathbf{q}_{1j} \geq \mathbf{q}_{2j} \geq \mathbf{p}_j$ or $\mathbf{q}_{1j} \leq \mathbf{q}_{2j} \leq \mathbf{p}_j$, and for at least one issue j, the inequality between $\mathbf{q}_{1j}$ and $\mathbf{q}_{2j}$ is strict. A utility-model function U is said to be *multi-dimensional single-peaked* if whenever $\mathbf{q}_2$ is closer to $\mathbf{p}$ than $\mathbf{q}_1$, it holds that $U(\mathbf{p}, \mathbf{q}_2) > U(\mathbf{p}, \mathbf{q}_1)$. In Appendix E.2 we prove that MDSP is indeed stronger than star-shaped.

3. Peak-linear. Peak-linearity captures the idea that moving halfway toward one's ideal budget should yield exactly half the gain in utility compared to moving all the way. Brandt et al. [6] define a utility function as *peak-linear* if for any distribution $\mathbf{q}$ and $\lambda \in [0, 1]$, $U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}) = \lambda U(\mathbf{p}, \mathbf{p}) + (1 - \lambda)U(\mathbf{p}, \mathbf{q})$. Their definition relies on the numeric value of the utility, which we have no way to check. Hence we give a more general definition, which relies only on ordinal comparisons. We say that a utility function as *peak-linear* if for any two distributions $\mathbf{q}_1, \mathbf{q}_2$ and $\lambda \in [0, 1]$,

$$U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}_1) \geq U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}_2) \tag{1}$$

$$\iff U(\mathbf{p}, \mathbf{q}_1) \geq U(\mathbf{p}, \mathbf{q}_2). \tag{2}$$

It is easy to check that ℓ_p metrics are peak-linear according to both definition. However, a utility function such as $\sum_j (p_j - q_j)^2$ is peak-linear according to our definition and not according to the definition in Brandt et al. [6], although such a utility function is clearly equivalent to ℓ_2.

[3] In preliminary experiments we tested a fifth utility model which we called *Anti-Leontief*, which reverses the logic of Leontief utilities by aiming to minimize the *largest* ratio between actual and ideal allocations, representing satisfaction driven by the most overfunded issue:

$$U_{\text{Anti-Leontief}}(p, q) = -\max_{j \in A} \left(\frac{q_j}{p_j} \right)$$

However, almost no subjects were even mildly consistent with this model, so we decided to drop it from our results.

In Appendix E.1 we prove that peak-linear is stronger than star-shaped (assuming the utility function is continuous). However, peak-linearity and MDSP are independent — none of them implies the other (See Appendix E for a proof).

Leontief utilities are peak-linear too KL utilities are multi-dimensional single-peaked, but not peak-linear (see Appendix E for full proofs).

5.3.1 Pair-generation algorithms.

We present a pair-generation algorithm for each of the three monotonicity properties.

1. Star-shaped. The algorithm generates $k = 10$ pairs, one for each weight $\lambda \in \{0.1, \ldots, 0.9\}$ (with 0.5 appearing twice). For each λ, the algorithm generates a random budget-allocation vector $\mathbf{q}$, computes the convex combination $\mathbf{q}_\lambda := \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}$, and adds $(\mathbf{q}, \mathbf{q}_\lambda)$ to the list of pairs. The pseudo-code for the algorithm is provided in Appendix D, and an example appears in Table 6.

For most values of λ, the convex combination has entries that are not multiples of 5%. We suspected that this might create some mental bias. Therefore, we implemented a variant of this algorithm, which rounds all entries in the convex combination to the nearest multiple of 5. The algorithm and an example can be found in Algorithm 3 and in Table 9.

2. Multi-dimensional single-peaked. Here we used a simple random-search algorithm: (1) Generate a random pair $(\mathbf{q}_1, \mathbf{q}_2)$; (2) If $\mathbf{q}_1$ is closer to $\mathbf{p}$ than $\mathbf{q}_2$ or vice-versa, then add $(\mathbf{q}_1, \mathbf{q}_2)$ to the list of pairs; (3) Repeat until k pairs have been added. The pseudo-code is provided in Algorithm 4.

3. Peak-linear. The pair-generation algorithm for peak-linearity is more involved, as peak-linearity requires utilities to be equal, and we cannot query numeric utility values using our pairwise-comparison framework. We coped with this issue in the following way. (1) The algorithm generates three pairs comparing three extreme vectors. To avoid zero-bias, we chose "extreme vectors" without zero coordinates: $\mathbf{v}_1 = [10, 10, 80]$, $\mathbf{v}_2 = [10, 80, 10]$, $\mathbf{v}_3 = [80, 10, 10]$. (2) For each weight $\lambda \in \{0.25, 0.5, 0.75\}$, the algorithm generates three convex combinations $\mathbf{q}_i := \lambda\mathbf{p} + (1 - \lambda)\mathbf{v}_i$ for $i \in \{1, 2, 3\}$, and adds the three pairwise comparisons among them. This algorithm yields 12 pairs overall. A subject with a peak-linear utility function should rank $\mathbf{q}_i$ and $\mathbf{q}_j$ exactly the same as $\mathbf{v}_i$ and $\mathbf{v}_j$, for every λ. The pseudo-code for this algorithm, as well as an illustrating example, appears in appendix D.

5.3.2 Results.

The results for all three properties were quite positive, with over $3/4$ of the participants showing consistency.

1. Star-shaped. Overall, 88.8% of the answers (out of the $10n$ questions) were consistent with star-shapedness, that is, preferred the convex combination vector over the random vector. 75% of the users (63 out of 84) exhibited consistency levels of at least 90%. See Table 7 for complete results. The results when the vector components were rounded to multiples of 5% were very similar.

We also examined consistency levels across different values of λ. For $\lambda = 0.1$, the overall consistency level was slightly lower than the average (77.46% of the answers); this is expected, as for such a small λ, the convex combination is very similar to the random vector, and might be considered by some subjects as equivalent to it. For larger λ values the overall percentages remained high, though we did not observe a clear increase in consistency as λ grew larger. See Table 8 Appendix D for complete results.

2. Multi-dimensional single-peaked. The results here were even more striking than for star-shaped: over 97% of the answers were consistent with MDSP, that is, participants preferred the vector that is closer to the peak. Over 75% of the subjects showed 100% consistency, and the remaining subjects showed 90% consistency. The full results are in table 10.

To explain why MDSP exhibits higher consistency than the mathematically weaker star-shapedness property (88.8%), we analyzed the utility "gap" between presented options. In the star-shapedness test, the distance between the random vector $\mathbf{q}$ and the weighted average $\mathbf{c}_\lambda$ is scaled by λ. When λ is low, the alternatives become visually and mathematically similar, increasing cognitive noise.

In contrast, the MDSP algorithm does not use a scaling parameter like λ; it generates independent vectors and filters for strict dominance, typically resulting in much larger gaps.

Because the MDSP test provides a significantly clearer signal of improvement across all categories, participants are far less likely to make noisy mistakes, leading to higher observed consistency.

5.4 Checking symmetry

In the next set of polls, our aim was to check whether people's utility functions are compatible with *any* ℓ_p metric. The ℓ_p metrics are *symmetric* in the sense that only the size of the deviation matters — not where in the budget it occurs, nor whether it represents an increase or decrease in funding. This symmetry consists of two independent properties:

- *Issue Symmetry* — adding x to issue i is equivalent to adding x to issue j for all i, j.
- *Sign Symmetry* — adding x to issue i is equivalent to subtracting x from the same issue.

A numeric example is given in Appendix F in the appendix.

5.4.1 Pair-generation algorithms. We designed an algorithm for each of the two symmetry properties. Both algorithms follow the same idea: construct sets S_1, S_2 of vectors, such that the vectors in each S_i should have the same utility according to the tested symmetry property; then compare vectors in S_1 with corresponding vectors in S_2. A subject whose utility function satisfies the tested symmetry property should rank all pairs $\mathbf{q}_1 \in S_1$ and $\mathbf{q}_2 \in S_2$ in the same way.

- *Issue Symmetry*: sets S_i were constructed by sampling a random allocation $\mathbf{q}$, computing $\mathbf{d} = \mathbf{q} - \mathbf{p}$, and generating all $m-1$ cyclic rotations of $\mathbf{d}$. Each rotated vector $\mathbf{d}'$ was added as $\mathbf{d}' + \mathbf{p}$ to S_i. We constructed 4 set-pairs, each yielding $m = 3$ comparisons (12 pairs total).See Algorithm 6 for pseudo-code.
- *Sign Symmetry*: sets S_i were constructed by sampling $\mathbf{q}$, computing $\mathbf{d} = \mathbf{q} - \mathbf{p}$, and adding both $\mathbf{q} = \mathbf{d} + \mathbf{p}$ and $\mathbf{q}' = -\mathbf{d} + \mathbf{p}$ to S_i. We constructed 6 set-pairs, each yielding 2 comparisons (12 pairs total). See Algorithm 7 for pseudo-code.

A full illustration of the Issue Symmetry pair-generation process appears in Example F.1 in Appendix F.

5.4.2 Results. Our results for both symmetry properties were negative:

- Issue Symmetry: 10% (4/40) consistent in all four groups; 15% in at least three.
- Sign Symmetry: no subject consistent across all six groups; only 2/31 consistent in five groups.

See Table 17, Table 18, and Table 19 in Appendix F for complete results and an example of inconsistency.

This inconsistency suggest that the ℓ_p model, as well as any other utility model that treats issues symmetrically or treats increases and decreases symmetrically, may not adequately represent people's preferences.

5.4.3 Is the asymmetry in issues caused by asymmetry in amounts? The asymmetry among issues could be explained in two ways: (1) People assign different values for cuts in different issues (e.g. a cut of 10 in Defense is different than a cut of 10 in Education). (2) People assign different values for cuts in different initial amounts (e.g. a cut of 10 in an ideal budget of 20 is different than a cut of 10 in an ideal budget of 30). The second explanation would imply the following slight generalization of the ℓ_p utility model: $U(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^{m} D(p_j, |p_j - q_j|)^r$, where D is a "deformation function" that modifies the difference $|p_j - q_j|$ based on the initial amount.

To test this possibility, we conducted an additional targeted poll. In this poll, we filtered and retained only participants whose allocations assigned *identical amounts* to two different issues. The budget of the third issue was fixed at its original value, while the two identical budgets were systematically varied across alternatives. The formal procedure used to construct the comparison pairs is described in Algorithm 8 in Appendix F.

As an example, a subject with an ideal budget of $[40, 30, 30]$ could be asked to compare $[40, 15, 45]$ with $[40, 45, 15]$. A subject with a "deformation-based" utility model would be indifferent between these two vectors. In general, such a subject would be indifferent between adding x to issue 2 and subtracting x to issue 3, and vice-versa. Hence, over 10 pairs, such a subject's answers would be close to random (near 50% percent supporting an increase in issue 2).

The actual results were quite different (can be seen in Table 20 in Appendix F): Out of the 31 participants, 16 (51%) exhibited more than 90% consistency, including 13 participants who demonstrated perfect (100%) consistency across all questions. In contrast, only 6 (19%) participants showed low consistency levels, with less than 60% consistent choices. These results suggest that subjects' decisions are influenced not only by the magnitude of budget changes but also by the specific issue being modified. In particular, the observed asymmetric preferences indicate that symmetric utility models, such as ℓ_1, may be insufficient to fully capture participants' behavior in this setting, even when accounting for a "deformation effect".

5.5 Checking Consistency in Asymmetry

Following the negative results of Section 5.4, we checked whether subjects' utility functions are consistent with a generalization of an ℓ_p metric, which allows asymmetry in issues or signs.

(1) A utility-model function such as $U(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^{m} a_j \cdot |p_j - q_j|^r$ exhibits a consistent asymmetry between issues, represented by the issue-specific weights a_j;

(2) A utility-model function such as $U(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^{m} a \cdot \max(p_j - q_j, 0)^r + b \cdot \max(q_j - p_j, 0)^r$ exhibits a consistent asymmetry between signs, represented by the sign-specific weights a, b;

5.5.1 *Pair-generation algorithms.*

We designed an algorithm for each of the weighted utility models.

1. Issue-specific weights. The algorithm picks a positive value x and generated $m = 3$ difference vectors varying by rotation, namely $\mathbf{d}_1 = [2x, -x, -x]$ and $\mathbf{d}_2 = [-x, 2x, -x]$ and $\mathbf{d}_3 = [-x, -x, 2x]$. The weighted utility corresponding to $\mathbf{d}_1$ is $a_1(2x)^r + a_2(x)^r + a_3(x)^r = x^r \cdot (a_1 2^r + a_2 + a_3)$. Similarly, the weighted utility corresponding to $\mathbf{d}_2$ is $x^r \cdot (a_1 + 2^r a_2 + a_3)$, and the weighted utility corresponding to $\mathbf{d}_3$ is $x^r \cdot (a_1 + a_2 + 2^r a_3)$. Hence, the ranking between these three difference vectors should be the same regardless of x. In other words, if a subject prefers a concentrated increase in issue 1 ($[2x, -x, -x]$) to a concentrated increase in issue 2 ($[-x, 2x, -x]$), then the same should hold for any x.

We generated 4 triplets of vectors, corresponding to two positive and two negative values of x.

Instead of asking the subjects three questions per triplet (one for each pairwise comparison), we decided it was simpler to ask them to rank the three vectors; see Figure 4 for GUI example. See Algorithm 9 for the pseudo-code and Table 21 for a numeric example of the resulting allocations.

2. Sign-specific weights. The algorithm picks a positive value x, and generated two difference vectors varying by sign, namely $\mathbf{d}_1 = [x, x, -2x]$ and $\mathbf{d}_2 = [-x, -x, 2x]$. The weighted utility corresponding to $\mathbf{d}_1$ is $a(2x)^r + b(x)^r + b(x)^r = x^r \cdot (2^r \cdot a + 2b)$. Similarly, the weighted utility corresponding to $\mathbf{d}_2$ is $x^r \cdot (2^r \cdot b + 2a)$. Hence, subjects with this utility model should rank this pair in the same way for all x. We generated 6 pairs of vectors, corresponding to two different values of x and $m = 3$ rotations of the difference vectors, for a total of 12 pairwise comparisons. See Algorithm 10 for the pseudo-code. Note that the algorithm can fail to find valid budget-allocation vectors, particularly when an ideal budget is close to an extreme (for these cases we have a fallback procedure, which is detailed in Algorithm 11). In this poll, participants who allocated a budget of zero to any issue were excluded and prevented from proceeding to the comparison questions, as the algorithm requires non-zero values for all the issues to generate valid pairs.

5.5.2 *Results.*

1. Issue-specific weights. In this poll, a total of 37 subjects took part. Consistency with a utility-model with issue-specific weights would imply that, for each pair of i, j of issues (1 vs 2, 2 vs 3, 3 vs 1), the subject would consistently prefer a concentrated increase in i to a concentrated increase in j, or consistently prefer the other way around. In fact, Only 7 (less than 20% of the subjects) showed a consistent ranking among all three pairs. See Appendix G for complete results.

2. Sign-specific weights. We presented the results for each participant in the form of a preference matrix, that is, a table where the rows represent the topics and the columns represent the magnitude levels. Within each cell of the matrix, we indicated whether the participant preferred a distributed decrease (orange) or a concentrated decrease (blue) for a given topic at a given level. Examples of participants' preference matrices are shown in Figures 5 to 7.

Out of 33 participants, not a single one showed full consistency among all 12 pairs. See Appendix G for complete results.

