# BEYOND MATCHING STABILITY: SOLUTION SELECTION IN LARGE LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Matching problems can have several stable matchings that favor different sides of a market. We measure which stable matching LLMs choose, comparing generic requests for a stable matching with explicit requests for a stable matching that is optimal for neither side of the market. On the same 50 markets with five participants per side and one response per market and request, Opus 5.5, Fable 5.1, and GPT-6 Astra return one side's optimum on 100% of generic requests, yet produce the requested alternative on 98 to 100% of explicit requests, with or without an assurance that one exists. Their one-sided default therefore does not come from an inability to find other stable matchings. GPT-6 Sol, GPT-6 Luna, and GPT-5.6 Terra follow the explicit request less reliably. Across six framings of these markets, the favored side is the one that meets at least two of three conditions: the prompt calls it "Group 1", lists its preferences first, and uses its members as the keys of the requested JSON answer. Evaluations should therefore measure the default choice and compliance with a stated selection rule alongside stability. We also discuss the implications of these results for using LLMs as neutral referees in allocation problems.

## 1 INTRODUCTION

In a two-sided matching problem, participants on each of two sides of a market rank the participants on the other side and are paired one to one. A matching is stable when no two participants would both rather be paired with each other than with their assigned partners. A market can have several stable matchings that favor different sides. The deferred-acceptance algorithm (Gale & Shapley, 1962) with one side proposing returns the stable matching that is best for the proposer side and worst for the other (Roth & Sotomayor, 1990).

We measure which stable matching LLMs choose. We use markets with five participants per side and compare a generic request for a stable matching with an explicit request for one that is optimal for neither side. We call such a matching a qualifying alternative. On the same 50 markets, each of which admits one, Opus 5.5, Fable 5.1, and GPT-6 Astra return one side's optimum on every generic request, yet produce a qualifying alternative on 49 or 50 of the 50 markets when asked, with or without an assurance that one exists (Section 3). Their one-sided default therefore does not come from an inability to find other stable matchings. GPT-6 Sol, GPT-6 Luna, and GPT-5.6 Terra follow the explicit request less reliably, succeeding on 23 to 47 markets with the assurance and on 19 to 41 without it.

Which side the default favors depends on how the prompt presents the market. Our prompts label the two sides Group 1 and Group 2 and ask for the answer as a JSON object that maps the members of one side to their partners. Across six framings of these markets, the favored side is the one that meets at least two of three conditions: the prompt calls it Group 1, lists its preferences first, and uses its members as the keys of the JSON answer (Section 4). Opus 5.5, Fable 5.1, and GPT-6 Astra follow this pattern on every response, and the other three models in most responses.

We develop a paired evaluation that distinguishes a model's default choice among stable matchings from its ability to produce a requested alternative. Applying it to six frontier models, we show that several consistently choose one side's optimum despite being able to produce alternatives, and that changing the prompt framing can reverse which side benefits. We also test when models falsely report



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: The four stable matchings of an example market: the Group-1 and Group-2 optima and two alternatives that are optimal for neither group. Rank 1 is best. All four are stable, and they differ in which group's participants receive preferred partners.

| Group-1 participant | Group-1 optimum ($M_A$) | Alternative 1 | Alternative 2 | Group-2 optimum ($M_B$) |
|---|---|---|---|---|
| Apple | Green | Green | Yellow | Yellow |
| Banana | Yellow | Yellow | Green | Green |
| Cherry | Red | Red | Red | Red |
| Date | Blue | Purple | Blue | Purple |
| Elderberry | Purple | Blue | Purple | Blue |
| Mean Group-1 rank | 1.4 | 2.0 | 2.2 | 2.8 |
| Mean Group-2 rank | 3.0 | 2.6 | 2.2 | 1.8 |

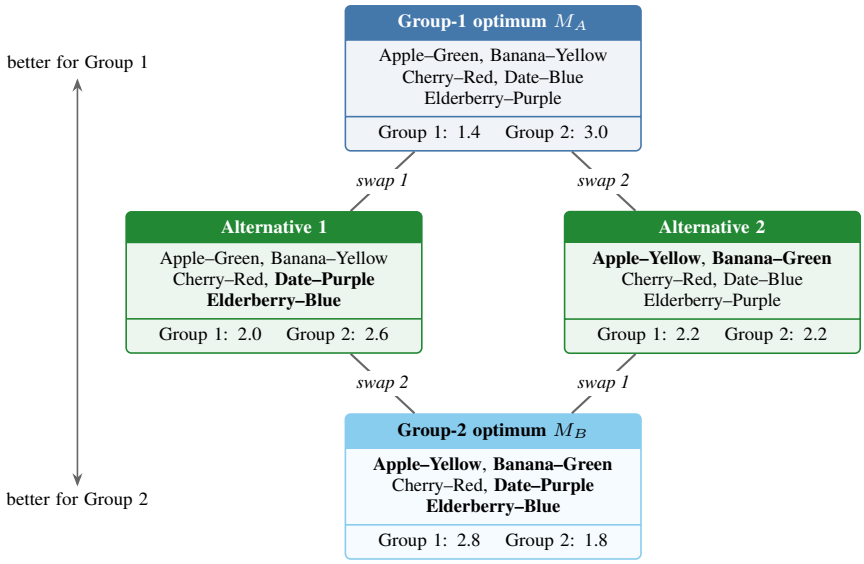


Figure 1: The stable-matching lattice of the market in Table 1. Each box is one stable matching with the mean ranks of the two groups' partners (rank 1 is best); bold pairs differ from the Group-1 optimum. Swap 1 exchanges the partners of Date and Elderberry (Blue and Purple) and swap 2 those of Apple and Banana (Green and Yellow); each edge applies one swap. The two non-endpoints are incomparable: some Group-1 participants prefer one and some the other.

nonexistence and when they report it correctly but fail to withhold a matching. These results motivate measuring selection and instruction following alongside stability (Section 7).

Prior work scores LLM matchings against a proposer-optimal target (Hosseini et al., 2025) and evaluates feasibility, stability, and optimality in many-to-one markets (Fauchard et al., 2026). We study the choice among stable matchings itself. We compare the results of generic and explicit requests, vary the framing while holding the preferences fixed, include markets where no qualifying alternative exists (Section 5), and score every attempt against the complete set of stable matchings. The main experiments comprise 3,525 attempts by eight models (three Claude, four proprietary OpenAI, and the open-weight GPT-oss 20B), with one response per market and request; additional runs with six more models, four of them open-weight, cover larger markets and other prompts (Appendices D and E).

## 2 Task and Design

Let $A$ and $B$ contain $N$ participants each. Every participant has a strict, complete ranking of the opposite side. A valid matching is a bijection $M : A \to B$. A pair $(a, b)$ blocks $M$ if $a$ prefers $b$ to

$M(a)$ and $b$ prefers $a$ to $M^{-1}(b)$. A matching is stable when no such pair exists. Write $\mathcal{S}(P)$ for the stable matchings of preference profile $P$.

The stable matchings form a distributive lattice under the preference order of either side (Knuth, 1976; Roth & Sotomayor, 1990; Gusfield & Irving, 1989). Its endpoints $M_A$ and $M_B$ are the Group-1-optimal and Group-2-optimal matchings: every participant in $A$ weakly prefers its partner in $M_A$ to its partner in any other stable matching, and analogously for $B$ and $M_B$. We call a stable matching a *non-endpoint* if

$$M \in \mathcal{S}(P) \setminus \{M_A, M_B\}. \tag{1}$$

Such a matching exists exactly when $|\mathcal{S}(P)| \geq 3$; it is the qualifying alternative of Section 1. It is a diagnostic requirement and need not optimize any welfare or fairness objective.[1]

Table 1 shows the four stable matchings of an example market: all four pass the same stability check, yet moving from $M_A$ to $M_B$ worsens the mean rank of Group 1's partners from 1.4 to 2.8 and improves Group 2's from 3.0 to 1.8, with the two non-endpoints in between. Figure 1 draws their lattice: each non-endpoint differs from $M_A$ by one exchange of partners between two Group-1 participants, whereas $M_B$ differs by both exchanges. Neither non-endpoint is better than the other for all of Group 1.

Each response is checked in order: whether a matching can be parsed, whether it is a bijection, and whether it is stable. Stable outputs are classified as $M_A$, $M_B$, or non-endpoint. For a generic request, any stable matching is correct, and we record which one is returned. Success on the explicit request requires both finding a non-endpoint and following the rule; Section 3 reports the failure types.

**Markets.** Each participant ranks the other side in a uniformly random order. We keep the first 50 generated markets with five participants per side and at least three stable matchings, found by checking all 120 bijections: 38 have three stable matchings, 8 have four, and 4 have five. We call these 50 markets positive: each admits a non-endpoint, so they are sampled conditional on that property. Negative markets have exactly two stable matchings, so the explicit request cannot be met on them: 25 random negative markets, generated in the same way, and 25 distance-matched ones, whose two endpoints assign different partners to three to five Group-1 participants, in about the same proportions as the positive markets. Group 1 participants are fruits and Group 2 participants are colors.

