ISLANDS OF STABILITY: TRACTABLE LATTICE STRUCTURES FOR NO-REGRET LEARNING OF SUPER-STABLE MATCHINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

The stable matching problem has long served as a foundational model for two-sided market design, yet classical solutions assume that complete preference orderings are known in advance. We study an online version of this setting, motivated by the challenges of learning a stable matching in a repeated game—specifically as it relates to super stability and efficient learning under partial information. Our model is designed to operate directly on pairwise comparative feedback, making it practical in settings where agents observe only comparative outcomes. This forms a more natural representation of preference elicitation, obviating the need to assign ordinal scores to outcomes (common in previous work). Operating on direct pairwise feedback, we first present a learning algorithm, P2ETG, with provable sublinear regret guarantees with respect to achieving any stable matching. Next, as a crucial extension, we present PrefLID, an algorithm which exploits the lattice structure of stable matchings and the concept of super-stability, connecting common stable matchings among different refinements of a bipartite partial order into connected graphs, called islands. We demonstrate that these islands reveal structural properties of the preference-refinement space that can be exploited for more efficient super-stable matching search over partial orders. This allows PrefLID to sample more efficiently, making only the pairwise comparisons needed to resolve the problem structure within a single island. We provide a tractable means to construct this island structure, enabling its use in an efficient online setting and demonstrating that it achieves no-regret performance while avoiding full preference resolution. We illustrate the applicability of our approach through a natural application to LLM routing, where agents must be matched to tasks in a streaming setting, and further provide corresponding empirical results to corroborate the theoretical properties.

1 INTRODUCTION

How can a two-sided market learn a stable matching from a multitude of potential solutions when neither side knows its own preferences and the only available feedback is the outcome of a noisy comparison between two alternatives? One classical answer, built on the competing-bandits formulation (Liu et al., 2020), couples preference learning across agents through competition for partners; subsequent work has extended the programme to decentralized markets (Liu et al., 2021), two-sided learning (Zhang & Fang, 2024), many-to-one structures (Kong & Li, 2024), and sharper regret analysis (Kong et al., 2024; Li et al., 2025). This line of work, however, assumes each agent observes an *absolute* stochastic reward for every potential partner. In many modern systems — routing queries among large language models, matching users to services, assigning tasks to workers — such a scalar is never observed. What is observed is comparative: this model answered better than that one, this worker was preferred to that one. Noisy pairwise comparisons of this kind are the currency of *duelling bandits* and pure-exploration rank identification (Sui et al., 2018; Busa-Fekete et al., 2014; Chen et al., 2020), and most recently of matching markets with relative feedback (Kong et al., 2026), which identifies the player-optimal stable matching with sample complexity governed by the smallest gap among *all* comparisons needed to recover the full ranking. Across all of these formulations one assumption persists: the learner must resolve *every* agent's complete ranking before a stable matching can be certified.

The assumption is expensive, and it is also unnecessary. Resolving a pairwise comparison to high confidence costs samples inversely proportional to its preference gap, so recovering a complete ranking is bottlenecked by the single hardest comparison in the market — even when that comparison is irrelevant to the matching ultimately chosen. Combinatorial work has long exploited this slack. The theory of stable marriage with incomplete

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.

information defines weak, strong, and *super-stability* for partially specified preferences (Irving, 1994), and possible and certain stability refine partial orderings into strict preferences (Aziz et al., 2020; 2022). Preference elicitation shows that a stable matching can often be determined without eliciting every agent's full ranking (Drummond & Boutilier, 2014), and matchings robust to preference perturbations remain stable across entire families of profiles (Mai & Vazirani, 2018b; Gangam et al., 2026b). The statistical translation is direct: if the observations induce a set of rankings that contains the true ranking with high probability, and a candidate matching is stable under *every* ranking in that set, then the matching is stable with high probability. A purely deterministic check of robust stability thereby becomes a sound statistical stopping rule, and the learning goal shifts from resolving complete rankings to resolving the *sufficient* partial rankings that certify a stable outcome.

This paper develops that observation into an online learning framework for two-sided matching under pairwise uncertainty. The learner maintains the uncertainty explicitly as a partial ordering, resolves comparisons adaptively, and stops as soon as the induced lattice of candidate stable matchings certifies a single outcome — without requiring any agent's full ranking. The framework is naturally suited to AI systems that route queries among multiple large language models, where preferences over models are learned from noisy pairwise comparisons or user context feedback (Hu et al., 2024; Liu & Etesami, 2026). Fully ranking every model for every query type is costly; focusing on the comparisons needed to certify a stable assignment lets the system stop exploring earlier with high-probability guarantees, while ensuring no user context or model has a clear incentive to switch.

Literature Comparison: Table 1 summarizes the closest neighbouring research directions. The final row describes a target research formulation suggested by their intersection, rather than an established result.

Work	Online	Two-Sided Unknown	Relative Feedback	Partial Prefs.	Stable Structure	Primary Objective
Liu et al. (2020)	✓	✗	✗	✗	Limited	Stable Regret
Liu et al. (2021)	✓	✗	✗	✗	Limited	Decentralized Regret
Kong & Li (2023a)	✓	✗	✗	✗	Gale-Shapley	Player-Optimal Regret
Kong et al. (2024)	✓	✗	✗	✗	✓	Improved Stable Regret
Hosseini et al. (2024)	✓	✗	✗	Partial	✓	Stable Identification
Zhang & Fang (2024)	✓	✓	✗	✗	Gale-Shapley	Two-Sided Learning
Kong et al. (2026)	✓	Partial	✓	Confidence-Based	Gale-Shapley	Stable Identification
Drummond & Boutilier (2014)	Elicitation	✓	Queries	✓	Stability-Based	Minimize Elicitation
Aziz et al. (2020)	✗	Uncertain	✗	✓	Super-Stability	Certain Stability
Aziz et al. (2022)	✗	Uncertain	Pairwise Probs.	✓	✗	Certain Stability
Mai & Vazirani (2018a)	✗	Perturbations	✗	Profiles	✓	Robust Stability
Gangam et al. (2026a)	✗	Changing Profiles	✗	Profiles	✓	Robust Stability
Our Formulation	✓	✓	✓	✓	✓	Stability Certification with Selective Learning

Table 1: Schematic comparison of research most closely related to learning stable matchings under pairwise preference uncertainty.