5.5.3 *Satisfaction-based model.*

Gourvès et al. [17] present a *satisfaction-based* utility model. According to their model, the utility of an agent with ideal budget $\mathbf{p}$ from actual budget $\mathbf{q}$ is determined by the number of issues j for which $q_j \geq p_j$. Such a user would always prefer a large decrease and two small increases, over a large increase and two small decreases. Our results provide only weak support for this model: In our poll, the overall summary of participants' choices showed that 50.5% of responses corresponded to concentrated decreases, while 49.5% corresponded to distributed increases.

5.5.4 *Monotonicity of inconsistency.*

We also investigated whether those who did not display Issue Symmetry still exhibit monotonicity. That is, while their preference between a concentrated or distributed change may not be consistent across different issues, it might be monotonic with respect to the magnitude of the change (i.e., they might prefer a concentrated decrease when the magnitude is small, but switch to preferring a concentrated increase when the magnitude grows too large). Among those who were not consistent, 11 displayed full monotonicity — meaning they changed their direction of preference at most once (Figure 6 in Appendix G is an example of a participant who exhibits monotonicity, while Figure 7 belongs to a participant who does not exhibit monotonicity). If we also include the fully consistent participants, we find that 22 out of 34 people exhibited monotonicity. This insight may be useful for designing more general utility models in future work.

5.5.5 *An even more general utility model.*

In this section, we tested utility model functions that are asymmetric in sign or in issue, but not both. A more general utility-model function, that allows both types of asymmetry, is $U(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^{m} [a_j \cdot \max(0, p_j - q_j)^r + b_j \cdot \max(0, q_j - p_j)^r]$. Currently, we do not know how to test if subjects' utility functions are consistent with this form. We leave this question to future work.

5.6 Biennial Budgets and the Triangle Inequality

Our motivation for the next poll was to test whether participants' preferences correspond to *any* metric.A key property of any metric is the triangle inequality: for any points A, B, C, $d(A, B) + d(B, C) \geq d(A, C)$. In utility terms ($U = -d$), this becomes:

$$U(\mathbf{q}_A, \mathbf{q}_B) + U(\mathbf{q}_B, \mathbf{q}_C) \leq U(\mathbf{q}_A, \mathbf{q}_C). \tag{3}$$

We cannot test this directly because utilities are always evaluated relative to a fixed peak $\mathbf{p}$. Therefore, we focus on *norm-based metrics*.

Recall that a *norm* is a function from a vector space to $\mathbb{R}_+$, usually denoted by $\|\cdot\|$, that satisfies three conditions: Homogeneity, Positiveness, and Triangle inequality. A metric is called *norm-based* if there exists some norm $\|\cdot\|$ such that the distance between every two point A and B is equal to the norm of the difference vector, $\|A - B\|$. Every ℓ_p metric is norm-based (based on the so-called ℓ_p *norm*).

Suppose the utility-model function U is based on a norm-based metric with norm $\|\cdot\|$. Then $U(\mathbf{p}, \mathbf{q}) = -\|\mathbf{p} - \mathbf{q}\|$ for each vector $\mathbf{q}$. Let $\mathbf{d}_A, \mathbf{d}_B$ be two difference-vectors (vectors whose components sum up to 0), and let $\mathbf{d}_C := \mathbf{d}_A + \mathbf{d}_B$. Let $\mathbf{q}_j = \mathbf{p} + \mathbf{d}_j$ for all $j \in \{A, B, C\}$. Then $U(\mathbf{p}, \mathbf{q}_j) = -\|\mathbf{d}_j\|$. The triangle inequality implies that $\|\mathbf{d}_A\| + \|\mathbf{d}_B\| \geq \|\mathbf{d}_C\|$. This implies the following for utilities:

$$U(\mathbf{p}, \mathbf{q}_A) + U(\mathbf{p}, \mathbf{q}_B) \leq U(\mathbf{p}, \mathbf{q}_C). \quad (4)$$

Inequality (4) is more convenient to test than (3), as it involves only utilities with respect to the same peak $\mathbf{p}$. However, it still requires to compute a sum of utilities.

Comparing sums of utilities is challenging in our framework, so we propose a preliminary approach based on biennial budgets: The idea is to consider the budget over two consecutive years. Suppose the budget in year A is $\mathbf{q}_A$ and the budget in year B is $\mathbf{q}_B$. If a subject evaluates each year independently of the other year, then the subject's total utility from the two years will be the sum $U(\mathbf{p}, \mathbf{q}_A) + U(\mathbf{p}, \mathbf{q}_B)$.

The assumption of independence between years is a strong one. An alternative reasonable assumption is that a subject consider consecutive years to be *complementary*: if in year A the budget deviated from $\mathbf{p}$ to one direction, then the subject would prefer the budget in year B to deviate from $\mathbf{p}$ in the opposite direction, so that the two-year average equals the ideal budget $\mathbf{p}$.

Therefore, before actually testing the triangle inequality, we conducted a preliminary experiment in which we compared the above two assumptions: independence (implying additivity) versus complementarity.

5.6.1 Biennial budgets: independent or complementary? The preliminary poll consisted of a simple repeated choice task. In each task, participants were asked to choose which budget they preferred for the current year, while the not-chosen budget would be automatically allocated in the subsequent year.

The preliminary poll included 12 questions across three sub-polls testing temporal preferences and cross-year trade-offs. The detailed generation procedure and an example appears in Algorithm 12 in Appendix H.

The results are mixed (see Appendix H for detailed results): sub-polls 2–3 show over 60% support for the ideal allocation (consistent with additivity), while sub-poll 1 shows a bias toward earlier realization of the ideal budget.

5.6.2 Triangle Inequality. Based on these results, we want to examine the primary objective of the triangle inequality. We construct a new poll consisting of 14 questions. The purpose of the first two questions is to filter out participants who balance the budgets across the two years — that is, the poll will only include individuals who do **not** balance between years (meaning they choose their ideal budget in one year and a random budget in the other, rather than selecting a random budget in one year and a budget that balances to the average in the other). After restricting the poll to these participants, we present the remaining 12 questions of the poll. Each question asks to compare two biennial budgets: Option 1 is a *concentrated change*, where the entire deviation from the ideal budget occurs within a single year. Option 2 is a *distributed change*, where the same overall deviation is divided evenly between the two years, so that each year deviates only partially from the ideal.

The algorithm tests the triangle inequality by presenting choices that compare a single concentrated change to a split change: Each question compares a concentrated deviation $(\mathbf{p}, \mathbf{p} + \mathbf{d}_C)$ with a distributed deviation $(\mathbf{p} + \mathbf{d}_A, \mathbf{p} + \mathbf{d}_B)$ where $\mathbf{d}_C = \mathbf{d}_A + \mathbf{d}_B$. The choice reveals whether the triangle inequality holds. The generation procedure, including the precise algorithm and an example, is provided in Appendix H.

Across all participants, 64.4% of choices favored the *distributed change* option, compared to 35.6% favoring the concentrated option. This tendency strengthens as the level of consistency increases. While participants with lower consistency levels (50-75%) exhibit relatively balanced preferences between the two options, participants with higher consistency levels display a preference for distributed changes. In particular, among participants with consistency levels above 80%, the distributed option is chosen in over 75% of the cases, reaching roughly 80% for participants with consistency levels above 90%. Full results are in Table 24 in Appendix H.

These results indicate that, in general, most subjects' utility models do *not* satisfy the triangle inequality. Hence, any norm-based metric might not be a good representation of agents' utilities. Interestingly, the majority shows convexity in preferences over the distances from the ideal budget: two small changes are preferable to one large change. Finally, as a point of comparison, we note that in the municipal budgeting setting, preferences were approximately evenly split between concentrated and distributed options. A possible explanation is that, when decisions involve less critical domains, participants tend to exhibit more indifferent behavior. They are more willing to accept concentrated budgets, even when this entails substantial losses in specific issues.

6 STORY EFFECTS

Municipal experiments yielded qualitatively identical results: core axioms (star-shaped, MDSP, and peak-linear) held in both settings. We observed notable quantitative shifts: municipal preferences were sharper, with higher star-shaped (80% vs. 75%) and Identity Asymmetry (85% vs. 51%) consistency. However, municipal symmetry was lower (42.4% vs. 61.8%), possibly because local budgets feel less critical. Thus, framing affects utility intensity but not structural properties (See Appendix I for a detailed comparison).

7 DISCUSSION AND FUTURE WORK

We introduced a low-burden framework for eliciting budget preferences through pairwise comparisons. Our findings emphasize that utility assumptions must be tested empirically rather than assumed. Our current results are limited in scope, as the data was collected within a single country and does not yet account for demographic variables such as education or political orientation. Future work should also extend our framework to examine additional utility properties, including ranking-based preferences, and to better understand the theoretical boundaries of the approach. In particular, an important open question is which classes of utility functions can be reliably tested using pairwise comparisons alone. Finally, incorporating qualitative feedback, such as open-ended questions asking participants to explain their choices, may provide valuable insights that could help refine and improve automated models.

ACKNOWLEDGMENTS

We are grateful to Prof. Ralph L. Keeney for his advice on the initial poll design and to Prof. Klaus Nehring for the idea to check the Kullback-Leibler utility model. We are also grateful to participants of the Technion game theory seminar and Israeli algorithmic game theory seminar, particularly Ran Smorodinsky, Reshef Meir, Tal Grinshpoun and Lihi Dery for their helpful criticism and interesting.

REFERENCES

[1] Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. 2019. Fair mixing: the case of dichotomous preferences. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 753–781.

[2] Gerdus Benadè, Nevo Itzhak, Nisarg Shah, Ariel D. Procaccia, and Ya'akov Gal. 2018. Efficiency and Usability of Participatory Budgeting Methods. In *Empirical Studies in Participatory Budgeting (Proceedings of the PB Conference / Workshops)*. Unpublished manuscript / Academic preprint, 1–8. https://procaccia.info/wp-content/uploads/2018/03/pb19.pdf Empirical study with over 1,200 voters comparing input formats for PB.

[3] Florian Brandl and Felix Brandt. 2024. A natural adaptive process for collective decision-making. *Theoretical Economics* 19, 2 (2024), 667–703.

[4] Florian Brandl, Felix Brandt, Dominik Peters, and Christian Stricker. 2021. Distribution rules under dichotomous preferences: two out of three ain't bad. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. 158–179.

[5] Felix Brandt, Matthias Greger, Erel Segal-Halevi, and Warut Suksompong. 2025. Coordinating charitable donations with Leontief preferences. *Journal of Economic Theory* (2025), 106096.

[6] Felix Brandt, Matthias Greger, Erel Segal-Halevi, and Warut Suksompong. 2025. Optimal budget aggregation with star-shaped preference domains. *Mathematics of Operations Research* (2025).

[7] Ioannis Caragiannis, George Christodoulou, and Nicos Protopapas. 2022. Truthful Aggregation of Budget Proposals with Proportionality Guarantees. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 5 (2022), 4917–4924. https://doi.org/10.1609/aaai.v36i5.20421 arXiv:2203.09971.

[8] Mark de Berg, Rupert Freeman, Ulrike Schmidt-Kraepelin, and Markus Utke. 2024. Truthful Budget Aggregation: Beyond Moving-Phantom Mechanisms. *arXiv preprint arXiv:2405.20303* (2024). https://arxiv.org/abs/2405.20303

[9] Edith Elkind, Warut Suksompong, and Nicholas Teh. 2023. Settling the Score: Portioning with Cardinal Preferences. In *ECAI 2023 (Frontiers in Artificial Intelligence and Applications)*. IOS Press, 621–628. https://doi.org/10.3233/FAIA230324 arXiv:2307.15586.

[10] Roy Fairstein, Gerdus Benadè, and Kobi Gal. 2023. Participatory Budgeting Design for the Real World. *arXiv preprint arXiv:2302.13316* (2023). https://arxiv.org/abs/2302.13316

[11] Piotr Faliszewski and Nimrod Talmon. 2018. A framework for approval-based budgeting methods. *arXiv preprint arXiv:1809.04382* (2018).

[12] Rupert Freeman, David M. Pennock, Dominik Peters, and Jennifer Wortman Vaughan. 2019. Truthful Aggregation of Budget Proposals. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. Association for Computing Machinery, New York, 751–752. https://doi.org/10.1145/3328526.3329557 arXiv:1905.00457.

[13] Rupert Freeman, David M. Pennock, Dominik Peters, and Jennifer Wortman Vaughan. 2021. Truthful aggregation of budget proposals. *Journal of Economic Theory* 193 (2021), 105234. https://doi.org/10.1016/j.jet.2021.105234

[14] Rupert Freeman and Ulrike Schmidt-Kraepelin. 2023. Project-Fair and Truthful Mechanisms for Budget Aggregation. arXiv:2309.02613 [cs.GT]

[15] Nikhil Garg, Vijay Kamble, Ashish Goel, David Marn, and Kamesh Munagala. 2019. Iterative Local Voting for Collective Decision-making in Continuous Spaces. *Journal of Artificial Intelligence Research* 64 (2019), 315–355.

[16] Ashish Goel, Anilesh K. Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. 2019. Knapsack Voting for Participatory Budgeting. *ACM Transactions on Economics and Computation* 7, 2 (2019), 8:1–8:27. https://doi.org/10.1145/3340230 arXiv:2009.06856.

[17] Laurent Gourvès, Michael Lampis, Nikolaos Melissinos, and Aris Pagourtzis. 2025. Satisfactory Budget Division. *arXiv preprint arXiv:2502.00484* (2025). https://arxiv.org/abs/2502.00484

[18] M. D. Intriligator. 1973. A Probabilistic Model of Social Choice. *The Review of Economic Studies* 40, 4 (1973), 553–560. https://doi.org/10.2307/2296588

[19] H. Moulin. 1980. On strategy-proofness and single peakedness. *Public Choice* 35, 4 (1980), 437–455. https://doi.org/10.1007/BF00128122

[20] Simon Rey, Felicia Schmidt, and Jan Maly. 2025. The (Computational) Social Choice Take on Indivisible Participatory Budgeting.

[21] Ariel Rosenfeld and Nimrod Talmon. 2021. What Should We Optimize in Participatory Budgeting? An Experimental Study. *CoRR* abs/2111.07308 (2021). arXiv:2111.07308 [cs.MA] https://arxiv.org/abs/2111.07308

[22] Chris D. Skedgel, Allan J. Wailoo, and Ron L. Akehurst. 2015. Choosing vs. Allocating: Discrete Choice Experiments and Constant-Sum Paired Comparisons for the Elicitation of Societal Preferences. *Health Expectations* 18, 5 (2015), 1227–1240. https://doi.org/10.1111/hex.12098

[23] Piotr Skowron, Arkadii Slinko, Stanisław Szufa, and Nimrod Talmon. 2020. Participatory Budgeting with Cumulative Votes. *arXiv preprint arXiv:2009.02690* (2020). https://arxiv.org/abs/2009.02690

[24] Warut Suksompong and Nicholas Teh. 2026. Voting in Divisible Settings: A Survey. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI-2026)*.