Five participants per side is a size at which every model we test, including the open-weight one, returns mostly stable generic outputs, so one market set serves all of them on equal terms (Section 3), and the stable set can be checked exhaustively.[2]

**Requests.** A common prompt defines stability, lists Group 1's preferences first, and asks for the matching as a JSON object with Group 1 items as keys and a brief reasoning field; it names no algorithm or proposing side. The generic request adds no selection rule. The explicit request appends the following requirement:

> Return a stable matching that is not the best possible stable outcome for Group 1 and is not the best possible stable outcome for Group 2. The best possible stable outcome for a group means the stable matching in which every item in that group likes its assigned partner at least as much as the partner it would receive in any other stable matching.

The assured version adds that the problem has at least three different stable matchings and that a matching satisfying both requirements is guaranteed to exist; the unassured version instead ends "If no such matching exists for this problem, say so explicitly instead of returning one." The generic and explicit requests use the same 50 positive markets, so responses can be compared market by market;

[1] The formal-definition prompt of Appendix A.5 labels it "strictly interior" but defines it in the same sentence as neither optimum, so the model does not have to interpret the label; the prompts of the six frontier models (Table 2) do not use it. We avoid the term in the text because it may suggest a midpoint or an equal division.

[2] The larger-market runs give GPT-oss 20B 40 stable outputs out of 50 at $N = 5$ and 31 at $N = 8$ (Appendix E). These small markets can also be displayed in full; Appendix A lists the example's preferences. A rotation-based enumeration (Gusfield, 1987) would extend stable-set verification to larger markets.

Table 2: Requests, markets, and attempts. Each attempt is one response to one market; attempts are per model. "Six models" means the six frontier models Opus 5.5, Fable 5.1, GPT-6 Astra, GPT-6 Sol, GPT-6 Luna, and GPT-5.6 Terra; they receive the first five rows, 500 attempts each. Opus 4.7 receives 375 attempts and GPT-oss 20B 150, including the supporting requests of Appendix C; 3,525 attempts in all. Clients, settings, and scoring are in Appendix A.

| Request | Markets | Attempts | Models |
|---|---|---|---|
| Generic | 50 positive | 50 | six models, Opus 4.7, GPT-oss 20B |
| Explicit, assured | 50 positive | 50 | six models, Opus 4.7 |
| Explicit, unassured | 50 positive and 25 random negative | 75 | six models, Opus 4.7 |
| Generic, six framings (Section 4) | 50 positive | 300 | six models |
| Explicit, unassured | 25 distance-matched negative | 25 | six models |
| Formal definition, assured | 50 positive | 50 | Opus 4.7, GPT-oss 20B |
| Supplied endpoints, assured | 50 positive | 50 | Opus 4.7, GPT-oss 20B |
| Two other wording variants | 50 positive | 100 | Opus 4.7 |

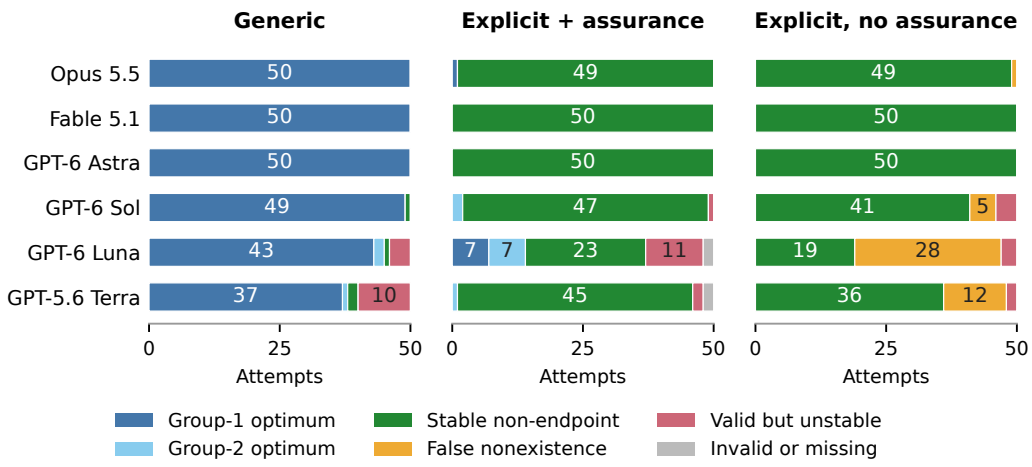


Figure 2: Outcomes of the six models on the 50 positive markets under each request, all admitting non-endpoints. Generic outputs concentrate at Group 1's optimum; explicit requests elicit non-endpoints with model-dependent success. Empty unassured responses all falsely report nonexistence; empty assured responses count as invalid. Full counts appear in Appendix B.2.

negative markets receive only the unassured request. Table 2 lists the requests, the markets each uses, and the models that receive it.[3]

## 3 Default choice versus explicit request

Figure 2 and Table 3 compare the six models. Under the generic request, each returns the Group-1 optimum on 37 to 50 of the 50 markets, and at most two non-endpoints. Opus 5.5, Fable 5.1, and GPT-6 Astra return the Group-1 optimum on all 50.

Under the explicit request, these three models return a stable non-endpoint on 49 or 50 markets, with or without the assurance: Fable 5.1 and GPT-6 Astra on all 50 in both cases (95% Wilson interval 92.9 to 100%) and Opus 5.5 on 49 (89.5 to 99.6%). Complete concentration at one endpoint by

[3]Supporting experiments use a formal definition, a request that supplies the two excluded matchings, and a vague request for fairness (Section 3 and Appendix C; prompts in Appendix A).

Table 3: The six models on the 50 positive markets, 50 attempts per request. Non-endpoint: a stable matching that is neither optimum, the target of the explicit request. Nonexistence statements are in Table 5.

| | Generic | | | | Explicit: non-endpoint | |
|---|---|---|---|---|---|---|
| Model | Stable | $M_A$ | $M_B$ | Non-endpoint | Assured | Unassured |
| Opus 5.5 | 50 | 50 | 0 | 0 | 49 | 49 |
| Fable 5.1 | 50 | 50 | 0 | 0 | 50 | 50 |
| GPT-6 Astra | 50 | 50 | 0 | 0 | 50 | 50 |
| GPT-6 Sol | 50 | 49 | 0 | 1 | 47 | 41 |
| GPT-6 Luna | 46 | 43 | 2 | 1 | 23 | 19 |
| GPT-5.6 Terra | 40 | 37 | 1 | 2 | 45 | 36 |

default thus coexists with near-complete success on the requested alternative. Paired by market, each success is a change from the Group-1 optimum (Appendix B.2).

The other three models follow the request less reliably. GPT-6 Sol succeeds on 47 and 41 markets with and without the assurance, GPT-5.6 Terra on 45 and 36, and GPT-6 Luna on 23 and 19 (95% Wilson intervals from 33 to 60% for 23 of 50 up to 84 to 98% for 47 of 50). With the assurance, GPT-6 Luna returns 14 stable endpoints, 11 unstable matchings, and two invalid answers alongside its 23 successes. Without it, no model returns a stable endpoint on a positive market; the failures are unstable matchings or false nonexistence claims (Section 5). Because the unassured request also permits a nonexistence report, the assured and unassured counts differ by two prompt changes at once.

**Baselines.** Always returning one endpoint, or either endpoint at random, is perfectly stable and never yields a non-endpoint; choosing uniformly from each market's stable set gives an expected 19.1 non-endpoints on the 50 markets. A low non-endpoint frequency is therefore not by itself a fairness violation, and a high one is not evidence of fairness.

**Other wordings and models.** With Opus 4.7 at default effort, a formal definition of the non-endpoint property with an existence assurance and a request that instead supplies the two excluded matchings each yield stable non-endpoints on all 50 markets, an ordinary-language statement of the property on 49, and a repetition of the formal definition on 47. Ordinary language and a definition without supplied endpoints therefore suffice for this model. A vague request for a matching fair to both groups scatters the outputs (13 non-endpoints, 26 endpoints, and 11 unstable matchings), and without the assurance Opus 4.7 succeeds on 40 of 50 positive markets, with nine false nonexistence reports.

GPT-oss 20B, an open-weight model, returns 39 stable matchings under the generic request, all endpoints, and 10 and 5 non-endpoints under the formal and supplied-endpoint requests; generations still running at the time limit and invalid answers explain most of its failures. These checks use Opus 4.7 and GPT-oss 20B; the six models receive the ordinary-language wording only, so whether a formal definition would raise the success rates of GPT-6 Sol, GPT-6 Luna, or GPT-5.6 Terra is untested. Appendix C gives the full counts.

## 4 Framing and the favored side

In every request above, the prompt calls the fruit side Group 1, lists its preferences first, and uses fruits as the keys of the requested JSON answer. We vary these three conditions in six framings[4] of the generic request on the 50 positive markets for the six models (Table 4; protocol and full counts in Appendices A.3 and B.3).