Table 1 highlights that while individual dimensions have been explored in isolation, no existing work in online stable matching literature combines *relative feedback*, explicit *partial preferences*, and *stable structures* to achieve stability certification with selective sampling. Our work fills this gap by developing a framework where pairwise comparisons are resolved adaptively based on their relevance to stability, and corresponding sample-complexity guarantees. Our algorithmic contributions in this work to the online stable matching literature are primarily:

Natural Pairwise Feedback: We introduce an algorithm for online stable matching designed to naturally leverage pairwise feedback dynamics during online interactions. (Sec. 4.1.)

Dynamic Entry and Exit: The proposed architecture seamlessly integrates agents entering and leaving the system dynamically, operating effectively under heterogeneous information constraints. (Sec. 4.3.)

Structural Insights and Sample Complexity: We reveal core structural properties of the stable matching search space, yielding a geometric framework that isolates critical sampling regions to substantially minimize sample complexity in online learning. (Sec. 3.)

2 Preliminaries

Stability & Matching: Let $\mathcal{A}$ and $\mathcal{B}$ be two finite, disjoint sets with $|\mathcal{A}| = N$, $|\mathcal{B}| = K$. Denote the set of all agents by $\mathcal{A} \cup \mathcal{B}$. Each agent $a \in \mathcal{A}$ truly possesses a strict, complete preference relation over the set $\mathcal{B}$, which

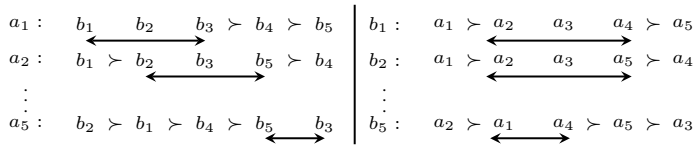


Figure 1: **Bipartite Ranking Topology $\widehat{\succ}_{AB}$:** We represent a bipartite partial ordering. Each agent $a \in \mathcal{A}$ and $b \in \mathcal{B}$ has an ordering over the other side. However, due to uncertainty, some of the preferences are not fully resolved, leaving sections of sequences unordered. For example, from the perspective of a_1, they are indifferent to the ordering of b_1, b_2, b_3, but prefer $b_4 \succ b_5$, and prefer any $b \in \{b_1, b_2, b_3\}$ over b_4 or b_5.

we denote as $\succ_a$. For $b, b' \in \mathcal{B}$, we write $b \succ_a b'$ to indicate that a strictly prefers b to b'. Analogously, each agent $b \in \mathcal{B}$ has a strict, complete preference relation $\succ_b$ over $\mathcal{A}$. Completeness ensures that every agent ranks all members of the opposite set, and strictness precludes ties. We denote $\succ_a$ as a strict ordering for agent $a \in \mathcal{A}$ over the set $\mathcal{B}$, and $\succ_b$ vice versa, for N agents on the A-side and, K agents on the B-side.

Definition 2.1 (Matching)**.** *A **matching** is a bijection $\rho : \mathcal{A} \to \mathcal{B}$ (or $\rho : \mathcal{B} \to \mathcal{A}$) such that* I) *For every $a \in \mathcal{A}$, $\rho(a) \in \mathcal{B}$;* II) *For every $b \in \mathcal{B}$, $\rho(b) \in \mathcal{A}$;* III) *For every $x \in \mathcal{A} \cup \mathcal{B}$, $\rho(\rho(x)) = x$. Equivalently, a matching is a set of N disjoint pairs (a, b) with $a \in \mathcal{A}$ and $b \in \mathcal{B}$ that partitions $\mathcal{A} \cup \mathcal{B}$.*

Remark 2.1. ***Uneven Matching:*** *In practice $N \leq K$ to ensure each agent in $\mathcal{A}$ is matched. Intuitively, a matching assigns each agent in $\mathcal{A}$ to a unique partner in $\mathcal{B}$, and vice versa. If $N = K$ them no agent remains unmatched.*

Stability: The notion of stability captures the absence of mutually beneficial deviations. A pair of agents who are not matched to each other forms a *blocking pair* if they would both prefer to be together over their current assignments.

Definition 2.2 (Blocking Pair)**.** *Given a matching ρ, an unordered pair $(a, b) \in \mathcal{A} \times \mathcal{B}$ with $\rho(a) \neq b$ is a **blocking pair** if: I) $b \succ_a \rho(a)$ (agent a prefers b to his or her current match); and II) $a \succ_b \rho(b)$ (agent b prefers a to his or her current match).*

Definition 2.3 (Stable Matching)**.** *A matching ρ is **stable**, denoted as $\widetilde{\mathcal{M}}$, if there exists no blocking pair. Equivalently, for every $(a, b) \in A \times B$ with $\rho(a) \neq b$, $\neg\big(b \succ_a \rho(a)$ and $a \succ_b \rho(b)\big)$.*

Existence of Stable Matchings: This is the central solution concept in two-sided matching theory. An equivalent formulation is often useful: for all $a \in \mathcal{A}$ and $b \in \mathcal{B}$, if $b \succ_a \rho(a)$ then necessarily $\rho(b) \succ_b a$. In words, if an agent would prefer to be matched with someone else, that someone else must prefer their own current partner. A fundamental result due to (Gale & Shapley, 1962) is that a stable matching always exists. The proof is constructive and given by the Gale-Shapley algorithm, also known as deferred acceptance. In the version where agents in A propose, the Gale-Shapley (GS) algorithm illustrated in Alg. 5 finds at least one stable matching which happens to be the proposer optimal, receiver pessimal stable matching. The algorithm terminates after at most $|\mathcal{A}| \cdot |\mathcal{B}|$ proposals and produces a stable matching[1]. Moreover, it yields the matching that is optimal for the proposing side: each agent in $\mathcal{A}$ receives the best possible partner among all stable matchings, while each agent in $\mathcal{B}$ receives the worst possible partner among all stable matchings.

2.1 Matchings on Partial Orderings

We denote the set of bipartite partial orderings as the *bipartite ranking topology*, $\widehat{\succ}_{AB}$, representing a set of orderings constituting refinements of single bipartite partial ordering, with members indexed by c, such that $\succ_{AB}^{(c)} \in \widehat{\succ}_{AB}$ is a fully-resolved order. Between the strongest and weakest notions of stability under such ties lie several intermediate concepts, each balancing permissiveness with robustness against alternative tie-breakings.