[25] Joshua C. Yang, Carina I. Hausladen, Dominik Peters, Evangelos Pournaras, Regula Hänggli Fricker, and Dirk Helbing. 2024. Designing Digital Voting Systems for Citizens: Achieving Fairness and Legitimacy in Participatory Budgeting. *Digital Government: Research and Practice* 5, 1 (2024), 1–16. https://doi.org/10.1145/3665332

APPENDIX

A POLL INTERFACE AND QUESTION DESIGN

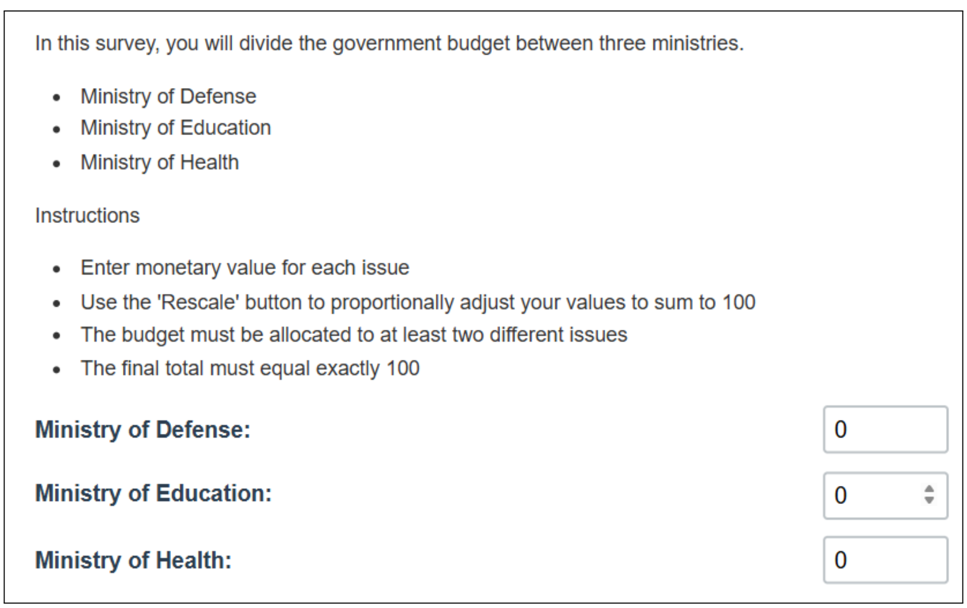

Figure 1: Initial screen where participants enter their ideal budget allocation.

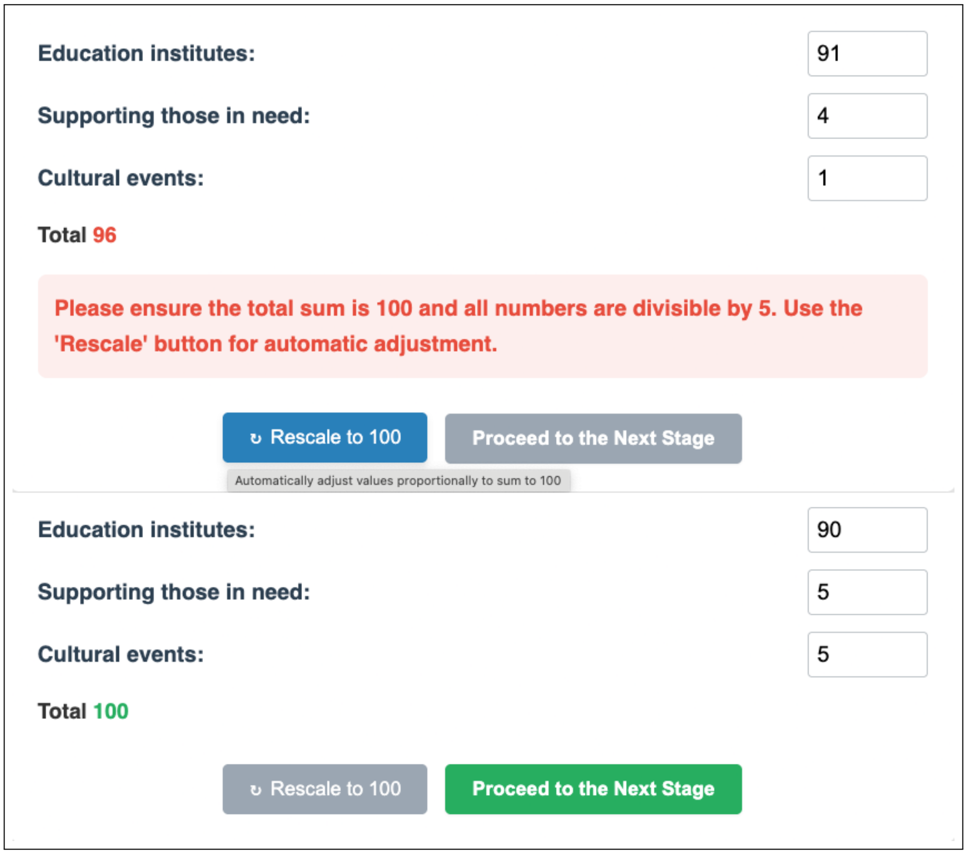

Figure 2: An example of the automatic budget rescaling feature. (Top) A participant's initial allocation that does not sum to 100. (Bottom) The allocation after using the "Rescale" button, which automatically adjusts the values to meet the poll's constraints while preserving the user's proportional intent.

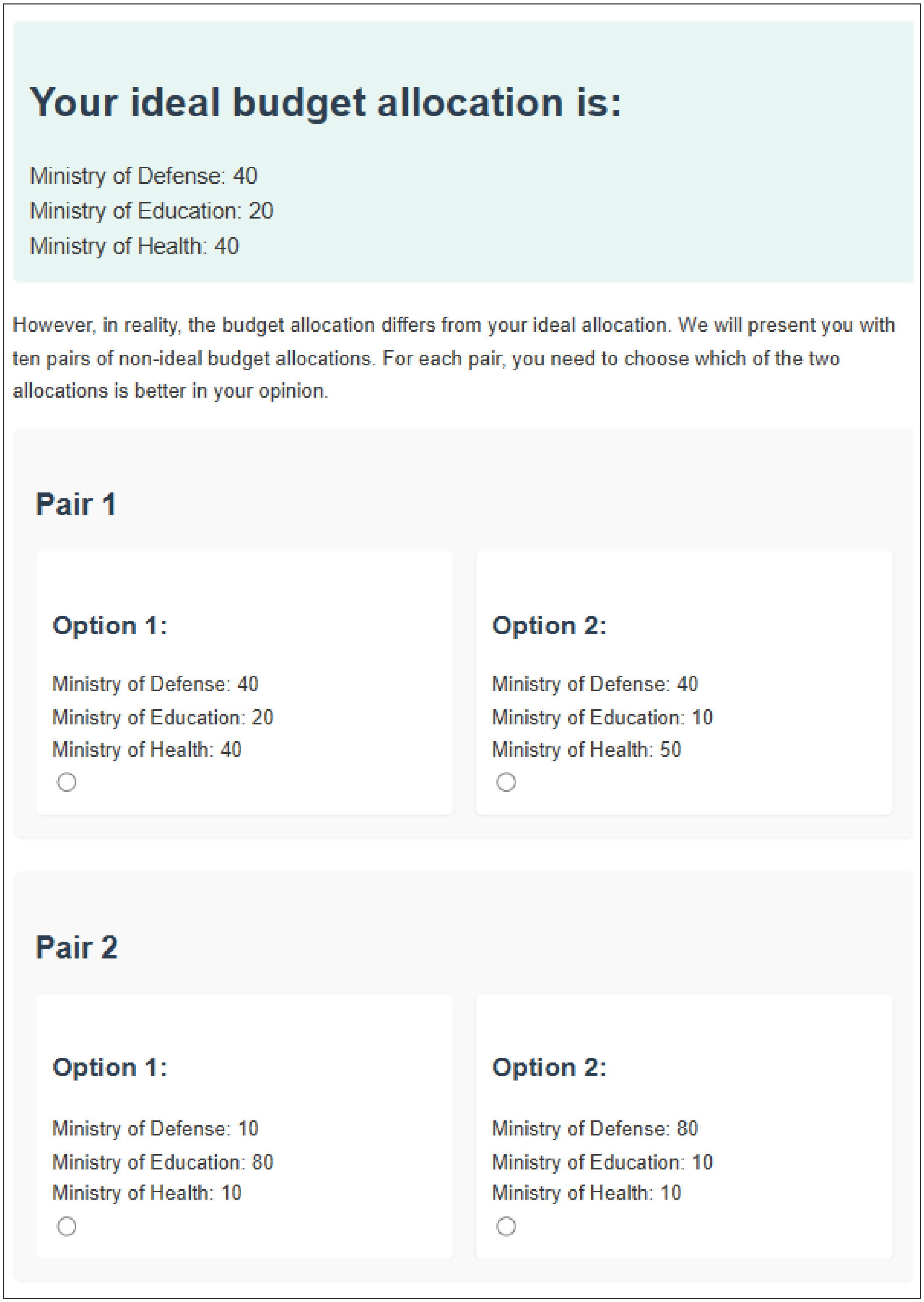

Figure 3: Example of a pairwise comparison question, where participants are asked to choose between two alternative allocations. Pair 1 is an alertness check: Option 1 is identical to the ideal budget, so a user choosing Option 2 will be filtered out.

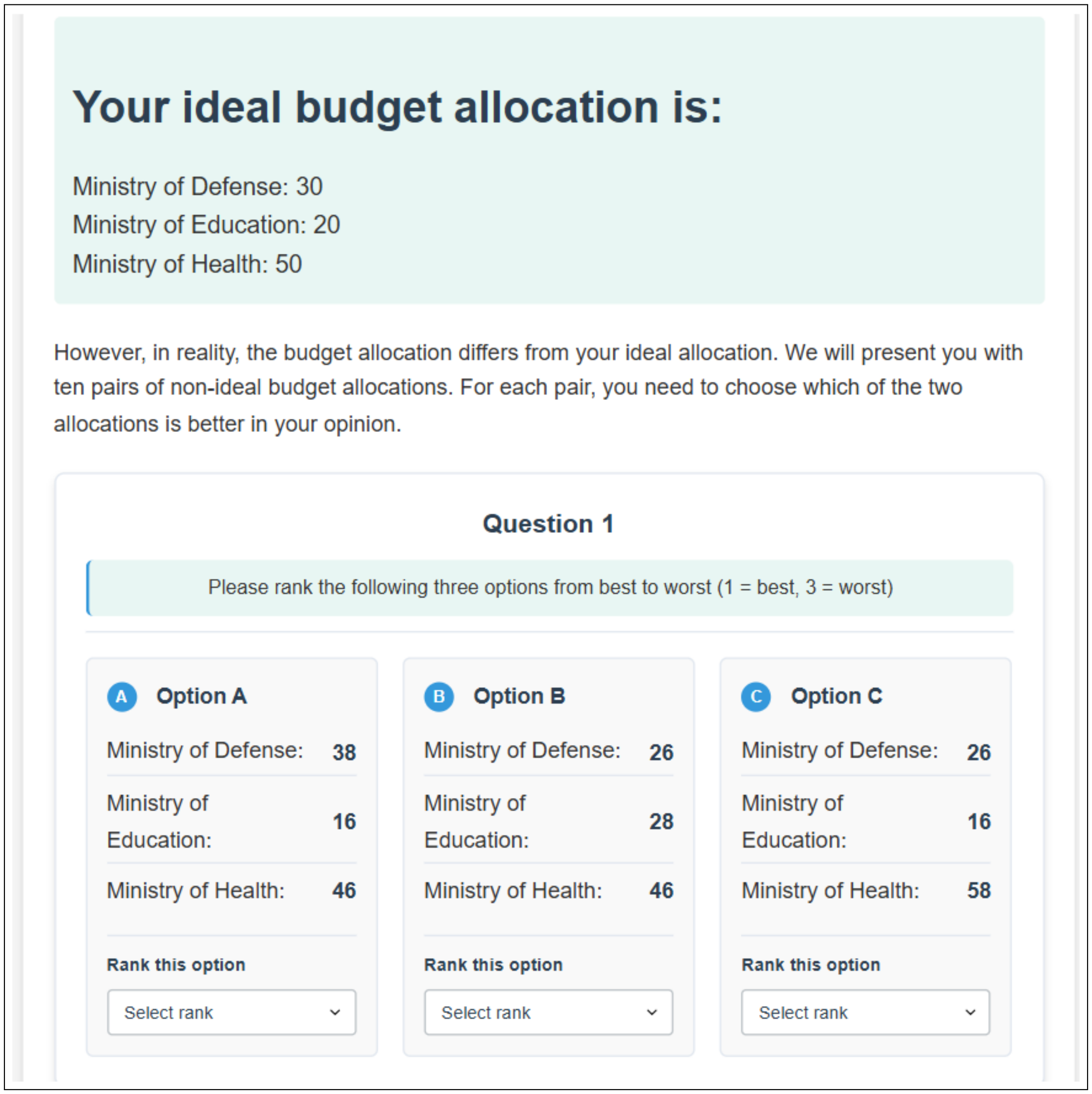

Figure 4: A question where participants are required to rank the three options.

B APPENDIX TO SECTION 5.1: DISTRIBUTION OF PEAK ALLOCATIONS

:

Table 1: Most frequent peak allocations

Optimal allocation	Dimension	Frequency
$[40, 30, 30]$	3	321
$[50, 25, 25]$	3	199
$[60, 20, 20]$	3	104
$[40, 40, 20]$	3	73
$[35, 35, 30]$	3	64
$[30, 40, 30]$	3	61
$[25, 25, 25, 25]$	4	60
$[35, 30, 35]$	3	58
$[20, 20, 20, 20, 20]$	5	53
$[40, 20, 40]$	3	52
$[40, 20, 20, 20]$	4	45
$[30, 35, 35]$	3	43
$[30, 30, 40]$	3	37
$[50, 20, 20, 10]$	4	23
$[20, 20, 60]$	3	22
$[40, 20, 10, 10, 10]$	5	22
$[30, 20, 20, 20, 10]$	5	22
$[50, 50]$	2	21
$[30, 25, 25, 20]$	4	21
$[70, 15, 15]$	3	20
$[30, 20, 30, 20]$	4	20

C APPENDIX TO SECTION 5.2: COMPARING SPECIFIC UTILITY MODELS

Algorithm 1 Pair generation for comparing two utility models

Require: Utility models U_1, U_2; number of projects m; ideal budget $\mathbf{p}$; number of pairs k; minimum allocation ℓ
Ensure: A set S of k informative comparison pairs

$V \leftarrow \textsc{GenerateFeasibleBudgets}(m, \ell)$
for all $v \in V$ **do**
  $u_1(v) \leftarrow U_1(\mathbf{p}, v)$
  $u_2(v) \leftarrow U_2(\mathbf{p}, v)$
end for
$r_1 \leftarrow \textsc{RankNormalize}(\{u_1(v)\}_{v \in V})$
$r_2 \leftarrow \textsc{RankNormalize}(\{u_2(v)\}_{v \in V})$ ▷ values in $[0, 1]$
$P \leftarrow \emptyset$
for all (v_i, v_j) with $i < j$ **do**
  if $r_1(v_i) > r_1(v_j)$ **and** $r_2(v_i) < r_2(v_j)$ **then**
    score $\leftarrow \min(r_1(v_i) - r_1(v_j),\ r_2(v_j) - r_2(v_i))$
    $P \leftarrow P \cup \{(v_i, v_j, \text{score})\}$
  else if $r_1(v_i) < r_1(v_j)$ **and** $r_2(v_i) > r_2(v_j)$ **then**
    score $\leftarrow \min(r_1(v_j) - r_1(v_i),\ r_2(v_i) - r_2(v_j))$
    $P \leftarrow P \cup \{(v_j, v_i, \text{score})\}$
  end if
end for
Sort P by decreasing score
$S \leftarrow$ first k pairs in P
return S

- $\textsc{GenerateFeasibleBudgets}(m, \ell)$ generates the set V of all feasible budget vectors over m issues. Each budget allocates a percentage to every issue such that (i) the total allocation sums to 100, (ii) each allocation is a multiple of 5, and (iii) each issue receives at least ℓ.
- $\textsc{UtilityByU1}(v, \mathbf{p})$ and $\textsc{UtilityByU2}(v, \mathbf{p})$ compute the utility of a budget vector v relative to the agent's ideal budget $\mathbf{p}$ according to utility models U_1 and U_2, respectively.
- $\textsc{RankNormalize}(\{u(v)\}_{v \in V})$ takes the utilities of all budget vectors in V, ranks them from lowest to highest, and maps these ranks linearly to the interval $[0, 1]$. The least-preferred vector receives value 0, the most-preferred vector receives value 1, and intermediate vectors are assigned proportionally spaced values. This normalization makes utilities from different models comparable while preserving ordinal preferences.