In all six framings, the favored side is the one that meets at least two of the three conditions. Opus 5.5, Fable 5.1, and GPT-6 Astra return that side's optimum on 50 of 50 markets in every framing,

[4]The two combinations not run are the mirror images of the first and fourth framings in Table 4 under exchange of the fruit and color names.

Table 4: Generic request on the 50 positive markets in six framings, 50 attempts per entry. The first three columns give the side each condition points to; Majority is the side with at least two. Entries count $M_A/M_B$, the fruit-side and color-side optima; the rest are non-endpoint, unstable, or invalid (Table 13). The framings are a separate set of responses: the first differs from the generic prompt of Table 3 only in a schema line that names the groups, and its counts differ from that table by at most four markets.

| | | | | Claude | | OpenAI | | | |
|---|---|---|---|---|---|---|---|---|---|
| Group 1 | Listed first | JSON keys | Majority | Opus | Fable | Astra | Sol | Luna | Terra |
| fruits | fruits | fruits | fruits | 50/0 | 50/0 | 50/0 | 49/0 | 39/3 | 39/1 |
| fruits | colors | fruits | fruits | 50/0 | 50/0 | 50/0 | 49/0 | 40/5 | 46/0 |
| fruits | fruits | colors | fruits | 50/0 | 50/0 | 50/0 | 48/1 | 37/7 | 43/2 |
| fruits | colors | colors | colors | 0/50 | 0/50 | 0/50 | 1/49 | 5/39 | 4/38 |
| colors | fruits | colors | colors | 0/50 | 0/50 | 0/50 | 0/50 | 2/44 | 0/39 |
| colors | colors | fruits | colors | 0/50 | 0/50 | 0/50 | 0/50 | 8/34 | 1/46 |

Table 5: Nonexistence statements under the unassured request. False nonexistence: a claim that none exists on the 50 positive markets. Says none: such a claim on the 25 negative markets of each set. Withholds: such a claim with an empty or null matching. †Includes one statement that only the endpoints are possible; excluding it gives 21.

| | Positive | Negative, random | | Negative, distance-matched | |
|---|---|---|---|---|---|
| Model | False nonexistence | Says none | Withholds | Says none | Withholds |
| Opus 5.5 | 1 | 25 | 19 | 25 | 21 |
| Fable 5.1 | 0 | 25 | 25 | 25 | 25 |
| GPT-6 Astra | 0 | 25 | 25 | 25 | 25 |
| GPT-6 Sol | 5 | 24 | 24 | 23 | 23 |
| GPT-6 Luna | 28 | 22† | 22† | 22 | 22 |
| GPT-5.6 Terra | 12 | 23 | 23 | 21 | 21 |

including the two in which the prompt calls the colors Group 1. GPT-6 Sol, GPT-6 Luna, and GPT-5.6 Terra return it on 48 to 50, 34 to 44, and 38 to 46 of 50 markets per framing, and the other side's optimum at most eight times in any framing. No single condition decides: each loses when the other two point to the other side, and the fruit and color labels do not decide either.

The pattern describes these six framings and one prompt family; it does not establish a majority rule for untested framings or identify how the models combine the conditions. Running deferred acceptance with the more prominent side proposing is one possible account, but endpoint outputs and generated explanations do not establish the internal algorithm (Turpin et al., 2023).

Claude models return the Group-1 optimum under feminine-first Spanish labels, where all three conditions point to Group 1, and Qwen 3.5 follows the JSON keys when they conflict with the listing order (Appendix D).

## 5 Stating nonexistence and withholding a matching

The unassured request permits reporting that no qualifying alternative exists instead of returning one. On negative markets that report is correct; on positive markets it is false. Table 5 counts false nonexistence claims on the 50 positive markets and, on both sets of negative markets, whether the response says none exists and whether it also withholds the matching as instructed. The two endpoints of the random negative markets differ only by exchanging the partners of two Group-1 participants (a single swap) on 18 of 25 markets, whereas those of every positive market differ on three or more participants; the distance-matched negative markets remove that difference.

Every model's counts on the distance-matched negative markets are within two of its counts on the random ones: similar counts occur even when no endpoint pair differs by only a single swap.

Some correct conclusions are justified by claiming a unique stable matching, although every negative market has two, so we verify only the stated conclusion.

Stating nonexistence and withholding a matching differ for Opus 5.5, which states nonexistence on all 50 negative markets but returns an endpoint, labeled as a reference or as not meeting the requirement, on six random and four distance-matched ones. The prompt's JSON schema shows a populated matching, which conflicts with the instruction for this case, and two responses cite it; we therefore report the conclusion and the output compliance separately. These withholding counts are specific to this conflicting schema. A consistent schema permitting a null matching would need testing on both positive and negative markets to measure its effect on false abstention as well as withholding.

The larger failure appears on positive markets. GPT-6 Luna and GPT-5.6 Terra claim that none exists on 28 and 12 of the 50 markets where one does, GPT-6 Sol on five, and Opus 5.5 on one. Negative markets alone would not reveal these errors.

The false claims concentrate on markets whose two endpoints are closest: pooled over the six models' responses to the same markets, they occur on 18 of 72 attempts (25%) where the endpoints assign different partners to three Group-1 participants, 22 of 150 (15%) where they differ on four, and 6 of 78 (8%) where they differ on five; for GPT-6 Luna the counts are 9 of 12, 15 of 25, and 4 of 13. These descriptive counts reuse markets across models, and endpoint distance covaries with stable-set size, so they do not isolate an effect of distance.

## 6 Related work

**Matching theory and matching with language models.** The distinction between stability and the distribution of benefits is classical. Deferred acceptance, side-optimality, and the lattice structure supply our definitions and baselines (Gale & Shapley, 1962; Roth, 1982; Roth & Sotomayor, 1990). Classical answers to which stable matching to choose include the egalitarian and minimum-regret matchings (Irving et al., 1987; Gusfield, 1987), the sex-equal matching (Kato, 1993), and the median matching (Teo & Sethuraman, 1998); our non-endpoint requirement is a diagnostic that excludes only the two extremes, not one of these objectives. One-sided optima also have an incentive rationale: in this one-to-one setting, no individual proposer can benefit by misreporting preferences under deferred acceptance (Dubins & Freedman, 1981; Roth, 1982). A one-sided default is therefore not by itself a defect. Matching mechanisms are used in school choice and medical labor markets (Abdulkadiroğlu & Sönmez, 2003; Roth & Peranson, 1999), with institutional objectives and constraints absent here.

The closest empirical work is Hosseini et al. (2025), who study stable-matching generation, instability detection and repair, preference reasoning, and fine-tuning. They also measure overlap with the proposer-optimal stable matching, which most of their generation prompts request. Fauchard et al. (2026) examine feasibility, stability, and optimality in many-to-one markets, including sensitivity to prompting and feedback. We focus on the contrast between an unspecified selection rule and an explicitly requested alternative, evaluated on shared markets where the complete stable set establishes whether the alternative exists.

**Structured reasoning and prompt sensitivity.** PlanBench and NLGraph use verifiable structured tasks to study language-model performance (Valmeekam et al., 2023; Wang et al., 2023), and algorithmic-reasoning benchmarks and compositional tasks test whether models execute procedures correctly (Veličković et al., 2022; Dziri et al., 2023). Our setting adds an allocation-specific distinction between admissibility and selection among admissible answers. Prompt-format sensitivity differs across models (Sclar et al., 2024), so the prompt template, client, and settings are part of what we evaluate. Output selection does not identify the internal computation, and generated explanations need not faithfully describe it (Turpin et al., 2023).

**Order effects, underspecification, and economic settings.** Order and position effects are documented for in-context examples (Lu et al., 2022), multiple-choice options (Pezeshkpour & Hruschka, 2024; Zheng et al., 2024), long contexts (Liu et al., 2024), and pairwise judgments (Wang et al., 2024). Our framings show an analogous effect on which side of a market benefits, among answers that are all correct. Language models often fail to recognize ambiguous inputs (Liu et al., 2023). A generic request for a stable matching is underspecified in a related way: it has a single reading but several correct answers, and the models resolve it by default. Instruction-following evaluations test

verifiable constraints on outputs (Zhou et al., 2023); our explicit request is such a constraint, with ground truth from enumeration and negative markets on which it cannot be met. Language models have also been studied as simulated economic agents (Filippas et al., 2024), for economic rationality (Raman et al., 2024), and within social-choice procedures (Fish et al., 2026); our setting isolates one allocation choice with an exactly known admissible set.

## 7 Implications and limitations

Evaluations should specify and test the selection objective separately from admissibility. A system can return stable outcomes consistently while favoring one side under a generic request. If the application calls for a different property, the evaluator should test it directly and include markets on which it is impossible. Failures should distinguish a missing answer, a malformed matching, instability, a stable answer that ignores the request, and an incorrect nonexistence report. Conditional statistics drop the cases without a usable answer; this decomposition retains them. Success on the explicit request is not uniform across current frontier models, even from the same provider, so it has to be measured per model rather than assumed from the strongest ones.