Definition 2.4. ***Super-Stability:*** *A pair (a, b) is a* super-blocking pair *for matching ρ if a and b are unmatched to each other and each weakly prefers the other to their partner. ρ is* super-stable *if it has no super-blocking pairs.*

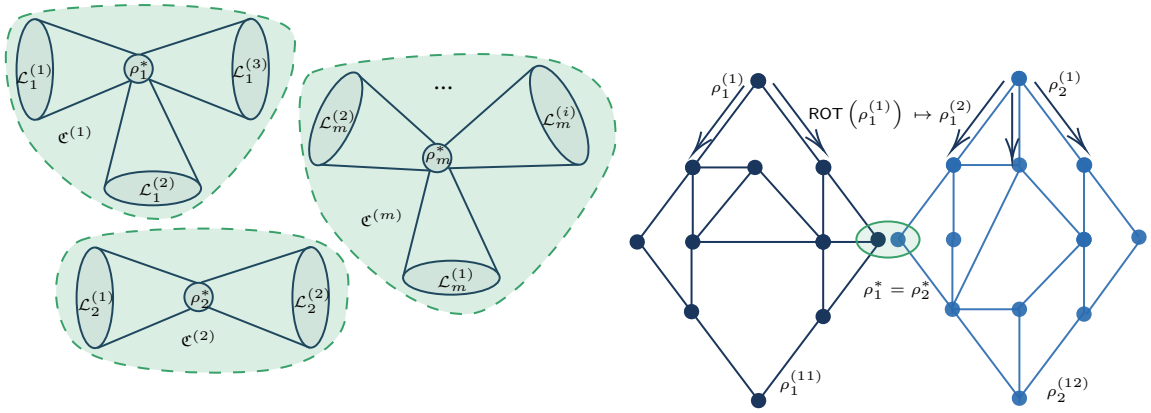
Super-stability (Def. 2.4) is the strongest of these notions: a matching remains stable under every refinement of the ties into strict orders. For any bipartite partially ordered instance $\widehat{\succ}_{AB}$, a super-stable matching ρ^* need not

[1] In our case we may be left with $\max\{N - K, K - N\}$ unmatched agents if $N \neq K$.

exist; however (Irving, 1994) provided a polynomial-time algorithm that either finds such an ρ^* or determines that none exists.

Weak & Strong Stability: In the domain of bipartite partial orderings, finer concepts of stability emerge. Weak stability permits blocking only when both agents strictly gain, ensuring existence via tie-breaking but allowing indifference-based blocks; strong stability forbids any pair where at least one strictly benefits and the other does not lose, yielding a distributive lattice when existent. Together with super-stability, these notions form the strict hierarchy super-stability $\Rightarrow$ strong stability $\Rightarrow$ weak stability. Super-stability (Def. 2.4) guarantees stability under the broadest interpretation of indifference. By contrast, strong stability requires stability for *some* refinement of a partial ordering, while weak stability may fail for refinements that resolve ties unfavourably.

3 Structural Properties & Algorithms



(a) **Illustration of Island Lattice Structure:** Via the blocking function Γ, a set of lattices $\{\mathscr{L}\}$ can be mapped into a set of islands $\mathfrak{C}$, where within each island $\mathfrak{C}^{(m)}$ a common centroidal matching ρ_m^* is shared between the lattices $\mathscr{L}$.

(b) **Lattice of Stable Matchings:** Two lattices share an intersection via matching ρ^*. A lattice is constructed from each starting configuration ρ_i and ρ_j, and via GS we can arrive at another stable matching.

Figure 2: **Construction of Lattice Island Structure $\mathfrak{C}$:** Fundamentally, every lattice island on the lattice island structure $\mathfrak{C}$ is formed by intersection of a common stable matching within the lattices in the lattice set.

With respect to super-stability (Def. 2.4), some restricted preference models can allow for complex stable matching structures. More broadly, super-stability is closely related to lattice structure. Robust stable matchings, which remain stable across nearby instances, are known to form well-behaved structures under certain conditions, with sharp thresholds where this structure breaks down (Cheng & Rosenbaum, 2023; Gangam et al., 2026b; Mai & Vazirani, 2018b).

Definitions and Notation: To combat the combinatorial complexity of exploring the full space of two-sided preference orderings, we introduce a computational budget B_Σ and a family of *partition functions* $\{\Psi : \widehat{\succ}_{AB} \mapsto \widehat{\succ}'_{AB}\}$ that coarsen a bipartite ordering topology into a new element of the same domain. For a single agent a, we write $\Psi_a : \succ_a \mapsto \succ'_a$ for the partition function acting on a's ordering, and symmetrically Ψ_b for b. The integer N_Ψ denotes the number of partitions induced by Ψ on a given ordering.

For each agent we let $\mathcal{P}^{(n)}$ denote the collection of partitions belonging to agent $n \in [N]$ on the A-side (and $\mathcal{P}^{(k)}$ for $k \in [K]$ on the B-side); the size of the i-th partition is $|\mathcal{P}^{(i)}|$. We write $\succ_a$ for the total ordering of an A-side agent a over the K B-side agents, and $\succ_b$ for the symmetric B-side ordering. The *partial* ordering of a over the B-side is denoted $\widehat{\succ}_a$, and similarly $\widehat{\succ}_b$; the hat denotes that only some pairwise relations have been established with high confidence, so the ordering is a refinement-in-progress of the true $\succ_a$. The completely uninformed ordering — a single block containing all opposing agents, with no relations resolved — is denoted $\breve{\succ}_i$; this is the starting state of any learning agent and corresponds to $C_\Psi = 0$ cuts.