Illustrative Example

To illustrate the algorithm, consider a participant whose ideal allocation is $\mathbf{p} = [40, 30, 30]$. They might be presented with the following pair of alternative allocations with Pair Score of 0.34. Note that Pair Score = min(L1 advantage, Leontief advantage). Advantage = how much better one vector is than the other in each metric. Higher score is a clearer choice.

Poll	**Allocation A: [55, 35, 10]**	**Allocation B: [20, 15, 65]**
ℓ_1 vs. Leontief	$\ell_1 = 40$, Leontief = 0.5	$\ell_1 = 70$, Leontief = 0.33

C.1 Aggregate Preference Results

Table 2, Table 3, and Table 4 summarize the aggregate results comparing the proportion of participants whose preferences are aligned with each utility model at varying consistency thresholds.

Table 2: Summary of participant preferences by model comparison and consistency threshold, for three issues (percentages out of all participants in each comparison).

Comparison	60%	70%	80%	90%	100%	Total Participants
ℓ_1 over ℓ_2	6.5% (2)	12.9% (4)	6.5% (2)	3.2% (1)	-	31
ℓ_2 over ℓ_1	22.6% (7)	12.9% (4)	19.4% (6)	3.2% (1)	-	31
ℓ_1 over Leontief	15.6% (5)	21.9% (7)	9.4% (3)	12.5% (4)	21.9% (7)	32
Leontief over ℓ_1	12.5% (4)	6.3% (2)	0.0% (0)	0.0% (0)	0.0% (0)	32
KL over ℓ_1	19.4% (6)	16.1% (5)	16.1% (5)	3.2% (1)	-	31
ℓ_1 over KL	12.9% (4)	3.2% (1)	3.2% (1)	3.2% (1)	-	31
KL over ℓ_2	12.9% (4)	9.7% (3)	16.1% (5)	6.5% (2)	3.2% (1)	31
ℓ_2 over KL	19.4% (6)	19.4% (6)	6.5% (2)	0.0% (0)	0.0% (0)	31
ℓ_2 over Leontief	10.0% (3)	13.3% (4)	20.0% (6)	20.0% (6)	26.7% (8)	30
Leontief over ℓ_2	3.3% (1)	0.0% (0)	3.3% (1)	0.0% (0)	0.0% (0)	30
KL over Leontief	6.3% (2)	12.5% (4)	18.8% (6)	18.8% (6)	28.1% (9)	32
Leontief over KL	6.3% (2)	3.1% (1)	0.0% (0)	0.0% (0)	0.0% (0)	32

Table 3: Summary of participant preferences by model comparison and consistency threshold, for four issues.

Comparison	60%	70%	80%	90%	100%	Total
KL over Leontief	3.1% (1)	18.8% (6)	12.5% (4)	9.4% (3)	40.6% (13)	32
Leontief over KL	6.3% (2)	3.1% (1)	0.0% (0)	3.1% (1)	0.0% (0)	32
KL over ℓ_1	12.5% (4)	12.5% (4)	12.5% (4)	12.5% (4)	12.5% (4)	32
ℓ_1 over KL	12.5% (4)	3.1% (1)	3.1% (1)	0.0% (0)	0.0% (0)	32
KL over ℓ_2	20.0% (6)	10.0% (3)	6.7% (2)	13.3% (4)	20.0% (6)	30
ℓ_2 over KL	10.0% (3)	0.0% (0)	6.7% (2)	0.0% (0)	3.3% (1)	30
ℓ_1 over Leontief	23.3% (7)	13.3% (4)	16.7% (5)	10.0% (3)	16.7% (5)	30
Leontief over ℓ_1	3.3% (1)	0.0% (0)	3.3% (1)	0.0% (0)	0.0% (0)	30
ℓ_1 over ℓ_2	3.3% (1)	0.0% (0)	3.3% (1)	0.0% (0)	3.3% (1)	32
ℓ_2 over ℓ_1	15.6% (5)	21.8% (7)	15.6% (5)	9.4% (3)	28.1% (9)	32
ℓ_2 over Leontief	0.0% (0)	16.1% (5)	9.7% (3)	6.5% (2)	41.9% (13)	31
Leontief over ℓ_2	3.2% (1)	0.0% (0)	6.5% (2)	3.2% (1)	9.7% (3)	31

Table 4: Summary of participant preferences by model comparison and consistency threshold, for five issues.

Comparison	60%	70%	80%	90%	100%	Total
KL over Leontief	9.4% (3)	6.3% (2)	12.5% (4)	9.4% (3)	31.3% (10)	32
Leontief over KL	0.0% (0)	9.4% (3)	3.1% (1)	6.3% (2)	0.0% (0)	32
KL over ℓ_1	6.3% (2)	12.5% (4)	18.8% (6)	9.4% (3)	25.0% (8)	32
ℓ_1 over KL	3.1% (1)	3.1% (1)	9.4% (3)	3.1% (1)	3.1% (1)	32
KL over ℓ_2	28.1% (9)	9.4% (3)	9.4% (3)	0.0% (0)	12.5% (4)	32
ℓ_2 over KL	9.4% (3)	12.5% (4)	3.1% (1)	3.1% (1)	3.1% (1)	32
ℓ_1 over Leontief	10.0% (3)	13.3% (4)	10.0% (3)	23.3% (7)	16.7% (5)	30
Leontief over ℓ_1	3.3% (1)	0.0% (0)	3.3% (1)	3.3% (1)	6.7% (2)	30
ℓ_1 over ℓ_2	3.2% (1)	3.2% (1)	0.0% (0)	0.0% (0)	0.0% (0)	31
ℓ_2 over ℓ_1	12.9% (4)	19.4% (6)	16.1% (5)	6.5% (2)	35.5% (11)	31
ℓ_2 over Leontief	9.7% (3)	9.7% (3)	9.7% (3)	9.7% (3)	32.3% (10)	31
Leontief over ℓ_2	0.0% (0)	6.5% (2)	0.0% (0)	6.5% (2)	9.7% (3)	31

Table 5: Summary of attention check failures across polls

Poll	Total Participants	Failed	Exclusion Rate
3 Topics	1,439	524	36.41%
4 Topics	527	252	47.82%
5 Topics	586	293	50.00%

C.2 Extending Budget Vectors Beyond Three Categories

As the number of issues increases, the computational complexity of generating and evaluating comparison pairs grows substantially. In particular, naively computing distances or identifying informative pairs across all alternatives induces a quadratic dependence on the number of issues, resulting in an $O(n^2)$ complexity that quickly becomes computationally heavy in higher dimensions. To address this challenge, we developed a dedicated algorithmic procedure that significantly reduces the effective computational burden by avoiding exhaustive pairwise comparisons. This approach enables scalable distance evaluation and pair generation even as the dimensionality of the budget vector increases, thereby preserving the practical feasibility of the framework for settings with many issues.

Finding the k Most Different Pairs. [4] for this algorithm.

We are given n items, where each item i is associated with two real-valued attributes (a_i, b_i). For each pair (i, j), the difference score is defined as

$$d(i, j) := \min\left(|a_i - a_j|,\ |b_i - b_j|\right).$$

The goal is to identify the $k \ll n^2$ pairs with the highest difference scores, without explicitly enumerating all $\binom{n}{2}$ pairs.

Key idea. Instead of ranking all pairs, the algorithm searches for the largest threshold D such that there exist at least k pairs (i, j) satisfying

$$d(i, j) \geq D.$$

Equivalently, such pairs must satisfy both $|a_i - a_j| \geq D$ and $|b_i - b_j| \geq D$.

Binary search over D. The algorithm performs a binary search over possible values of D. For each candidate D, it checks whether the number of pairs with $d(i, j) \geq D$ is at least k. Since there are at most $O(n^2)$ distinct values of $d(i, j)$, this requires $O(\log n)$ iterations.

Counting pairs for a fixed D. Items are first sorted by their a-values. Using a two-pointers technique, for each item i we maintain a set of items j such that $a_j - a_i \geq D$. These items are stored in a balanced search tree ordered by their b-values. For each i, we count how many such j satisfy either

$$b_j \leq b_i - D \quad \text{or} \quad b_j \geq b_i + D.$$

To count efficiently, we maintain, in each node in the tree, the number of elements smaller and larger than its element. Then, we search for $b_i - D$ and $b_i + D$ in the tree. Each query and update takes $O(\log n)$ time, yielding a total running time of $O(n \log n)$ for counting pairs for a fixed D.

Selecting the pairs. After finding the maximal threshold D^* such that at least k pairs satisfy $d(i, j) \geq D^*$, the same procedure is run again to explicitly enumerate qualifying pairs. Any k of these pairs may be returned.

Complexity. The overall running time of the algorithm is

$$O(n \log^2 n + k),$$

which is significantly faster than the naive $O(n^2)$ approach when $k \ll n^2$.

[4]We are grateful to
Siddhanth Ramakrishnan from computer science stackexchange (https://cs.stackexchange.com/a/175997/1342)

D APPENDIX TO SECTION 5.3: MONOTONICITY PROPERTIES

Algorithm 2 Pair-generation for checking star-shapedness

Require: Participant's ideal budget $\mathbf{p}$; weights $\Lambda = \{0.1, 0.2, \ldots, 0.9\}$
Ensure: A set S of budget allocation questions

```
1: Initialize empty set of questions S ← ∅
2: for each λ ∈ Λ do
3:     Let q be a random vector representing a budget allocation
4:     Construct convex combination: q_λ ← λp + (1 − λ)q
5:     Add (q, q_λ) to the poll set S
6: end for
7: return S
```

Example Calculation

Table 6: An example calculation for a participant with an ideal allocation of $[30, 40, 30]$**.**

λ	q	Calculation	Weighted vector
0.1	$[20, 60, 20]$	$0.1 \cdot [30, 40, 30] + 0.9 \cdot [20, 60, 20]$	$[21, 58, 21]$
0.2	$[25, 35, 40]$	$0.2 \cdot [30, 40, 30] + 0.8 \cdot [25, 35, 40]$	$[26, 36, 38]$
0.3	$[40, 20, 40]$	$0.3 \cdot [30, 40, 30] + 0.7 \cdot [40, 20, 40]$	$[37, 26, 37]$
0.4	$[10, 70, 20]$	$0.4 \cdot [30, 40, 30] + 0.6 \cdot [10, 70, 20]$	$[18, 58, 24]$
0.5	$[50, 30, 20]$	$0.5 \cdot [30, 40, 30] + 0.5 \cdot [50, 30, 20]$	$[40, 35, 25]$
0.5	$[60, 15, 25]$	$0.5 \cdot [30, 40, 30] + 0.5 \cdot [60, 15, 25]$	$[45, 27.5, 27.5]$
0.6	$[35, 45, 20]$	$0.6 \cdot [30, 40, 30] + 0.4 \cdot [35, 45, 20]$	$[32, 42, 26]$
0.7	$[40, 50, 10]$	$0.7 \cdot [30, 40, 30] + 0.3 \cdot [40, 50, 10]$	$[33, 43, 24]$
0.8	$[20, 40, 40]$	$0.8 \cdot [30, 40, 30] + 0.2 \cdot [20, 40, 40]$	$[28, 40, 32]$
0.9	$[45, 25, 30]$	$0.9 \cdot [30, 40, 30] + 0.1 \cdot [45, 25, 30]$	$[31.5, 38.5, 30]$

Distribution of participants by consistency level

Consistency Level	# of Participants	Percentage of Participants
60%	4	4.8%
70%	6	7.1%
80%	7	8.3%
90%	23	27.4%
100%	40	47.6%

Table 7: Distribution of participants by consistency level

Consistency results per λ

Table 8: Consistency results per λ.

λ	Average Consistency (%)	Total Pairs
0.1	77.46	71
0.2	90.14	71
0.3	92.96	71
0.4	84.51	71
0.5	91.55	142
0.6	91.55	71
0.7	90.14	71
0.8	88.73	71
0.9	91.55	71

Algorithm 3 Pair generation for checking star-shapedness with rounded values

Require: Participant ideal budget $\mathbf{p}$; weights $\Lambda = \{0.1, 0.2, \ldots, 0.9\}$; number of projects m
Ensure: A set S of questions consisting of random vectors and their adjusted convex combinations

```
S ← ∅
for all λ ∈ Λ do
    Let q be a random budget allocation vector
    q_λ ← λp + (1 − λ)q
    for i = 1 to m − 1 do
        Round q_λ[i] to the nearest integer (ties to even)
    end for
    q_λ[m] ← 100 − Σ_{i=1}^{m−1} q_λ[i]
    for i = 1 to m do
        Round q_λ[i] to the nearest multiple of 5
    end for
    if Σ_{i=1}^{m} q_λ[i] ≠ 100 then
        j ← arg max_i q_λ[i]
        q_λ[j] ← 100 − Σ_{i≠j} q_λ[i]
    end if
    S ← S ∪ {(q, q_λ)}
end for
return S
```

Note that the rounded vectors are not exact convex-combinations anymore, but they are close to convex combinations.

An example demonstrating how the convex combination vector $\mathbf{q}_\lambda$ was constructed for a participant with a given ideal and random vector. The process illustrates how intermediate adjustments ensure the total sum equals 100 and values are rounded to meaningful units.

	Vector
Ideal vector p	$[30, 40, 30]$
Random vector $\mathbf{q}$	$[45, 25, 30]$
$\lambda = 0.9$ convex combination	$[31.5, 38.5, 30]$
Rounded to nearest integer	$[30, 38, 30]$
Adjusted last project to sum 100	$[30, 38, 32]$
Rounded to nearest multiple of 5	$[30, 40, 30]$
Final $\mathbf{q}_\lambda$	$[30, 40, 30]$

Table 9: Example illustrating the construction of $\mathbf{q}_\lambda$ for a peaked participant.

Algorithm 4 Pair-generation for testing Multi-Dimensional Single-Peakedness (MDSP)

Require: Participant's ideal allocation $\mathbf{p}$; target number of questions k
Ensure: A set S of questions where one allocation is directionally closer to the peak

```
Initialize empty set of questions S ← ∅
while |S| < k do
    Sample two random allocations q_1, q_2
    if ∀i : (q_1^i − p^i)(q_2^i − p^i) ≥ 0 and ∀i : |q_2^i − p^i| ≤ |q_1^i − p^i| and ∃j : |q_2^j − p^j| < |q_1^j − p^j| then
                                  ▷ q_2 moves weakly toward the peak in all dimensions and strictly in at least one
        Add ordered pair (q_1, q_2) to S
    end if
end while
return S
```

Importantly, closeness is defined directionally: the preferred allocation must lie on the same side of the peak in every dimension and move weakly toward it, rather than merely being closer in absolute distance.

