EFFICIENT ENSEMBLE SELECTION FROM BINARY AND PAIRWISE FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

To choose a small ensemble of models, prompts, or solvers, we must learn not only how well each candidate performs but also which tasks it solves that the others miss, and this information is costly to obtain. We ask how much feedback such a choice requires. Tasks are drawn from an unknown distribution, each query reveals either one candidate's correctness on one task or one pairwise comparison, and a committee of k candidates is judged by its best member on each task.

For binary feedback, we give a conditional greedy algorithm that evaluates possible additions only on tasks that the current committee misses. It attains coverage at least $(1 - 1/e)\mathrm{OPT}_k - \varepsilon$ with high probability, and its query bound includes the cost of finding the missed tasks. For a fixed gap in success rate between two possible additions to a fixed committee, theoretical bounds show that agreement on missed tasks reduces the number of queries needed. For pairwise feedback, majority cycles can make every committee lose to another of the same size on nearly all tasks. We therefore ask a committee to beat every fixed omitted candidate on a large fraction of tasks. With complete rankings, this objective admits a polynomial-time approximation scheme (PTAS) but, under randomized Gap-ETH, no efficient one (EPTAS). Allowing for lotteries over committees changes the picture: by expressing weighted comparisons with candidates as coverage and reweighting these candidates with multiplicative weights, we approximate the best lottery over size-k committees to within a factor $1 - 1/e$, up to additive error ε, using polynomially many comparisons. Experiments on controlled synthetic populations illustrate the savings from agreement, the limits of conditional sampling, and the gain from randomization.

1 INTRODUCTION

When candidate solutions to a mathematics problem can be verified, a single correct one is enough. Similarly, a user choosing a logo for a website can review several designs and keep the one they like best. In such settings, a collection of models, prompts, or solvers should be judged by the best output it produces, not by the average quality of its outputs.

A good *ensemble* combines candidates that succeed on *different* tasks. Suppose two candidates solve the same 60% of tasks and a third solves exactly the remaining 40%. Together, the two strongest candidates still fail on 40% of tasks, whereas either of them paired with the weaker third solves every task. Individual performance says little about the value of a group; what matters is *complementarity*. Similarly, teams of diverse problem solvers can outperform teams of individually stronger ones (Hong & Page, 2004; LiCalzi & Surucu, 2012; Marcolino et al., 2013).

Complementarity motivates algorithm portfolios, ensemble selection, and model routing (Gomes & Selman, 2001; Caruana et al., 2004; Jiang et al., 2023; Chen et al., 2024; Ong et al., 2025). It matters more as organizations deploy several foundation models side by side (Wang et al., 2025), and more still as the gap between leading models on aggregate leaderboards narrows (Maslej et al., 2025). Individual scores do not show complementarity. It appears only in joint outcomes, which must be learned by evaluating candidates on shared tasks (via queries), as model selection guides from cloud providers recommend (Amazon Web Services, 2023; Microsoft Azure Architecture Center, 2026). Every such evaluation has a cost. The form of feedback may also be limited: some tasks admit a binary correct/incorrect label, whereas for others we can only ask which of two outputs is better.

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.

We therefore study how to select complementary candidates when both the amount and the form of feedback are constrained.

We call a selected set of k candidates a *committee*. Tasks are drawn independently from an unknown distribution, and a committee is judged by its best member on each task. The *win rate* of one committee against another is the probability that its best outcome is better, plus half the probability of a tie. With binary feedback, a committee covers a task when at least one member solves it. With pairwise feedback, we compare the committee with each candidate left out. Both objectives measure whether a useful answer is *available* among the committee's outputs; choosing which answer to return at deployment is a separate problem. Each query reveals one correctness bit or one comparison on one task, and we count every query. Our central question is:

> *How many queries does it take to find a near-optimal committee, and what guarantees are achievable with binary and pairwise feedback?*

1.1 Our Contributions

Binary feedback. Our objective is *coverage*, i.e., the probability that at least one selected candidate solves a random task. In social choice, maximizing coverage is known as the approval Chamberlin–Courant rule (Chamberlin & Courant, 1983; Skowron & Faliszewski, 2017). Every committee that maximizes coverage has win rate at least $1/2$ against every committee of the same size. Coverage is also submodular, so greedy selection achieves a $(1 - 1/e)$-approximation when marginal gains are known (Nemhauser et al., 1978). The challenge is to learn which candidate to add at each step without evaluating every candidate on every task.

Our *conditional greedy* algorithm first tests the current committee and evaluates possible additions only on tasks it misses. It eliminates clearly inferior candidates early and uses coarser conditional estimates when few tasks remain uncovered. The algorithm attains coverage at least $(1-1/e)\mathrm{OPT}_k-\varepsilon$ with high probability, with a query bound that counts both candidate evaluations and the queries needed to find missed tasks (Section 3.3). The bound is smaller in two cases. When marginal gaps are large, clearly inferior candidates are eliminated early. When coverage is high, coarser estimates suffice on the few tasks that remain uncovered.

We also study the cost of comparing two possible additions to a fixed committee. On a missed task where both candidates succeed or both fail, they add the same amount to the committee's coverage, so only disagreements distinguish them. For a fixed gap in marginal gain, we give upper and lower bounds that match up to logarithmic factors. They show that the number of paired observations needed scales linearly with the candidates' disagreement probability on missed tasks (Section 3.2). Thus, candidates that agree more often on missed tasks can be cheaper to compare.

Pairwise feedback. With pairwise feedback, there may be no committee with win rate at least $1/2$ against every committee of the same size, because majority preferences can cycle (the Condorcet paradox). For example, suppose the rankings $a \succ b \succ c$, $b \succ c \succ a$, and $c \succ a \succ b$ each occur on a third of tasks. Then every singleton loses to another singleton on two thirds of tasks. We therefore compare a committee with each fixed omitted candidate, which we call a rival, and maximize the smallest of these win rates (Elkind et al., 2011).

Our guarantees depend on whether deployment requires a single committee or permits a *lottery*, that is, a probability distribution over committees. Suppose we need a single committee and have n complete task rankings. A result of Charikar et al. (2025) then yields a $(1 - \gamma)$-approximation in $m^{O(1/\gamma)}\,\mathrm{poly}(m, n)$ time, which is a polynomial-time approximation scheme (PTAS). We give a reduction showing that, under randomized Gap-ETH, no efficient PTAS (EPTAS) exists. Such a scheme would run in $f(1/\gamma)\,\mathrm{poly}(m, n)$ time (Section 4.2). The algorithm also applies to sampled rankings with polynomially many comparisons, but its computational exponent still depends on the desired accuracy.

Randomization helps. In the example above, choosing each singleton with probability $1/3$ matches or beats every fixed rival with probability $2/3$. For lotteries, we maximize the smallest win-or-tie rate against any fixed rival. Each rate is averaged over both the committee draw and the task draw. We express weighted comparisons with rivals as a coverage problem and solve it with conditional greedy. Multiplicative weights then shifts weight toward rivals that earlier committees handle poorly.

The resulting algorithm approximates the best lottery to within a factor $1 - 1/e$, up to additive error ε. It uses $O(\varepsilon^{-2} \log m)$ weighted greedy calls and polynomially many comparisons in total (Theorem 4.5). Unlike the approximation scheme, it requires neither complete rankings nor enumeration of committees.

Experiments. Experiments on controlled synthetic populations illustrate these results (Section 5). When two possible additions rarely disagree, paired tests compare them with far fewer queries than separate estimates of their success rates. At disagreement probability 0.08, the disagreement test uses a median of 59,925 queries, compared with 1,541,100 for separate estimates. Under a fixed query budget, conditional greedy outperforms full evaluation when specialist tasks are rare, but not when they are common. On cyclic rankings, multiplicative weights with exact population values nearly reaches the best lottery's value (0.663 versus 0.667). That is about twice the value of the best single candidate (0.338).

1.2 Related Work

Our algorithms build on greedy submodular maximization, action elimination in bandits, and adaptive sampling for noisy submodular maximization (Nemhauser et al., 1978; Even-Dar et al., 2006; Singla et al., 2016). Our query accounting differs in two ways. We charge for the queries spent finding tasks on which an addition can matter. We also show that the cost of a paired comparison scales with the candidates' disagreement probability. For single committees under pairwise feedback, our approximation scheme applies the existence theorem of Charikar et al. (2025). Our contribution there is a reduction showing that, under randomized Gap-ETH, the scheme's running time cannot be improved to that of an EPTAS. For lotteries, we adapt weighted methods for robust optimization (Krause et al., 2008; Chen et al., 2017) to task-level comparisons whose costs we count explicitly. Appendix A discusses related work on voting under incomplete information, algorithm portfolios, ensemble pruning, and pure exploration.

2 Model and Objectives

We select a committee of k *candidates* from a set $C = [m]$. Define $\mathcal{S}_k = \{S \subseteq C : |S| = k\}$ as the possible committees, and assume $1 \le k < m$; when $k = m$, we can return C without querying. *Tasks* are independent draws from an unknown distribution $\mathcal{D}$. Drawing a task reveals nothing by itself; the algorithm chooses which candidates to query on the task, and it may store the task and query it again later. Our guarantees therefore count every query, including those spent learning how the current committee performs. We consider two forms of feedback.

Binary outcomes. For a task d and candidate c, one query reveals the correctness indicator $u(d, c) \in \{0, 1\}$. A committee solves a task whenever at least one of its members does, and we measure its value by its *coverage*, the probability that it solves a random task. Formally, the task outcome, coverage, and optimal coverage at size k are $U(d, S) = \max_{c \in S} u(d, c)$, $v(S) = \Pr_{d \sim \mathcal{D}}[U(d, S) = 1]$, and $\mathrm{OPT}_k = \max_{S \in \mathcal{S}_k} v(S)$, respectively. Let $U(d, \varnothing) = v(\varnothing) = 0$. Although tasks are drawn independently, candidate outcomes on the same task may be arbitrarily dependent. This dependence is the heart of the problem: two candidates with the same success rate may solve the same tasks or entirely different ones, and only their joint outcomes reveal which.

Pairwise outcomes. Each task induces a strict ranking π of C, so the task distribution induces a distribution $P_{\mathcal{D}}$ over rankings. A query reveals which of two candidates ranks higher on the task, but not their numerical ranks. Write $\mathrm{rank}_\pi(c)$ for the rank of c, with smaller numbers better, and $r_\pi(S) = \min_{c \in S} \mathrm{rank}_\pi(c)$ for the rank of the committee's best member, with $r_\pi(\varnothing) = +\infty$. All queries on a task refer to the same ranking, so answering one comparison also determines its reverse. Across tasks, rankings may vary arbitrarily. Comparing committees through their best members extends pairwise model evaluation, as in Chatbot Arena (Chiang et al., 2024), to ensembles.

Win rates. For committees $S, S' \subseteq C$, the win rate of S against S' gives one unit of credit when S has the better outcome and half a unit when the outcomes tie. Under binary feedback, it is

$$\mathrm{WIN}(S, S') = \Pr_{d \sim \mathcal{D}}[U(d, S) > U(d, S')] + \tfrac{1}{2} \Pr_{d \sim \mathcal{D}}[U(d, S) = U(d, S')].$$

Under pairwise feedback, it is

$$\mathrm{WIN}(S, S') = \Pr_{\pi \sim P_{\mathcal{D}}}[r_\pi(S) < r_\pi(S')] + \tfrac{1}{2} \Pr_{\pi \sim P_{\mathcal{D}}}[r_\pi(S) = r_\pi(S')].$$

Binary outcomes tie when both committees solve the task or both do not. With strict rankings, nonempty committees tie only when their best member belongs to both. In either model, $\mathrm{WIN}(S, S') + \mathrm{WIN}(S', S) = 1$, so a win rate of at least $1/2$ means that S wins at least as often as it loses.

Probability and computation. We use $\varepsilon \in (0, 1)$ for additive accuracy and $\delta \in (0, 1)$ for the permitted error probability. A high-probability query bound holds on the same event as the accompanying value guarantee, whereas an expected query bound averages over all observations and internal randomness. We also study *full-information* problems, in which all outcomes or rankings on a finite sample of n tasks are known in advance. These problems separate the computational difficulty of selection from the statistical difficulty of acquiring feedback.

3 Binary Feedback

Binary feedback arises whenever an output can be checked automatically, as in question answering with reference answers, mathematics with known solutions, or code with unit tests. Many benchmarks and evaluation frameworks score models in this way (Wang et al., 2024; White et al., 2025; UK AI Security Institute, 2024), and routing benchmarks evaluate many models on shared tasks because no single model is best on all of them (Hu et al., 2024). The coverage of a committee is then the fraction of tasks that a router, verifier, or ensemble could solve by using the right member. Each correctness bit, however, requires running a candidate on a task and grading its output, so with many candidates the evaluations themselves become the main cost of selection.

Coverage determines the win rate against every competing committee. Since $U(d, S), U(d, S') \in \{0, 1\}$, their difference is 1 when S wins, -1 when S' wins, and 0 when they tie. Taking expectations and using the half credit for ties gives

$$\mathrm{WIN}(S, S') = \frac{1}{2} + \frac{v(S) - v(S')}{2}. \tag{1}$$

Thus every coverage maximizer over $\mathcal{S}_k$ has win rate at least $1/2$ against every other size-k committee.

A candidate added to a committee helps precisely on tasks that it solves and the current committee misses. As the committee grows, fewer such tasks remain, so coverage has diminishing returns: for $S \subseteq S'$ and $c \notin S'$, the marginal gain $\Delta(c \mid S) = v(S \cup \{c\}) - v(S)$ satisfies $\Delta(c \mid S) \geq \Delta(c \mid S')$. This property, *submodularity*, makes greedy selection the natural starting point: if all marginal gains are known, repeatedly adding a candidate with maximum gain achieves coverage at least $(1 - 1/e)\mathrm{OPT}_k$ (Nemhauser et al., 1978). Our goal is to approximate these choices while evaluating as few outcomes as possible. We begin with a full-evaluation baseline, study how agreement affects the cost of comparing two candidates, and then analyze conditional greedy, which selects among all remaining candidates but does not use the two-candidate disagreement test.

For a committee S, define its *miss probability* by $\rho(S) = 1 - v(S)$ and, when $\rho(S) > 0$, a candidate's *conditional success probability* by $q(c \mid S) = \Pr[u(d, c) = 1 \mid U(d, S) = 0]$. The marginal gain satisfies

$$\Delta(c \mid S) = \rho(S) q(c \mid S). \tag{2}$$

3.1 A reference method using full evaluation

The simplest approach gathers a complete sample first and selects a committee afterwards. Evaluate all m candidates on T tasks $d_1, \ldots, d_T$, define the empirical coverage $\widehat{v}_T(S) = T^{-1} \sum_{t=1}^{T} U(d_t, S)$, and run greedy on $\widehat{v}_T$ to obtain $\widehat{S}$. With enough tasks, empirical and true coverage are close for every size-k committee simultaneously. Uniform concentration over the $\binom{m}{k}$ committees shows that $T = O((\log \binom{m}{k} + \log(1/\delta))/\varepsilon^2)$ tasks suffice for $v(\widehat{S}) \geq (1 - 1/e)\mathrm{OPT}_k - \varepsilon$ with probability $1 - \delta$, at a cost of mT queries (Appendix B.1). The drawback is that full evaluation

tests every candidate even on tasks that the committee already solves, where no addition can help. We use full evaluation as a baseline for assessing the cost of adaptive sampling. Some dependence on both m and ε is unavoidable: even for $k = 1$, additive-ε selection requires $\Omega(m/\varepsilon^2)$ expected queries in the worst case (Theorem B.2), mirroring the classical lower bound for identifying a near-best arm in a bandit (Mannor & Tsitsiklis, 2004).