Preference Configurations: A *preference configuration* $\succ_{ab}^{(c)}$ for $c \in 1 \dots |\widehat{\succ}_{AB}|$ is a fully-resolved two-sided ordering compatible with the current partial order $\widehat{\succ}_{AB}$. Equivalently, the set of configurations is precisely the set of refinements of the partial order $\widehat{\succ}_{AB}$. Each agent n carries a set of N_Ψ partitions $\mathcal{P}^{(n)}$; the number of ways to fully order a single partition of size $|\mathcal{P}^{(n)}|$ is $|\mathcal{P}^{(n)}|!$, so the total number of configurations is,

$$|\widehat{\succ}_{AB}| = \underbrace{\left(\prod_{n=1}^{N}\prod_{i=1}^{|\Psi_n(B)|}|\mathcal{P}_i^{(n)}(B)|!\right)}_{\text{A side}}\underbrace{\left(\prod_{k=1}^{K}\prod_{j=1}^{|\Psi_k(A)|}|\mathcal{P}_j^{(k)}(A)|!\right)}_{\text{B side}}, \tag{1}$$

where $\Psi_n(B)$ denotes the partition of the members on the B side from the perspective of member n, i.e., $\Psi_n(B) := \{\mathcal{P}_1^{(n)}(B), \dots, \mathcal{P}_n^{(|\Psi_n|)}(B)\}$, such that $\bigcup_i \mathcal{P}_i^{(n)}(B) = K$. Similarly, $\Psi_k(A)$ denotes the partition of the members on the A side from the perspective of member k.

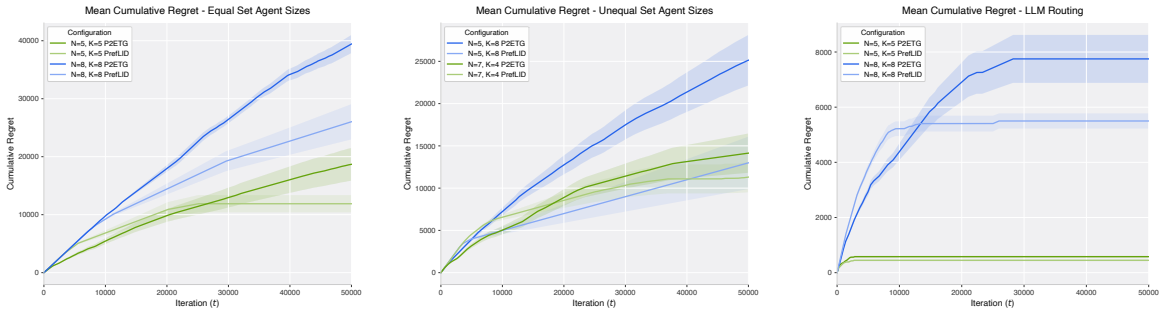
Subsequently, in the learning setting (see Sec. 4), $\widehat{\succ}_{AB}$ is not given explicitly but is derived from the current confidence intervals: for each agent, we sort the opposing agents by their estimated Bradley–Terry parameters $\widehat{\theta}_{a,b}$ in descending order, and we place a *cut* between consecutive agents b_i, b_{i+1} wherever their confidence intervals are disjoint. Within a partition, the intervals overlap and the internal order remains unresolved (refer to Diagram 1). Although the size of the space of configurations grows exponentially with the number of agents, we can compute the $|\widehat{\succ}_{AB}|$, the cardinality of $\widehat{\succ}_{AB}$, in polynomial time. (See Lemma A.1 in §A.1.)

Rotational Exploration and Lattice Construction: Applying Gale–Shapley to a fully-resolved configuration $\succ_{ab}^{(c)} \in \widehat{\succ}_{AB}$ yields one stable matching ρ_m; the map $\{\mathsf{GS} : \widehat{\succ}_{AB} \to \widetilde{\mathcal{M}},\ \widehat{\succ}_{AB}^{(c)} \mapsto \rho_m\}$ returns a stable matching from a fully-resolved preference, i.e., $\mathsf{GS}(\widehat{\succ}_{AB}^{(c)}) = \rho_m$, where $\mathcal{M}$ denotes the set of matchings and $\widetilde{\mathcal{M}} \in \mathcal{M}$ the stable subset. From any $\rho_m \in \mathcal{M}$, remaining stable matchings compatible with the same configuration arise via rotations, producing a lattice $\mathscr{L}^{(c)}$ among stable matchings. Each rotation is computable in polynomial time, and the number of unique rotations is upper-bounded by $\mathsf{nCr}(\bar{N}, 2)$ (Gusfield, 1988), where $\bar{N} \stackrel{\text{def}}{=} \max\{N, K\}$ (w.l.o.g. $N = K$ here for complexity analysis). A series of rotations generates a lattice; denote each rotation as $\{\mathsf{ROT} : \rho_m^{(i)} \to \rho_m^{(j)}\}$. Thus a lattice $\mathscr{L}^{(c)}$ is associated to each preference profile $\widehat{\succ}_{AB}^{(c)}$. Importantly, two different preference profiles need not share the same rotation structure: even if their lattices share matchings, their rotations (and partial orders among rotations) can differ. We write $\{\mathscr{L}\}$ for the set of all such lattices and denote the map $\mathsf{LAT} : \widehat{\succ}_{AB} \to \{\mathscr{L}\}$ sending any fully-resolved configuration to its corresponding lattice.

Intersection of Lattices and Islands $\mathsf{ConstructIslands}(\widehat{\succ}_{AB})$**:** Within $\{\mathscr{L}\}$, distinct lattices may share vertices. Any matching in the intersection of two or more lattices is a *centroidal lattice matching*, ρ^*. When $V(\mathscr{L}^{(i)}) \cap V(\mathscr{L}^{(j)}) \neq \emptyset$, the lattices belong to the same *island*, $\mathfrak{C}$. Precisely, $\mathfrak{C}^{(m)} \stackrel{\text{def}}{=} \{\mathscr{L}_m^{(i)}, \mathscr{L}_m^{(j)}, \dots\}$ is a set of lattices connected by a single ρ_m^*. A lattice may belong to multiple islands. We define a blocking function $\{\Gamma : \{\mathscr{L}\} \mapsto \mathfrak{C}\}$ mapping lattices to islands. For instance, $\mathfrak{C} = \Gamma(\{\mathscr{L}\})$ might be $\{\{\mathscr{L}_m^{(i)}, \mathscr{L}_m^{(j)}\}, \{\mathscr{L}_n^{(i)}, \mathscr{L}_n^{(j)}, \mathscr{L}_n^{(k)}\}, \dots, \{\mathscr{L}_o^{(i)}\}\}$. The blocking function partitions $\{\mathscr{L}\}$ into islands, each sharing a common ρ^*. In the worst case each lattice is its own island; in the best case all form one island. It involves searching all intersecting vertices and runs in polynomial time (Lem A.2; Algorithm 6 in §E). The key advantage: given any $\widehat{\succ}_{AB}$, if $|\mathfrak{C}| = 1$, the centroidal matching is a super stable matching (obviating resolution over all strict preferences). The number of intersecting matchings per lattice is also computable in polynomial time (Lemma A.2 in §A.7). Thus $\mathfrak{C}$ is constructible, though worst-case configuration count scales exponentially with $|\widehat{\succ}_{AB}| \leq \bar{N}^2$, where $\bar{N} \stackrel{\text{def}}{=} \max\{N, K\}$. We denote the resulting procedure $\{\mathsf{ConstructIslands} : \widehat{\succ}_{AB} \to \mathfrak{C}\}$.