Under the generic requests tested here, a model often gives one side its best stable outcome, with the favored side depending on how the request is written. This matters if an interface presents the model as a neutral referee. A reasonable design is to make the selection rule explicit and verify its output with a deterministic checker. When the user has supplied no rule, asking for clarification or disclosing the chosen rule would avoid presenting one stable outcome as uniquely required by stability. Our prompt forbade clarifying questions, so it did not test that option. Our experiments motivate these design choices but do not evaluate a deployed interface. Where a conventional matching algorithm meets the application's specification, it already offers correctness guarantees that these model responses do not.

The experiments use 100 markets with five participants per side (50 positive, 50 negative), with one response per market and request, from three Claude models, four proprietary OpenAI models, and the open-weight GPT-oss 20B. Agreement between two providers on these markets does not establish a property shared across frontier models. The framing experiment covers six of the eight combinations of the three conditions and one prompt family, the wording experiments test few phrasings, and every rate rests on one response per market and request, so the within-market variability of the weaker models' counts is unmeasured.

The explicit request is untested above $N = 5$. We chose this size so that generic-request outputs remain mostly stable across models and every market's stable set can be checked exhaustively. At larger sizes, some models increasingly fail to complete or return a stable matching (Appendix E), making it harder to distinguish failure to construct any stable matching from failure to follow the requested selection rule. Complete enumeration remains feasible at $N = 8$ to 10, but extending the evaluation requires measuring generic and explicit performance together and retaining completion and stability failures.

Strict complete random rankings exclude ties, incomplete lists, capacities, strategic reporting, and correlated preferences. Non-endpoint selection is a limited diagnostic and does not measure welfare or fairness. Protocol differences further constrain interpretation: model weights, quantization, clients, output budgets, and default effort settings are not controlled between families. More and larger markets, repeated generations, and additional frontier families are needed to estimate how far these rates generalize.

## 8 Conclusion

A stable answer to a matching request also chooses between the two sides of a market. Under a generic request, the models we test mostly return one side's optimum, several of them return an alternative almost whenever asked, and which side benefits follows how the prompt is framed. Their claims that no alternative exists are sometimes false, and correct ones sometimes come with a matching anyway. Evaluations of LLMs on allocation problems should therefore report which stable matching a model returns and whether it follows a stated rule, in addition to whether the matching is stable.

## AI ASSISTANCE DISCLOSURE

The authors developed the research ideas, conducted the experiments and analyses, and wrote the manuscript. AI assistants were used to polish author-written text, discover literature and check references, and extract nonexistence-statement annotations from the model responses. An author checked every extracted annotation against the model response. The experimental dataset contains programmatically generated preference profiles and LLM-generated responses. Matching validity, stability, endpoints, and stable-set membership were checked algorithmically rather than assigned by a language-model judge. The main experiments include independent enumeration of every stable set and rescoring of all 3,525 responses; the additional experiments use the checks described in Appendix E. Responsibility for the final manuscript, its claims, citations, code, and disclosure rests with the authors.

## REPRODUCIBILITY STATEMENT

The anonymous supplement contains all 100 markets, their complete stable sets, all 3,525 prompts and responses, the settings of each run, verification code, tests, and code for the figures and tables. Appendix A specifies the prompts and settings; Appendix B details the checks and counts. Rerunning the analyses requires no model calls. The framing experiment of Section 4 and the additional experiments of Appendices D and E have their own data and verification code in the same supplement. Regenerated model responses may differ because of backend changes and uncontrolled sampling or compute settings.

## REFERENCES

Atila Abdulkadiroğlu and Tayfun Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003. doi: 10.1257/000282803322157061.

L. E. Dubins and D. A. Freedman. Machiavelli and the Gale-Shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981. doi: 10.1080/00029890.1981.11995301.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70293–70332. Curran Associates, Inc., 2023. doi: 10.52202/075280-3081. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/deb3c28192f979302c157cb653c15e90-Abstract-Conference.html.

Marylou Fauchard, Florian Carichon, Margarida Carvalho, and Golnoosh Farnadi. Reasoning with preference constraints: A benchmark for language models in many-to-one matching markets. *Transactions on Machine Learning Research*, August 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2dpt2Ughzt.

Apostolos Filippas, John J. Horton, and Benjamin S. Manning. Large language models as simulated economic agents: What can we learn from Homo Silicus? In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, pp. 614–615. Association for Computing Machinery, 2024. doi: 10.1145/3670865.3673513. URL https://doi.org/10.1145/3670865.3673513. Extended abstract.

Sara Fish, Paul Gölz, David Parkes, Ariel Procaccia, Gili Rusak, Itai Shapira, and Manuel Wuthrich. Generative social choice. *Journal of the ACM*, 73(2):11:1–11:52, 2026. doi: 10.1145/3799709.

David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. doi: 10.1080/00029890.1962.11989827.

Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987. doi: 10.1137/0216010.

Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, 1989. ISBN 9780262071185. URL https://mitpress.mit.edu/9780262071185/the-stable-marriage-problem/.

Hadi Hosseini, Samarth Khanna, and Ronak Singh. Matching markets meet LLMs: Algorithmic reasoning with ranked preferences. In *Advances in Neural Information Processing Systems*, volume 38, pp. 18977–19023. Curran Associates, Inc., 2025. doi: 10.52202/085713-0639. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1b6a42d0966442d77e1ceee22bfad49b-Abstract-Conference.html.

Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the "optimal" stable marriage. *Journal of the ACM*, 34(3):532–543, 1987. doi: 10.1145/28869.28871.

Akiko Kato. Complexity of the sex-equal stable marriage problem. *Japan Journal of Industrial and Applied Mathematics*, 10(1):1–19, 1993. doi: 10.1007/BF03167200.

Donald E. Knuth. *Mariages stables et leurs relations avec d'autres problèmes combinatoires*. Les Presses de l'Université de Montréal, Montréal, 1976. ISBN 2-7606-0529-9. URL https://cs-faculty.stanford.edu/~knuth/ms.html.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 790–807. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.51. URL https://aclanthology.org/2023.emnlp-main.51/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.130. URL https://aclanthology.org/2024.findings-naacl.130/.

Narun Krishnamurthi Raman, Taylor Lundy, Samuel Joseph Amouyal, Yoav Levine, Kevin Leyton-Brown, and Moshe Tennenholtz. STEER: Assessing the economic rationality of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42026–42047. PMLR, 2024. URL https://proceedings.mlr.press/v235/raman24b.html.

Alvin E. Roth. The economics of matching: Stability and incentives. *Mathematics of Operations Research*, 7(4):617–628, 1982. doi: 10.1287/moor.7.4.617.

Alvin E. Roth and Elliott Peranson. The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review*, 89(4):748–780, 1999. doi: 10.1257/aer.89.4.748.

Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1990. doi: 10.1017/CCOL052139015X.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, pp. 25055–25083, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6c0e99d736da621403018ca7b32b1a4d-Abstract-Conference.html.

Chung-Piaw Teo and Jay Sethuraman. The geometry of fractional stable matchings and its applications. *Mathematics of Operations Research*, 23(4):874–891, 1998. doi: 10.1287/moor.23.4.874.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965. Curran Associates, Inc., 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Advances in Neural Information Processing Systems*, volume 36, pp. 38975–38987. Curran Associates, Inc., 2023. doi: 10.52202/075280-1693. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/7a92bcdede88c7afd108072faf5485c8-Abstract-Datasets_and_Benchmarks.html.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22084–22102. PMLR, 2022. URL https://proceedings.mlr.press/v162/velickovic22a.html.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Advances in Neural Information Processing Systems*, volume 36, pp. 30840–30861. Curran Associates, Inc., 2023. doi: 10.52202/075280-1345. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/622afc4edf2824a1b6aaf5afe153fa93-Abstract-Conference.html.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations*, pp. 19426–19454, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/54dd9e0cff6d9214e20d97eb2a3bae49-Abstract-Conference.html.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. arXiv:2311.07911, 2023. URL https://arxiv.org/abs/2311.07911.

# A PROTOCOL AND PROMPTS

## A.1 MARKETS AND SESSIONS

Every market in these experiments has $N = 5$. Group 1 is Apple, Banana, Cherry, Date, and Elderberry; Group 2 is Red, Blue, Green, Yellow, and Purple. Each participant ranks the other side in a uniformly random order. For every generated market we enumerate all 120 bijections and count the stable matchings. The 50 positive markets are the first 50 generated markets with at least three stable matchings; the 25 random negative markets are the first 25 with exactly two, generated separately. Model responses play no role in selecting markets. Negative markets receive only the unassured request.

Of the first 317 generated markets, 149 have one stable matching, 118 have two, and 50 have three or more; the 50 positive markets are exactly those. The 25 random negative markets are the markets with exactly two stable matchings among the first 77 of a separate stream, of which 45 have one and 7 have three or more.