3.2 Comparing candidates that usually agree

On a task the committee misses, two potential additions to the committee contribute the same amount whenever both succeed or both fail. The difference in their marginal gains therefore depends only on the tasks where they disagree.

Fix a committee S with $i = |S|$ and $\rho = \rho(S) > 0$. For two additions $a, b \notin S$, define their disagreement probability and conditional success gap as $d_{ab} = \Pr[u(d, a) \neq u(d, b) \mid U(d, S) = 0]$ and $g_{ab} = |q(a \mid S) - q(b \mid S)|$.

The *disagreement test* is based on McNemar's paired test (McNemar, 1947). For each task it draws, it queries committee members until one succeeds or all fail. When all fail, it queries both candidates, and when the candidates disagree, it records which one succeeded. The test stops once a confidence interval for the probability that a wins a disagreement lies entirely above or entirely below $1/2$.

Theorem 3.1. *If $g_{ab} > 0$, the disagreement test selects the better addition with probability at least $1-\delta$ using $O((2+i/\rho)\cdot(d_{ab}/g_{ab}^2)\cdot\log(ed_{ab}/(\delta g_{ab})))$ queries in expectation, counting all committee and candidate queries.*

A matching lower bound on the number of paired observations holds up to the logarithmic factor. Appendix B.2 proves both bounds.

With the gap g_{ab} fixed, candidates that agree more often on missed tasks need fewer queries. They agree more often exactly when d_{ab} is small, and the bound scales linearly with d_{ab}. Intuitively, the better candidate wins each disagreement with probability $1/2 + g_{ab}/(2d_{ab})$. Detecting this bias takes about $(d_{ab}/g_{ab})^2$ disagreements, which is about d_{ab}/g_{ab}^2 missed tasks. Each missed task costs at most $2+i/\rho$ queries in expectation. Conditional greedy, introduced next, compares all candidates approximately rather than in pairs.

3.3 Conditional greedy and its total query cost

Conditional greedy (CG) builds a committee one candidate at a time. At step i, it estimates the current committee's miss probability $\rho_i = \rho(S_i) = 1 - v(S_i)$. If sufficiently few tasks remain uncovered, CG fills the remaining places arbitrarily and stops. Otherwise, it compares unselected candidates on missed tasks, using confidence intervals to eliminate clearly inferior choices. The comparison accuracy is chosen so that the selected candidate's marginal gain is within ε/k of the best, with high probability.

CG obtains missed tasks by rejection sampling: draw a task, query committee members until one succeeds or all fail, and retain the task only if all fail. Unselected candidates are queried only on retained tasks. All queries, including committee tests, are charged. Algorithm 1 in Appendix B gives the implementation.

Let $\tau \leq k$ be the number of selection calls, and define the *marginal gap*

$$\Delta_i(c) = \max_{a \notin S_i} \Delta(a \mid S_i) - \Delta(c \mid S_i).$$

This is the marginal gain lost by choosing c instead of the best addition.

Theorem 3.2. *With probability at least $1-\delta$, CG returns $\widehat{S} \in \mathcal{S}_k$ with $v(\widehat{S}) \geq (1-1/e)\mathrm{OPT}_k - \varepsilon$ and total query count*

$$Q = O\left(\log\frac{emk^2}{\delta\varepsilon}\left[\underbrace{\frac{k^2}{\varepsilon}}_{\text{early stopping}} + \sum_{i<\tau}\left(\underbrace{\sum_{c\notin S_i}\frac{\rho_i^2}{\max\{\Delta_i(c),\varepsilon/k\}^2}}_{\text{candidate queries}} + \underbrace{\frac{i\rho_i k^2}{\varepsilon^2}}_{\text{committee queries}}\right)\right]\right).$$

Ignoring logarithmic factors, the three terms correspond to:

1. **Early stopping.** The preliminary test at a step where CG stops early uses at most $O(k/\varepsilon)$ task draws, each requiring at most k committee queries. This gives the k^2/ε term.
2. **Candidate queries.** This term comes from evaluating unselected candidates on tasks that were missed. By (2), a conditional error becomes a marginal error once it is multiplied by ρ_i. It is therefore enough to estimate conditional accuracy to within order $\varepsilon/(k\rho_i)$, which takes on the order of $\rho_i^2 k^2/\varepsilon^2$ samples. When a candidate's marginal gap is large, it can be eliminated earlier, and this is where the gap in the denominator comes from. Gaps smaller than ε/k need not be resolved.
3. **Committee queries.** This term covers the cost of finding missed tasks. Selection needs at most order $\rho_i^2 k^2/\varepsilon^2$ missed tasks. Finding each one takes $1/\rho_i$ draws in expectation, and each draw uses at most i committee queries. Multiplying these gives $i\rho_i k^2/\varepsilon^2$. The same term also covers the preliminary tests run at steps that continue.

We defer the proof to Appendix B.4. The bound identifies where CG can save queries. Large marginal gaps allow early elimination. A small miss probability means that rejection sampling skips candidate evaluations on most tasks, coarser comparisons suffice on the remaining tasks, and early stopping may avoid further selections. We compare the methods under fixed query budgets in Section 5.

4 Pairwise Feedback

Many tasks have no answer that can be checked automatically, such as open-ended dialogue or summarization. Evaluation then typically asks which of two outputs is better: Chatbot Arena collects such judgments from users and fits a Bradley–Terry model to rank models (Bradley & Terry, 1952; Chiang et al., 2024; Ameli et al., 2025), and pairwise preferences are also used to train routers and to rank candidate outputs within ensembles (Ong et al., 2025; Jiang et al., 2023). Comparisons pooled over heterogeneous tasks, however, need not be transitive; in fact, any pattern of majority preferences can arise (McGarvey, 1953). This possibility has led recent work to evaluate AI systems with tools from social choice (Lanctot et al., 2025; Haghtalab et al., 2026; Hays et al., 2026; Khalaf et al., 2026), and it shapes our objective as well.

Equation (1) suggests asking for a committee whose win rate is at least $1/2$ against every committee of the same size. However, pairwise feedback need not admit such a committee: the three-candidate cycle in the introduction already rules it out for $k = 1$. The limitation is more severe for larger collections of candidates. For every $m \geq 2k$, the uniform distribution over the cyclic rotations of $1 \succ 2 \succ \cdots \succ m$ has the following property: for every $S \in \mathcal{S}_k$, some disjoint $S' \in \mathcal{S}_k$ satisfies $\mathrm{WIN}(S, S') \leq k/m$. For fixed k, this bound tends to zero as m grows (Appendix C.1).

We therefore compare a committee with one omitted candidate at a time and maximize its least win rate against these rivals. This asks whether the ensemble provides a better outcome than each available alternative used alone. We first study this objective for a single committee, and then show that randomizing over committees allows a query-efficient algorithm.

4.1 A guarantee against every omitted candidate

For a nonempty proper committee S, a *rival* is a candidate $x \notin S$. Since rankings are strict and x is omitted, the best member of S cannot tie with x, so $\mathrm{WIN}(S, \{x\}) = \Pr_{\pi \sim P_{\mathcal{D}}}[r_\pi(S) < \mathrm{rank}_\pi(x)]$. Following Elkind et al. (2011), we evaluate the committee by its least favorable comparison.

Definition 4.1 (Winning value)**.** The *winning value* of S is $\theta(S) = \min_{x \notin S} \mathrm{WIN}(S, \{x\})$, and the best value at size k is $\theta_k^* = \max_{S \in \mathcal{S}_k} \theta(S)$. We set $\theta(C) = 1$.

We maximize this value without requiring it to reach $1/2$. The order of the minimum and the probability matters. The condition $\theta(S) \geq 0.8$ means that *each fixed* rival is beaten on at least 80% of tasks. Different rivals may be beaten on different tasks, so the committee need not contain the overall best candidate on 80% of tasks. A guarantee against individual rivals also extends to small competing committees, by a union bound (Lemma C.1).

4.2 Selecting a committee when all rankings are known

Even with complete rankings, the greedy argument from the binary case breaks down, because θ is not submodular (Proposition C.2). The minimum over rivals is responsible: improving a comparison that is not currently the worst leaves θ unchanged, yet the same improvement can become valuable once the worst rival has been handled. Nevertheless, rankings have enough structure to approximate the best committee arbitrarily well. In the next theorem, winning values refer to the uniform distribution over the given rankings. Its lower bound assumes randomized Gap-ETH: distinguishing satisfiable 3-CNF formulas from those in which a constant fraction of clauses must be violated requires exponential time, even for bounded-error randomized algorithms (Manurangsi, 2020).

Theorem 4.2. *Given n complete task rankings and any $\gamma \in (0, 1)$, a committee $S \in \mathcal{S}_k$ satisfying $\theta(S) \geq (1 - \gamma)\theta_k^*$ can be found in $m^{O(1/\gamma)}\mathrm{poly}(m, n)$ time. Under randomized Gap-ETH, no algorithm achieves this guarantee in $f(1/\gamma)\mathrm{poly}(m, n)$ time for any computable f.*

The first statement gives a *polynomial-time approximation scheme*: for every fixed γ, the running time is polynomial, although the exponent grows as γ shrinks. An *efficient* approximation scheme would keep the exponent fixed and confine the dependence on accuracy to a factor $f(1/\gamma)$. The second statement rules out such running times under randomized Gap-ETH. The approximation scheme relies on a result of Charikar et al. (2025): every ranking profile admits a committee of size $O(1/\gamma)$ with winning value at least $1 - \gamma$, so it suffices to enumerate committees of size at most $\min\{k, O(1/\gamma)\}$ and add arbitrary members, which cannot decrease the winning value. The lower bound is a reduction from a parameterized maximum-coverage problem (Appendix C.3).

The scheme also applies when the distribution is unknown: comparison sorting recovers the rankings of $T = O(\varepsilon^{-2}(\gamma^{-1} \log m + \log(1/\delta)))$ sampled tasks with $O(Tm \log m)$ comparisons, and running the scheme on them loses at most an additional ε (Corollary C.3). The query bound is polynomial in $1/\varepsilon$ and $1/\gamma$, whereas the computational exponent depends on $1/\gamma$. We next show that randomizing over committees changes both the objective and the available algorithms.

4.3 Lotteries over committees

Every committee may have some rival it handles poorly. A lottery over committees can spread these weaknesses so that no single rival exploits all of them, much as maximal lotteries avoid majority cycles among single alternatives (Fishburn, 1984).

To evaluate a lottery against a fixed rival, we must also say what happens when the drawn committee contains that rival. We count a task as covered whenever the committee offers an outcome at least as good as the rival's, and so extend the comparison value to every $x \in C$ by $g_x(S) = \Pr[r_\pi(S) \leq \mathrm{rank}_\pi(x)]$. In other words, $g_x(S)$ is the Win-or-tie rate against x. It agrees with the original comparison value when $x \notin S$, and it equals 1 when $x \in S$. The winning value of a fixed committee is therefore still $\theta(S) = \min_{x \in C} g_x(S)$. For a lottery p over $\mathcal{S}_k$, we average over committees first and then minimize over rivals: $\theta(p) = \min_{x \in C} \mathbb{E}_{S \sim p}[g_x(S)]$.

In the example, fix any rival. The three singletons then have values 1, $2/3$, and $1/3$ against it, so the uniform lottery has average Win-or-tie rate $2/3$ against every rival, and $\theta(p) = 2/3$.

Remark 4.3. The lottery value is not the same as $\mathbb{E}_{S \sim p}\theta(S)$, which equals $1/3$ in the example because every singleton has winning value $1/3$. The two differ in when the rival is chosen. The lottery guarantee fixes the rival before the committee and the task are drawn, so the rival cannot depend on which committee is realized. Drawing a fresh committee for each task achieves this guarantee on average over both draws. Drawing one committee and keeping it does not.

4.4 Reduction to weighted coverage

To optimize $\theta(p)$ without listing all committees, we start with a simpler problem: maximizing a weighted average over rivals. For a probability vector λ over C, let $\Phi_\lambda(S) = \sum_{x \in C} \lambda_x g_x(S)$. Let $P_\pi(x) = \{c : \mathrm{rank}_\pi(c) \leq \mathrm{rank}_\pi(x)\}$. A committee covers the task–rival pair (π, x) exactly when it intersects $P_\pi(x)$, so Φ_λ is a coverage function and hence submodular.

To sample, draw a task and an independent rival $x \sim \lambda$. If $x \in S$, the pair is covered at no cost. Otherwise, compare members of S with x until one ranks higher or all rank lower. Testing

whether an addition c covers an uncovered pair takes one comparison, or none if $c = x$. With $\rho_\lambda(S) = 1 - \Phi_\lambda(S)$ and $q_\lambda(c \mid S)$ the probability that $c \in P_\pi(x)$ given that S misses the pair,

$$\Phi_\lambda(S \cup \{c\}) - \Phi_\lambda(S) = \rho_\lambda(S) q_\lambda(c \mid S), \tag{3}$$

which is (2) for task–rival pairs.

Theorem 4.4. *For fixed λ, conditional greedy returns $\widehat{S} \in \mathcal{S}_k$ with $\Phi_\lambda(\widehat{S}) \geq (1 - 1/e) \max_{S \in \mathcal{S}_k} \Phi_\lambda(S) - \varepsilon$ with probability at least $1 - \delta$. Its total comparison cost satisfies the bound of Theorem 3.2 with weighted miss probabilities and marginal gaps.*

This bound counts every comparison used to find uncovered pairs, so the weighted objective can be optimized without recovering complete rankings.

4.5 Learning the lottery by reweighting rivals

We can now call weighted selection repeatedly, each time shifting weight toward rivals that earlier committees handle poorly. We do this with *multiplicative weights*. Start with $\lambda_1(x) = 1/m$. In round t, run weighted conditional greedy to choose S_t. Then estimate each $g_x(S_t)$ by $\widehat{g}_t(x)$ on fresh tasks, using $m - 1$ comparisons per task, and update $\lambda_{t+1}(x) \propto \lambda_t(x) \exp(\eta_{\mathrm{MW}}[1 - \widehat{g}_t(x)])$, where $\eta_{\mathrm{MW}} > 0$ is the learning rate. After R rounds, return the uniform lottery over $S_1, \ldots, S_R$ (Algorithm 4, Appendix D.3).

This works because, for any weights, some committee has weighted value at least $\max_{p'} \theta(p')$. Each round therefore gets a $(1 - 1/e)$ fraction of this value, up to estimation error. The reweighting then ensures that every rival receives nearly this much on average over the rounds.

Theorem 4.5. *With probability at least $1 - \delta$, Algorithm 4 returns a lottery p with $\theta(p) \geq (1 - 1/e) \max_{p'} \theta(p') - \varepsilon$, where p' ranges over all lotteries on $\mathcal{S}_k$. It uses $R = O(\varepsilon^{-2} \log m)$ rounds of weighted selection and $O(Rm\varepsilon^{-2} \log(mR/\delta))$ evaluation comparisons. On the same event, the total comparison count, including weighted selection, is $O(mk^3 \varepsilon^{-4} \log m \log(emk^2/(\delta\varepsilon)))$.*

With pairwise feedback, we thus obtain two distinct guarantees. The approximation scheme approaches the best single committee arbitrarily closely, at a computational cost that grows with the accuracy. The lottery algorithm uses polynomially many comparisons and approximates a different, possibly much larger optimum, whose value averages over the committee draw.