Consistency Results for Multi-Dimensional Single-Peaked Preferences

Result Category	Percentage
Perfect consistency	76.5%
Consistency ≥ 90%	23.5%
Closer vector chosen	97.6%
Farther vector chosen	2.4%

Table 10: Consistency results supporting the multi-dimensional single-peaked assumption

MDSP results per number of topics:

Table 11: MDSP Results for 3, 4, and 5 Topics: Overall Survey Statistics

# Topics	Consistency Level	# of Users	Far Vector	Near Vector
3 Topics	90.0%	8	10.0%	90.0%
	100.0%	26	0.0%	100.0%
	Total	34	2.4%	97.6%
4 Topics	50.0%	1	50.0%	50.0%
	80.0%	1	20.0%	80.0%
	90.0%	7	10.0%	90.0%
	100.0%	26	0.0%	100.0%
	Total	35	4.0%	96.0%
5 Topics	60.0%	1	40.0%	60.0%
	70.0%	1	30.0%	70.0%
	80.0%	1	20.0%	80.0%
	90.0%	6	10.0%	90.0%
	100.0%	27	0.0%	100.0%
	Total	36	4.2%	95.8%

Algorithm 5 Pair-generation for testing peak-linearity

Require: Participant's ideal budget p; weights $\Lambda = \{0.25, 0.5, 0.75\}$
Ensure: A set S of comparison questions based on convex combinations

1: Set $S := \emptyset$
2: Define extreme vectors:
   $\mathbf{v}_A := [10, 10, 80]$
   $\mathbf{v}_B := [10, 80, 10]$
   $\mathbf{v}_C := [80, 10, 10]$
3: Add the pairs $(\mathbf{v}_A, \mathbf{v}_B)$, $(\mathbf{v}_A, \mathbf{v}_C)$, $(\mathbf{v}_B, \mathbf{v}_C)$ to S
4: **for** each $\lambda \in \Lambda$ **do**
5: Compute convex combinations:
   $\mathbf{q}_A := \lambda p + (1-\lambda)\mathbf{v}_A$
   $\mathbf{q}_B := \lambda p + (1-\lambda)\mathbf{v}_B$
   $\mathbf{q}_C := \lambda p + (1-\lambda)\mathbf{v}_C$
6: Add the pairs $(\mathbf{q}_A, \mathbf{q}_B)$, $(\mathbf{q}_A, \mathbf{q}_C)$, $(\mathbf{q}_B, \mathbf{q}_C)$ to S
7: **end for**
8: **return** S

Table 12: Consistency of pairwise comparisons for different weight percentiles

Weight (λ)	**A vs. B**	**A vs. C**	**B vs. C**	**Average Consistency**
25% ($\lambda = 0.25$)	68% (30/44)	73% (32/44)	70% (31/44)	70% (93/132)
50% ($\lambda = 0.5$)	80% (35/44)	84% (37/44)	80% (35/44)	81% (107/132)
75% ($\lambda = 0.75$)	91% (40/44)	80% (35/44)	80% (35/44)	83% (110/132)
All percentiles	80% (105/132)	79% (104/132)	77% (101/132)	**78% (310/396)**

To illustrate the algorithm, we present below an example of answers that were inconsistent with peak-linearity.

Table 13: An example of inconsistency is a participant whose ideal budget is $[30, 20, 50]$. This example illustrates an inconsistent choice pattern, as the participant's preferences over the λ-weighted averages do not consistently mirror the ranking of the original extreme vectors, contrary to what would be expected under a peak-linear utility function.

Pair / λ	Option A	Option B	Participant Choice
Extreme Vectors	[10, 10, 80]	[10, 80, 10]	A
$\lambda = 0.25$	[24, 18, 58]	[25, 35, 40]	B
$\lambda = 0.5$	[20, 15, 65]	[20, 50, 30]	A
$\lambda = 0.75$	[16, 12, 72]	[15, 65, 20]	B

Consistency Across Participant Groups

Metric	All users	Star-shaped users
Users	44	22
Overall consistency	78.3%	**84.1%**
Transitivity rate	**96%**	94%
Order consistency	70.1%	**78.6%**

Table 14: Consistency metrics across all participants and among those preferring weighted vectors

Table 15: Distribution of transitivity consistency levels in poll 3.

Transitivity Level	100%	75%	50%
All users	39	5	1
Star-Shaped users	18	3	1

Example of intransitive choices

Table 16 presents an example of a participant whose pairwise selections violate transitivity. Although allocation A is preferred over B, and B over C, the participant ultimately prefers C over A, creating an intransitive cycle:

Table 16: Example of an intransitive preference cycle from poll 3.

Pair	**Option 1**	**Option 2**	**Chosen Allocation**
#1	$[42, 32, 26]$	$[25, 50, 25]$	$[25, 50, 25]$
#2	$[42, 32, 26]$	$[25, 32, 43]$	$[42, 32, 26]$
#3	$[25, 50, 25]$	$[25, 32, 43]$	$[25, 32, 43]$

E RELATIONS BETWEEN MONOTONICITY PROPERTIES: PROOFS

E.1 Peak-Linearity as a Stronger Condition than Star-Shapedness

PROPOSITION E.1. *If a continuous utility function U is* peak-linear *around the peak* $\mathbf{p}$*, then it is* star-shaped *around* $\mathbf{p}$*. The converse does not hold: there exist star-shaped utility functions that are not peak-linear.*

PROOF. Assume by contradiction that U is peak-linear and continuous, but not strictly star-shaped. Since U is not strictly star-shaped, there exists a distribution $\mathbf{q} \neq \mathbf{p}$ and a scalar $\lambda \in (0, 1)$ such that moving towards $\mathbf{p}$ does not strictly increase the utility:

$$U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}) \leq U(\mathbf{p}, \mathbf{q}).$$

Let us define a sequence of distributions $\{\mathbf{q}_n\}_{n=0}^{\infty}$ recursively:

$$\mathbf{q}_0 = \mathbf{q}$$
$$\mathbf{q}_{n+1} = \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}_n$$

By our initial assumption, $U(\mathbf{p}, \mathbf{q}_1) \leq U(\mathbf{p}, \mathbf{q}_0)$.

According to the definition of peak-linearity, for any two distributions x, y and a scalar $\alpha \in (0, 1)$, we have

$$U(\mathbf{p}, \alpha\mathbf{p} + (1 - \alpha)x) \leq U(\mathbf{p}, \alpha\mathbf{p} + (1 - \alpha)y) \iff U(\mathbf{p}, x) \leq U(\mathbf{p}, y).$$

Applying this property recursively for all n, we obtain a monotonically non-increasing sequence of utilities:

$$U(\mathbf{p}, \mathbf{q}_0) \geq U(\mathbf{p}, \mathbf{q}_1) \geq U(\mathbf{p}, \mathbf{q}_2) \geq \cdots \geq U(\mathbf{p}, \mathbf{q}_n) \geq \ldots$$

Notice that the distance between $\mathbf{q}_n$ and $\mathbf{p}$ shrinks by a factor of $(1 - \lambda)$ at each step. Since $\lambda \in (0, 1)$, as $n \to \infty$, the sequence of distributions $\mathbf{q}_n$ converges to the peak $\mathbf{p}$.

Now, substitute these specific terms into the definition of peak-linearity. By setting $\alpha = \lambda$, $x = \mathbf{q}_1$, and $y = \mathbf{q}_0$, the left-hand side of the equivalence yields:

$$U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}_1) \leq U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}_0)$$

Here, examining the arguments within the utility function and recalling the recursive construction of our sequence, we can see that the argument on the right-hand side is exactly the definition of $\mathbf{q}_1$. Similarly, the argument on the left-hand side represents a further contraction from $\mathbf{q}_1$ towards the peak, which is precisely the definition of $\mathbf{q}_2$. Therefore, the inequality translates directly to:

$$U(\mathbf{p}, \mathbf{q}_2) \leq U(\mathbf{p}, \mathbf{q}_1)$$

We have thus established that $\mathbf{q}_2$ yields a utility less than or equal to that of $\mathbf{q}_1$. We can now repeat this exact process: by substituting $\mathbf{q}_2$ and $\mathbf{q}_1$ as x and y respectively in the peak-linearity definition, we obtain $U(\mathbf{p}, \mathbf{q}_3) \leq U(\mathbf{p}, \mathbf{q}_2)$. Applying this property recursively chains these inequalities into a single monotonically non-increasing sequence:

$$U(\mathbf{p}, \mathbf{q}_0) \geq U(\mathbf{p}, \mathbf{q}_1) \geq U(\mathbf{p}, \mathbf{q}_2) \geq \cdots \geq U(\mathbf{p}, \mathbf{q}_n) \geq \ldots$$

Because the utility function $U(\mathbf{p}, \cdot)$ is continuous, the limit of the utilities must equal the utility of the limit point:

$$\lim_{n \to \infty} U(\mathbf{p}, \mathbf{q}_n) = U(\mathbf{p}, \mathbf{p}).$$

Since the sequence $U(\mathbf{p}, \mathbf{q}_n)$ is monotonically non-increasing to this limit, every element in the sequence must be greater than or equal to the limit. In particular for the first element:

$$U(\mathbf{p}, \mathbf{q}) \geq U(\mathbf{p}, \mathbf{p}).$$

However, $\mathbf{p}$ is the unique global maximum, meaning $U(\mathbf{p}, \mathbf{p}) > U(\mathbf{p}, \mathbf{q})$ for all $\mathbf{q} \neq \mathbf{p}$. This is a direct contradiction. Therefore, our assumption must be false, and U must be strictly star-shaped. □

E.1.1 The Role of Continuity. Without continuity, the above proposition does not hold.

Example E.2. Consider an indicator utility function where the agent only derives utility from their exact ideal distribution:

$$U(\mathbf{p}, \mathbf{q}) = \begin{cases} 1 & \text{if } \mathbf{q} = \mathbf{p} \\ 0 & \text{if } \mathbf{q} \neq \mathbf{p} \end{cases}$$

This function is peak-linear: For any $\mathbf{q}_1, \mathbf{q}_2$ and $\lambda \in (0, 1)$, the condition holds trivially. For instance, if $\mathbf{q}_1 = \mathbf{p}$ and $\mathbf{q}_2 \neq \mathbf{p}$, both sides of the equivalence evaluate to $1 \geq 0$. If both $\mathbf{q}_1, \mathbf{q}_2 \neq \mathbf{p}$, both sides evaluate to $0 \geq 0$.

However, this function violates the strict star-shaped property. For any $\mathbf{q} \neq \mathbf{p}$ and $\lambda \in (0, 1)$, the intermediate distribution $\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}$ is strictly not equal to $\mathbf{p}$. Thus:

$$U(\mathbf{p}, \lambda\mathbf{p} + (1 - \lambda)\mathbf{q}) = 0 \not> 0 = U(\mathbf{p}, \mathbf{q}).$$

Moving strictly closer to the peak does not strictly increase utility.

PROOF. **Star-shaped does not imply peak-linear.** Consider the utility function

$$U(\mathbf{q}) = -\big((\mathbf{q}_1 - \mathbf{p}_1)^2 + (\mathbf{q}_2 - \mathbf{p}_2)^4\big).$$

This function is strictly maximized at $\mathbf{p}$.

Star-shapedness. For any $\lambda \in (0, 1)$,

$$U(\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}) = -\big((\lambda\mathbf{p}_1 + (1 - \lambda)\mathbf{q}_1 - \mathbf{p}_1)^2 + (\lambda\mathbf{p}_2 + (1 - \lambda)\mathbf{q}_2 - \mathbf{p}_2)^4\big)$$

$$= -\big(((\lambda - 1)\mathbf{p}_1 + (1 - \lambda)\mathbf{q}_1)^2 + ((\lambda - 1)\mathbf{p}_2 + (1 - \lambda)\mathbf{q}_2)^4\big)$$

$$= -\big(((1 - \lambda)(\mathbf{q}_1 - \mathbf{p}_1))^2 + ((1 - \lambda)(\mathbf{q}_2 - \mathbf{p}_2))^4\big) = -\big((1 - \lambda)^2(\mathbf{q}_1 - \mathbf{p}_1)^2 + (1 - \lambda)^4(\mathbf{q}_2 - \mathbf{p}_2)^4\big).$$

Since $(1 - \lambda)^2 < 1$ and $(1 - \lambda)^4 < 1$, we obtain

$$U(\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}) > U(\mathbf{q}),$$

for every $\mathbf{q} \neq \mathbf{p}$. Hence U is star-shaped.

Failure of peak-linearity. Fix $\mathbf{p} = (0, 0)$ and consider

$$\mathbf{q}^{(1)} = (0, 2), \qquad \mathbf{q}^{(2)} = (1, 1).$$

Then

$$U(\mathbf{q}^{(1)}) = -16, \qquad U(\mathbf{q}^{(2)}) = -2,$$

so

$$U(\mathbf{q}^{(2)}) > U(\mathbf{q}^{(1)}).$$

However, taking $\lambda = 0.9$,

$$U(\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}^{(1)}) = -0.0016, \qquad U(\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}^{(2)}) = -0.0101,$$

and therefore

$$U(\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}^{(1)}) > U(\lambda\mathbf{p} + (1 - \lambda)\mathbf{q}^{(2)}),$$

Thus peak-linearity fails. □

E.2 Multi-Dimensional Single-Peaked as a Stronger Condition than Star-Shapedness

PROPOSITION E.3. *If the utility function U is* multi-dimensional single-peaked *around the peak* $\mathbf{p}$*, then U is also* star-shaped *around* $\mathbf{p}$.

PROOF. For any allocation $\mathbf{q}$ and any coefficient $\alpha \in [0, 1]$, define

$$\mathbf{q}^{\alpha} := \mathbf{p} + \alpha(\mathbf{q} - \mathbf{p}).$$

To prove star-shapedness around $\mathbf{p}$, we must show that for every $\mathbf{q} \neq \mathbf{p}$ and every $\alpha \in (0, 1)$,

$$U(\mathbf{q}^{\alpha}) > U(\mathbf{q}),$$

That is, moving toward the peak strictly increases utility.

Recall the definition of *multi-dimensional single-peakedness*: $\mathbf{q}_2$ is said to be *closer to* $\mathbf{p}$ *than* $\mathbf{q}_1$ if for every dimension j, either

$$q_{1j} \geq q_{2j} \geq p_j \quad \text{or} \quad q_{1j} \leq q_{2j} \leq p_j,$$

and in at least one dimension, the inequality between q_{1j} and q_{2j} is strict. Whenever $\mathbf{q}_2$ is closer to $\mathbf{p}$ than $\mathbf{q}_1$, multi-dimensional single-peakedness requires that

$$U(\mathbf{q}_2) > U(\mathbf{q}_1).$$

Now take $\mathbf{q}_1 = \mathbf{q}$ and $\mathbf{q}_2 = \mathbf{q}^{\alpha}$ for some $\alpha \in (0, 1)$. Fix any coordinate j. There are three cases:

(1) If $q_j = p_j$, then $\mathbf{q}^{\alpha}_j = p_j$ as well.

(2) If $q_j > p_j$, then

$$q_j - p_j > 0 \quad \text{and} \quad \mathbf{q}^{\alpha}_j - p_j = \alpha(q_j - p_j),$$

so

$$q_j > \mathbf{q}^{\alpha}_j > p_j.$$

(3) If $q_j < p_j$, then

$$q_j - p_j < 0 \quad \text{and} \quad \mathbf{q}^{\alpha}_j - p_j = \alpha(q_j - p_j),$$

so

$$q_j < \mathbf{q}^{\alpha}_j < p_j.$$

Thus, for every coordinate j, $\mathbf{q}^{\alpha}_j$ lies weakly between q_j and p_j in the same direction from the peak, and whenever $q_j \neq p_j$ the inequality between q_j and $\mathbf{q}^{\alpha}_j$ is strict. If $\mathbf{q} \neq \mathbf{p}$, there is at least one such coordinate, and therefore $\mathbf{q}^{\alpha}$ is *closer to* $\mathbf{p}$ than $\mathbf{q}$ in the sense of the definition.