Theorem 1. ***Super-Stability and the Single Island Rule:*** *Given any bipartite ordering topology $\widehat{\succ}_{AB}$, if the corresponding lattice topology constitutes a singleton set $|\Gamma(\{\mathscr{L}\}_{ab})| = 1$ then effectively the all centroidal lattice matchings in $\mathfrak{C}$ are stable matchings. Therefore, $|\Gamma(\{\mathscr{L}\}_{ab})| = 1 \iff \rho^* \in \widetilde{\mathcal{M}}$.*

Proof. For any $\mathfrak{C}^{(c)} \in \mathfrak{C}$, the centroidal lattice matching ρ_m^* is a stable matching for every refinement of every configuration $\widehat{\succ}_{AB} \in \widehat{\succ}_{AB}$ corresponding to a lattice in $\mathfrak{C}^{(c)}$. Therefore, if $|\mathfrak{C}| = 1$, every refinement of $\widehat{\succ}_{AB}$ yields the same stable matching, which is *super-stable* per Def. 2.4. □



(a) Scenario: $N = K$. (b) Scenario: $N \neq K$. (c) Scenario: LLM Routing.

Figure 3: **Comparison of Cumulative Regret:** We evaluate P2ETG and PrefLID on both synthetically generated Bradley-Terry parameters, as well as LLM routing (see Sec. 4.4). Across all instances, the cumulative regret performance of PrefLID versus P2ETG demonstrates lower cumulative regret for the same configuration of matching agents (N, K). Shaded regions illustrate uncertainty up to one standard deviation. We present a more comprehensive ablation study in §F.

3.1 Tractability of the Lattice Island Construction

Eq. (1) shows that in the worst case a combinatorially large number of configurations leads to a combinatorially large number of candidate matchings. To control this growth, we impose a budget B_Σ, and require that the number of configurations enumerated satisfies $|\{\mathscr{L}\}| \leq B_\Sigma$. At each iteration, if the size of the candidate set of lattices is within the budget, we may enumerate all configurations explicitly. Otherwise, we subsequently break down the problem via partitioning until the configuration count falls below the budget (see §4.3).

Tractibility via Optimal Partitioning: Although enumerating all orderings in $\widehat{\succ}_{AB}$ explodes factorially as demonstrated in Eq. (1), sufficient partitioning can control the problem: generating tractable sub-problems can reduce complexity more than the factorial explosion, while incrementally resolving preference certainty among one agent at a time sequentially controls combinatorial explosion. In the worst case, $\mathsf{ConstructIslands}(\widehat{\succ}_{AB})$ scales exponentially with $|\widehat{\succ}_{AB}|$, therefore additional measures are needed to enforce tractability. In §A.3, we demonstrate that complexity can be made tractable given a sufficient partition function (specifically Thm. 6, §A.3). As we introduce online learning in Sec. 4, the intuitive idea that sampling leads to partitions of $\widehat{\succ}_{AB}$ and it is cheaper than offline combinatorial search, and the two complement each other (i.e. sample sufficiently to reduce the problem search space over $\mathfrak{C}$, and use the island structure to speed up sampling vice-versa).

4 Online Learning

We extend optimisation under perfect information to a setting with parameter uncertainty, where both sides may not know their preferences yet must learn them and obtain a stable matching. Typically this uses an explore-then-commit (ETC) framework (Kong & Li, 2023b; Kong et al., 2024; Liu et al., 2021), drawing enough samples so that an estimate $\widehat{\succ}$ reflects the true preferences $\succ$ with high probability. We present our initial baseline algorithm, P2ETG a key extension of an ETC bandit framework with additional two-sided uncertainty, and also adapts the Bradley–Terry pairwise comparison model directly into the learning protocol (see Sec. 4.2). We further devise PrefLID, an improvement over P2ETG, which reduces the sample complexity of ETC framework via construction of the island structure $\{\mathsf{ConstructIslands} : \widehat{\succ}_{AB} \to \mathfrak{C}\}$ and targeted sampling (see Sec. 4.3).

Interaction Protocol: We now extend from optimisation under perfect information to parameter uncertainty, where both sides may not know their own preferences and must learn them while obtaining a stable matching. The literature typically uses an explore-then-commit framework, drawing enough samples so that an estimate $\widehat{\succ}_a$ reflects the true preferences $\succ_a$ with high probability (Kong & Li, 2023b; Kong et al., 2024; Liu et al., 2021). We adopt this framework with two extensions: we allow *both* sides to be uncertain, and we embed the Bradley–Terry pairwise comparison model directly into the online learning algorithm and its regret analysis. Works such as (Kong et al., 2024) hinge on observability of a utility function, but in matching a utility can typically only be compared via a measure over an outcome (e.g. from a's perspective, matching b_i yields 7.7 points in expectation versus 5.8 for b_j), which induces a ranking (e.g. $7.7 > 5.8 \implies b_i \succ b_j$). Our framework obviates the need for a direct measure of the matching outcome.