5 Experiments

Our experiments address three questions. How does agreement between two possible additions affect the cost of comparing them? When does a lottery achieve a higher winning value than any single committee? And when does selective binary evaluation improve coverage under a fixed query budget? All experimental populations are specified synthetically. We distinguish measured query costs from objective values computed with full information. Appendix E gives the constructions, parameters, and additional results.

Agreement saves queries at a fixed mean gap. We hold the candidates' conditional success rates fixed and vary how often they disagree. Figure 1 (left) shows that agreement reduces the query cost of the paired Bernstein and disagreement tests, but not of separate mean estimates. At disagreement probability 0.08, their median costs are 320,772 and 59,925, compared with 1,541,100 for separate means, including all committee tests. All 6,000 runs across the parameter settings stop by their statistical rule and select the better candidate. These results concern learning which addition is better, not the benefit of selecting redundant candidates.

Randomization improves the objective, not every realized committee. Figure 1 (right) varies the weight on cyclic rankings. At full cyclic weight, the best singleton has value 0.338, while the best lottery has value 0.667. Multiplicative weights reaches 0.663 after 100 rounds using exact population values. This is an illustration of the lottery objective and convergence, not of query efficiency: the winning value of an individual committee drawn from the lottery remains substantially lower.

Selective evaluation does not always outperform full evaluation. We compare four methods on populations with one generalist, two specialists, and 58 decoys, and on a population of independent

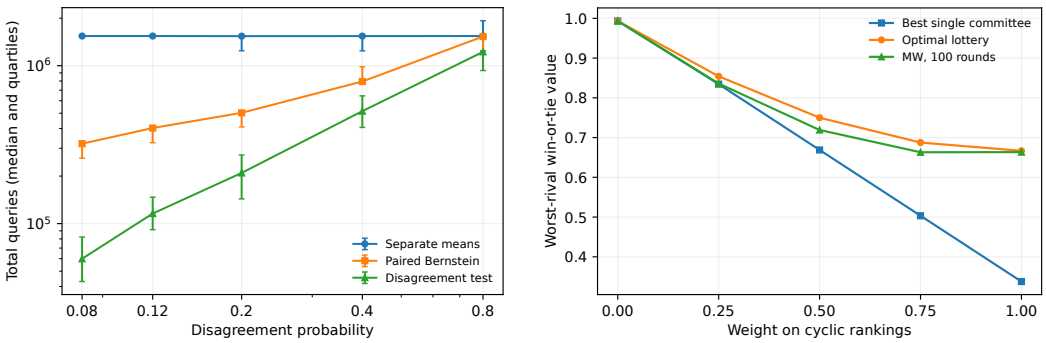


Figure 1: Agreement and randomization in controlled examples. **Left:** at a fixed mean gap, agreement reduces the cost of comparing two additions. Points are medians over 100 independent runs; bars span the middle half of the query counts. **Right:** as more weight is placed on cyclic rankings, the best lottery pulls away from the best single committee.

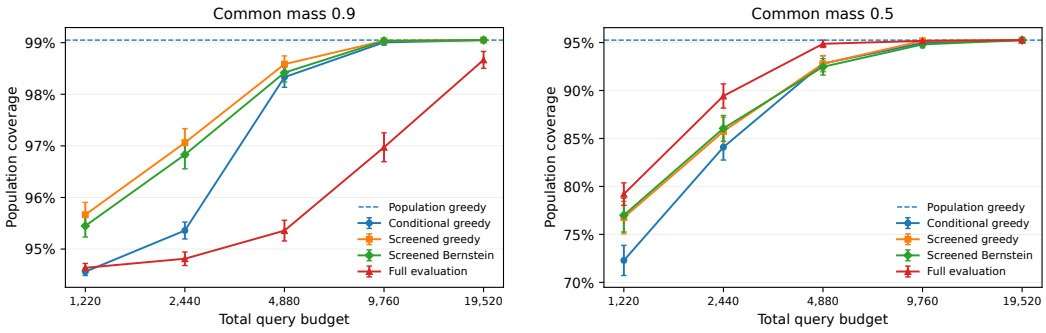


Figure 2: Binary selection under fixed query budgets, with $m = 61$ and $k = 3$. **Left:** rare specialist tasks, with common mass 0.9. **Right:** more frequent specialist tasks, with common mass 0.5. All queries are charged, including committee tests. Bars show 1.96 standard errors over 200 repetitions; dashed population-greedy values use information unavailable to the query algorithms. Appendix E.3 reports all four populations.

candidates. At a budget of 4,880 queries, conditional greedy attains mean coverage 0.9833, compared with 0.9536 for full evaluation, when the common task type has probability 0.9 (Figure 2, left). At common mass 0.5, the comparison reverses: the means are 0.9281 and 0.9487 (right). Screened baselines can match or outperform conditional greedy. Every adaptive trial includes at least one budget-forced addition, so these results compare coverage under a fixed budget, not confidence-certified query complexity.

6 Conclusion

Selecting an ensemble requires learning how candidates complement the members already chosen. With binary feedback, this leads to evaluating additions only on tasks the current committee misses. We accounted for the cost of finding missed tasks and showed that agreement between two additions makes them cheaper to compare. With pairwise feedback, the achievable guarantee depends on whether we return one committee or a lottery. A single committee admits an approximation scheme but, under randomized Gap-ETH, no efficient one, whereas a lottery admits a query-efficient $(1 - 1/e)$ approximation to a different, possibly larger optimum.

Deployment also requires a rule for choosing among the committee's answers. Our experiments use controlled synthetic populations; performance on model-derived or human-preference data remains to be established. A next question is whether selection can exploit agreement among many candidates while handling exact ties and counting every query.

AI USE STATEMENT

The problem formulation, the models, the algorithms, the theorem statements and their proofs, the experimental design, and the interpretation of the results are the authors' own; we did not use generative AI tools for any of these. The one task requiring disclosure under ICLR policy for which we used generative AI tools is implementing the experiments: coding assistants helped write, edit, and debug the experimental code and rerun the experiments. Generating synthetic datasets, translation, dataset cleaning, and qualitative data analysis do not apply: the synthetic populations in Appendix E are produced by seeded samplers that we specified, not by a generative model, and we use no external data. We also used generative AI tools to polish the writing, including the exposition of the proofs, to help find related work, to draft the documentation for the supplementary code, and to prepare the submission files. These edits changed presentation, not content.

The authors reviewed all AI-assisted work. We checked every suggested edit against our intended meaning, revised or rejected edits where needed, and confirmed that no mathematical statement or proof step was altered. We read and tested the code through unit tests, an independent implementation of the coverage formulas, seeded replays, and primal–dual checks on every linear program (Appendix E.4); a fresh rerun from scratch reproduces every stored trial record. We checked each reference, and each claim about prior work, against the original source. We take responsibility for the final content of this work, including text, claims, code, and results produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendices B–D state all assumptions and give proofs of every theoretical result. Appendix E specifies the synthetic populations, selection rules, query budgets, and summary statistics, and Appendix E.4 lists software versions, random seeds, and the exact commands. The supplementary material contains the code, every trial record, and scripts that regenerate all figures, Table 1, and every number reported in the text; a single command reruns all trials from scratch in a fresh directory and compares them with the stored records. Independent verification code recomputes each committee's coverage, checks query accounting and summary statistics, replays seeded runs, and re-solves the linear programs for the cyclic populations. A full rerun matches all 16,000 binary, 6,000 paired, and 180 cyclic records, with stored tables agreeing to within 10^{-12}. The experiments run on a CPU; no GPU, external datasets, model API calls, or credentials are required.

REFERENCES

Amazon Web Services. Evaluate, compare, and select the best fms for your use case in amazon bedrock (preview). https://aws.amazon.com/about-aws/whats-new/2023/11/evaluate-compare-select-fms-use-case-amazon-bedrock/, 2023.

Siavash Ameli, Siyuan Zhuang, Ion Stoica, and Michael W. Mahoney. A statistical framework for ranking LLM-based chatbots. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

Nima Anari, Nika Haghtalab, Seffi Naor, Sebastian Pokutta, Mohit Singh, and Alfredo Torrico. Structured robust submodular maximization: Offline and online algorithms. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3128–3137, 2019.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, 2010.

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.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pp. 18, 2004.

John R. Chamberlin and Paul N. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.

Moses Charikar, Alexandra Lassota, Prasanna Ramakrishnan, Adrian Vetta, and Kangning Wang. Six candidates suffice to win a voter majority. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1590–1601, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Robert Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 379–387, 2014.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 8359–8388, 2024.

Davin Choo, Paul W. Goldberg, and Nicholas Teh. Learning unanimously acceptable lotteries via queries. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Edith Elkind, Jérôme Lang, and Abdallah Saffidine. Choosing collectively optimal sets of alternatives based on the Condorcet criterion. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 186–191, 2011.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

Peter C. Fishburn. Probabilistic social choice based on simple voting comparisons. *The Review of Economic Studies*, 51(4):683–692, 1984.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2697–2705, 2013.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Carla P. Gomes and Bart Selman. Algorithm portfolios. *Artificial Intelligence*, 126(1–2):43–62, 2001.

Nika Haghtalab, Ariel D. Procaccia, Han Shao, Serena Lutong Wang, and Kunhe Yang. Pluralistic leaderboards. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Daniel Halpern, Gregory Kehne, Ariel D. Procaccia, Jamie Tucker-Foltz, and Manuel Wüthrich. Representation with incomplete votes. *Theory and Decision*, 100:257–296, 2026.

Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. In *Proceedings of the 30th Annual Conference on Learning Theory (COLT)*, pp. 1069–1122, 2017.

Chris Hays, Rachel Li, Bailey Flanigan, and Manish Raghavan. Strategic candidacy in generative AI arenas. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Lu Hong and Scott E. Page. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences*, 101(46):16385–16389, 2004.

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.

Aviram Imber, Jonas Israel, Markus Brill, and Benny Kimelfeld. Approval-based committee voting under incomplete information. *Artificial Intelligence*, 347:104381, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 14165–14178, 2023.

Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1):3–45, 2019.

Hadi Khalaf, Serena L. Wang, Daniel Halpern, Itai Shapira, Flavio du Pin Calmon, and Ariel D. Procaccia. Robust AI evaluation through maximal lotteries. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Lars Kotthoff. Algorithm selection for combinatorial search problems: A survey. *AI Magazine*, 35: 48–60, 2014.

Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(93):2761–2801, 2008.

Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. SpringerBriefs in Intelligent Systems. Springer, 2023.

Marc Lanctot, Kate Larson, Michael Kaisers, Quentin Berthet, Ian Gemp, Manfred Diaz, Roberto-Rafael Maura-Rivero, Yoram Bachrach, Anna Koop, and Doina Precup. Soft Condorcet optimization for ranking of general agents. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1253–1262, 2025.

Marco LiCalzi and Oktay Surucu. The power of diversity over large solution spaces. *Management Science*, 58(7):1408–1421, 2012.

Feline Lindeboom, Martijn Brehm, Davide Grossi, and Pradeep Murukannaiah. Diverse committees with incomplete or inaccurate approval ballots. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2026.

Tyler Lu and Craig Boutilier. Multi-winner social choice with incomplete preferences. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 263–270, 2013.

Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

Pasin Manurangsi. Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 62–81, 2020.

Leandro Soriano Marcolino, Albert Xin Jiang, and Milind Tambe. Multi-agent team formation: Diversity beats strength? In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 279–285, 2013.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Toby Walsh, Armin Hamrah, Lapo Santarlasci, Julia Betts Lotufo, Alexandra Rome, Andrew Shi, and Sukrut Oak. Artificial intelligence index report 2025. Technical report, AI Index Steering Committee, Institute for Human-Centered AI, Stanford University, Stanford, CA, April 2025.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

David C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4): 608–610, 1953.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157, 1947.

Microsoft Azure Architecture Center. Choose the right AI model for your workload. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/choose-ai-model, 2026.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, pp. 34433–34448, 2025.

John R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

Adish Singla, Sebastian Tschiatschek, and Andreas Krause. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2037–2043, 2016.

Piotr Skowron and Piotr Faliszewski. Chamberlin–Courant rule with approval ballots: Approximating the MaxCover problem with bounded frequencies in FPT time. *Journal of Artificial Intelligence Research*, 60:687–716, 2017.

Artin Tajdini, Lalit Jain, and Kevin Jamieson. Nearly minimax optimal submodular maximization with bandit feedback. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 96254–96281, 2024.

UK AI Security Institute. Inspect AI: A framework for large language model evaluations. https://github.com/UKGovernmentBEIS/inspect_ai, 2024.

Sarah Wang, Shangda Xu, Justin Kahl, and Tugce Erten. How 100 enterprise CIOs are building and buying Gen AI in 2025. https://a16z.com/ai-enterprise-2025/, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 95266–95290, 2024.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32:565–606, 2008.

Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 2012.

Itay Asher Zimet, Shiri Alouf-Heffetz, and Nimrod Talmon. Query-based committee selection. *arXiv preprint arXiv:2603.29729*, 2026.

A Additional Related Work

Committee selection and partial feedback. Our binary objective is the maximum-coverage objective, known in voting as approval Chamberlin–Courant: each task approves the candidates that solve it and is represented if the committee contains at least one of them (Chamberlin & Courant, 1983; Skowron & Faliszewski, 2017; Lackner & Skowron, 2023). The $(1 - 1/e)$ approximation guarantee and its worst-case computational limit are classical (Nemhauser et al., 1978; Feige, 1998). Work on incomplete and queried preferences asks related questions about selection with limited information (Lu & Boutilier, 2013; Imber et al., 2025; Lindeboom et al., 2026; Halpern et al., 2026; Zimet et al., 2026). We specify the cost at the level of one correctness bit or one pairwise comparison on a task, including the queries needed to find tasks on which an addition can help. Robust submodular optimization likewise uses distributions over feasible sets to obtain guarantees for several objectives (Anari et al., 2019). Lotteries are also central to probabilistic social choice, where maximal lotteries exist even when majority preferences cycle (Fishburn, 1984), and recent work learns unanimously acceptable lotteries from queries (Choo et al., 2026); our lottery objective instead asks for a guarantee against every omitted candidate.

Algorithm portfolios and per-instance selection. A closely related question predates LLMs: how should one choose a solver, or a portfolio of solvers, for instances from a given problem family (Gomes & Selman, 2001; Kotthoff, 2014; Kerschke et al., 2019; Rice, 1976; Xu et al., 2008)? The benefit of a portfolio comes from complementary abilities; a solver need not be best on every instance to be useful. Much of this literature learns a selector or solver schedule for each instance from training data with rich instance features. We ask how many binary or pairwise queries are needed before deployment to choose a small fixed committee with high value on future tasks.

Ensemble pruning and library selection. Classical ensemble learning likewise selects useful subsets from large collections of trained models (Caruana et al., 2004; Kuncheva, 2004; Zhou, 2012). These methods typically seek to improve prediction accuracy on a validation set. We retain the subset-selection problem while limiting access to the validation outcomes: the algorithm may reveal only some entries of the evaluation matrix. Our goal is to bound the number of queries needed to select a high-value committee, rather than to prescribe how its predictions should be weighted or averaged.