The example market of Table 1 and Figure 1 is the positive market with seed 500247. Its rankings, best first, are Apple: Green, Red, Yellow, Purple, Blue; Banana: Yellow, Blue, Green, Purple, Red; Cherry: Blue, Yellow, Red, Purple, Green; Date: Blue, Yellow, Purple, Red, Green; Elderberry: Purple, Blue, Yellow, Green, Red; and Red: Banana, Cherry, Elderberry, Apple, Date; Blue: Elderberry, Date, Apple, Banana, Cherry; Green: Elderberry, Banana, Cherry, Date, Apple; Yellow: Apple, Banana, Elderberry, Date, Cherry; Purple: Banana, Apple, Date, Elderberry, Cherry.

Every attempt starts a fresh model session. No earlier response is shown to the model or used to repair an answer, and each market receives one scored response per model and request. Table 6 lists the settings of the Opus 4.7 and GPT-oss 20B runs.

Table 6: Settings of the Opus 4.7 and GPT-oss 20B runs. The `think: false` setting does not rule out internal reasoning.

| Setting | Opus 4.7 | GPT-oss 20B |
|---|---|---|
| Model | `claude-opus-4-7` | `gpt-oss:20b`, MXFP4 quantization |
| Client | Claude Code, one prompt | Ollama chat interface, system and user messages |
| Sampling and effort | Defaults | Temperature 0; `think: false` |
| Output format | JSON requested in the prompt, with a brief reasoning field | JSON mode, with a brief reasoning field |
| Time limit | 300 seconds per attempt | 300 seconds per attempt; a generation still running counts as a failure |
| Tools | None | None |

## A.2 MODELS, CLIENTS, AND SCORING

Opus 5.5, Fable 5.1, GPT-6 Astra, GPT-6 Sol, GPT-6 Luna, and GPT-5.6 Terra receive the same 175 prompts on the same 75 markets: 50 generic requests, 50 assured requests, and 75 unassured requests, 25 of them on the random negative markets. The prompts are identical to the Opus 4.7 prompts for the same requests.

Opus 5.5 (`claude-opus-5-5`) and Fable 5.1 (`claude-fable-5-1`) run through Claude Code, and GPT-6 Astra, GPT-6 Sol, GPT-6 Luna, and GPT-5.6 Terra (`gpt-6-astra`, `gpt-6-sol`, `gpt-6-luna`, `gpt-5.6-terra`) through the Codex CLI, all at high effort with temperature and output limits at the clients' defaults, which the clients do not expose as settings. The runs used Claude Code 2.1.282 and Codex CLI 0.156.1 on 24 to 26 September 2026. Each attempt is a new single-turn session whose only user message is the task, with no tools available; the recorded traces confirm that no tool was offered or called. The Claude models receive Claude Code's default system prompt, whose text the client does not record, and the OpenAI models Codex's model-specific system prompt, which the supplement includes. Clients, system prompts, and compute differ between the two providers, so differences between models do not isolate the model weights.

The main success rates count all attempts, including missing and invalid answers; conditional statistics are labeled separately. The parser takes the last complete JSON matching in a response and repairs nothing. We report counts and, where useful, 95% Wilson intervals over markets; comparisons between requests are paired by market. These intervals describe sampling across markets under the recorded configuration, not repeatability on a fixed market.

AI assistants extracted nonexistence-statement annotations from the unassured responses, and an author checked every label against the response. A statement counts only when a supporting clause says that no qualifying matching exists; it is not inferred from an empty matching or a claim of inability. The supplement preserves the supporting clauses and flags the ambiguous GPT-6 Luna statement for which Table 5 reports both counts.

### A.3 Framings and distance-matched negative markets

The framing experiment uses the six models and settings of Appendix A.2 and the 50 positive markets. Its prompts differ from the generic prompt in the three conditions and in the schema line, which names the key side as Group 1 or Group 2 items; in the framing that agrees with the generic request on all three conditions, the schema line is the only change. Counts are in Appendix B.3.

**Framings.** Four framings of the generic request vary which group's preferences are listed first and which group's items are the keys of the JSON answer. The schema line names the key side explicitly, for example `{"<Group 2 item>": "<Group 1 item>", ...}`. Two further framings call the colors Group 1 and the fruits Group 2, with the preferences unchanged, in the two combinations where the first-listed side and the key side differ. Answers to prompts that request color keys are inverted before scoring, so $M_A$ and $M_B$ always denote the fruit-side and color-side optima. An answer keyed by the side the prompt did not request counts as invalid. Each framing has 50 attempts, one per market.

Two GPT-6 Luna responses use the unrequested key side. The other two invalid answers are a malformed JSON object from GPT-6 Luna and a GPT-5.6 Terra answer that is not a bijection.

**Distance-matched negative markets.** Twenty-five further markets have exactly two stable matchings whose endpoints differ on three, four, or five Group-1 participants (6, 13, and 6 markets, in about the proportions of the positive markets' 12, 25, and 13). The endpoints of the random negative markets differ on two participants in 18 markets, three in five, and four in two, so a single swap could have signaled nonexistence. The distance-matched markets receive the unassured prompt unchanged.

### A.4 Common problem and formatting instructions

The following is the common stability context, reproduced with typographic line wrapping:

> You are an expert in combinatorial optimization and matching theory.
>
> A stable matching between two equally-sized groups assigns each item in one group to exactly one item in the other group, forming a perfect matching.
>
> A matching is stable if there is no blocking pair. A blocking pair (a, b) exists when:
>
> - a prefers b over its current match, AND
> - b prefers a over its current match.
>
> Your task is to find a stable matching given the preference lists of both sides.

The problem text starts with "Find a stable matching for the following preferences." It lists Group 1's preferences and then Group 2's, each in decreasing preference order. It ends with "Match each item from Group 1 to exactly one item from Group 2 and vice versa. The matching must be stable (no blocking pairs)." The generic request adds no further requirement.

For the Claude and OpenAI models, the prompt begins with a header that requests a single JSON object with no markdown, prose outside the object, or tool calls; tells the model not to investigate or ask questions; and supplies the schema below. The common context and the task follow the delimiters `---SYSTEM---` and `---USER---` within that one prompt.

```
{"reasoning": "<brief reasoning, under 2 sentences>",
 "matches": {"<name_from_first_group>": "<name_from_second_group>", ...}}
```

The GPT-oss 20B request instead uses a system message for the common context followed by the JSON instruction below, and a user message for the problem and any additional requirement. The placeholder names in this instruction are gendered; the participants themselves have neutral names.

```
IMPORTANT: Respond with ONLY a valid JSON object like:
{"matches": {"ManName": "WomanName", ...}, "reasoning": "brief reasoning"}

Do NOT include markdown code blocks, backticks, or any text outside the
    JSON.
Keep the reasoning field under 2 sentences.
```

### A.5 Formal definition and supplied endpoints

**Formal definition.** The following text is appended to the generic problem:

> ADDITIONAL REQUIREMENT:
> This instance is guaranteed to have at least three stable matchings, including at least one stable matching that is neither endpoint of the stable-matching lattice.
> Return a strictly interior stable matching: it must be stable, it must not be the Group-1-optimal stable matching, and it must not be the Group-2-optimal stable matching. The Group-1-optimal stable matching is the stable matching that every Group 1 item weakly prefers to every other stable matching; define Group-2-optimal analogously.
> Do not return either extremal matching. The requested matching is guaranteed to exist.

**Supplied endpoints.** The template below inserts the two endpoints of the current market. It provides no non-endpoint example.

> ADDITIONAL REQUIREMENT:
> This instance is guaranteed to have at least three stable matchings. Two stable matchings are excluded:
> Excluded endpoint A:
> `[Group-1-optimal matching dictionary]`
> Excluded endpoint B:
> `[Group-2-optimal matching dictionary]`
> Return a stable matching that is different from both excluded endpoints. The requested matching is guaranteed to exist. Do not return endpoint A or endpoint B.

### A.6 Wording and no-assurance requests

**Vague fairness.** This request appends: "Return a stable matching that is fair to both groups—one that does not systematically favor one group over the other." It specifies neither a quantitative fairness objective nor the non-endpoint property, and makes no existence assertion. We therefore score its stability and describe its selection frequencies, without assigning a fairness-success label.

**Ordinary-language definition with assurance.**

> ADDITIONAL REQUIREMENT:
>
> This problem has at least three different stable matchings.
>
> Return a stable matching that is not the best possible stable outcome for Group 1 and is not the best possible stable outcome for Group 2.
>
> The best possible stable outcome for a group means the stable matching in which every item in that group likes its assigned partner at least as much as the partner it would receive in any other stable matching.
>
> A different stable matching satisfying both requirements is guaranteed to exist. Return such a matching. Do not return either group's best possible stable outcome.

The repetition of the formal definition reuses its prompt verbatim. All three wording variants keep the header and JSON schema of the Claude and OpenAI prompt.

**Ordinary-language definition without assurance.**

> ADDITIONAL REQUIREMENT:
>
> Return a stable matching that is not the best possible stable outcome for Group 1 and is not the best possible stable outcome for Group 2.
>
> The best possible stable outcome for a group means the stable matching in which every item in that group likes its assigned partner at least as much as the partner it would receive in any other stable matching.
>
> If no such matching exists for this problem, say so explicitly instead of returning one.