Feedback Mechanism: Building on (Kong & Li, 2023b; Kong et al., 2024; Liu et al., 2021), the bandit-feedback model no longer adheres to real-valued stochastic feedback. Instead of a sub-Gaussian reward $\mu_a : (a, b) \mapsto \mathbb{R}$ with a stationary mean $\mathbb{E}[\mu_a(a, b)]$ when (a, b) is matched, a more natural approach governs the interaction. When a match between (a, b) occurs, it results in two outcomes $\left(o_{(a,b)},\, o_{(b,a)}\right)_t$ which each side can distinguish in terms of preference. Let us focus on the left $\mathcal{A}$-side outcomes $o_{(a,\cdot)}$. Suppose the *arm* (a, b_i) is played concerning a match for both sides a and b. This outcome could also be compared to another set of outcomes featuring (a, b_j). For example, suppose at two separate time steps a match was made $(o_{a,b_i})_{t=1}$ and $(o_{a,b_j})_{t=2}$. The player a can then evaluate their preferences on a probabilistic basis, $\mathbb{P}\big((o_{(a,i)})_{t=1} \succ (o_{(a,j)})_{t=2}\big)$, governed by their internal preference parameters θ_a in accordance with the Bradley–Terry model (see §4.1). We denote this as $\mathbb{P}_a(i \succ j)$. Evidently, each agent has an internal set of parameters θ_a which governs their preference function on each specific outcome from matching with other agents.

Definition of Regret: Previous work in this area typically defines regret with respect to the proposer-optimal, receiver-pessimal stable matching ρ_{GS}, as given by the Gale–Shapley algorithm. Any misspecification can drive the algorithm to produce an alternative matching ρ', regardless of whether ρ' is stable. We cast our regret notion in a similar fashion,

$$\texttt{Reg} \stackrel{\text{def}}{=} \sum_{t=1}^{T} \mathbb{1}\left(\rho_t \notin \widetilde{\mathcal{M}}\right). \tag{2}$$

Regret declares a maximum loss of 1 each time the algorithm fails to obtain the correct stable matching under perfect information when estimated preferences are input at each time step.

Definition 4.1. ***Stopping Time:*** *The stopping time $T_{\texttt{stop}}$ is the number of samples required such that, with probability at least $1 - \eta$, the learner can commit to a fixed strategy of play for all iterations $t > T_{\texttt{stop}}$.*

4.1 The Bradley–Terry Model Model for Preference Relation

The Bradley–Terry model is a probability model for paired comparisons (Bradley & Terry, 1952). It provides a principled framework for estimating the relative strengths or abilities of items (players, teams, products, etc.) based on observed pairwise contest outcomes. For a pair of items i and j with positive strength parameters θ_i and θ_j, the model specifies the probability that i beats j in a contest as $\mathbb{P}(i \succ j) = \theta_i/(\theta_i + \theta_j)$ (Simons & Yao, 1999). It follows that $\theta_i > \theta_j$ implies $i \succ j$ in the preference ranking over $\mathcal{B}$ for the corresponding agent. However, since $\mathbb{P}(i \succ j)$ is a probability, the closer θ_i and θ_j are to each other, the more samples are required to distinguish the minute preference. We refer to this as the *preference parameter gap*, Δ. In our two-sided setting, the parameters are indexed by both the agent and the partner, for example each A-side agent $a \in \mathcal{A}$ carries a vector $\theta_a \in \mathbb{R}^{A}_{>0}$, with coordinate $\theta_{a,b}$ describing a's latent affinity for partner $b \in \mathcal{B}$, and mutatis mutandis for the B-side. Within agent a's own ranking, the comparison between two partners $b_1, b_2 \in \mathcal{B}$ is then governed by,

$$\mathbb{P}_a(b_1 \succ b_2) = \frac{\theta_{a,b_1}}{\theta_{a,b_1} + \theta_{a,b_2}}, \tag{3}$$

and symmetrically for the B-side[2]. Because the two sides hold *independent* parameter vectors, the learning problem is genuinely two-sided: no side has access to the ground truth. The standard method to estimate Bradley Terry parameters via maximum liklihood estimation is presented in §C.2.

Assumption 4.1. ***Bradley–Terry Parameter Normalisation.*** *The parameters of each agent are normalised so that $\sum_{b\in\mathcal{B}} \theta_{a,b} = 1$ for every $a \in \mathcal{A}$, and symmetrically $\sum_{a\in\mathcal{A}} \theta_{b,a} = 1$ for every $b \in \mathcal{B}$.*

4.2 Pairwise Two-Sided Explore-then-Gale-Shapley Algorithm

We now describe the *Pairwise Two-Sided Explore-then-Gale–Shapley* algorithm (P2ETG). Fundamentally, this is an adaptation of the explore-then-commit (ETC) algorithm adapted to pairwise comparisons via the Bradley Terry framework. The procedure runs in doubling epochs $\ell = 0, 1, 2, \ldots$, where in epoch ℓ the target

[2]Often in literature, we can also express $\mathbb{P}_a(b_1 \succ b_2) = (1 + \exp(\gamma_{a,b_1}\gamma_{a,b_1} + \gamma_{a,b_2}))^{-1}$, here by convention we use $\theta_{a,b} \stackrel{\text{def}}{=} \exp(\gamma_{a,b})$, aligning with the representation in Simons & Yao (1999).

number of pairwise replications per agent–pair is $R_\ell = 2^\ell$. Within each epoch, all agents on both sides perform round-robin pairwise comparisons. After each epoch the Bradley–Terry maximum-likelihood estimates are refreshed, confidence intervals are recomputed, the empirical preference profile is reconstructed by sorting $\hat{\theta}$ in descending order, and Gale–Shapley is run on the estimated profile. If the confidence intervals of every pair of potential partners have become disjoint for every agent, the algorithm commits to the current Gale–Shapley matching and performs no further exploration. (We present the explicit outline of this Alg. 2 in §B.2.)

Theorem 2. P2ETG ***Regret.*** *Pairwise Two-Sided Explore-then-Gale-Shapley Algorithm (*P2ETG*) experiences an expected regret of,*

$$\mathtt{Reg} \in \mathsf{O}\left(\frac{N}{\Delta^2}\log\left(\frac{T}{K\Delta^2}\right)\right)$$

with respect to the any stable matching is. (Proof is provided in §B.3.)

4.3 Preference Resolution via Lattice Island Detection (PrefLID)

PrefLID is a two-sided, explore-then-commit learner that certifies *global* stability of a committed matching by verifying that all configurations compatible with the current uncertainty lead to a single lattice island. It improves on the sample complexity of P2ETG by certifying a matching at the level of refinements rather than at the level of individual pairs. The stopping condition triggers when $|\mathfrak{C}| = 1$, i.e., when only one island exists in the partitions of $\{\{\mathscr{L}\}\}$; this implies, with probability $1-\eta$, that the candidate matchings $\{\rho_m^*, \rho_n^*, \dots \rho_l^*\}$ corresponding to that singleton island are the true stable matchings. In other words, all rotations starting from the various possible estimated preference configurations share a common destination matching. (A detailed exposition of PrefLID is provided in §B.4 along with full algorithmic details.)