Pure exploration with structured feedback. Each greedy step in our binary algorithm resembles pure exploration in bandits: the aim is to identify a candidate with large marginal gain, rather than to maximize reward during the search (Audibert et al., 2010; Even-Dar et al., 2006). Combinatorial pure exploration studies identification problems in which the answer is a structured set of arms (Chen et al., 2014). Here, candidates are evaluated on shared tasks, and their marginal values depend on what the current committee already solves. These two features explain why our query bounds depend on committee miss probabilities and marginal gaps, as well as on the statistical difficulty of comparing candidates.

Adaptive submodular and bandit submodular optimization. Coverage is submodular, and our algorithms choose which observations to obtain as they proceed. This connects our setting to adaptive submodularity and to submodular maximization with noisy or bandit feedback (Golovin & Krause, 2011; Gabillon et al., 2013; Hassidim & Singer, 2017; Tajdini et al., 2024). In our problem, the adaptive decisions determine which correctness bits or pairwise comparisons to reveal before selecting the final committee. We use this choice to focus queries on tasks where a candidate can improve the committee and to reuse comparisons when evaluating several committees.

B Binary Feedback: Proofs and Further Results

This appendix proves the binary-feedback results in the order in which the analysis needs them. Appendix B.1 analyzes the full-evaluation baseline. Appendix B.2 studies the comparison of two additions to a fixed committee. Appendix B.3 gives the elimination routine that selects one addition, and Appendix B.4 combines it with the miss-rate estimate to prove Theorem 3.2. We close with a lower bound showing that the elimination routine's dependence on candidate gaps is necessary, even

Algorithm 1 Conditional greedy (CG)

Require: Committee size k, accuracy ε, error probability δ.
$r_0 \leftarrow \lceil 6\log(8k/\delta)\rceil$; $M_0 \leftarrow \lfloor 2r_0k/\varepsilon\rfloor$; $S_0 \leftarrow \varnothing$.
for $i = 0, \dots, k-1$ **do**
  Test S_i on fresh tasks until r_0 misses or M_0 draws. Keep the missed tasks.
  if fewer than r_0 misses occur **then**
    Add arbitrary candidates to reach size k and **return** .
  end if
  Let N_i be the draw count; set $\widetilde{\rho}_i = \min\{1, 2r_0/N_i\}$.
  Set $\eta_i = \varepsilon/(k\widetilde{\rho}_i)$ and select an addition to conditional accuracy η_i,
    with error probability $\delta/(4k)$, reusing the missed tasks first.
  Add the selected candidate to obtain S_{i+1}.
end for
return S_k.

when missed tasks are supplied directly. For reference, Algorithm 1 states conditional greedy as described in Section 3.3.

B.1 Full Evaluation and a Lower Bound

Evaluating every candidate on every sampled task gives an estimate of the coverage of every committee. This allows us to separate the number of queries needed for accurate estimation from the computation needed to choose a committee. Let $\widehat{v}_T$ be the empirical coverage from Section 3.1, and let $\widehat{S}_{\mathrm{ERM}} \in \arg\max_{S\in\mathcal{S}_k} \widehat{v}_T(S)$ maximize it. We also consider the size-k committee $\widehat{S}_{\mathrm{gr}}$ obtained by greedy selection on the same empirical function. Exact maximization need not be efficient, whereas greedy selection is.

Our first result shows that uniform accuracy over size-k committees suffices to make the empirical maximizer nearly optimal.

Theorem B.1. *Fix $\varepsilon, \delta \in (0,1)$. If $T \geq \frac{2}{\varepsilon^2}(\log\binom{m}{k} + \log(2/\delta))$, then, with probability at least $1-\delta$, $v(\widehat{S}_{\mathrm{ERM}}) \geq \mathrm{OPT}_k - \varepsilon$.*

Proof. We bound the estimation error for every size-k committee and then compare the empirical maximizer with a true optimum. For any fixed $S \in \mathcal{S}_k$, the variables $U(d_t, S)$ are independent Bernoulli variables with mean $v(S)$. Hoeffding's inequality gives

$$\Pr[|\widehat{v}_T(S) - v(S)| > \varepsilon/2] \leq 2e^{-T\varepsilon^2/2}.$$

Under the stated sample size, a union bound over $\mathcal{S}_k$ makes every estimate accurate to within $\varepsilon/2$ with probability at least $1-\delta$. On this event, the definition of $\widehat{S}_{\mathrm{ERM}}$ gives, for an optimal committee $S^* \in \mathcal{S}_k$,

$$v(\widehat{S}_{\mathrm{ERM}}) \geq \widehat{v}_T(\widehat{S}_{\mathrm{ERM}}) - \varepsilon/2 \geq \widehat{v}_T(S^*) - \varepsilon/2 \geq \mathrm{OPT}_k - \varepsilon. \qquad \square$$

Is the dependence on m caused only by full evaluation? It is not. Even when only one candidate is selected, an algorithm must find a single slightly better candidate hidden among many equally good ones, and it learns about each candidate only by querying that candidate.

Theorem B.2. *Assume $m \geq 2$, $0 < \varepsilon \leq 1/8$, and $k = 1$. Any adaptive algorithm that returns $\widehat{c} \in C$ with $v(\{\widehat{c}\}) \geq \mathrm{OPT}_1 - \varepsilon$ with probability at least $2/3$ on every instance has worst-case expected query complexity $\Omega(m/\varepsilon^2)$.*

Proof. We compare a distribution in which all candidates are equally good with m alternatives, each of which slightly improves a different candidate. The strategy is to show that a correct algorithm must distinguish the base distribution from most alternatives, and that each such distinction requires $\Omega(1/\varepsilon^2)$ queries to the improved candidate. Our model allows any joint distribution of the outcomes on a task, so we may take them to be independent across candidates. Let $\mathcal{D}^{(0)}$ make every outcome Bernoulli$(1/2)$. For each $i \in C$, let $\mathcal{D}^{(i)}$ change only candidate i's outcome to Bernoulli$(1/2+2\varepsilon)$.

Task profiles are independent across draws. Under $\mathcal{D}^{(i)}$, only candidate i has value at least $\mathrm{OPT}_1-\varepsilon$, so the algorithm must output i with probability at least $2/3$.

Let P_0, P_i be the laws of the algorithm's observations, queries, internal randomness, and output under these distributions. The Kullback–Leibler divergence KL, also called relative entropy, measures how well the observations distinguish two distributions. Because output probabilities sum to one, at least $m-1$ candidates satisfy $P_0(\widehat{c}=i) \le 1/2$. For each such i, the required change in output probability and Pinsker's inequality give

$$\mathrm{KL}(P_0 \| P_i) \ge 2(P_i(\widehat{c}=i) - P_0(\widehat{c}=i))^2 \ge 1/18.$$

We next relate this statistical distinction to the query count. Let N_i be the number of queries to candidate i, and let $Q = \sum_i N_i$. A repeated query to the same task–candidate pair reveals nothing new, so we may reuse its earlier answer and assume there are no repetitions. All unqueried outcomes are independent; consequently, only queries to i distinguish the two distributions. The adaptive chain rule for relative entropy gives

$$\mathrm{KL}(P_0 \| P_i) = \mathbb{E}_0[N_i]\,\mathrm{KL}\left(\mathrm{Bernoulli}(1/2) \,\|\, \mathrm{Bernoulli}(1/2+2\varepsilon)\right) \le \tfrac{32}{3}\varepsilon^2 \mathbb{E}_0[N_i].$$

The last inequality follows from $-\frac{1}{2}\log(1-16\varepsilon^2) \le (32/3)\varepsilon^2$ for $\varepsilon \le 1/8$. If the expected query count is infinite, the lower bound already holds. Otherwise, the chain rule follows by stopping after $Q \wedge n$ queries and taking $n \to \infty$. Thus $\mathbb{E}_0[N_i] \ge 1/(192\varepsilon^2)$ for at least $m-1$ candidates. Summing these bounds proves the claim. □

Full evaluation also permits an efficient choice of committee. The next result replaces exact empirical maximization with greedy selection and obtains the usual $1-1/e$ factor, with the same order of query cost.

Theorem B.3. *Fix $\varepsilon, \delta \in (0,1)$. Let $\widehat{S}_{\mathrm{gr}} \in \mathcal{S}_k$ be the committee returned by empirical greedy on $\widehat{v}_T$, padded arbitrarily to size k if greedy stops early. If $T \ge \frac{(2-1/e)^2}{2\varepsilon^2}(\log\binom{m}{k} + \log(2/\delta))$, then, with probability at least $1-\delta$, $v(\widehat{S}_{\mathrm{gr}}) \ge (1-1/e)\mathrm{OPT}_k - \varepsilon$. The procedure uses $Q = mT$ queries.*

Proof. The proof combines the submodular greedy guarantee with uniform estimation accuracy. For every task, $S \mapsto U(d,S)$ is normalized, monotone, and submodular: adding c contributes $u(d,c)$ if S misses the task, and contributes zero otherwise. Increasing the committee can only reduce this contribution. Expectations and empirical averages preserve these properties, so both v and $\widehat{v}_T$ are normalized, monotone, and submodular.

The concentration argument from Theorem B.1, with the accuracy adjusted to account for the greedy factor, gives $|\widehat{v}_T(S) - v(S)| \le \varepsilon/(2-1/e)$ for all $S \in \mathcal{S}_k$ with probability at least $1-\delta$ under the stated sample size. On this event, apply the empirical greedy guarantee and compare with a committee S^* of optimal true coverage:

$$v(\widehat{S}_{\mathrm{gr}}) \ge \widehat{v}_T(\widehat{S}_{\mathrm{gr}}) - \frac{\varepsilon}{2-1/e} \ge (1-1/e)\widehat{v}_T(S^*) - \frac{\varepsilon}{2-1/e} \ge (1-1/e)\mathrm{OPT}_k - \varepsilon.$$

Padding does not reduce coverage. Full evaluation uses exactly mT queries. □

B.2 Comparing Additions through Their Disagreements

Theorem 3.1 compares two additions to a fixed committee. When both additions give the same answer on a missed task, their contributions on that task are equal, so we base the decision only on tasks where they disagree while still paying for every query. The proof has three steps: we bound the number of disagreements needed for a reliable comparison, translate that bound into total query cost, and finally show that the dependence on d_{ab} and g_{ab} cannot be improved for paired observations.

The stopping rule. Since $g_{ab} > 0$, we have $d_{ab} > 0$; write $h = g_{ab}/d_{ab} \in (0,1]$. On each new task, query the members of S in a fixed order until one is correct or all have been tested. If all are incorrect, query a and b. When these two outcomes differ, retain the sign 1 if a is correct and 0 if b is correct. The retained signs $Z_1, Z_2, \ldots$ are independent Bernoulli variables with mean $p = \frac{1}{2} + \frac{q(a|S)-q(b|S)}{2d_{ab}}$. Thus the better candidate is determined by whether p is above or below $1/2$.

To estimate this direction, after n signs set $\overline{Z}_n = n^{-1}\sum_{j=1}^{n} Z_j$ and $\mathrm{rad}_n = \sqrt{\frac{\log(4n^2/\delta)}{2n}}$. Stop and select a if $\overline{Z}_n - \mathrm{rad}_n > 1/2$, or select b if $\overline{Z}_n + \mathrm{rad}_n < 1/2$. The rule does not need to know ρ, d_{ab}, or g_{ab}.

Correctness and expected sign count. We choose the confidence radius so that the estimates are accurate at every possible stopping time. Hoeffding's inequality and a union bound give

$$\Pr\left[\exists n \geq 1 : \ |\overline{Z}_n - p| > \mathrm{rad}_n\right] \leq \sum_{n\geq 1} \frac{\delta}{2n^2} < \delta.$$

On the complementary event, the confidence interval always contains p, so the rule cannot select the worse candidate. We also need an expected query bound that includes the remaining outcomes. Assume without loss of generality that $p = 1/2 + h/2$, and let N be the number of retained signs at stopping. There is a universal constant K such that, for

$$n_0 = \left\lceil Kh^{-2}\log\frac{e}{\delta h}\right\rceil, \quad \mathrm{rad}_n \leq h/4 \quad \text{for all } n \geq n_0.$$

For example, substitution shows that $K = 1024$ suffices. If the rule has not stopped by such an n, then $\overline{Z}_n \leq 1/2 + \mathrm{rad}_n$, so $p - \overline{Z}_n \geq h/4$. Another application of Hoeffding's inequality gives

$$\Pr(N > n) \leq e^{-nh^2/8} \quad (n \geq n_0).$$

Summing these tail probabilities proves

$$\mathbb{E}N = O\left(h^{-2}\log\frac{e}{\delta h}\right). \tag{4}$$

In particular, the test stops almost surely. The same order holds when the test checks at an unbounded sequence of predetermined sign counts, starting at a fixed constant and increasing by at most a factor of two between checks, as in the experiments. Applying the tail bound at the preceding checkpoint changes it by at most a constant factor in the exponent.

Counting all queries. Each retained sign requires us to find a missed task on which the two additions disagree. Conditional on a committee miss, a disagreement occurs with probability d_{ab}, and its sign is independent of the number of preceding agreements. Thus, if M is the number of missed tasks examined before stopping, the independent geometric waiting times give $\mathbb{E}M = \mathbb{E}N/d_{ab}$. This identity holds for the random stopping count N because the stopping rule depends only on the retained signs. Similarly, a new task produces both a committee miss and a disagreement with probability ρd_{ab}, independently of the resulting sign. The total number T of task draws therefore satisfies $\mathbb{E}T = \mathbb{E}N/(\rho d_{ab})$. Each missed task requires two candidate queries, and each draw requires at most i committee queries, giving

$$\mathbb{E}Q \leq 2\mathbb{E}M + i\mathbb{E}T = \left(2 + \frac{i}{\rho}\right)\frac{\mathbb{E}N}{d_{ab}}.$$

Substituting (4) proves the upper bound in Theorem 3.1. On tasks that the committee solves, the test may query fewer than i members, which only lowers this cost.

A matching dependence on disagreement. To see why the factor d_{ab}/g_{ab}^2 is necessary, we compare two distributions that have the same disagreement probability but in which the better candidate differs. For $0 < g_{ab} \leq d_{ab}/2$ and $d_{ab} \leq 1$, define the following laws for the pair of bits $(u(d,a), u(d,b))$ on a missed task d:

	$(1,0)$	$(0,1)$	$(0,0)$	$(1,1)$
P_+	$(d_{ab}+g_{ab})/2$	$(d_{ab}-g_{ab})/2$	$(1-d_{ab})/2$	$(1-d_{ab})/2$
P_-	$(d_{ab}-g_{ab})/2$	$(d_{ab}+g_{ab})/2$	$(1-d_{ab})/2$	$(1-d_{ab})/2.$

Both laws have disagreement probability d_{ab} and absolute mean gap g_{ab}. Their Kullback–Leibler divergence measures how much a single paired observation reveals about which candidate is better:

$$\mathrm{KL}(P_+\|P_-) = g_{ab}\log\frac{d_{ab}+g_{ab}}{d_{ab}-g_{ab}} \leq \frac{8g_{ab}^2}{3d_{ab}}.$$

The inequality follows by integrating $2/(1-x^2) \le 8/3$ for $0 \le x \le g_{ab}/d_{ab} \le 1/2$.