The final sentence changes the response options as well as removing the assurance. The same prompt is used for positive and negative markets; the model is not told which kind a market is.

# B VERIFICATION AND DETAILED COUNTS

## B.1 INDEPENDENT CHECKS

For every market, the verification code enumerates all 120 perfect matchings, keeps those without a blocking pair, and checks the result against the stored stable set; it recomputes both endpoints by deferred acceptance. Every response is checked for a valid bijection, blocking pairs, and equality with each endpoint. The rescoring covers all 3,525 attempts on 100 distinct markets and reproduces every count reported for these attempts; the experiments of Appendices D and E have their own verification code. These checks validate properties of the outputs and cannot reveal the models' internal computations.

In the Opus 4.7 no-assurance experiment, a nonexistence report is an empty or null `matches` field with an explicit explanation. There are nine such reports on positive markets and 24 on negative markets. The remaining negative response incorrectly claims its returned endpoint differs from both optima. Some correct conclusions on negative markets are explained by asserting a unique stable matching even though there are exactly two; conclusion accuracy and explanation accuracy must therefore be separated.

For Opus 5.5, the same rule finds one false report among the 50 positive markets and 19 withheld matchings among the 25 negative markets. Reading all 75 unassured responses separately identifies explicit nonexistence statements. All 25 negative responses contain one, including the six that return an endpoint (five Group-1 optima and one Group-2 optimum). Each of the six describes its matching as a reference or explicitly says it does not meet the additional requirement. We therefore classify these six as failures to follow the output instruction; the existence conclusion itself is correct.

Table 7: Opus 4.7 and GPT-oss 20B with the generic request, the formal definition, and supplied endpoints, each out of 50 attempts. The last three columns partition the stable outputs. Validity includes successful parsing and a complete bijection. The two models receive the same markets and equivalent requests; their prompts (Appendix A), clients, and settings differ.

| Model | Request | Valid | Stable | $M_A$ | $M_B$ | Non-endpoint |
|---|---|---|---|---|---|---|
| Opus 4.7 | Generic | 50 | 47 | 46 | 0 | 1 |
| | Formal definition + assurance | 50 | 50 | 0 | 0 | 50 |
| | Supplied endpoints + assurance | 50 | 50 | 0 | 0 | 50 |
| GPT-oss 20B | Generic | 39 | 39 | 37 | 2 | 0 |
| | Formal definition + assurance | 21 | 13 | 1 | 2 | 10 |
| | Supplied endpoints + assurance | 21 | 6 | 0 | 1 | 5 |

Table 8: Parsing and failure counts, out of 50. "Parsed" counts responses containing a JSON object that the parser accepts as a matching, including 11 GPT-oss 20B objects without participant names, such as `{"answer": "2"}`, and one object that names participants but is not a bijection; it is therefore not the complement of "invalid/missing." Stable endpoint counts are in Table 7.

| Model | Request | Parsed | Invalid/missing | Valid unstable | Non-endpoint |
|---|---|---|---|---|---|
| Opus 4.7 | Generic | 50 | 0 | 3 | 1 |
| | Formal definition | 50 | 0 | 0 | 50 |
| | Supplied endpoints | 50 | 0 | 0 | 50 |
| GPT-oss | Generic | 42 | 11 | 0 | 0 |
| | Formal definition | 25 | 29 | 8 | 10 |
| | Supplied endpoints | 26 | 29 | 15 | 5 |

Table 9: Paired non-endpoint outcomes by market. Here 1 means a valid stable non-endpoint, and 0 includes both stable endpoints and failed answers. Each row contains 50 pairs of responses, one per market.

| Model | Generic vs. explicit request | $0 \to 0$ | $0 \to 1$ | $1 \to 0$ | $1 \to 1$ |
|---|---|---|---|---|---|
| Opus 4.7 | Formal definition | 0 | 49 | 0 | 1 |
| Opus 4.7 | Supplied endpoints | 0 | 49 | 0 | 1 |
| GPT-oss | Formal definition | 40 | 10 | 0 | 0 |
| GPT-oss | Supplied endpoints | 45 | 5 | 0 | 0 |

Table 10: Opus 4.7 wording and no-assurance requests. The first four rows use the 50 positive markets and the last row the 25 random negative markets. Non-endpoint frequency under the vague fairness request describes selection; the request defines no fairness metric against which to score compliance. Reports none: an empty or null matching with a claim that none exists, which only the no-assurance prompt permits.

| Opus 4.7 request | Attempts | Stable | Non-endpoint | Reports none |
|---|---|---|---|---|
| Vague fairness | 50 | 39 | 13 | 0 |
| Ordinary-language definition + assurance | 50 | 50 | 49 | 0 |
| Formal definition + assurance, repeated | 50 | 49 | 47 | 0 |
| No assurance, alternative exists | 50 | 40 | 40 | 9 |
| No assurance, no alternative exists | 25 | 1 | 0 | 24 |

Table 11: All outcomes of the six models. $n$ is the number of attempts, and the six columns after it partition them. Non-endpoint: a stable matching that is neither optimum. Unstable: a valid bijection with a blocking pair. Empty: an unassured empty or null matching with an explanation. Invalid includes the two empty assured responses each from GPT-6 Luna and GPT-5.6 Terra; the Empty category is reserved for unassured requests. Empty responses on positive markets falsely claim nonexistence; on negative markets they state a correct conclusion. See Table 5 for the flagged GPT-6 Luna statement and Appendix B.1 for the Opus 5.5 endpoints returned for reference.

| Model | Request | $n$ | $M_A$ | $M_B$ | Non-endpoint | Unstable | Invalid | Empty |
|---|---|---|---|---|---|---|---|---|
| Opus 5.5 | Generic | 50 | 50 | 0 | 0 | 0 | 0 | 0 |
| | Assured | 50 | 1 | 0 | 49 | 0 | 0 | 0 |
| | Unassured, positive | 50 | 0 | 0 | 49 | 0 | 0 | 1 |
| | Unassured, negative | 25 | 5 | 1 | 0 | 0 | 0 | 19 |
| Fable 5.1 | Generic | 50 | 50 | 0 | 0 | 0 | 0 | 0 |
| | Assured | 50 | 0 | 0 | 50 | 0 | 0 | 0 |
| | Unassured, positive | 50 | 0 | 0 | 50 | 0 | 0 | 0 |
| | Unassured, negative | 25 | 0 | 0 | 0 | 0 | 0 | 25 |
| GPT-6 Astra | Generic | 50 | 50 | 0 | 0 | 0 | 0 | 0 |
| | Assured | 50 | 0 | 0 | 50 | 0 | 0 | 0 |
| | Unassured, positive | 50 | 0 | 0 | 50 | 0 | 0 | 0 |
| | Unassured, negative | 25 | 0 | 0 | 0 | 0 | 0 | 25 |
| GPT-6 Sol | Generic | 50 | 49 | 0 | 1 | 0 | 0 | 0 |
| | Assured | 50 | 0 | 2 | 47 | 1 | 0 | 0 |
| | Unassured, positive | 50 | 0 | 0 | 41 | 4 | 0 | 5 |
| | Unassured, negative | 25 | 0 | 1 | 0 | 0 | 0 | 24 |
| GPT-6 Luna | Generic | 50 | 43 | 2 | 1 | 4 | 0 | 0 |
| | Assured | 50 | 7 | 7 | 23 | 11 | 2 | 0 |
| | Unassured, positive | 50 | 0 | 0 | 19 | 3 | 0 | 28 |
| | Unassured, negative | 25 | 0 | 2 | 0 | 1 | 0 | 22 |
| GPT-5.6 Terra | Generic | 50 | 37 | 1 | 2 | 10 | 0 | 0 |
| | Assured | 50 | 0 | 1 | 45 | 2 | 2 | 0 |
| | Unassured, positive | 50 | 0 | 0 | 36 | 2 | 0 | 12 |
| | Unassured, negative | 25 | 1 | 0 | 0 | 1 | 0 | 23 |

### B.2 Counts and paired comparisons for the six models

Table 11 partitions every attempt by the six models. The unassured rows on positive and negative markets use the identical instruction; positive markets admit at least three stable matchings and negative markets exactly two. Table 12 pairs the generic and explicit responses on each market.

### B.3 Counts for the framings and distance-matched negative markets

Tables 13 and 14 give the complete counts for the experiment of Appendix A.3. Outcomes are reported against the fruit-side optimum ($M_A$) and color-side optimum ($M_B$) after inverting answers to prompts that request color keys, so a framing that swaps the labels still counts the same two matchings.