Targeted Round-Robin Sampling (RRT): PrefLID employs *Targeted Round-Robin (RRT) Sampling* (see §7) rather than sampling every pair uniformly. Specifically, each round centers on a single agent c, who performs a full round-robin against the opposite side ($K(K-1)/2$ comparisons per round), refining only c's partition structure.

Theorem 3. ***Regret and Stopping Dominance of*** PrefLID***:*** *Under the same preference–feedback model and confidence sequences as* P2ETG*,*

$$\mathtt{Reg}^{\mathsf{PrefLID}}(T) = O\big(\mathtt{Reg}^{\mathsf{P2ETG}}(T)\big), \qquad T_{stop}^{\mathsf{PrefLID}} \le T_{stop}^{\mathsf{P2ETG}}, \qquad \Delta^{\mathsf{PrefLID}} \le \Delta^{\mathsf{P2ETG}},$$

where $\Delta^{\mathsf{PrefLID}}, \Delta^{\mathsf{P2ETG}}$ are the smallest preference gaps that must be resolved before the respective stopping conditions are certified.

Proof. PrefLID shares the sampling rule of P2ETG, but resolves only comparisons that can distinguish between islands. Pairs whose ordering is irrelevant to the surviving centroidal matching may remain unresolved indefinitely

Algorithm 1 Abridged Version of Preference Resolution via Lattice Island Detection (PrefLID) - See §B.4.

```
Initialize the unresolved profile ≻̂_AB and all comparison statistics.
loop
    if the compatible lattice representation exceeds B_Σ then
        Choose the next entering center agent c.
        Perform RRT(c); update Bradley–Terry estimates, confidence intervals, and the blocks of ≻̂_AB.
        continue
    end if
    {ℒ} ← {LAT(≻_ab) :≻_ab is a strict refinement of ≻̂_AB}.
    𝔠 ← ConstructIslands({ℒ}).
    if all surviving lattices share a certificate ρ* then
        return ρ*.
    end if
    Select an unresolved comparison associated with competing islands (using the widest-confidence rule).
    Sample it until the relation is resolved with confidence 1 − η; update ≻̂_AB.
end loop
```

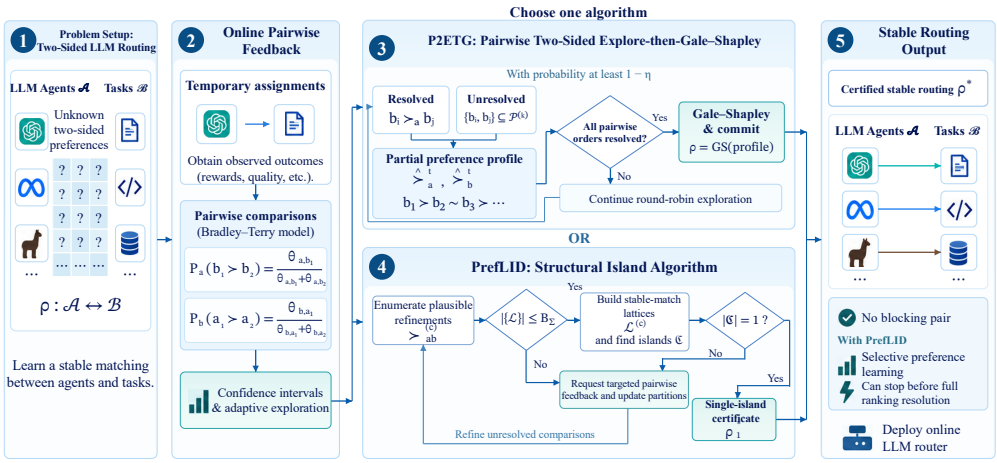


Figure 4: **The LLM Routing Workflow:** Both tasks and LLM agents have reciprocal preferences about which model is most appropriate for a given task. The LLM router's goal is to match the optimal agent to each incoming task. Both P2ETG and PrefLID are benchmarked within this experimental setting.

without affecting the certified output. Hence the set of gaps PrefLID must resolve is a subset of those required by P2ETG, giving $T_{\text{stop}}^{\text{PrefLID}} \le T_{\text{stop}}^{\text{P2ETG}}$ and $\Delta^{\text{PrefLID}} \le \Delta^{\text{P2ETG}}$. Moreover, a single island ($|\mathfrak{C}| = |\Gamma(\{\mathscr{L}\})| = 1$) certifies by Theorem 1 that the shared centroidal matching ρ_1^* is super-stable, so PrefLID stops as soon as this holds. Since per-sample regret is identical under the shared sampling rule and PrefLID stops no later, $\texttt{Reg}^{\text{PrefLID}}(T) \le C\,\texttt{Reg}^{\text{P2ETG}}(T)$ for a universal constant C, yielding the stated regret order. □

4.4 LLM Routing Markets - A Modern Application of Online Stable Matching

We provide a novel application for the stable matching framework within the context of LLM routing, specifically instantiated from the `LLMRouterBench` evaluation suite (Li et al., 2026). Here N models (side A) and K tasks (side B) rank each other by a shared compatibility, making their preferences reciprocal—$\text{rank}_a(b) = \text{rank}_b(a)$ for all a, b—since competence is mutual, simplifying the preference structure. This reciprocity collapses the ambiguity into a single unique stable matching, which the Gale–Shapley algorithm solves deterministically, returning the same unique outcome regardless of which side proposes. Computationally, the problem thus reduces to finding a singleton $\widetilde{\mathcal{M}}$. Therefore, the goal is to find $\{\rho^*\} = \widetilde{\mathcal{M}}$ in which no unpaired model–task pair prefers each other to their current partners (whereas arbitrary preferences can yield a rich lattice of stable matchings and task-optimal extremes). A workflow of this experimental setting is presented in Fig. 4, and an extensive manifest of the experiments can be found in §F.1.