Suppose a test observes complete independent pairs and selects the better candidate with probability at least $1-\delta$ under both laws. Let M be the number of pairs it observes. Its output must distinguish the two laws. The chain rule for relative entropy, applied up to the stopping time, and the event that the output is a give

$$\mathbb{E}_+[M]\,\mathrm{KL}(P_+\|P_-) \ge \mathrm{kl}(1-\delta,\delta) = (1-2\delta)\log\frac{1-\delta}{\delta} \ge \frac{1}{4}\log\frac{1}{\delta}$$

for $\delta \le 1/4$. Here kl is relative entropy between Bernoulli distributions. For a finite expected sample size, the stopped chain rule follows by truncation; an infinite expected sample size already satisfies the lower bound. Consequently,

$$\mathbb{E}_+[M] \ge \frac{3d_{ab}}{32g_{ab}^2}\log\frac{1}{\delta}.$$

The lower bound thus matches the dependence on d_{ab} and g_{ab} for conditional paired observations. It does not address the cost of finding missed tasks or of selecting all k committee members.

Relation to paired binary tests. Discarding agreements when comparing correlated binary outcomes is the classical idea behind McNemar's test (McNemar, 1947). Here we analyze the query cost of applying that idea during committee selection: we count the work needed both to find a missed task and to find a disagreement on such a task. The lower bound explains the resulting dependence on d_{ab} and g_{ab} for the paired comparison.

B.3 Conditional Candidate Selection

We now turn from comparing a pair of additions to choosing among all of them. When $\rho(S) = 0$, no addition can improve coverage and conditional selection is unnecessary.

Fix S with $\rho(S) > 0$ and a nonempty $A \subseteq C \setminus S$. Write $q^* := \max_{a\in A} q(a \mid S)$ and $\Delta_q(c) := q^* - q(c \mid S)$. Thus $\Delta_q(c)$ measures how far c is from the best conditional success rate. Algorithm 2 queries the remaining candidates on independent tasks drawn from $\mathcal{D} \mid U(d,S) = 0$, obtained by rejection sampling: test S on fresh tasks and keep those it misses. It removes a candidate when its upper confidence bound is more than η below the largest lower confidence bound. It stops once one candidate remains or the confidence intervals are sufficiently narrow. This allows clearly worse candidates to be removed early, without requiring the algorithm to distinguish candidates whose values are nearly equal.

Algorithm 2 Conditional-Elim(S, A, η, δ): η-optimal selection on tasks missed by S

Input: committee S with $\rho(S) > 0$; nonempty candidate set $A \subseteq C \setminus S$; accuracy $\eta \in (0,1]$; error probability $\delta \in (0,1)$.
Output: a candidate $\widehat{c} \in A$.

$A_1 \leftarrow A$; initialize $\widehat{q}_0(c) \leftarrow 0$ for all $c \in A$.
for $r = 1, 2, \ldots$ **do**
  Draw $d_r \sim \mathcal{D}$ conditioned on $U(d_r, S) = 0$ (via rejection sampling).
  for all $c \in A_r$ **do**
    Query $X_{r,c} \leftarrow u(d_r, c) \in \{0,1\}$ and set $\widehat{q}_r(c) \leftarrow \frac{(r-1)\widehat{q}_{r-1}(c) + X_{r,c}}{r}$.
  end for
  $\mathrm{rad}_r \leftarrow \sqrt{\frac{1}{2r}\log(4|A|r^2/\delta)}$.
  Let $c_r^* \in \arg\max_{c\in A_r} \widehat{q}_r(c)$.
  $A_{r+1} \leftarrow \left\{c \in A_r : \widehat{q}_r(c) + \mathrm{rad}_r \ge \widehat{q}_r(c_r^*) - \mathrm{rad}_r - \eta\right\}$.
  if $\mathrm{rad}_r \le \eta/4$ **or** $|A_{r+1}| = 1$ **then**
    return any $\widehat{c} \in \arg\max_{c\in A_{r+1}} \widehat{q}_r(c)$.
  end if
end for

The next theorem makes this advantage precise. The total number of retained tasks is determined by the requested accuracy, whereas a candidate with a large gap is queried on only a few of them.

Theorem B.4. *Fix $\eta \in (0, 1]$ and $\delta \in (0, 1)$. The routine uses $O(\log(e|A|/(\delta\eta))/\eta^2)$ retained tasks in the worst case. With probability at least $1 - \delta$, $q(\widehat{c} \mid S) \geq q^* - \eta$, and the number of candidate queries is $Q_{\mathrm{cand}} = O(\sum_{c \in A} \log(e|A|/(\delta\eta))/\max\{\Delta_q(c), \eta\}^2)$.*

Proof. We first show that accurate confidence intervals keep an optimal candidate in the active set. We then bound how long each other candidate can remain. For the analysis, consider independent tasks $d_1, d_2, \ldots$ from $\mathcal{D} \mid U(d, S) = 0$, including all candidate outcomes whether queried or not. For fixed c, let $\widehat{q}_r(c) := r^{-1} \sum_{s=1}^{r} u(d_s, c)$. Since active sets only shrink, a candidate queried in round r was also queried in every earlier round. This expression is therefore exactly its observed empirical mean whenever it is active. With $\mathrm{rad}_r := \sqrt{\log(4|A|r^2/\delta)/(2r)}$, Hoeffding's inequality and a union bound imply

$$\Pr[\exists c \in A,\ r \geq 1 : |\widehat{q}_r(c) - q(c \mid S)| > \mathrm{rad}_r] \leq \sum_{c \in A} \sum_{r \geq 1} \frac{\delta}{2|A|r^2} < \delta.$$

On the complementary event, all confidence intervals are accurate. Fix $c^* \in \arg\max_{c \in A} q(c \mid S)$. This candidate is never eliminated: its upper confidence bound is at least q^*, whereas every candidate's lower confidence bound is at most q^*. Moreover, the empirical maximizer always remains after the update, so the active set is nonempty even when some confidence intervals are inaccurate.

If only one candidate remains, it must be optimal on the concentration event. Otherwise, the algorithm stops once $\mathrm{rad}_r \leq \eta/4$ and returns an empirical maximizer among the remaining candidates. Because c^* is among them,

$$q(\widehat{c} \mid S) \geq \widehat{q}_r(\widehat{c}) - \mathrm{rad}_r \geq \widehat{q}_r(c^*) - \mathrm{rad}_r \geq q^* - 2\mathrm{rad}_r \geq q^* - \eta.$$

To bound the time until stopping, take $r = \lceil K \log(e|A|/(\delta\eta))/\eta^2 \rceil$ for a sufficiently large universal constant K. Then $\log r = O(\log(e|A|/(\delta\eta)))$, and substitution in the confidence radius gives $\mathrm{rad}_r \leq \eta/4$. The algorithm therefore uses at most this many retained tasks on every outcome.

A candidate with a large gap can be eliminated sooner. Fix c with $\Delta_q(c) \geq 2\eta$. Once $\mathrm{rad}_r < \Delta_q(c)/8$,

$$\widehat{q}_r(c) + \mathrm{rad}_r \leq q^* - \Delta_q(c) + 2\mathrm{rad}_r < q^* - \eta - 2\mathrm{rad}_r \leq \widehat{q}_r(c^*) - \mathrm{rad}_r - \eta,$$

so the elimination rule removes c. The same confidence-radius calculation bounds its observations by $O(\log(e|A|/(\delta\eta))/\Delta_q(c)^2)$. For $\Delta_q(c) < 2\eta$, we use the overall stopping bound. It differs by at most a constant factor from the claimed expression with denominator $\max\{\Delta_q(c), \eta\}^2$. Summing over candidates proves the query bound. The proof uses independence across tasks but does not require independence among candidates on the same task. □

B.4 Proof of Theorem 3.2

We now analyze Algorithm 1. The analysis has two parts. For coverage, we show that the miss-rate estimate sets a suitable accuracy for each conditional selection call. For query cost, we count both candidate evaluations and the committee evaluations needed to obtain missed tasks. Both parts rely on one structural fact: using the draw count to choose an accuracy does not change the distribution of the retained tasks.

Reusing the stored missed tasks. To establish this fact, fix S with $\rho(S) > 0$, draw independent tasks, and let $N_r(S)$ be the number of draws needed to observe r tasks with $U(d, S) = 0$.

Conditional on the entire sequence of acceptance indicators $\mathbf{1}\{U(d_t, S) = 0\}$, the retained task profiles are independent and each has law $\mathcal{D} \mid U(d, S) = 0$. This law does not depend on when the misses occurred. In particular, conditioning on $N_{r_0}(S)$, or on whether it exceeds M_0, leaves the distribution of the retained profiles unchanged. The algorithm may therefore use the draw count to choose its accuracy, query the stored missed tasks in order, and then continue with fresh retained tasks. Section 2 permits these later queries to stored tasks.

Accuracy of the miss-rate estimate. The preliminary test estimates the miss probability from the number of draws needed to see r misses. With high probability, this number is within a factor of

two of $r/\rho(S)$: for every integer $r \geq 1$,

$$\Pr\left[N_r(S) < \frac{r}{2\rho(S)}\right] \leq e^{-r/6}, \quad \Pr\left[N_r(S) > \frac{2r}{\rho(S)}\right] \leq e^{-r/4}. \tag{5}$$

For the first inequality, the expected number of misses after $\lfloor r/(2\rho(S))\rfloor$ draws is at most $r/2$. The event in question requires at least r misses by then, so the upper-tail Chernoff bound gives $e^{-r/6}$. If the number of draws is zero, the event is empty. For the second inequality, the expected number of misses after $\lceil 2r/\rho(S)\rceil$ draws is at least $2r$. The event in question allows at most r misses by that time; equality includes the possibility that the rth miss occurs on the last draw. The lower-tail Chernoff bound therefore gives $e^{-r/4}$. Combining the two inequalities yields

$$\Pr\left[\rho(S) \leq \frac{2r}{N_r(S)} \leq 4\rho(S)\right] \geq 1 - 2e^{-r/6}.$$

Correctness. Consider a step i that the algorithm reaches, and condition on all earlier observations, so that S_i is fixed. If $\rho_i = 0$, the algorithm stops after M_0 draws. Otherwise, the choice of r_0 makes the preceding estimate valid for S_i and $r = r_0$ with conditional probability at least $1 - \delta/(4k)$. On this event, an early stop means that few tasks remain uncovered. More precisely, it implies $\rho_i \leq \varepsilon/k$: if $\rho_i > \varepsilon/k$, then $N_{r_0}(S_i) \leq 2r_0/\rho_i < 2r_0k/\varepsilon$, and integrality would imply $N_{r_0}(S_i) \leq M_0$, contradicting the stop. If the step continues, then $N_i = N_{r_0}(S_i) \leq M_0$, so

$$\rho_i \leq \widetilde{\rho}_i \leq 4\rho_i, \quad \widetilde{\rho}_i \geq \varepsilon/k, \quad \frac{\varepsilon}{4k\rho_i} \leq \eta_i \leq \frac{\varepsilon}{k\rho_i}.$$

In particular, $\eta_i \in (0, 1]$.

To apply the conditional selection theorem, condition further on the acceptance indicators from the preliminary test, without conditioning on unqueried candidate outcomes. This fixes η_i, while the stored tasks and subsequent retained tasks have the conditional distribution established above. Theorem B.4 thus applies with error probability $\delta/(4k)$. There are at most k preliminary tests and k selection calls, so a union bound gives probability at least $1 - \delta/2$ that all estimates and selections satisfy their bounds.

On this joint event, an early stop followed by padding gives $v(\widehat{S}) \geq 1 - \varepsilon/k \geq \mathrm{OPT}_k - \varepsilon$. If the algorithm completes all k selections, each addition loses at most ε/k relative to the largest marginal gain:

$$\Delta(\widehat{c}_{i+1} \mid S_i) \geq \max_{c \notin S_i} \Delta(c \mid S_i) - \rho_i\eta_i \geq \max_{c \notin S_i} \Delta(c \mid S_i) - \varepsilon/k.$$

To translate this local comparison into a coverage guarantee, let $S^* \in \mathcal{S}_k$ be optimal. Submodularity bounds the remaining difference from its value by

$$\mathrm{OPT}_k - v(S_i) \leq \sum_{c \in S^* \setminus S_i} \Delta(c \mid S_i) \leq k \max_{c \notin S_i} \Delta(c \mid S_i).$$

Thus each selection multiplies the remaining difference by at most $1 - 1/k$, apart from the additive error ε/k. Iterating this recurrence gives

$$\mathrm{OPT}_k - v(S_k) \leq (1 - 1/k)^k \mathrm{OPT}_k + \frac{\varepsilon}{k} \sum_{j=0}^{k-1} (1 - 1/k)^j \leq e^{-1}\mathrm{OPT}_k + \varepsilon.$$

This proves the approximation guarantee.

Candidate-query count. We first count queries to possible additions. At each step that performs conditional selection, candidate c has conditional gap $\Delta_q(c) = \Delta_i(c)/\rho_i$. The lower bounds $\eta_i \geq \varepsilon/(4k\rho_i)$ and $\eta_i \geq \varepsilon/k$ allow us to express the cost from Theorem B.4 in terms of the marginal gaps:

$$Q^{(i)}_{\mathrm{cand}} = O\left(\sum_{c \notin S_i} \frac{\log(emk^2/(\delta\varepsilon))}{\max\{\Delta_i(c)/\rho_i, \varepsilon/(4k\rho_i)\}^2}\right) = O\left(\sum_{c \notin S_i} \frac{\rho_i^2 \log(emk^2/(\delta\varepsilon))}{\max\{\Delta_i(c), \varepsilon/k\}^2}\right).$$

Summing over selection steps gives the candidate-query term in the bound of Theorem 3.2. The preliminary test may retain more missed tasks than the selection call needs. No candidate queries are made on these unused tasks; the committee queries spent finding them are included next.

A high-probability bound for all queries. We now count the committee evaluations used to obtain missed tasks. Fix a step reached by the algorithm and condition on the preceding history. When the preliminary estimate is accurate and the step continues, we have $\rho_i \geq \varepsilon/(4k)$, and the preliminary test uses at most $2r_0/\rho_i$ draws. Conditional on the preliminary test, the deterministic stopping bound for elimination is

$$R_i = O\left(\frac{\widetilde{\rho}_i^2 k^2}{\varepsilon^2} \log \frac{emk^2}{\delta\varepsilon}\right)$$

retained tasks. For an upper bound on further draws, we may ignore the stored tasks and count the draws needed to obtain R_i new missed tasks.

The number of draws needed to obtain a fixed number of misses also concentrates. For any positive integer R and miss probability $\rho > 0$, a Chernoff bound gives

$$\Pr\left[N_R > \left\lceil \frac{8(R + \log(4k/\delta))}{\rho} \right\rceil\right] \leq \frac{\delta}{4k}.$$

Indeed, the number of misses in the indicated number of draws has mean at least $8(R + \log(4k/\delta))$, and the Chernoff lower-tail bound limits the probability of seeing fewer than R of them as claimed. Subsequent draws are independent of the preliminary test, so the bound still applies after conditioning on that test.

Substituting the bound on R_i and using $\widetilde{\rho}_i \leq 4\rho_i$ and $\rho_i \geq \varepsilon/(4k)$, we can bound both the preliminary draws and the further draws by

$$O\left(\frac{\rho_i k^2}{\varepsilon^2} \log \frac{emk^2}{\delta\varepsilon}\right)$$

at every continued step, with an additional probability loss of at most $\delta/4$ after a union bound over steps. Each draw requires at most i committee queries. The algorithm can also stop during one preliminary test; that test uses at most M_0 draws and at most $kM_0 = O(k^2\varepsilon^{-1}\log(ek/\delta))$ committee queries. Adding these committee costs to the candidate-query bound proves the query bound of Theorem 3.2. The approximation and query guarantees hold together with probability at least $1 - \delta$. Reusing previously observed answers can only reduce these counts.