By multi-dimensional single-peakedness, it follows that

$$U(\mathbf{q}^{\alpha}) > U(\mathbf{q}).$$

Since this holds for every $\mathbf{q} \neq \mathbf{p}$ and every $\alpha \in (0, 1)$, we conclude that utility strictly increases as one moves along the line segment from $\mathbf{q}$ toward $\mathbf{p}$. This is exactly the definition of a *star-shaped* utility function around $\mathbf{p}$.

□

E.3 Logical Relation Between Peak-Linear and Multi-Dimensional Single-Peakedness

PROPOSITION E.4. *Peak-linearity and Multi-Dimensional Single-Peakedness are independent: neither property implies the other in general.*

Below, we give a short justification and counterexample:

MDSP does not *imply Peak-linear:* Let $\mathbf{p} \in \mathbb{R}^2$ be fixed and define

$$U(q) = -\Big((\mathbf{q}_1 - \mathbf{p}_1)^2 + (\mathbf{q}_2 - \mathbf{p}_2)^4\Big).$$

(1) U satisfies MDSP.

Suppose $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$ lie on the same orthant relative to $\mathbf{p}$ and that $\mathbf{q}^{(2)}$ is coordinatewise closer to $\mathbf{p}$ than $\mathbf{q}^{(1)}$, with strict inequality in at least one coordinate. Then

$$|\mathbf{q}^{(2)}_1 - \mathbf{p}_1| < |\mathbf{q}^{(1)}_1 - \mathbf{p}_1|, \qquad |\mathbf{q}^{(2)}_2 - \mathbf{p}_2| \leq |\mathbf{q}^{(1)}_2 - \mathbf{p}_2|.$$

Since both $x \mapsto x^2$ and $x \mapsto x^4$ are strictly increasing on $\mathbb{R}_{\geq 0}$, we obtain

$$(\mathbf{q}^{(2)}_1 - \mathbf{p}_1)^2 + (\mathbf{q}^{(2)}_2 - \mathbf{p}_2)^4 < (\mathbf{q}^{(1)}_1 - \mathbf{p}_1)^2 + (\mathbf{q}^{(1)}_2 - \mathbf{p}_2)^4,$$

and therefore

$$U(\mathbf{q}^{(2)}) > U(\mathbf{q}^{(1)}).$$

Thus U satisfies MDSP.

(2) U is not peak-linear.

Let $\mathbf{p} = (0, 0)$ and consider

$$\mathbf{q}^{(1)} = (0, 2), \qquad \mathbf{q}^{(2)} = (1, 1).$$

Then

$$U(\mathbf{q}^{(1)}) = -16, \qquad U(\mathbf{q}^{(2)}) = -2,$$

so

$$U(\mathbf{q}^{(2)}) > U(\mathbf{q}^{(1)}).$$

Now take $\lambda = 0.9$. Then

$$U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}^{(1)}) = -0.0016, \qquad U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}^{(2)}) = -0.0101,$$

and hence

$$U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}^{(1)}) > U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}^{(2)}),$$

so the ordering reverses. Therefore, the ordinal equivalence required by peak-linearity fails.

Peak-linear does not *imply MDSP:.* We present a counterexample in which the utility function is peak-linear yet violates multi-dimensional single-peakedness:

Let $\mathbf{p} \in \mathbb{R}^m$ be fixed and define

$$U(\mathbf{q}) = -\max_{j=1,\dots,m} |q_j - p_j|.$$

(1) U is peak-linear.

For any $\mathbf{q}$ and any $\lambda \in [0, 1]$,

$$\lambda\mathbf{p} + (1-\lambda)\mathbf{q} - \mathbf{p} = (1-\lambda)(\mathbf{q} - \mathbf{p}),$$

and therefore

$$U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}) = -\max_j |(1-\lambda)(q_j - p_j)| = (1-\lambda)U(\mathbf{q}).$$

Hence, for any $\mathbf{q}(1), \mathbf{q}^{(2)}$,

$$U(\mathbf{q}^{(1)}) \geq U(\mathbf{q}^{(2)}) \iff U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}^{(1)}) \geq U(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}^{(2)}),$$

so U satisfies peak-linearity.

(2) U violates MDSP.

Let $m = 2$ and take

$$\mathbf{p} = (0, 0), \qquad \mathbf{q}^{(1)} = (2, 2), \qquad \mathbf{q}^{(2)} = (1, 2).$$

Then $\mathbf{q}^{(2)}$ is coordinatewise closer to $\mathbf{p}$ than $\mathbf{q}^{(1)}$ with strict inequality in the first coordinate. However,

$$U(\mathbf{q}^{(1)}) = -2, \qquad U(\mathbf{q}^{(2)}) = -2.$$

Thus utility does not strictly increase when moving closer in every coordinate, and MDSP fails.

Therefore,

$$\text{Peak-linear} \not\Rightarrow \text{MDSP}.$$

Overall, these two properties are independent: neither one implies the other.

E.4 Leontief utilities are peak-linear

PROOF. Let $U(p, q) = \min_{j \in A} \left(\frac{q_j}{p_j}\right)$. For any alternative distribution q and $\lambda \in [0, 1)$, we evaluate the utility of the mixed distribution:

$$U(p, \lambda p + (1-\lambda)q) = \min_{j \in A} \left(\frac{\lambda p_j + (1-\lambda)q_j}{p_j}\right) = \min_{j \in A} \left(\lambda + (1-\lambda)\frac{q_j}{p_j}\right).$$

Since λ and $(1-\lambda)$ are non-negative constants, this affine transformation preserves the order of the elements inside the minimum operator. Thus, we can extract the constants:

$$U(p, \lambda p + (1-\lambda)q) = \lambda + (1-\lambda)\min_{j \in A} \left(\frac{q_j}{p_j}\right) = \lambda + (1-\lambda)U(p, q).$$

Now, for any two distributions q_1, q_2 and $\lambda \in (0, 1)$:

$$U(p, q_1) \geq U(p, q_2) \iff (1-\lambda)U(p, q_1) \geq (1-\lambda)U(p, q_2) \iff \lambda + (1-\lambda)U(p, q_1) \geq \lambda + (1-\lambda)U(p, q_2).$$

$$U(p, \lambda p + (1-\lambda)q_1) \geq U(p, \lambda p + (1-\lambda)q_2) \iff \lambda + (1-\lambda)U(p, q_1) \geq \lambda + (1-\lambda)U(p, q_2).$$

Because $1 - \lambda > 0$, we can subtract λ and divide both sides by $1 - \lambda$ without changing the inequality's direction. This simplifies exactly to $U(p, q_1) \geq U(p, q_2)$, perfectly satisfying the ordinal definition of peak-linearity. □

E.5 KL-Divergence Utility implies Multi-Dimensional Single-Peak

PROOF. Recall that the KL-based utility is given by:

$$U(\mathbf{p},\mathbf{q}) = -\sum_{j\in A} p_j \cdot \ln\left(\frac{p_j}{q_j}\right) = \sum_{j\in A} p_j \ln(q_j) - \sum_{j\in A} p_j \ln(p_j).$$

Note that for the KL utility function to be well-defined, we assume that all allocations and peaks are strictly positive (i.e., $q_j > 0$ and $p_j > 0$ for all $j \in A$). This means we are avoiding division by zero and undefined logarithmic values.

Since the natural logarithm is a strictly concave function, U is strictly concave with respect to $\mathbf{q}$. Recall the gradient inequality for a strictly concave function f: for any two distinct points x and y, we have:

$$f(x) < f(y) + \nabla f(y) \cdot (x - y)$$

Rearranging this inequality to isolate the difference gives:

$$f(y) - f(x) > \nabla f(y) \cdot (y - x)$$

Let $\mathbf{q}_1, \mathbf{q}_2 \in \Delta$ be two distinct allocations, where $\mathbf{q}_2$ is closer to $\mathbf{p}$ than $\mathbf{q}_1$ according to the MDSP definition. By substituting $x = \mathbf{q}_1$ and $y = \mathbf{q}_2$ into our inequality, we obtain:

$$U(\mathbf{p},\mathbf{q}_2) - U(\mathbf{p},\mathbf{q}_1) > \sum_{j\in A} \frac{\partial U}{\partial q_j}(\mathbf{q}_2) \cdot (q_{2j} - q_{1j}) = \sum_{j\in A} \frac{p_j}{q_{2j}} (q_{2j} - q_{1j}).$$

Because both $\mathbf{q}_1$ and $\mathbf{q}_2$ are valid allocations, the sum of their coordinates must equal the total budget B. Therefore, the sum of their component-wise differences is zero:

$$\sum_{j\in A} (q_{2j} - q_{1j}) = 0.$$

We can subtract this sum (which is exactly zero) from our right-hand side without changing its value:

$$\sum_{j\in A} \frac{p_j}{q_{2j}} (q_{2j} - q_{1j}) - \sum_{j\in A} 1 \cdot (q_{2j} - q_{1j}) = \sum_{j\in A} \left(\frac{p_j}{q_{2j}} - 1\right) (q_{2j} - q_{1j}).$$

Recall the definition of MDSP: *Let* $\mathbf{q}_1$ *and* $\mathbf{q}_2$ *be two alternative distributions. We say that* $\mathbf{q}_2$ *is* closer to $\mathbf{p}$ than $\mathbf{q}_1$ *if for every issue* j*, either* $q_{1j} \geq q_{2j} \geq p_j$ *or* $q_{1j} \leq q_{2j} \leq p_j$*, and for at least one issue* j*, the inequality between* q_{1j} *and* q_{2j} *is strict. A utility-model function* U *is said to be multi-dimensional single-peaked if whenever* $\mathbf{q}_2$ *is closer to* $\mathbf{p}$ *than* $\mathbf{q}_1$*, it holds that* $U(\mathbf{p},\mathbf{q}_2) > U(\mathbf{p},\mathbf{q}_1)$.

This means that for every coordinate j, one of the following cases holds:

- $q_{1j} \leq q_{2j} \leq p_j$: In this case, $(q_{2j} - q_{1j}) \geq 0$. Furthermore, since $q_{2j} \leq p_j$, we have $\frac{p_j}{q_{2j}} \geq 1$, which implies $\left(\frac{p_j}{q_{2j}} - 1\right) \geq 0$. The product of two non-negative terms is non-negative.
- $q_{1j} \geq q_{2j} \geq p_j$: In this case, $(q_{2j} - q_{1j}) \leq 0$. Furthermore, since $q_{2j} \geq p_j$, we have $\frac{p_j}{q_{2j}} \leq 1$, which implies $\left(\frac{p_j}{q_{2j}} - 1\right) \leq 0$. The product of two non-positive terms is non-negative.

In all cases, every term in the summation is non-negative. Therefore, the entire sum is greater than or equal to zero:

$$\sum_{j\in A} \left(\frac{p_j}{q_{2j}} - 1\right) (q_{2j} - q_{1j}) \geq 0.$$

Combining this non-negative sum with our strict gradient inequality from earlier, we finally get:

$$U(\mathbf{p},\mathbf{q}_2) - U(\mathbf{p},\mathbf{q}_1) > \sum_{j\in A} \left(\frac{p_j}{q_{2j}} - 1\right) (q_{2j} - q_{1j}) \geq 0 \implies U(\mathbf{p},\mathbf{q}_2) > U(\mathbf{p},\mathbf{q}_1).$$

This proves that the KL utility model is multi-dimensional single-peaked. □

E.6 KL-Divergence Utility is not Peak-Linear

PROOF. Let $m = 2$ and assume a peak:

$$\mathbf{p} = (0.8, 0.2).$$

Take two alternative allocations:

$$\mathbf{q}_1 = (0.5, 0.5), \qquad \mathbf{q}_2 = (0.97, 0.03), \qquad \lambda = 0.9.$$

Using the KL-based utility

$$U(\mathbf{p},\mathbf{q}) = -\sum_{j\in A} p_j \ln\left(\frac{p_j}{q_j}\right) = \sum_j p_j \ln(q_j) - \sum_j p_j \ln(p_j),$$

the constant term for our peak is $\sum p_j \ln(p_j) = 0.8\ln(0.8) + 0.2\ln(0.2) \approx -0.5004$. We compute the utilities for $\mathbf{q}_1$ and $\mathbf{q}_2$:

$$U(\mathbf{p}, \mathbf{q}_1) = (0.8 \ln(0.5) + 0.2 \ln(0.5)) - (-0.5004) \approx -0.1927$$

$$U(\mathbf{p}, \mathbf{q}_2) = (0.8 \ln(0.97) + 0.2 \ln(0.03)) - (-0.5004) \approx -0.2253$$

Hence, initially,

$$U(\mathbf{p}, \mathbf{q}_1) > U(\mathbf{p}, \mathbf{q}_2).$$

We compute the interpolated allocations with $\lambda = 0.9$:

$$\tilde{\mathbf{q}}_1 = \lambda \mathbf{p} + (1 - \lambda)\mathbf{q}_1 = 0.9(0.8, 0.2) + 0.1(0.5, 0.5) = (0.77, 0.23)$$

$$\tilde{\mathbf{q}}_2 = \lambda \mathbf{p} + (1 - \lambda)\mathbf{q}_2 = 0.9(0.8, 0.2) + 0.1(0.97, 0.03) = (0.817, 0.183)$$

Now, we recalculate the utilities for the interpolated points:

$$U(\mathbf{p}, \tilde{\mathbf{q}}_1) = (0.8 \ln(0.77) + 0.2 \ln(0.23)) - (-0.5004) \approx -0.002624$$

$$U(\mathbf{p}, \tilde{\mathbf{q}}_2) = (0.8 \ln(0.817) + 0.2 \ln(0.183)) - (-0.5004) \approx -0.000944$$

Thus,

$$U(\mathbf{p}, \tilde{\mathbf{q}}_2) > U(\mathbf{p}, \tilde{\mathbf{q}}_1),$$

which is a strict reversal of the original preference.

We have shown that

$$U(\mathbf{p}, \mathbf{q}_1) > U(\mathbf{p}, \mathbf{q}_2) \quad \text{but} \quad U(\mathbf{p}, \lambda \mathbf{p} + (1 - \lambda)\mathbf{q}_2) > U(\mathbf{p}, \lambda \mathbf{p} + (1 - \lambda)\mathbf{q}_1).$$

This contradicts the definition of peak-linearity. Hence, KL utilities are not peak-linear. $\square$

F APPENDIX TO SECTION 5.4: CHECKING SYMMETRY

Symmetry Example

Let the ideal budget be $[27, 33, 40]$. The table below shows two illustrative cases: one for Issue Symmetry and one for Sign Symmetry. In each case, the alternatives have identical ℓ_p distances from the ideal allocation, despite differences in the positions or signs of the deviations.