5 Conclusion

We developed an online learning framework for two-sided stable matching via pairwise comparison. By representing uncertainty explicitly as a partial ordering and exploiting the lattice–island structure of the induced stable matching space, we showed that a stable outcome can be certified without resolving every agent's complete ranking: our algorithm PrefLID resolves only those comparisons that distinguish between competing islands and thereby dominates the full-resolution baseline P2ETG in both stopping time and the smallest parameter gap that must be overcome, directly improving on cumulative performance. Our framework is particularly natural for modern AI systems that route queries among large language models, where preferences are learned from noisy pairwise or contextual feedback and fully ranking every model for every query type is prohibitively costly. Several directions remain open, such as, extending our framework to transient preferences with cycles, where pairwise comparisons may induce intransitive relations, would require rethinking both the partial-ordering representation and the notion of stability itself. Also, a finer parameterisation of regret via graph-structural properties of the island—such as its diameter, and connectivity—could yield sharper instance-dependent bounds and guide the design of even more sample-efficient exploration schedules.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, literature and citation checking, proof checking, and experimental design. AI-generated suggestions were independently reviewed and verified by the authors. All mathematical claims, proofs, algorithms, experiments, citations, and conclusions are the responsibility of the authors.

REFERENCES

Haris Aziz, Péter Biró, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. Stable matching with uncertain linear preferences. *Algorithmica*, 82:1410–1433, 2020. doi: 10.1007/s00453-019-00650-0.

Haris Aziz, Péter Biró, Tamás Fleiner, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. Stable matching with uncertain pairwise preferences. *Theoretical Computer Science*, 909:1–11, 2022. doi: 10.1016/j.tcs.2022.01.028.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Róbert Busa-Fekete, Balázs Szörényi, and Eyke Hüllermeier. Pac rank elicitation through adaptive sampling of stochastic pairwise preferences. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. doi: 10.1609/aaai.v28i1.8978.

Wei Chen, Yihan Du, Longbo Huang, and Haoyu Zhao. Combinatorial pure exploration for dueling bandit. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1531–1541, 2020.

Christine T Cheng and Will Rosenbaum. Stable matchings with restricted preferences: Structure and complexity. *ACM Transactions on Economics and Computation*, 10(3):1–45, 2023.

Joanna Drummond and Craig Boutilier. Preference elicitation and interview minimization in stable matchings. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 645–653, 2014. doi: 10.1609/aaai.v28i1.8829.

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.

Rohith Reddy Gangam, Tung Mai, Nitya Raju, and Vijay V. Vazirani. Robust stable matchings: Dealing with changes in preferences. *arXiv preprint arXiv:2601.07959*, 2026a.

Rohith Reddy Gangam, Tung Mai, Nitya Raju, and Vijay V Vazirani. Robust stable matchings: Dealing with changes in preferences. *arXiv preprint arXiv:2601.07959*, 2026b.

Dan Gusfield. The structure of the stable roommate problem: Efficient representation and enumeration of all stable assignments. *SIAM Journal on Computing*, 17(4):742–769, August 1988. ISSN 0097-5397. doi: 10.1137/0217048.

Hadi Hosseini, Sanjukta Roy, and Duohan Zhang. Putting gale & shapley to work: Guaranteeing stability through learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Robert W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48(3):261–272, 1994. doi: 10.1016/0166-218X(92)00179-P.

Fang Kong and Shuai Li. Player-optimal stable regret for bandit learning in matching markets. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1512–1522, 2023a. doi: 10.1137/1.9781611977554.ch55.

Fang Kong and Shuai Li. Improved bandits in many-to-one matching markets with incentive compatibility. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i12.29226. URL https://doi.org/10.1609/aaai.v38i12.29226.

Fang Kong, Zilong Wang, and Shuai Li. Improved analysis for bandit learning in matching markets. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a730abbcd6cf4a371ca9545db5922442-Abstract-Conference.html.

Fang Kong, Xiaoxi Zhang, Xiao Huang, Lijun Zhang, and Shuai Li. Bandit learning in matching markets with relative feedback. *Theoretical Computer Science*, 1071:115846, 2026. doi: 10.1016/j.tcs.2026.115846.

Fangyuan Kong and Shuai Li. Bandit learning in many-to-one matching markets. In *International Conference on Machine Learning*, 2023b. URL https://proceedings.mlr.press/v202/kong23a.html.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, Zhen Wang, and Shuyue Hu. Llmrouterbench: A massive benchmark and unified framework for llm routing. *arXiv preprint arXiv:2601.07206*, 2026.

Shuai Li, Zilong Wang, and Fang Kong. A survey on bandit learning in matching markets. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 10546–10554, 2025. doi: 10.24963/ijcai.2025/1171.

Larkin Liu and Jalal Etesami. Online mixture of experts: No-regret learning for optimal collective decision-making. *Advances in Neural Information Processing Systems*, 38:166827–166873, 2026.

Lydia T. Liu, Horia Mania, and Michael I. Jordan. Competing bandits in matching markets. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1618–1628. PMLR, 2020.

Lydia T. Liu, Feng Ruan, Horia Mania, and Michael I. Jordan. Bandit learning in decentralized matching markets. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LPJ_akD-cB.

Tung Mai and Vijay V. Vazirani. Finding stable matchings that are robust to errors in the input. In *26th Annual European Symposium on Algorithms*, Leibniz International Proceedings in Informatics, 2018a. doi: 10.4230/LIPIcs.ESA.2018.60.

Tung Mai and Vijay V Vazirani. Finding stable matchings that are robust to errors in the input. *arXiv preprint arXiv:1804.00553*, 2018b.

Gordon Simons and Yi-Ching Yao. Asymptotics when the number of parameters tends to infinity in the bradley-terry model for paired comparisons. *The Annals of Statistics*, 27(3):1041–1060, 1999. doi: 10.1214/aos/1018031267. URL https://projecteuclid.org/journals/annals-of-statistics/volume-27/issue-3/Asymptotics-when-the-number-of-parameters-tends-to-infinity-in-the/10.1214/aos/1018031267.full.

Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 5502–5510, 2018. doi: 10.24963/ijcai.2018/776.

Ernst Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29:436–460, 1929. doi: 10.1007/BF01180541. URL https://doi.org/10.1007/BF01180541.

YiRui Zhang and Zhixuan Fang. Decentralized two-sided bandit learning in matching market. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL https://openreview.net/forum?id=RUazoBVXIN.