Expected task draws and committee-evaluation cost. The high-probability analysis does not by itself bound the expectation, because it leaves open the cost when an estimate is inaccurate. We therefore give a separate expectation bound over all outcomes. Condition on the history before a reached step and fix the committee-testing order σ_i. The preliminary test uses $\min\{N_{r_0}(S_i), M_0\}$ draws, whose expectation is at most

$$\min\{r_0/\rho_i, M_0\} = O\left(\frac{\log(ek/\delta)}{\max\{\rho_i, \varepsilon/k\}}\right).$$

For $\rho_i = 0$, use the bound M_0 directly; no further draws occur.

Suppose $\rho_i > 0$ and write $N = N_{r_0}(S_i)$. Conditional on the preliminary draw count, Theorem B.4 bounds the total number of retained tasks by $O(\widetilde{\rho}_i^2 k^2 \log(emk^2/(\delta\varepsilon))/\varepsilon^2)$. Obtaining this many new retained tasks requires their number divided by ρ_i fresh draws in expectation, which is an upper bound even without reusing the stored tasks. It remains to average this expression over the estimated miss rate. Since $r_0 > 2$, the negative-binomial law for N gives

$$\mathbb{E}\left[\frac{1}{(N-1)(N-2)}\right] = \sum_{n=r_0}^{\infty} \frac{\binom{n-1}{r_0-1}\rho_i^{r_0}(1-\rho_i)^{n-r_0}}{(n-1)(n-2)} = \frac{\rho_i^2}{(r_0-1)(r_0-2)}.$$

For the last equality, cancel the two factors in the denominator and recognize the sum of a negative-binomial probability mass function with $r_0 - 2$ required misses. This identity controls the squared estimate of the miss rate: because $N^{-2} \leq ((N-1)(N-2))^{-1}$,

$$\mathbb{E}\left[\mathbf{1}\{N \leq M_0\} \min\{1, 2r_0/N\}^2\right] \leq \frac{4r_0^2\rho_i^2}{(r_0-1)(r_0-2)} = O(\rho_i^2).$$

Thus the expected number of further draws is $O(\rho_i k^2 \log(emk^2/(\delta\varepsilon))/\varepsilon^2)$, without conditioning on a high-probability event.

Finally, the order of committee evaluation may make a draw cheaper than testing all i members. For a fixed order σ_i, let $Q_{\mathrm{eval}}(d, S_i, \sigma_i)$ be the number of members queried up to the first correct answer or until all members have been tested. This cost lies between zero and i. The task draws are independent, and the decision to request the next draw depends only on earlier observations. Wald's identity therefore converts the preceding draw bounds into query bounds by multiplying them by $\mathbb{E}[Q_{\mathrm{eval}}(d, S_i, \sigma_i)]$. Alternatively, we may assume that after the preliminary test the algorithm always obtains the deterministic maximum number of retained tasks, which can only increase the cost. Conditional on the history before step i, the expected number of committee queries is consequently

$$O\left(\mathbb{E}[Q_{\mathrm{eval}}(d, S_i, \sigma_i)]\left[\frac{\log(ek/\delta)}{\max\{\rho_i, \varepsilon/k\}} + \frac{\rho_i k^2 \log(emk^2/(\delta\varepsilon))}{\varepsilon^2}\right]\right).$$

Removing the factor $\mathbb{E}[Q_{\mathrm{eval}}(d, S_i, \sigma_i)]$ from this expression gives the bound on task draws. Both expectation bounds include outcomes on which the confidence estimates are inaccurate.

B.5 Lower Bound for Conditional Selection

The query bound of Theorem B.4 charges each candidate the inverse square of its gap, down to the requested accuracy. We now show that this charge is necessary for candidates whose gaps exceed the accuracy, even when missed tasks are supplied directly. Fix S with $\rho(S) > 0$ and a nonempty candidate set $A \subseteq C \setminus S$, and let q^* and Δ_q be as in Appendix B.3.

Theorem B.5. *Fix $\eta \in (0, 1)$, $\delta \in (0, 1/4]$, and gaps in $(0, 1/4]$ for all but one candidate in A. Any algorithm that, for every conditional outcome distribution, returns $\widehat{c} \in A$ satisfying $q(\widehat{c} \mid S) \geq q^* - \eta$ with probability at least $1 - \delta$ has an instance with these gaps on which $\mathbb{E}[Q_{\mathrm{cand}}] = \Omega(\sum_{a \in A : \Delta_q(a) > \eta} \log(1/\delta)/\Delta_q(a)^2)$.*

Proof. We construct one distribution with prescribed candidate gaps, then change the mean of each candidate whose gap exceeds η in turn. A uniformly correct algorithm must query each such candidate often enough to distinguish the original distribution from the one where that candidate becomes the only acceptable choice.

Let c^* be the candidate without a prescribed gap, which will be optimal, and let $\gamma_a \in (0, 1/4]$ be the prescribed gap of each $a \in A \setminus \{c^*\}$. The bound is nonzero when at least one gap exceeds η. Set every outcome outside A to zero. On each sampled task, independently across candidates, let

$$u(d, c^*) \sim \mathrm{Bernoulli}(1/2), \quad u(d, a) \sim \mathrm{Bernoulli}(1/2 - \gamma_a) \quad (a \neq c^*).$$

Because $A \cap S = \varnothing$, the committee misses every task, so this is a valid conditional instance with $\rho(S) = 1$, $q^* = 1/2$, and $\Delta_q(a) = \gamma_a$.

For each c with $\gamma_c > \eta$, consider the alternative distribution that changes only its mean to $1/2 + \gamma_c$. Under this alternative, c is the only candidate within η of optimal; under the original distribution, it is more than η below optimal. Let P_0, P_c be the laws of the algorithm's complete observations and output in these two cases, and let $E_c := \{\widehat{c} = c\}$. Correctness on both distributions requires $P_0(E_c) \leq \delta$ and $P_c(E_c) \geq 1 - \delta$. The Bretagnolle–Huber and Pinsker inequalities give, respectively,

$$\mathrm{KL}(P_0 \| P_c) \geq \log(1/(4\delta)), \quad \mathrm{KL}(P_0 \| P_c) \geq 2(1 - 2\delta)^2 \geq 1/2.$$

The first bound grows as δ decreases, and the second remains positive throughout the stated range. Together they imply $\mathrm{KL}(P_0 \| P_c) \geq \log(1/\delta)/(2\log 4 + 1)$ for $\delta \leq 1/4$.

Only queries to c can distinguish the original distribution from this alternative. Let N_c count these queries, including repetitions. The adaptive chain rule, justified by truncation as in Theorem B.2, gives

$$\mathrm{KL}(P_0 \| P_c) \leq \mathbb{E}_0[N_c]\, \mathrm{KL}\left(\mathrm{Bernoulli}(1/2 - \gamma_c) \,\|\, \mathrm{Bernoulli}(1/2 + \gamma_c)\right) \leq \tfrac{64}{3}\gamma_c^2 \mathbb{E}_0[N_c].$$

Indeed, queries to other candidates have identical conditional laws, and repeated queries reveal nothing new. The Bernoulli bound follows from $\mathrm{KL}(\mathrm{Bernoulli}(p) \| \mathrm{Bernoulli}(q)) \leq (p-q)^2/[q(1-q)]$ and $q(1-q) \geq 3/16$. It follows that $\mathbb{E}_0[N_c] = \Omega(\log(1/\delta)/\gamma_c^2)$ for every c with $\gamma_c > \eta$. Since all these bounds concern the same original distribution, we may sum them. Using $Q_{\mathrm{cand}} \geq \sum_{c \in A} N_c$ then proves the theorem. □

C PAIRWISE FEEDBACK: SINGLE COMMITTEES

This appendix studies a single fixed committee. We first show how cyclic rankings affect comparisons with other committees and what a guarantee against individual rivals implies. We then prove the approximation scheme, the lower bounds, and the procedures for evaluating and selecting single committees.

C.1 COMPARING WITH OTHER COMMITTEES

Cyclic rankings. Let $m \geq 2k$, and put equal probability on the m cyclic rotations of $1 \succ 2 \succ \cdots \succ m$. Fix any $S \in \mathcal{S}_k$. For each maximal consecutive block of members of S around the cycle, put the candidate immediately preceding that block in S'. There are at most k such candidates, all outside S, so we can add omitted candidates until $|S'| = k$. Whenever the first candidate in a ranking lies outside S, one of these predecessors appears before the first member of S. Since S and S' are disjoint, their best members cannot tie. Thus S wins only when the ranking starts with a member of S, giving

$$\mathrm{WIN}(S, S') = \Pr[r_\pi(S) < r_\pi(S')] \leq k/m.$$

The competing committee S' therefore wins on at least a $1 - k/m$ fraction of tasks. For fixed k, this fraction approaches one as m grows.

From individual rivals to competing committees. The winning value compares a committee with each individual rival. What does this imply about its win rate against another committee? If the other committee wins, at least one of its members must beat every selected candidate. A union bound therefore gives the following guarantee, which becomes weaker as the competing committee grows.

Lemma C.1. *If $S \subsetneq C$ is nonempty, then $\Pr_{\pi \sim P_{\mathcal{D}}}[r_\pi(S) \leq r_\pi(S')] \geq 1 - |S'|(1 - \theta(S))$ for every committee $S' \subseteq C$. If $S \cap S' = \varnothing$, the same bound holds for $\mathrm{WIN}(S, S')$.*

Proof. We bound the probability that S' has a better-ranked member than S. Whenever $r_\pi(S) > r_\pi(S')$, the best candidate in S' belongs to $S' \setminus S$ and ranks above every member of S. Thus

$$\{r_\pi(S) > r_\pi(S')\} \subseteq \bigcup_{x \in S' \setminus S} \{\mathrm{rank}_\pi(x) < r_\pi(S)\}.$$

Because the rankings are strict, each event on the right has probability at most $1 - \theta(S)$. Taking a union bound yields

$$\Pr[r_\pi(S) > r_\pi(S')] \leq |S' \setminus S|(1 - \theta(S)) \leq |S'|(1 - \theta(S)).$$

Taking complements proves the first claim. If S and S' are disjoint and nonempty, their best ranks cannot tie. They also cannot tie when $S' = \varnothing$, since $r_\pi(S') = +\infty$ and $S \neq \varnothing$. The probability in the lemma is therefore $\mathrm{WIN}(S, S')$ whenever the committees are disjoint. □

C.2 MONOTONICITY AND NON-SUBMODULARITY

Adding candidates cannot hurt the winning value, but the minimum over rivals can destroy diminishing returns. The following example explains why the binary greedy guarantee does not apply directly.

Proposition C.2. *On nonempty proper committees, θ is monotone but not submodular.*

Proof. Adding members helps in two ways: it cannot decrease the probability of beating any remaining rival, and it removes the new members from the set of rivals. Formally, if $S \subseteq S' \subsetneq C$, then every comparison against $x \notin S'$ can only improve, and $C \setminus S' \subseteq C \setminus S$. Taking the minimum therefore gives $\theta(S') \geq \theta(S)$.

To see why diminishing returns need not hold, take $C = \{a, b, c, d\}$ and assign probability $2/3$ to $b \succ a \succ d \succ c$ and $1/3$ to $c \succ a \succ d \succ b$. The resulting values are

$$\theta(\{a\}) = \theta(\{a, c\}) = 1/3, \quad \theta(\{a, b\}) = 2/3, \quad \theta(\{a, b, c\}) = 1.$$

Adding c to $\{a\}$ does not improve the value because b remains the hardest rival to beat. After b has been added, however, adding c increases the value by $1/3$. All four committees are proper, so this example violates diminishing returns within the stated domain. □

C.3 Proof of Theorem 4.2

Throughout this subsection, $P_{\mathcal{D}}$ is uniform over the given finite profile $(\pi_1, \ldots, \pi_n)$, and θ, WIN, and θ_k^* retain their definitions. Both parts of the proof use the fact that adding candidates cannot decrease θ: every remaining rival becomes at least as easy to beat, and added candidates no longer count as rivals. The approximation scheme uses this monotonicity to pad a small committee. The lower bound constructs profiles in which every sufficiently good committee encodes a good solution to a maximum-coverage problem.

Proof. **Approximation scheme.** Let $s = \min\{k, \lceil 9.8217/\gamma \rceil\}$. Theorem 2 of Charikar et al. (2025) implies that every profile has a committee of at most $\lceil 9.8217/\gamma \rceil$ members with value at least $1-\gamma$, whenever $\lceil 9.8217/\gamma \rceil < m$. Thus, once the permitted committee size reaches this number, it is enough to find one of these small committees and add members.

If $s = k$, enumerate all size-k committees and return an optimum. Otherwise, enumerate all size-s committees, select one with maximum θ, and add arbitrary candidates until its size is k. Since $s < k < m$, the existence result applies, and monotonicity gives $\theta(S) \geq 1-\gamma \geq (1-\gamma)\theta_k^*$. In either case, we enumerate committees of size $s = O(1/\gamma)$ and evaluate each by scanning the n rankings. The total running time is $m^{O(1/\gamma)}\mathrm{poly}(m, n)$.

No efficient approximation scheme. We show that an efficient approximation scheme would solve a parameterized maximum-coverage problem in a running time ruled out by Gap-ETH. We use the promise version of Max κ-Coverage from Manurangsi (2020). As in that work and in Section 4.2, Gap-ETH refers to its randomized form.

Let N be the encoding size of the Max κ-Coverage instance. There is a constant $\eta \in (0, 1)$ such that, under Gap-ETH, no algorithm with running time $f(\kappa)\mathrm{poly}(N)$, for any computable f, can distinguish the following two cases. The input consists of a universe $\mathcal{U} = \{e_1, \ldots, e_h\}$, a family of sets $\mathcal{F} = \{F_c : c \in B\}$, and an integer κ. The two cases are

$$\textsc{Yes}: \quad \exists Y \subseteq B,\ |Y| \leq \kappa, \quad \bigcup_{c \in Y} F_c = \mathcal{U},$$

and

$$\textsc{No}: \quad \forall Y \subseteq B,\ |Y| \leq \kappa, \quad \left| \bigcup_{c \in Y} F_c \right| \leq (1-\eta)h.$$

For each element e_i, let B_i denote the set of indices of the sets containing it: $B_i := \{c \in B : e_i \in F_c\}$.

We may assume $h \geq 1$ and $1 \leq \kappa \leq |B|$, so κ is bounded by the input size. We now turn such an instance into a finite ranking profile. Candidates in B represent the original sets. The additional candidates make a high winning value possible only when the selected set candidates cover many elements. Let $q := \kappa + 3$ and $D := \{a_1, \ldots, a_q\}$, and introduce one additional candidate b. The candidate set is the disjoint union $C := B \dot\cup D \dot\cup \{b\}$. We set the committee size to $k := \kappa + 1$. The profile has two types of rankings. We use block notation: $X \succ Y$ means that every member of X is ranked above every member of Y, with an arbitrary but fixed order inside each block.