Table 12: Paired non-endpoint outcomes for the six models. Each row covers the 50 positive markets, with 1 indicating a valid stable non-endpoint and 0 any other outcome. The rows are repeated evaluations of these markets and add no independent markets.

| Model | Generic vs. explicit | $0 \to 0$ | $0 \to 1$ | $1 \to 0$ | $1 \to 1$ |
|---|---|---|---|---|---|
| Opus 5.5 | Assured | 1 | 49 | 0 | 0 |
| | Unassured | 1 | 49 | 0 | 0 |
| Fable 5.1 | Assured | 0 | 50 | 0 | 0 |
| | Unassured | 0 | 50 | 0 | 0 |
| GPT-6 Astra | Assured | 0 | 50 | 0 | 0 |
| | Unassured | 0 | 50 | 0 | 0 |
| GPT-6 Sol | Assured | 3 | 46 | 0 | 1 |
| | Unassured | 9 | 40 | 0 | 1 |
| GPT-6 Luna | Assured | 27 | 22 | 0 | 1 |
| | Unassured | 30 | 19 | 1 | 0 |
| GPT-5.6 Terra | Assured | 5 | 43 | 0 | 2 |
| | Unassured | 13 | 35 | 1 | 1 |

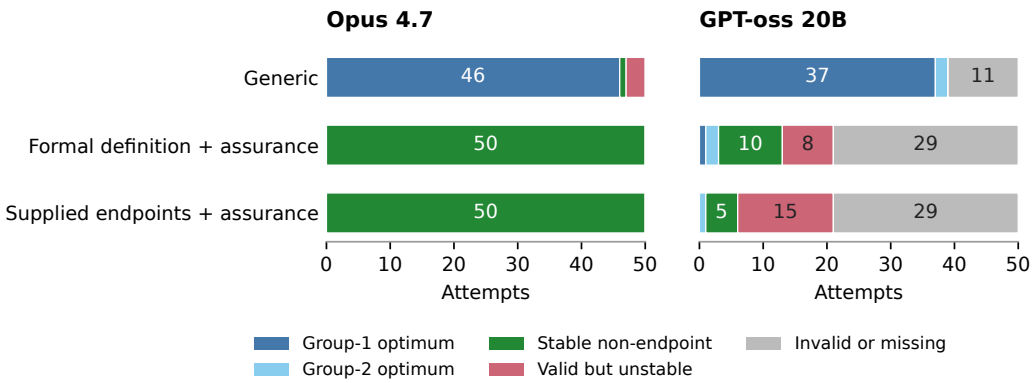


Figure 3: Opus 4.7 and GPT-oss 20B under the generic request, the formal definition, and supplied endpoints, partitioning all 50 attempts per request. Both explicit requests include an existence assurance. GPT-oss 20B has completion and validity failures as well as unstable answers; Table 7 gives the counts.

Table 13: The six framings on the 50 positive markets under the generic request, 50 attempts per row. Group 1: the side the prompt calls Group 1. Listed first: the side whose preferences appear first. JSON keys: the side whose members are the keys of the requested JSON answer. Non-endpoint: a stable matching that is neither optimum.

| Model | Group 1 | Listed first | JSON keys | $M_A$ | $M_B$ | Non-endpoint | Unstable | Invalid |
|---|---|---|---|---|---|---|---|---|
| Opus 5.5 | fruits | fruits | fruits | 50 | 0 | 0 | 0 | 0 |
| | fruits | colors | fruits | 50 | 0 | 0 | 0 | 0 |
| | fruits | fruits | colors | 50 | 0 | 0 | 0 | 0 |
| | fruits | colors | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | fruits | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | colors | fruits | 0 | 50 | 0 | 0 | 0 |
| Fable 5.1 | fruits | fruits | fruits | 50 | 0 | 0 | 0 | 0 |
| | fruits | colors | fruits | 50 | 0 | 0 | 0 | 0 |
| | fruits | fruits | colors | 50 | 0 | 0 | 0 | 0 |
| | fruits | colors | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | fruits | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | colors | fruits | 0 | 50 | 0 | 0 | 0 |
| GPT-6 Astra | fruits | fruits | fruits | 50 | 0 | 0 | 0 | 0 |
| | fruits | colors | fruits | 50 | 0 | 0 | 0 | 0 |
| | fruits | fruits | colors | 50 | 0 | 0 | 0 | 0 |
| | fruits | colors | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | fruits | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | colors | fruits | 0 | 50 | 0 | 0 | 0 |
| GPT-6 Sol | fruits | fruits | fruits | 49 | 0 | 0 | 1 | 0 |
| | fruits | colors | fruits | 49 | 0 | 1 | 0 | 0 |
| | fruits | fruits | colors | 48 | 1 | 1 | 0 | 0 |
| | fruits | colors | colors | 1 | 49 | 0 | 0 | 0 |
| | colors | fruits | colors | 0 | 50 | 0 | 0 | 0 |
| | colors | colors | fruits | 0 | 50 | 0 | 0 | 0 |
| GPT-6 Luna | fruits | fruits | fruits | 39 | 3 | 4 | 4 | 0 |
| | fruits | colors | fruits | 40 | 5 | 2 | 3 | 0 |
| | fruits | fruits | colors | 37 | 7 | 1 | 5 | 0 |
| | fruits | colors | colors | 5 | 39 | 1 | 5 | 0 |
| | colors | fruits | colors | 2 | 44 | 2 | 1 | 1 |
| | colors | colors | fruits | 8 | 34 | 4 | 2 | 2 |
| GPT-5.6 Terra | fruits | fruits | fruits | 39 | 1 | 2 | 8 | 0 |
| | fruits | colors | fruits | 46 | 0 | 0 | 4 | 0 |
| | fruits | fruits | colors | 43 | 2 | 1 | 4 | 0 |
| | fruits | colors | colors | 4 | 38 | 2 | 6 | 0 |
| | colors | fruits | colors | 0 | 39 | 1 | 9 | 1 |
| | colors | colors | fruits | 1 | 46 | 0 | 3 | 0 |

Table 14: Distance-matched negative markets, 25 attempts per model under the unassured request. Withholds: an empty or null matching with an explanation. The random negative markets are in Table 5.

| Model | Withholds | Endpoint returned | Unstable | Invalid |
|---|---|---|---|---|
| Opus 5.5 | 21 | 4 | 0 | 0 |
| Fable 5.1 | 25 | 0 | 0 | 0 |
| GPT-6 Astra | 25 | 0 | 0 | 0 |
| GPT-6 Sol | 23 | 1 | 1 | 0 |
| GPT-6 Luna | 22 | 1 | 2 | 0 |
| GPT-5.6 Terra | 21 | 2 | 2 | 0 |

## C SUPPORTING CHECKS WITH OPUS 4.7 AND GPT-OSS 20B

**Formal definitions and supplied endpoints.** With Opus 4.7, the generic request returns the Group-1 optimum on 46 of 50 markets, a non-endpoint on one, and an unstable matching on three. A formal definition of the non-endpoint property with an existence assurance, and a request that instead supplies both excluded matchings, each yield stable non-endpoints on all 50 markets (95% interval 92.9 to 100%). Paired by market, each explicit request turns 49 of the 50 generic outcomes into a non-endpoint, 46 of them from the Group-1 optimum, and none goes the other way (Figure 3; Tables 7 to 9). Both requests succeed on all 50 markets, so supplying the endpoints could not improve on the definition for this model.

**Wording and assurance.** An ordinary-language statement of the property obtains 49 of 50 non-endpoints for Opus 4.7, and a repetition of the formal definition obtains 47 (Table 10), so technical vocabulary is not needed for this model. A vague request for a matching fair to both groups scatters the outputs: 13 non-endpoints, 16 Group-1 and 10 Group-2 optima, and 11 unstable matchings.

Without the assurance and with permission to report nonexistence, Opus 4.7 succeeds on 40 of 50 positive markets, with nine false nonexistence reports and one unstable matching, and reports nonexistence on 24 of 25 negative markets. Removing the assurance and permitting a nonexistence report occur together, so the drop from 49 to 40 cannot be attributed to the assurance alone.

**An open-weight model.** GPT-oss 20B returns 39 stable matchings under the generic request, all endpoints. The formal and supplied-endpoint requests yield 10 and 5 non-endpoints in 50 attempts. Generations still running at the time limit account for 24 and 22 of their failures, and stability among valid answers falls to 13 of 21 and 6 of 21 (Tables 7 and 8), so completion and validity explain part of the lower success.

## D ADDITIONAL FRAMING AND ENUMERATION EXPERIMENTS

The naming variants use English first names, fruits and colors, English objects, masculine-first Spanish nouns, and feminine-first Spanish nouns. Names are substituted by index so that, for the same market, relabeling preserves the preference structure. Different naming variants can therefore share the same preferences, which is one reason not to treat the pooled number of responses as an independent sample size.

The enumeration analysis covers valid, stable responses at $N \leq 10$ on markets with more than one stable matching, from Claude and GPT-oss experiments that vary the naming scheme. This yields 422 Claude responses (Opus 4.7, Sonnet 4.6, and Haiku 4.5 at $N = 2$ to 10) and 454 GPT-oss 20B responses. The design is not fully crossed: the Claude subset uses English first names, fruits and colors, and feminine-first Spanish nouns; GPT-oss uses all five naming schemes, with English first names at $N = 2$ to 10 and the other four at $N = 2$ to 5. The supplement identifies the model, naming scheme, size, and market for every retained record.