Symmetry Type	Alternative	Allocation	Deviations from Ideal
Issue Symmetry	A	$[36, 33, 31]$	$[+9, 0, -9]$
	B	$[27, 42, 31]$	$[0, +9, -9]$
Sign Symmetry	A	$[26, 30, 44]$	$[-1, -3, +4]$
	B	$[28, 36, 36]$	$[+1, +3, -4]$

Algorithm 6 Pair-generation for testing Issue Symmetry

Require: An ideal budget $\mathbf{p}$; number of poll sets k; number of issues m
Ensure: A set S of budget allocation poll sets

Set $S := \emptyset$
while $|S| < mk$ **do**
  Let $\mathbf{q}_1, \mathbf{q}_2$ be random vectors representing budget allocations
  Compute differences from peak:
  $\mathbf{d}_1 := \mathbf{q}_1 - \mathbf{p}$, $\mathbf{d}_2 := \mathbf{q}_2 - \mathbf{p}$
  for $j = 1$ **to** $m - 1$ **do**
    Generate j-th rotation of differences: $\mathbf{d}_1^{(j)}, \mathbf{d}_2^{(j)}$
    Compute shifted allocations:
    $\mathbf{q}_1^{(j)} := \mathbf{p} + \mathbf{d}_1^{(j)}$, $\mathbf{q}_2^{(j)} := \mathbf{p} + \mathbf{d}_2^{(j)}$
    if all components of $\mathbf{q}_1^{(j)}$ and $\mathbf{q}_2^{(j)} \geq 0$ **then**
      Append $(\mathbf{q}_1^{(j)}, \mathbf{q}_2^{(j)})$ to S
    end if
  end for
  Add the original pair $(\mathbf{q}_1, \mathbf{q}_2)$ to S
end while
return S

Algorithm 7 Pair-generation for testing Sign Symmetry

Require: An ideal budget $\mathbf{p}$; number of poll sets k
Ensure: A set S of poll sets containing original and negated deviations

Set $S := \emptyset$
while $|S| < k$ **do**
  Let $\mathbf{q}_1, \mathbf{q}_2$ be random vectors representing budget allocations
  Compute deviations from peak:
  $\mathbf{d}_1 := \mathbf{q}_1 - \mathbf{p}$, $\mathbf{d}_2 := \mathbf{q}_2 - \mathbf{p}$
  Generate negated deviations:
  $\mathbf{d}'_1 := -\mathbf{d}_1$, $\mathbf{d}'_2 := -\mathbf{d}_2$
  if $\mathbf{p} + \mathbf{d}'_1 \geq 0$ **and** $\mathbf{p} + \mathbf{d}'_2 \geq 0$ **then**
    Set negated allocations:
    $\mathbf{q}'_1 := \mathbf{p} + \mathbf{d}'_1$, $\mathbf{q}'_2 := \mathbf{p} + \mathbf{d}'_2$
    Add the allocations $(\mathbf{q}_1, \mathbf{q}_2)$ and $(\mathbf{q}'_1, \mathbf{q}'_2)$ as a poll set to S
  end if
end while
return S

Example F.1. Suppose the ideal budget is $\mathbf{p} = [30, 30, 40]$. The pair-generation algorithm for Issue Symmetry could generate the following pairs.

	$\mathbf{q}_1$	$\mathbf{q}_2$
Original	[50, 34, 16]	[20, 25, 55]
Deviation	[20, 4, -24]	[-10, -5, 15]
Rotated Deviation 1	[-24, 20, 4]	[15, -10, -5]
Rotation 1	[6, 50, 44]	[45, 20, 35]
Rotated Deviation 2	[4, -24, 20]	[-5, 15, -10]
Rotation 2	[34, 6, 60]	[25, 45, 30]

Analysis of Symmetry in Preferences

The following tables presents the proportion of participants whose responses demonstrated project or Sign Symmetry at different levels of consistency.

Table 17: Number of participants by consistency level (Sign Symmetry)

Consistency	0	1/6	2/6	3/6	4/6	5/6	6/6
# Participants	1 (3.6%)	1 (3.6%)	4 (14.3%)	6 (21.4%)	8 (28.6%)	6 (21.4%)	2 (7.1%)

Table 18: Number of participants by consistency level (Issue Symmetry)

Consistency	0	1/4	2/4	3/4	4/4
# Participants	8 (21.1%)	8 (21.1%)	12 (31.6%)	6 (15.8%)	4 (10.5%)

Example of project inconsistency

Table 19: Example of project inconsistency for an ideal budget of $[30, 30, 40]$, showing two pairwise comparisons where the participant's choices do not follow a consistent project-level preference pattern

Pair	Option A	Deviation A	Option B	Deviation B	Participant Choice
Pair 1	[66, 5, 29]	[+36, -25, -11]	[6, 60, 34]	[-24, +30, -6]	A
Pair 2	[5, 19, 76]	[-25, -11, +36]	[60, 24, 16]	[+30, -6, -24]	B

This example demonstrates issue-level inconsistency. In both pairs, Option A exhibits the same deviation pattern, differing only by a permutation of deviations across issues. Likewise, Option B also follows an identical deviation pattern across pairs, again differing only in the assignment of deviations to specific issues.

Despite this structural equivalence, the participant chooses Option A in the first pair and Option B in the second pair. That is, for the same deviation patterns, applied to different issues, the participant's preference reverses.

Algorithm 8 Pair-generation for testing issue symmetry among issues with identical allocations

Require: Participant's budget vector $\mathbf{p} \in \mathbb{R}^M$; number of poll questions K
Ensure: A set S of comparison pairs

```
Initialize empty set of questions S := ∅
Identify two issues i, j such that p_i = p_j.
If multiple such pairs exist (possible when M ≥ 4), select the pair with the largest amount; break other ties arbitrarily.
Let r_1, ..., r_{M−2} be the remaining issues whose budget remains fixed.
Define magnitude step size: Δ := p_i/K
for t = 1 to K do
    x := round(tΔ)
    Construct option q_1 ∈ ℝ^M:
  q_{1,i} := p_i + x, q_{1,j} := p_j − x
  q_{1,r_k} := p_{r_k} for k = 1, ..., M − 2
    Construct option q_2 ∈ ℝ^M:
  q_{2,i} := p_i − x, q_{2,j} := p_j + x
  q_{2,r_k} := p_{r_k} for k = 1, ..., M − 2
    if q_1 and q_2 are valid budget allocations then
        Add pair (q_1, q_2) to S
    end if
end for
return S
```

Issue Asymmetry Results

Table 20: Number of Users by Consistency Level

Consistency Level (%)	# of Users
50.0	2 (6.5%)
60.0	4 (12.9%)
70.0	5 (16.1%)
80.0	4 (12.9%)
90.0	3 (9.7%)
100.0	13 (41.9%)
Total	31 (100.0%)

G APPENDIX TO SECTION 5.5: CONSISTENCY IN ASYMMETRY

Algorithm 9 Pair-generation for testing consistency in issue-asymmetry

Require: Participant's ideal budget $\mathbf{p}$; weights $\Lambda = \{0.2, 0.4\}$; vector length m
Ensure: A set S of questions with options generated by rotating difference vectors
Initialize empty set of questions $S := \emptyset$
Let $\min(\mathbf{p})$ denote the smallest component of $\mathbf{p}$
for each $\lambda \in \Lambda$ **do**
  Compute $X_\lambda := \max(1, \text{round}(\lambda \cdot \min(\mathbf{p})))$ ▷ Define base difference vectors
  $\mathbf{d}_p := ((m-1)X_\lambda, -X_\lambda, \ldots, -X_\lambda)$ ▷ Concentrated increase
  $\mathbf{d}_n := (-(m-1)X_\lambda, X_\lambda, \ldots, X_\lambda)$ ▷ Concentrated decrease
  for each $\mathbf{d} \in \{\mathbf{d}_p, \mathbf{d}_n\}$ **do**
    for $j = 1$ **to** m **do**
      Compute cyclic rotation: $\Delta^{(j)} := \mathit{cyclic\_shift}(\mathbf{d}, j)$
    end for
    Create a ranking question with the m options:
  $\mathbf{q}_1 := \mathbf{p} + \Delta^{(1)}, \ldots, \mathbf{q}_m := \mathbf{p} + \Delta^{(m)}$
    Add the question to S
  end for
end for
return S

Example of Generated Allocation Options

To illustrate how alternative allocations were generated for a given participant, Table 21 presents the options produced for an ideal vector of $\mathbf{p} = (85, 15, 5)$ under different values of λ. Each option is accompanied by its corresponding deviation vector Δ, showing the directional adjustment applied to the original ideal allocation.

Table 21: Generated allocation options for $\mathbf{p} = (85, 15, 5)$ under different λ values, with corresponding deviation vectors.

Question	λ	Option 1	$\mathbf{d}_1$	Option 2	$\mathbf{d}_2$	Option 3	Δ_3
1	0.2	$[87, 14, 4]$	$[2, -1, -1]$	$[84, 17, 4]$	$[-1, 2, -1]$	$[84, 14, 7]$	$[-1, -1, 2]$
2	0.4	$[89, 13, 3]$	$[4, -2, -2]$	$[83, 19, 3]$	$[-2, 4, -2]$	$[83, 13, 9]$	$[-2, -2, 4]$
3	0.2	$[83, 16, 6]$	$[-2, 1, 1]$	$[86, 13, 6]$	$[1, -2, 1]$	$[86, 16, 3]$	$[1, 1, -2]$
4	0.4	$[81, 17, 7]$	$[-4, 2, 2]$	$[87, 11, 7]$	$[2, -4, 2]$	$[87, 17, 1]$	$[2, 2, -4]$

Participants' Choice Consistency

	over $1/3$	over $2/3$	$3/3$ consistent
Number of Participants	27	16	7
Percentage	72.9%	43.2%	18.9%

Algorithm 10 Pair-generation for testing consistency in sign asymmetry

Require: Participant's ideal budget $\mathbf{p} \in \mathbb{R}^M$
Ensure: A set S of comparison pairs

Initialize empty set of questions $S := \emptyset$
Let $\min(\mathbf{p})$ denote the smallest component of $\mathbf{p}$
Define base magnitude: $X_{\text{base}} := \max\left(1, \left\lfloor \frac{\min(\mathbf{p})}{10} \right\rfloor\right)$
Define magnitude levels: $\Lambda := \{X_{\text{base}}, 2X_{\text{base}}, 3X_{\text{base}}, 4X_{\text{base}}\}$
for each target category $i \in \{1, \ldots, M\}$ **do**
  for each magnitude $X \in \Lambda$ **do**
    Define a **concentrated loss** vector $\mathbf{d}_1 \in \mathbb{R}^M$:
    $d_{1,i} := -(M-1)X, \quad d_{1,j} := X$ for all $j \neq i$
    Define a **concentrated gain** vector $\mathbf{d}_2 \in \mathbb{R}^M$:
    $d_{2,i} := (M-1)X, \quad d_{2,j} := -X$ for all $j \neq i$
    Construct options: $\mathbf{q}_1 := \mathbf{p} + \mathbf{d}_1$, $\mathbf{q}_2 := \mathbf{p} + \mathbf{d}_2$
    if $\mathbf{q}_1$ and $\mathbf{q}_2$ are valid budget allocations (all entries in $[0, 100]$) **then**
      Add pair $(\mathbf{q}_1, \mathbf{q}_2)$ to S
    end if
  end for
end for
return S

Algorithm 11 Fallback procedure for pair generation

Require: Participant's ideal budget p; target category i; magnitude index $k \in \{1, 2, 3, 4\}$; pre-defined fixed vectors F_k
Ensure: A set S of comparison pairs updated with fallback allocations ▷ Triggered when the primary method fails for a given target category and magnitude

Retrieve pre-defined difference vectors: $(fd_1, fd_2) := F_k$
Rotate fd_1 and fd_2 based on target category i to obtain fd'_1, fd'_2
Construct fallback options:
$q_{f1} := p + fd'_1, \quad q_{f2} := p + fd'_2$
if q_{f1} and q_{f2} are valid budget allocations **then**
  Add pair (q_{f1}, q_{f2}) to the poll set S
end if

Example of a generated comparison pair

Table 22: Example of a generated comparison pair for a participant with an ideal budget of $p = (60, 30, 10)$, using magnitude level 2 ($X = 2$). The concentrated change is applied to the first category, resulting in one option with a concentrated loss and another with a concentrated gain.

	Option A (Concentrated Loss)	**Option B** (Concentrated Gain)
Deviation Vector	$(-4, +2, +2)$	$(+4, -2, -2)$
Resulting Allocation	$(56, 32, 12)$	$(64, 28, 8)$

Participant Preference Matrix

Preference matrix for a participant where the rows correspond to topics, and the columns correspond to magnitude levels. Each cell is colored to indicate whether the participant preferred a distributed decrease (orange) or a concentrated decrease (blue).

Target Category	X=3	X=6	X=9	X=12	%
Category 0	●	●	●	●	0%
Category 1	●	●	●	●	100%
Category 2	●	●	●	●	100%
Total	67%	67%	67%	67%	67%

Figure 5: A participant's preference matrix showing full consistency choices for each row.

Target Category	X=3	X=6	X=9	X=12	%
Category 0	●	●	●	●	75%
Category 1	●	●	●	●	0%
Category 2	●	●	●	●	0%
Total	33%	33%	33%	0%	25%

Figure 6: A participant's preference matrix showing consistent choices for category 1 and 2 but inconsistent yet monotonic choices for category 0 across different budget change magnitudes.

Target Category	X=3	X=6	X=9	X=12	%
Ministry of Defense	●	●	●	●	0%
Ministry of Education	●	●	●	●	50%
Ministry of Health	●	●	●	●	25%
Total	33%	33%	0%	33%	25%

Figure 7: A participant's preference matrix showing consistent choices for Defense, but inconsistent choices for Education and Health across different budget change magnitudes.

Distribution of Participants by Concentrated and Distributed Levels

Table 23: Number of Participants Across Different Levels of Budget Concentration and Distribution

Percentage (%)	Concentrated	Distributed
16.7	1	0
25.0	3	1
33.3	3	7
41.7	4	6
50.0	6	5
58.3	4	5
66.7	6	3
75.0	1	3
83.3	0	1

H APPENDIX TO SECTION 5.6: COMPARING BIENNIAL BUDGETS

Algorithm 12 Pair-generation for testing preferences among biennial budgets

Require: Participant's ideal budget $\mathbf{p}$; number of repetitions per sub-poll $k = 4$
Ensure: A set S of 12 questions comparing biennial budget allocations

(x, y) denotes (year 1, year 2)
Initialize empty set of questions $S := \emptyset$
for $i = 1$ **to** k **do**
  Randomly generate a budget vector r_i ▷ Sub-poll 1: Ideal year 1 vs. Ideal year 2
  Add question to S: Option 1 $(\mathbf{p}, r_i)$, Option 2 $(r_i, \mathbf{p})$
  ▷ Sub-poll 2: Fixed year 1, Ideal year 2 vs. Balanced year 2
  Fix year 1 budget to r_i
  Option 1: $(r_i, \mathbf{p})$
  Option 2: $(r_i, \mathbf{q}_i)$ such that $\frac{r_i+q_i}{2} = \mathbf{p}$
  Add question to S
  ▷ Sub-poll 3: Fixed year 2, Ideal year 1 vs. Balanced year 1
  Fix year 2 budget to r_i
  Option 1: $(\mathbf{p}, r_i)$
  Option 2: $(\mathbf{q}_i, r_i)$ such that $\frac{q_i+r_i}{2} = \mathbf{p}$
  Add question to S
end for
return S

An example of questions with an ideal budget $\mathbf{p} = (50, 30, 20)$ across two years:

Sub-poll	Year 1	Year 2	Description
1	(50,30,20)	(40,25,35)	Ideal in year 1, random in year 2
1	(40,25,35)	(50,30,20)	Random in year 1, ideal in year 2
2	(40,25,35)	(50,30,20)	Ideal in year 2
2	(40,25,35)	(60,35,5)	Average = ideal
3	(50,30,20)	(40,25,35)	Ideal in year 1
3	(60,35,5)	(40,25,35)	Average = ideal

Biennial Poll Results

Sub-poll 1			
Consistency level	**Number of users**	**Ideal Year 1**	**Random**
50%	2	50.00%	50.00%
75%	12	66.7%	33.30%
100%	25	96.00%	4.00%
Total	39	84.60%	15.40%
Sub-poll 2			
Consistency level	**Number of users**	**Ideal Year 2**	**Balanced Year 2**
50%	2	50.00%	50.00%
75%	10	55.00%	45.00%
100%	27	100.00%	0.00%
Total	39	85.90%	14.10%
Sub-poll 3			
Consistency level	**Number of users**	**Ideal Year 1**	**Balanced Year 1**
50%	3	50.00%	50.00%
75%	10	60.00%	40.00%
100%	26	100.00%	0.00%
Total	39	85.90%	14.10%

Biennial Poll Results (Cumulative)

Sub-poll	**over 50%**	**over 75%**	**100%**	**Participants**
Sub-poll 1	100.00% (39)	94.87% (37)	64.10% (25)	39
Sub-poll 2	100.00% (39)	94.87% (37)	69.23% (27)	39
Sub-poll 3	100.00% (39)	92.31% (36)	66.67% (26)	39

Triangle Inequality

Algorithm 13 Pair-generation for testing the triangle inequality

Require: Participant's ideal budget $\mathbf{p}$; positive integer k (number of base change vectors per rotation)
Ensure: A set S of comparisons between concentrated and distributed changes

```
Initialize empty set of questions S := ∅
for each of k random base change vectors do
    Sample q = [x_1, x_2, x_3] such that ∑ x_i = 0, each x_i is multiple of 5, and q ≠ [0, 0, 0]
    Decompose q as q = q_1 + q_2, where
      q_1 := [x_1, 0, −x_1],    q_2 := [0, x_2, −x_2]
    Verify that q_1 ≠ [0, 0, 0] and q_2 ≠ [0, 0, 0]
    if all vectors (p ± q, p ± q_1, p ± q_2) result in valid budgets in [0, 100] then
        Add to S: (p, p + q) vs. (p + q_1, p + q_2)
        Add to S: (p, p − q) vs. (p − q_1, p − q_2)    ▷ Repeat for coordinate rotations [x_2, x_3, x_1] and [x_3, x_1, x_2]
        Repeat the same construction for the two coordinate rotations of q, adding their comparisons to S
    end if
end for
return S
```

Notes.