First, for every $i \in [h]$ and every $j \in [q]$, include one type-I ranking

$$B_i \succ a_j \succ D \setminus \{a_j\} \succ B \setminus B_i \succ b.$$

Second, for every $j \in [q]$, include hq copies of the type-II ranking

$$a_j \succ b \succ B \succ D \setminus \{a_j\}.$$

Thus the total number of rankings is $n = hq + q \cdot hq = hq(q+1)$. The construction has size polynomial in N. To understand the two types of rankings, consider a committee that omits b. A type-II ranking then prefers b to the committee unless the committee contains the auxiliary candidate a_j ranked first. A committee containing b, meanwhile, has only κ places left for set candidates. Each element left uncovered by these sets gives a type-I ranking on which an omitted auxiliary candidate wins. We make these two observations precise below.

Yes instances. Suppose there is $Y \subseteq B$, $|Y| \leq \kappa$, covering all elements of $\mathcal{U}$. Consider $S_0 := Y \cup \{b\}$, and add arbitrary candidates to obtain a committee $S \in \mathcal{S}_k$. By monotonicity, it is enough to bound $\theta(S_0)$ from below. We do so separately for set candidates and auxiliary candidates outside S_0.

For any $c \in B \setminus Y$, the committee member b ranks above c in every type-II ranking. These hq^2 rankings alone give

$$\mathrm{WIN}(S_0, \{c\}) \geq \frac{hq^2}{n} = \frac{q}{q+1}.$$

For any $a_j \in D \setminus S_0$, the committee wins on every type-I ranking: Y covers each element e_i, so some candidate in $Y \cap B_i$ ranks above a_j. Among the type-II rankings, b also ranks above a_j except in the hq copies that place a_j first. Combining these counts gives

$$\mathrm{WIN}(S_0, \{a_j\}) \geq \frac{hq + (q-1)hq}{hq(q+1)} = \frac{q}{q+1}.$$

These are all candidates outside S_0. Hence $\theta(S_0) \geq q/(q+1)$, and extending the committee preserves this lower bound:

$$\theta_k^* \geq \theta(S) \geq \theta(S_0) \geq \frac{q}{q+1}.$$

No instances. Now suppose every choice of at most κ sets leaves at least ηh elements uncovered. We show that every committee $S \in \mathcal{S}_k$ has strictly smaller winning value, whether or not it contains b.

If $b \notin S$, we can use b itself as the rival. The committee beats b in all hq type-I rankings, since b ranks last there. In a type-II ranking, however, the committee beats b only if it contains the first candidate a_j. Because $|S| = k$, this happens in at most khq type-II rankings. Consequently,

$$\mathrm{WIN}(S, \{b\}) \leq \frac{hq + khq}{n} = \frac{1+k}{q+1}.$$

Because $k = \kappa + 1 = q - 2$, we get

$$\mathrm{WIN}(S, \{b\}) \leq \frac{q-1}{q+1} \leq \frac{q}{q+1} - \frac{\eta}{q(q+1)},$$

where the last inequality uses $\eta \leq 1$.

If $b \in S$, we instead compare S with an omitted auxiliary candidate. Let $Y := S \cap B$. The committee has only κ places besides b, so $|Y| \leq \kappa$. The No-instance promise then gives at least ηh indices i for which $Y \cap B_i = \varnothing$. There is also an auxiliary candidate $a_j \in D \setminus S$, since $|S \cap D| \leq \kappa$ and $|D| = \kappa + 3$.

Fix one such a_j. It beats every committee member in the hq type-II rankings that place it first. For each element e_i left uncovered by Y, it also beats the committee in the type-I ranking for the pair (i, j). Indeed, no member of Y lies in B_i, every selected auxiliary candidate belongs to $D \setminus \{a_j\}$, and b ranks last. Thus a_j beats the committee in at least $hq + \eta h$ rankings, which implies

$$\mathrm{WIN}(S, \{a_j\}) \leq 1 - \frac{hq + \eta h}{hq(q+1)} = \frac{q}{q+1} - \frac{\eta}{q(q+1)}.$$

The same upper bound holds in both cases. Therefore every $S \in \mathcal{S}_k$ in a No instance satisfies

$$\theta(S) \leq \frac{q}{q+1} - \frac{\eta}{q(q+1)}.$$

Deriving the contradiction. The Yes-instance lower bound exceeds the No-instance upper bound by $\eta/[q(q+1)]$. Suppose an efficient polynomial-time approximation scheme (EPTAS) exists, and choose its accuracy so that its possible loss is smaller than this gap. Specifically, run it on the constructed profile with $\gamma := \frac{\eta}{2q^2}$. Its running time is

$$f(1/\gamma)\mathrm{poly}(m, n) = f(2q^2/\eta)\mathrm{poly}(m, n) = f'(\kappa)\mathrm{poly}(N),$$

because $q = \kappa + 3$, η is constant, and the constructed values of m and n are polynomially bounded in N.

Let $\widehat{S} \in \mathcal{S}_k$ be the returned committee. In a Yes instance, the approximation guarantee gives

$$\theta(\widehat{S}) \geq (1-\gamma)\theta_k^* \geq (1-\gamma)\frac{q}{q+1} = \frac{q}{q+1} - \frac{\eta}{2q(q+1)} > \frac{q}{q+1} - \frac{\eta}{q(q+1)}.$$

In a No instance, the upper bound above applies to $\widehat{S}$. We can compute $\theta(\widehat{S})$ exactly from the complete rankings in polynomial time. Comparing it with $q/(q+1) - \eta/[q(q+1)]$ therefore distinguishes the Yes and No cases of Gap Max κ-Coverage in $f'(\kappa)\mathrm{poly}(N)$ time. This contradicts the stated consequence of Gap-ETH and rules out an EPTAS under that assumption. □

C.4 Selection from Sampled Rankings

The full-information algorithm also works when the distribution is unknown, once we recover a sample of complete rankings. Let $s = \min\{k, \lceil 9.8217/\gamma \rceil\}$ as in Appendix C.3, and draw

$$T = \left\lceil \frac{2}{\varepsilon^2} \log \frac{2(m-s)\binom{m}{s}}{\delta} \right\rceil \tag{6}$$

independent tasks. Recover their rankings by comparison sorting, choose a size-s committee of maximum empirical winning value, and add arbitrary members until its size is k.

Corollary C.3. *The preceding procedure returns $\widehat{S} \in \mathcal{S}_k$ with $\theta(\widehat{S}) \geq (1-\gamma)\theta_k^* - \varepsilon$ with probability at least $1-\delta$, using $O(Tm \log m)$ comparisons.*

Since $s = O(1/\gamma)$, the sample size is $T = O(\varepsilon^{-2}(\gamma^{-1}\log m + \log(1/\delta)))$. Beyond the full-information argument, the proof must ensure that all committees considered by the algorithm have empirical values close to their population values.

Proof. Let s and T be as in (6), and let $\widehat{\theta}_T$ be the winning value under the uniform distribution over the T sampled rankings. Hoeffding's inequality and a union bound over the $(m-s)\binom{m}{s}$ size-s committee–rival pairs give $|\widehat{\theta}_T(S) - \theta(S)| \leq \varepsilon/2$ for every $S \in \mathcal{S}_s$ simultaneously, with probability at least $1-\delta$. We work on this event. As in the full-information algorithm, select a size-s committee with maximum empirical value and add arbitrary candidates to obtain $\widehat{S} \in \mathcal{S}_k$.

If $k < \lceil 9.8217/\gamma \rceil$, then $s = k$. The empirical value of an optimal committee is at most $\varepsilon/2$ below its population value, and the selected committee has at least that empirical value. Allowing another $\varepsilon/2$ when returning to population values gives $\theta(\widehat{S}) \geq \theta_k^* - \varepsilon$.

Otherwise, Theorem 2 of Charikar et al. (2025), together with monotonicity, gives a size-s committee whose value on the empirical profile is at least $1-\gamma$. The selected size-s committee therefore has population value at least $1-\gamma-\varepsilon/2$. Adding members cannot decrease θ, and $1-\gamma-\varepsilon/2 \geq (1-\gamma)\theta_k^* - \varepsilon$. This proves the value guarantee in both cases. Recovering the T rankings by comparison sorting costs $O(Tm \log m)$ queries. Once the rankings are known, enumerating all size-s committees and evaluating them takes $m^{O(s)}\mathrm{poly}(m, T)$ time and requires no further queries. □

C.5 Full Evaluation and Pairwise Lower Bounds

Recovering complete rankings provides a useful starting point for assessing the query cost of selection. Let $\widehat{S}_{\mathrm{ERM}}$ maximize over $\mathcal{S}_k$ the empirical winning value $\widehat{\theta}_T$ of T independent rankings. The next theorem bounds the number of comparisons needed for this empirical maximizer to have nearly optimal population value. Computing the maximizer itself may require enumeration.

Theorem C.4. *If $T \geq \frac{2}{\varepsilon^2}\log(2(m-k)\binom{m}{k}/\delta)$, then, with probability at least $1-\delta$, $\theta(\widehat{S}_{\mathrm{ERM}}) \geq \theta_k^* - \varepsilon$. The procedure uses $O(Tm \log m)$ pairwise comparisons.*

Proof. We first estimate every committee's win probability against each of its rivals. For $S \in \mathcal{S}_k$ and $x \notin S$, define

$$\widehat{w}_T(S, x) := \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\{r_{\pi_t}(S) < \mathrm{rank}_{\pi_t}(x)\}.$$

Each summand has mean $\mathrm{WIN}(S, \{x\})$. Hoeffding's inequality and a union bound over the $(m - k)\binom{m}{k}$ pairs show that, at the stated sample size, every estimate is accurate to within $\varepsilon/2$ with probability at least $1 - \delta$. Taking the minimum over rivals preserves this error bound, so $|\widehat{\theta}_T(S) - \theta(S)| \le \varepsilon/2$ for every $S \in \mathcal{S}_k$ on the same event.

Let S^* be a committee with population value θ_k^*. The empirical maximizer does at least as well as S^* on the sample. Accounting for the estimation error of each committee gives

$$\theta(\widehat{S}_{\mathrm{ERM}}) \ge \widehat{\theta}_T(\widehat{S}_{\mathrm{ERM}}) - \varepsilon/2 \ge \widehat{\theta}_T(S^*) - \varepsilon/2 \ge \theta_k^* - \varepsilon.$$

To obtain the sample, comparison sorting recovers each strict ranking using $O(m \log m)$ queries. All committee–rival values can then be computed from these rankings, so the total query cost is $O(Tm \log m)$. □

How much can adaptive querying improve on full ranking recovery? For singleton committees, by at most logarithmic factors in the worst case: the following lower bounds show that even singletons can require order m/ε^2 comparisons, both to estimate a specified committee's value and to select a nearly optimal one.

Theorem C.5. *Assume $m \ge 2$ and $0 < \varepsilon \le 1/16$. Then,*

(i) For a fixed singleton committee $S = \{s\}$, any adaptive algorithm that estimates $\theta(S)$ to additive error ε with probability at least $2/3$ on every ranking distribution has worst-case expected pairwise query complexity $\Omega((m-1)/\varepsilon^2)$.

(ii) Any adaptive algorithm that returns $\widehat{c} \in C$ with $\theta(\{\widehat{c}\}) \ge \theta_1^* - \varepsilon$ with probability at least $2/3$ on every ranking distribution has worst-case expected pairwise query complexity $\Omega(m/\varepsilon^2)$.

Proof. For each claim, we construct a base ranking distribution and alternatives that differ from it in only one candidate's random bit. A correct algorithm must distinguish the base distribution from many of these alternatives. Yet it learns about a candidate's bit only by making a comparison involving that candidate, which allows us to bound the total information obtained from its queries.

To generate strict rankings, we use independent binary variables and place candidates in the blocks specified below. Within each block, order candidates by independent, identically distributed continuous random priorities, independent of all bits. Each block is therefore uniformly ordered. In particular, comparisons not involving candidate i are independent of its bit B_i.

(i) Evaluating a fixed singleton. Fix $S = \{s\}$. Draw independent bits B_x for $x \ne s$ and rank the candidates in three blocks: those with $B_x = 0$, then s, then those with $B_x = 1$. Thus s beats x exactly when $B_x = 1$.

Under the base distribution, every bit is Bernoulli$(1/2)$ and $\theta(\{s\}) = 1/2$. For each $i \ne s$, define an alternative by changing only B_i to Bernoulli$(1/2 - 4\varepsilon)$. Candidate i then becomes the most difficult rival for s, and $\theta(\{s\}) = 1/2 - 4\varepsilon$.

An estimate accurate to within ε must distinguish these values. In particular, the event $\widehat{\theta} \ge 1/2 - 2\varepsilon$ has probability at least $2/3$ under the base distribution and at most $1/3$ under each alternative. Let P_0 and P_i be the corresponding distributions of all observations and the algorithm's output. Pinsker's inequality then gives $\mathrm{KL}(P_0 \| P_i) \ge 2/9$.

We next bound this divergence in terms of the queries involving i. Let M_i count the sampled tasks on which the algorithm makes at least one such query. For the analysis, add B_i to the observations at the first query involving i on each task, while keeping the algorithm's decisions unchanged. Adding this information can only increase the divergence. Earlier comparisons on the task are independent of B_i; once B_i is known, all remaining random variables on that task have the same conditional distribution in the two cases. Hence the adaptive chain rule and data processing give

$$\mathrm{KL}(P_0 \| P_i) \le \mathbb{E}_0[M_i]\, \mathrm{KL}\left(\text{Bernoulli}(1/2) \,\|\, \text{Bernoulli}(1/2 - 4\varepsilon)\right) \le \tfrac{128}{3}\varepsilon^2 \mathbb{E}_0[M_i].$$

For the final inequality, $\varepsilon \le 1/16$ implies $-\frac{1}{2}\log(1-64\varepsilon^2) \le (128/3)\varepsilon^2$. As in the binary lower bound, the desired conclusion holds immediately if the expected query count is infinite. When it is finite, truncation justifies applying the chain rule to an algorithm with a random stopping time. Combining the lower and upper bounds on divergence yields $\mathbb{E}_0[M_i] = \Omega(\varepsilon^{-2})$ for every $i \neq s$. If Q is the total number of comparisons, then $\sum_{i\neq s} M_i \le 2Q$, because one comparison involves at most two of these candidates. Summing over the $m-1$ rivals proves the first claim.

(ii) Selecting a singleton. For selection, we instead place candidates with bit 1 above those with bit 0, again using independent priorities within each block. The base distribution has independent Bernoulli$(1/2)$ bits for all candidates. Alternative i changes only B_i to Bernoulli$(1/2+4\varepsilon)$, making i the uniquely best choice. To see this, for every $j \neq i$,

$$\Pr[i \succ_\pi j] = \Pr[B_i = 1, B_j = 0] + \tfrac{1}{2}\Pr[B_i = B_j] = \tfrac{1}{2} + 2\varepsilon.$$

Every comparison between two candidates other than i has win probability $1/2$. It follows that $\theta(\{i\}) = 1/2 + 2\varepsilon$, whereas $\theta(\{j\}) = 1/2 - 2\varepsilon$ for every $j \neq i$. Only i is within ε of the optimum, so the algorithm must output it with probability at least $2/3$ under alternative i.