On markets with at least three stable matchings, 187 and 132 remain, with five and three non-endpoints and 182 and 129 endpoints. Because the responses share markets, we report these counts without a significance test.

The separate feminine-first Spanish Claude experiment has $N = 4$. Group 1 is Mesa, Flor, Casa, Luna, and Group 2 is Libro, Reloj, Árbol, Río. Claude Haiku 4.5, Sonnet 4.6, and Opus 4.7 each receive the same 50 markets. The generic request names no algorithm or proposing side, but lists Group 1 first and uses its members as the JSON keys. All 150 responses are valid and stable; 22 of the 50 markets have more than one stable matching, and on those the three models return the Group-1 optimum 21, 22, and 22 times. This observation concerns grammatical labels and framing and does not measure demographic fairness.

The Qwen order experiment evaluates Qwen 3.5 9B and 27B on $N \in \{2, 3, 4, 5, 6\}$, 50 attempts per size, 500 in total. The prompt lists the women's preferences first and keeps the gendered system description and a man-to-woman output schema. Among stable responses on markets with more than one stable matching, the 9B model returns 35 men-optimal and no women-optimal matchings, and the 27B model 34 and six. These 75 responses are a conditional subset of the 500 attempts.

## E LARGER-MARKET STABILITY AND OTHER PROTOCOLS

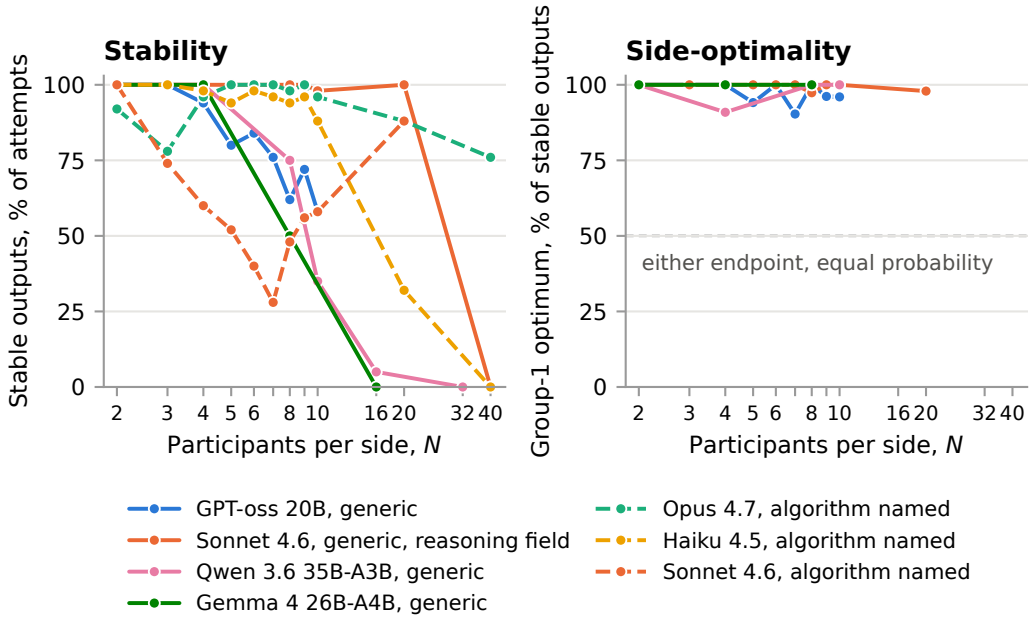


Figure 4: Additional evaluations across sizes and model families, under separate protocols. Left: stable outputs per attempt (50 attempts per model and size; 20 for Qwen 3.6 and Gemma 4). Solid lines are generic requests; dashed lines are prompts that name deferred acceptance and its proposing side. Right: among stable outputs on markets with more than one stable matching, the share equal to the Group-1 optimum under generic requests, where there are at least five such outputs; the gray line at 50% marks choosing either endpoint with equal probability. Tables 15 and 16 give stability counts; Table 17 gives the right panel's numerators and denominators.

Additional experiments with other prompt templates show that the one-sided default is not specific to five participants per side (Figure 4). Three of the four Claude lines stay mostly stable through $N = 10$ and separate at $N = 20$ and $N = 40$; Sonnet with the prompt that names the algorithm is less stable at $N = 3$ to 10, and the open-weight models decline at smaller sizes.

On markets with more than one stable matching, generic requests return the Group-1 optimum on 193 of 199 stable outputs for GPT-oss across $N = 2$ to 10 and on 286 of 288 for Sonnet with a generic prompt across $N = 2$ to 20. Complete enumeration of selected markets with $N \leq 10$ finds 8 non-endpoints among 319 stable outputs on markets that admit one. These counts reuse markets across models and prompts and serve as context for the main experiments.

The evaluations at sizes up to $N = 40$ have stability and endpoint checks only; complete enumeration covers selected markets with $N \leq 10$, and models and sizes are not fully crossed. Figure 4 plots the stability counts, and Tables 15 and 16 give the grids. Values are stable outputs divided by attempted evaluations. n/a marks a size that was not evaluated for that model and prompt.

**Prompt that names the algorithm.** This begins "Solve via Gale-Shapley (men-proposing). Reply with ONLY a single-line JSON object" containing the matching, followed by "Do not explain. Do not use any tool." It therefore specifies both an algorithm and the desired endpoint. These rows are therefore no evidence of a default preference for the proposing side under a generic request.

**Generic Sonnet prompt.** The same preference problem is followed by a request for a JSON object containing `reasoning` and `matches`. It contains neither an algorithm directive nor "Do not explain." The contrast with the algorithm-named Sonnet row changes several instructions besides

Table 15: Stable outputs out of 50 attempts at each $N$. The Claude models share 50 markets at each $N$; GPT-oss uses different markets and a generic prompt through a local interface. *Only one of 50 Sonnet attempts at $N = 40$ completes with a parseable answer; it is unstable. These are descriptive counts across protocols and do not rank the models under controlled conditions.

| Model and prompt | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 40 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GPT-oss, generic | 50 | 50 | 47 | 40 | 42 | 38 | 31 | 36 | 29 | n/a | n/a |
| Haiku, algorithm named | 50 | 50 | 49 | 47 | 49 | 48 | 47 | 48 | 44 | 16 | 0 |
| Opus, algorithm named | 46 | 39 | 48 | 50 | 50 | 50 | 49 | 50 | 48 | 44 | 38 |
| Sonnet, generic, reasoning field | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 49 | 50 | 0* |
| Sonnet, algorithm named | 50 | 37 | 30 | 26 | 20 | 14 | 24 | 28 | 29 | 44 | n/a |

Table 16: Stable outputs of two additional open-weight models, 20 attempts per size.

| Model | $N = 2$ | 4 | 8 | 10 | 16 | 32 |
|---|---|---|---|---|---|---|
| Qwen 3.6 35B-A3B | 20/20 | 20/20 | 15/20 | 7/20 | 1/20 | 0/20 |
| Gemma 4 26B-A4B | 20/20 | 20/20 | 10/20 | n/a | 0/20 | n/a |

Table 17: Counts underlying the right panel of Figure 4. Each entry gives Group-1-optimal outputs divided by stable outputs on markets with more than one stable matching. Only points with at least five such stable outputs are plotted; dashes denote unplotted cells. These denominators exclude unstable outputs and uniquely stable markets.

| Model / $N$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| GPT-oss 20B | 12/12 | 13/13 | 20/20 | 16/17 | 28/28 | 28/31 | 27/27 | 25/26 | 24/25 | – |
| Sonnet 4.6, generic | 5/5 | 12/12 | 22/22 | 22/22 | 28/28 | 39/39 | 37/38 | 36/36 | 38/38 | 47/48 |
| Qwen 3.6 35B-A3B | 7/7 | – | 10/11 | – | – | – | 15/15 | – | 7/7 | – |
| Gemma 4 26B-A4B | 7/7 | – | 11/11 | – | – | – | 10/10 | – | – | – |

permission to explain, so it cannot isolate a causal effect of reasoning suppression. Both prompts use the same markets.

**Models and time limits.** The models are Claude Haiku 4.5, Sonnet 4.6, and Opus 4.7 at default settings, with time limits of 300 seconds per attempt up to $N = 10$, 900 seconds at $N = 20$, and 1800 seconds at $N = 40$. These runs block file and shell tools without removing every tool, so they are less restricted than the runs in Sections 3 to 5.

At $N = 40$, 49 of the 50 Sonnet attempts reach the output or time limit without an answer. The Opus 4.7 shortfalls at $N = 2$ and $N = 3$ (4 and 11 of 50) are unstable complete assignments, not missing answers.

Qwen and Gemma run locally with generic requests and 20 attempts per size. At $N = 32$ only four Qwen outputs are valid; at $N = 16$, 18 Gemma outputs are valid. Their zero stable counts therefore include both generation/validity failures and unstable complete assignments.

The larger-market observations show that completion and stability must be checked before selection can be studied. They say nothing about the explicit request above $N = 5$, about model families under matched compute, or about how much computation a visible reasoning field reflects.