- Using $k = 2$ base vectors per rotation yields $2 \times 3 \times 2 = 12$ experimental comparisons (plus 2 initial screening questions).
- Sampling constraints (multiples of 5, sum zero) preserve interpretability and ensure all resulting budgets are valid.
- Both positive and negative variants of each change vector are included to examine symmetry with respect to the direction of change.

Illustrative Example of Concentrated and Distributed Budget Deviations

Assume the ideal budget is $\mathbf{p} = [30, 30, 40]$ and the base difference vector is $\mathbf{d}_C = [-20, 10, 10]$, which is decomposed as $\mathbf{d}_C = \mathbf{d}_A + \mathbf{d}_B$, where $\mathbf{d}_A = [-10,\ 10,\ 0], \mathbf{d}_B = [-10,\ 0,\ 10]$. Then:

- Option 1 (concentrated change) is: Year 1: $\mathbf{p} = [30, 30, 40]$, Year 2: $\mathbf{p} + \mathbf{q}_C = [10, 40, 50]$.
- Option 2 (distributed change) is: Year 1: $\mathbf{p} + \mathbf{q}_A = [20, 40, 40]$, Year 2: $\mathbf{p} + \mathbf{q}_B = [20, 30, 50]$.

Table 24: Distribution of Concentrated and Distributed Changes by Consistency Level

Consistency Level (%)	# of Users	Concentrated Change	Distributed Change
50.0	9	50.0%	50.0%
58.3	13	44.3%	55.7%
66.7	5	40.0%	60.0%
75.0	8	43.8%	56.2%
83.3	8	16.7%	83.3%
91.7	4	8.3%	91.7%
100.0	6	16.7%	83.3%
Total	53	34.8%	65.2%

Triangle Inequality Results.

I APPENDIX TO SECTION 6: MUNICIPAL VS. NATIONAL COMPARISONS

ℓ_1 vs ℓ_2 Rank Comparison

Table 25: Comparison of ℓ_1 vs ℓ_2 Rank Preferences Between Municipal and Government Budget Polls

Framing	Consistency Level	# Users	ℓ_1 (Rank)	ℓ_2 (Rank)	Neutral
Municipal Budget	50.0%	12	0.0%	0.0%	100.0%
	60.0%	11	54.5%	45.5%	0.0%
	70.0%	4	25.0%	75.0%	0.0%
	80.0%	4	75.0%	25.0%	0.0%
	90.0%	1	100.0%	0.0%	0.0%
	100.0%	2	50.0%	50.0%	0.0%
	Total	34	35.3%	29.4%	35.3%
Government Budget	50.0%	4	0.0%	0.0%	100.0%
	60.0%	9	22.2%	77.8%	0.0%
	70.0%	8	50.0%	50.0%	0.0%
	80.0%	8	25.0%	75.0%	0.0%
	90.0%	2	50.0%	50.0%	0.0%
	Total	31	29.0%	58.1%	12.9%

Star-Shaped Preference

Table 26: Comparison of Star-Shaped Preference Metrics Between Municipal and Government Budget Polls

Framing	Random	Weighted Average
Municipal Budget	9.0%	91.0%
Government Budget	11.2%	88.8%

Multi-Dimensional Single-Peaked

Table 27: Comparison of Multi-Dimensional Single-Peaked Test Results Between Municipal and Government Budget Polls

Framing	Consistency Level	# Users	Far Vector	Near Vector
Municipal Budget	60.0%	1	40.0%	60.0%
	80.0%	2	20.0%	80.0%
	90.0%	4	10.0%	90.0%
	100.0%	32	0.0%	100.0%
	Total	39	3.1%	96.9%
Government Budget	90.0%	8	10.0%	90.0%
	100.0%	26	0.0%	100.0%
	Total	34	2.4%	97.6%

Peak Linear

Table 28: Comparison of Peak-Linear Consistency Metrics Between Municipal and Government Budget Polls

Framing	Overall Consistency	Transitivity Rate	Order Consistency
Municipal Budget	92.0%	98.3%	87.8%
Government Budget	78.3%	96.0%	70.1%

Issue Symmetry

Table 29: Comparison of Component-Symmetric Consistency Between Municipal and Government Budget Polls

Framing	Average Consistency Rate
Municipal Budget	30.6%
Government Budget	42.5%

Sign Symmetry

Table 30: Comparison of Sign-Symmetry Consistency Between Municipal and Government Budget Polls

Framing	Average Consistency Rate
Municipal Budget	42.4%
Government Budget	61.8%

Identity Asymmetry

Table 31: Comparison of Identity Asymmetry Consistency Levels Between Municipal and Government Budget Polls

Framing	Consistency Level	# Users
Municipal Budget	50.0%	1 (3.2%)
	60.0%	2 (6.5%)
	70.0%	2 (6.5%)
	80.0%	0 (0.0%)
	90.0%	6 (19.4%)
	100.0%	20 (64.5%)
	Total	31 (100.0%)
Government Budget	50.0%	2 (6.5%)
	60.0%	4 (12.9%)
	70.0%	5 (16.1%)
	80.0%	4 (12.9%)
	90.0%	3 (9.7%)
	100.0%	13 (41.9%)
	Total	31 (100.0%)

Asymmetric Loss Distribution

Table 32: Comparison of Asymmetric Loss Distribution Preferences Between Municipal and Government Budget Polls

Framing	Concentrated (Target Decreases)	Distributed (Target Increases)
Municipal Budget	48.0%	52.0%
Government Budget	50.5%	49.5%

Preference Ranking

Table 33: Comparison of Preference Ranking Scores Between Municipal and Government Budget Polls

Framing	Final Score
Municipal Budget	45.3%
Government Budget	45.0%

Biennial Budget Preference

Table 34: Comparison of Biennial Budget Preferences Between Government and Municipal Budget Polls

Framing	Consistency Level	Number of Users	Ideal Year 1	Random
Government Budget	50%	2	50.0%	50.0%
	75%	9	63.9%	36.1%
	100%	25	96.0%	4.0%
	Total	36	85.4%	14.6%
Municipal Budget	50%	2	50.0%	50.0%
	75%	12	66.7%	33.3%
	100%	25	96.0%	4.0%
	Total	39	84.6%	15.4%

Triangle Inequality

Table 35: Comparison of Triangle Inequality Test Results Between Municipal and Government Budget Polls

Framing	Consistency Level	Number of Users	Concentrated Change	Distributed Change
Municipal Budget	50.0%	16	50.0%	50.0%
	58.3%	13	48.1%	51.9%
	66.7%	12	47.2%	52.8%
	75.0%	10	40.0%	60.0%
	83.3%	11	40.9%	59.1%
	91.7%	2	91.7%	8.3%
	100.0%	3	100.0%	0.0%
	Total	67	49.6%	50.4%
Government Budget	50.0%	10	50.0%	50.0%
	58.3%	13	44.3%	55.7%
	66.7%	5	40.0%	60.0%
	75.0%	8	43.8%	56.2%
	83.3%	8	16.7%	83.3%
	91.7%	4	8.3%	91.7%
	100.0%	8	25.0%	75.0%
	Total	56	35.6%	64.4%

J SYSTEM ARCHITECTURE AND REPRODUCIBILITY GUIDE

This appendix provides a practical guide for researchers wishing to replicate this study or utilize the open-source polling framework for new experiments. For comprehensive documentation, including detailed API endpoints, troubleshooting guides, and full database schemas, please refer to the `README.md` file located in the root of the repository, at URL https://github.com/ariel-research/budget-survey.git.

The system is designed using a modular *Strategy Pattern*, allowing researchers to inject new budget subjects (for example: municipal, national, or organizational budgets) and new comparison algorithms without modifying the frontend user interface. The system automatically adapts to the number of subjects (m) defined in the database, having been validated for $m \in \{3, 4, 5\}$.

J.1 Setup and Configuration

The system is containerized using Docker. The following steps outline the process from cloning the repository to configuring the environment.

Prerequisites: Docker and Docker Compose.

(1) **Clone the repository:**

```
git clone https://github.com/ariel-research/budget-survey.git
cd budget-survey
```

(2) **Environment Configuration:** Copy the example environment file and configure the critical application settings, including database credentials, the secret key, and the base URL.

```
cp .env.example .env
# Edit .env to set:
# - SURVEY_BASE_URL (Your hosting domain or localhost:5001)
# - FLASK_SECRET_KEY (For session security)
# - MYSQL_PASSWORD (Database credentials)
```

(3) **External Provider Integration:** The system is designed to work with external panel providers. The configuration in `config.py` defines the redirection logic based on the participant's completion status.

```
# config.py
EXTERNAL_PROVIDER_CONFIG = {
    "BASE_URL": "http://provider-url.com/status.php",
    "STATUS": {
        "COMPLETE": "finish",            # Successful completion
        "ATTENTION_FAILED": "filter",    # Failed attention checks (alertness tests)
        "FILTEROUT": "screenout",        # Failed pre-screening
```

```
        }
    }
```

J.2 System Deployment

To facilitate easy deployment, the repository includes a helper script (‘deploy.sh‘) that handles secret key generation and container orchestration.

Launch the environment: Use the deployment script to start the application. The ‘dev‘ argument enables hot-reloading for code editing, while ‘prod‘ optimizes for data collection and security.

```
# For Development (Coding/Testing):
./scripts/deploy.sh dev

# For Production (Running Experiments):
./scripts/deploy.sh prod
```

The survey interface will be available locally at `http://localhost:5001`.

J.3 Defining Survey Content

To add a new research topic, researchers insert a JSON-structured record into the `stories` table. The system allows multiple experimental conditions to run simultaneously on a single deployment.

Step 1: Define the Story (Subjects). Insert the narrative context and subjects (e.g., Education, Sanitation, Culture).

```
INSERT INTO stories (code, title, description, subjects)
VALUES (
    'municipal_2025',
    JSON_OBJECT('en', 'City Budget', 'loc', '...'),
    JSON_OBJECT('en', 'Allocate funds...', 'loc', '...'),
    JSON_ARRAY(
        JSON_OBJECT('en', 'Education', 'loc', '...'),
        JSON_OBJECT('en', 'Sanitation', 'loc', '...'),
        JSON_OBJECT('en', 'Culture', 'loc', '...')
    )
);
```

Step 2: Configure the Algorithm (The Survey). Create a survey entry linking the story to a specific algorithm strategy.

```
INSERT INTO surveys (id, story_code, active, pair_generation_config)
VALUES (
    114, -- Internal ID used for routing
    'municipal_2025',
    TRUE,
    JSON_OBJECT(
        'strategy', 'l1_vs_l2_rank_comparison',
        'params', JSON_OBJECT('num_pairs', 10)
    )
);
```

Step 3: Distribution. Participants are directed to specific experimental conditions using URL parameters. A valid URL requires three components:

```
.../take-survey/?userID=[UID]&surveyID=[SID]&internalID=[IID]
```

- **userID (UID):** A unique identifier for the participant (passed dynamically by the panel provider) to ensure data linkage and prevent duplicate submissions.
- **surveyID (SID):** An identifier used by the external panel provider to track the specific survey instance and link participant data.
- **internalID (IID):** The specific experimental condition ID (e.g., `114` from Step 2). This parameter forces the system to load the specific Story and Algorithm configuration defined for that ID, enabling precise A/B testing.

J.4 Implementing New Preference Algorithms

The framework supports extending research logic via Python classes.

J.4.1 Method A: Metric-Based Rankings. This method compares two mathematical models (e.g., testing *L1* vs. *L2*). The researcher defines the utility formula, and the system handles the grid search and normalization.

Step 1: Define the Utility Model.

Create a class in `application/services/algorithms/utility_models.py`. The example below implements the L_2 (Euclidean) metric.

```
class L2UtilityModel(UtilityModel):
    @property
    def name(self) -> str:
        return "l2"

    def calculate(self, user_vec: tuple, cand_vec: tuple) -> float:
        # Returns negative distance (higher score = better match)
        dist = np.sqrt(np.sum((np.array(user_vec) - np.array(cand_vec))**2))
        return -float(dist)
```

Step 2: Create the Strategy Wrapper.

Inherit from `GenericRankStrategy` in `rank_strategies.py`.

```
class L1VsL2RankStrategy(GenericRankStrategy):
    def __init__(self, grid_step=None):
        super().__init__(
            utility_model_a_class=L1UtilityModel,
            utility_model_b_class=L2UtilityModel,
            # grid_step: Defines the resolution of the discrete simplex.
            # e.g., step=5 generates vectors with multiples of 5 (0, 5, 10...).
            # Lower steps increase precision but increase computation cost.
            grid_step=grid_step,
            min_component=10 # Constraint: Min 10% per category
        )
```

Step 3: Registration. Register the new class in `__init__.py`:

```
StrategyRegistry.register(L1VsL2RankStrategy)
```

J.4.2 Method B: Custom Logic. For experiments requiring complex dynamic logic (e.g., temporal consistency or cyclic shifts), researchers can implement a fully custom strategy.

Step 1: Inherit from Base Strategy.

Create a new file in `application/services/pair_generation/` inheriting from `PairGenerationStrategy`.

```
class MyCustomLogicStrategy(PairGenerationStrategy):
    def get_strategy_name(self) -> str:
        return "my_custom_logic"

    def generate_pairs(self, user_vec, n, vec_size) -> list:
        pairs = []
        # Custom logic to generate 'n' pairs based on 'user_vec'
        # ...
        return pairs
```

Step 2: Registration. Register the strategy in `__init__.py` to make it callable via the database configuration.

```
StrategyRegistry.register(MyCustomLogicStrategy)
```

J.5 Localization

The system supports bilingual interfaces (e.g., English and a local language). While dynamic content (subjects, titles) is stored in the database as JSON objects, static UI labels (buttons, error messages) are managed in the application code (`application/translations.py`). Researchers adding new interface elements should add keys to the `TRANSLATIONS` dictionary.