Under the base distribution, at most one candidate can have output probability greater than $1/2$. Thus, for at least $m-1$ choices of i, the probability of outputting i changes by at least $1/6$ between the base distribution and alternative i. Pinsker's inequality gives $\mathrm{KL}(P_0 \| P_i) \ge 1/18$ for each such i. Repeating the information calculation from part (i), now with the bit bias increased rather than decreased, gives $\mathbb{E}_0[M_i] = \Omega(\varepsilon^{-2})$. Summing over these indices and using $\sum_i M_i \le 2Q$ yields $\mathbb{E}_0[Q] = \Omega(m/\varepsilon^2)$, as required. □

C.6 Committee Evaluation and Selection from a Finite List

Selecting a committee and estimating its value are different problems. A proposed committee can be evaluated without recovering a full ranking: find its highest-ranked member, denoted by $\mathrm{top}_\pi(S)$, and compare that member with every omitted candidate. The first step takes $|S|-1$ comparisons, and the second determines whether the committee beats each rival on the task. We first analyze a fixed number of evaluation tasks. We then show how to save comparisons by stopping early for rivals and committees that are clearly separated from the best choices.

Theorem C.6. *Fix $\varepsilon, \delta \in (0,1)$. A fixed nonempty proper committee S can be evaluated to additive error ε, with probability at least $1-\delta$, using $O(m\varepsilon^{-2}\log(2(m-|S|)/\delta))$ pairwise comparisons.*

More generally, let $\mathcal{F}$ be a nonempty finite family of nonempty proper committees, fixed independently of the evaluation tasks, and let $N_{\mathcal{F}} := \sum_{S\in\mathcal{F}}(m-|S|)$. Empirical maximization over $\mathcal{F}$ returns $\widehat{S} \in \mathcal{F}$ with $\theta(\widehat{S}) \ge \max_{S\in\mathcal{F}}\theta(S) - \varepsilon$ with probability at least $1-\delta$, using $O(m\log m\,\varepsilon^{-2}\log(2N_{\mathcal{F}}/\delta))$ comparisons.

Proof of Theorem C.6. The key observation is that one set of comparisons evaluates a committee against all of its rivals on a task. For a fixed committee S, draw independent tasks $\pi_1, \dots, \pi_T \sim P_{\mathcal{D}}$. On each task, find $\mathrm{top}_{\pi_t}(S)$ in $|S|-1$ comparisons and compare it with every $x \notin S$. The total is $m-1$ comparisons per task, and the answers give the observations whose averages define $\widehat{w}_T(S,x)$ in Appendix C.5. For

$$T \ge \frac{1}{2\varepsilon^2}\log\frac{2(m-|S|)}{\delta},$$

Hoeffding's inequality and a union bound give $\max_{x\notin S}|\widehat{w}_T(S,x) - \mathrm{WIN}(S,\{x\})| \le \varepsilon$ with probability at least $1-\delta$. Taking the minimum preserves this error bound, proving the claim for $\widehat{\theta}_T(S)$.

For a family $\mathcal{F}$, full rankings allow the same tasks to evaluate every proposed committee. Use $T \ge (2/\varepsilon^2)\log(2N_{\mathcal{F}}/\delta)$ fresh tasks and recover their rankings. A union bound over the $N_{\mathcal{F}}$ committee–rival pairs gives error at most $\varepsilon/2$ for every $\widehat{\theta}_T(S)$. Comparing the empirical maximizer with a true maximizer then loses at most ε, as in the proof of Theorem C.4. Sorting the rankings costs $O(Tm\log m)$ comparisons. If earlier data were used to generate the family, we first condition on those data; the fresh evaluation tasks remain independent, so the same argument applies. □

A fixed sample size spends the same effort on every rival, although many may be easy for the committee to beat. We can stop querying such a rival once its win probability is clearly above the minimum. The following bound makes this benefit precise: larger gaps Γ_x require fewer comparisons, while gaps below the requested accuracy ε need not be distinguished.

Theorem C.7. *Fix $\varepsilon, \delta \in (0,1)$ and a nonempty proper committee S. For $x \notin S$, let $\Gamma_x := \mathrm{WIN}(S, \{x\}) - \theta(S)$. There is an adaptive evaluation procedure whose confidence intervals for $\theta(S)$ are simultaneously valid with probability at least $1-\delta$. Stopping at interval width at most 2ε and returning the midpoint gives additive error at most ε. On the same event, its comparison count is*

$$O\left((|S|-1)\frac{\log(e(m-|S|)/(\delta\varepsilon))}{\varepsilon^2} + \sum_{x\notin S}\frac{\log(e(m-|S|)/(\delta\max\{\Gamma_x,\varepsilon\}))}{\max\{\Gamma_x,\varepsilon\}^2}\right).$$

Proof. We maintain confidence intervals for the active rivals and remove a rival only when another has a smaller win probability throughout their intervals. Initially all outside candidates are active. In round t, draw a fresh task, find $\mathrm{top}_{\pi_t}(S)$, and compare it with every active rival. A rival that is still active has been queried on all t tasks, so its observed mean is $\widehat{w}_t(S,x)$. Set

$$\mathrm{rad}_t := \sqrt{\frac{\log(4(m-|S|)t^2/\delta)}{2t}},$$

and define the interval endpoints by

$$\underline{\theta}_t := \min_{x \text{ active}} \widehat{w}_t(S,x) - \mathrm{rad}_t, \quad \overline{\theta}_t := \min_{x \text{ active}} \widehat{w}_t(S,x) + \mathrm{rad}_t.$$

Remove x if $\widehat{w}_t(S,x) - \mathrm{rad}_t > \overline{\theta}_t$. This rule compares the lower bound for x with the smallest upper bound among the active rivals.

As in Theorem B.4, Hoeffding's inequality and a union bound over all rivals and all t give an event of probability at least $1-\delta$ on which every empirical mean is within its stated radius. On this event, removing a rival implies that another rival has a strictly smaller true win probability. A minimizing rival therefore always remains active, and $\theta(S) \in [\underline{\theta}_t, \overline{\theta}_t]$ at every round. The midpoint has error at most ε when the interval has width at most 2ε.

To bound the comparisons, fix a minimizing rival z, which stays active. A rival x with a positive gap can be removed once the confidence radius is small relative to that gap. Specifically, if $\mathrm{rad}_t < \Gamma_x/4$, then

$$\widehat{w}_t(S,x) - \mathrm{rad}_t \geq \theta(S) + \Gamma_x - 2\mathrm{rad}_t > \theta(S) + 2\mathrm{rad}_t \geq \widehat{w}_t(S,z) + \mathrm{rad}_t \geq \overline{\theta}_t,$$

so x is removed. If it has not been removed sooner, the procedure stops once the common radius is at most ε, when the interval for $\theta(S)$ has width at most 2ε. The radius calculation in Theorem B.4 therefore bounds the comparisons to x in terms of $\max\{\Gamma_x, \varepsilon\}$. The procedure uses at most $O(\varepsilon^{-2}\log(e(m-|S|)/(\delta\varepsilon)))$ tasks, each requiring another $|S|-1$ comparisons to find the best committee member. Adding this cost to the rival comparisons gives the stated bound. □

We can also apply this reasoning when choosing from a finite list of committees. There are now two opportunities to stop querying: a rival may be clearly easier to beat than that committee's worst rival, or the committee may be clearly worse than another proposal. The next theorem expresses these two comparisons through the rival gap $\Gamma_{S,x}$ and the committee gap Λ_S.

Theorem C.8. *Fix $\varepsilon, \delta \in (0,1)$ and a nonempty finite family $\mathcal{F}$ of nonempty proper committees, independent of the evaluation tasks. Write $N_{\mathcal{F}} := \sum_{S\in\mathcal{F}}(m-|S|)$, $\Lambda_S := \max_{S'\in\mathcal{F}}\theta(S') - \theta(S)$, and $\Gamma_{S,x} := \mathrm{WIN}(S,\{x\}) - \theta(S)$. An adaptive procedure returns $\widehat{S} \in \mathcal{F}$ with $\theta(\widehat{S}) \geq \max_{S\in\mathcal{F}}\theta(S) - \varepsilon$ with probability at least $1-\delta$. On the same event, its comparison count is*

$$O\left(\sum_{S\in\mathcal{F}}(|S|-1)\frac{\log(eN_{\mathcal{F}}/(\delta\max\{\varepsilon,\Lambda_S\}))}{\max\{\varepsilon,\Lambda_S\}^2} + \sum_{S\in\mathcal{F}}\sum_{x\notin S}\frac{\log(eN_{\mathcal{F}}/(\delta\max\{\varepsilon,\Lambda_S,\Gamma_{S,x}\}))}{\max\{\varepsilon,\Lambda_S,\Gamma_{S,x}\}^2}\right).$$

Proof. We evaluate the committees to progressively finer accuracy, removing those whose upper bounds are sufficiently below another committee's lower bound. For each committee, use the preceding evaluation procedure with error probability $\delta/N_{\mathcal{F}}$. Since $N_{\mathcal{F}} \geq |\mathcal{F}|$, a union bound ensures that every interval is valid at every step with probability at least $1 - \delta$. Evaluations can be paused and resumed while retaining earlier observations and keeping removed rivals inactive. Work on this joint confidence event.

Initially all committees are active. At iteration $j = 0, 1, \ldots$, resume each active committee's evaluation until its interval $[\underline{\theta}_S, \overline{\theta}_S]$ has width at most 2^{-j-3}. Eliminate a committee when $\overline{\theta}_S < \max_{S' \text{ active}} \underline{\theta}_{S'} - \varepsilon$. Stop and return any remaining S with $\underline{\theta}_S \geq \max_{S' \text{ remaining}} \overline{\theta}_{S'} - \varepsilon$. An optimal committee is never eliminated, since its upper bound is at least the optimum and every lower bound is at most the optimum. At stopping, the upper bound of an optimal committee is at most $\underline{\theta}_S + \varepsilon \leq \theta(S) + \varepsilon$. Thus the returned committee is within ε of optimal.

To determine how long each evaluation runs, suppose S remains after iteration j. Comparing its upper bound with an optimal committee's lower bound gives $\Lambda_S \leq \varepsilon + 2^{-j-2}$. The procedure must also stop once $2^{-j-3} \leq \varepsilon$, because a committee with the largest lower bound then meets the stopping rule. When $\Lambda_S > 2\varepsilon$, elimination limits the required accuracy in terms of Λ_S; when $\Lambda_S \leq 2\varepsilon$, the stopping rule limits it in terms of ε. Thus evaluation of S only needs accuracy within a constant factor of $\max\{\varepsilon, \Lambda_S\}$. Successive precision targets differ by a factor of two, and resumed evaluations reuse all earlier observations, so Theorem C.7 applies at this final precision.

Finding the best member of S on each of its tasks gives the first sum in the theorem. Comparisons with a rival x stop when its committee is removed, when the rival itself is removed, or when the procedure terminates. These possibilities give the denominator determined by $\max\{\varepsilon, \Lambda_S, \Gamma_{S,x}\}$ in the second sum. Finally, $\log(e(m - |S|)N_{\mathcal{F}}/(\delta a)) \leq 2\log(eN_{\mathcal{F}}/(\delta a))$ for $0 < a \leq 1$. Allocating confidence separately to each committee therefore changes the bound only by universal constants, which proves the claimed comparison count. □

D Pairwise Feedback: Lotteries over Committees

We first prove that fixed rival weights give a coverage objective, then apply conditional greedy to this objective. Finally, we analyze the multiplicative-weights procedure that turns these weighted choices into a lottery with a guarantee against every fixed rival.

D.1 The Weighted Coverage Representation

For fixed rival weights, the objective instead averages events of the form "the committee contains a candidate at least as high as the rival." Each such event has diminishing returns, and averaging preserves this property.

Lemma D.1. *For every probability vector λ over C, the function Φ_λ is normalized, monotone, and submodular. Moreover, for every nonempty proper committee S, $\theta(S) = \min_\lambda \Phi_\lambda(S)$, where the minimum is over all such λ.*

Proof. Fix a task–rival pair (π, x). Adding a candidate to S changes $\mathbf{1}\{S \cap P_\pi(x) \neq \varnothing\}$ only when the committee does not yet intersect $P_\pi(x)$ and the new candidate belongs to that set. This is a normalized, monotone coverage function with diminishing returns. Taking expectations over π and nonnegative weighted averages over x preserves all three properties.

It remains to relate the weighted objective to θ. For a proper committee, $g_x(S) = 1$ when $x \in S$ and $g_x(S) = \mathrm{WIN}(S, \{x\})$ otherwise, so $\theta(S) = \min_{x \in C} g_x(S)$. A weighted average of these values is at least their minimum, and placing all weight on a minimizing candidate attains that minimum. Hence $\theta(S) = \min_\lambda \Phi_\lambda(S)$. □

D.2 Weighted Ordinal Greedy

The coverage representation lets us apply conditional greedy directly to task–rival pairs. Drawing a ranking in the algorithm below means drawing a task whose ranking remains hidden. To determine whether S covers (π, x), first check whether $x \in S$; in that case no query is needed. Otherwise,

compare members of S with x until one beats x or all have been queried. The algorithm retains uncovered pairs, on which x ranks above every member of S, and uses them to compare possible additions. If more such pairs are needed, it draws and tests fresh pairs in the same way.

Algorithm 3 Weighted ordinal feedback greedy (WOFG)

Require: Committee size k, rival weights λ, and $\varepsilon, \delta \in (0,1)$.
1: $r_0 \leftarrow \lceil 6\log(8k/\delta)\rceil$; $M_0 \leftarrow \lfloor 2r_0k/\varepsilon\rfloor$; $S_0 \leftarrow \varnothing$.
2: **for** $i = 0, \dots, k-1$ **do**
3: Draw fresh pairs $(\pi, x) \sim P_{\mathcal{D}} \times \lambda$ and test coverage by S_i, stopping after r_0 uncovered pairs or M_0 draws. Store the uncovered pairs in F_i and let N_i be the draw count.
4: **if** $|F_i| < r_0$ **then**
5: Pad S_i to size k and **return** the resulting committee.
6: **end if**
7: $\widetilde{\rho}_i \leftarrow \min\{1, 2r_0/N_i\}$; $\eta_i \leftarrow \varepsilon/(k\widetilde{\rho}_i)$.
8: Run CONDITIONAL-ELIM with accuracy η_i and error probability $\delta/(4k)$, using F_i first and drawing further uncovered pairs as needed. For candidate c, the observed bit is 1 when $c = x$ and otherwise $\mathbf{1}\{\mathrm{rank}_\pi(c) < \mathrm{rank}_\pi(x)\}$, obtained with one comparison.
9: Add the returned candidate to S_i to obtain S_{i+1}.
10: **end for**
11: **return** S_k.

Proof of Theorem 4.4. We identify the binary selection problem represented by the algorithm and then account for its comparisons. Treat $(\pi, x) \sim P_{\mathcal{D}} \times \lambda$ as a sampled task, with binary outcome $\mathbf{1}\{c \in P_\pi(x)\}$ for candidate c. Committee coverage in this binary problem is exactly Φ_λ. A new member covers a previously uncovered pair precisely when it belongs to $P_\pi(x)$, which gives (3). Thus Algorithm 3 applies Algorithm 1 to this distribution of binary profiles.

Each binary observation requires at most one pairwise comparison, and none when the candidate is the rival itself. Testing a size-i committee requires at most i comparisons per drawn pair. Conditional on the coverage indicators from the preliminary test, the retained pairs are independent and have the required conditional distribution, by the argument in Appendix B.4. We can therefore reuse these pairs and apply the same miss-probability estimate and conditional selection guarantee. Theorem 3.2 then gives both the value guarantee and the total comparison bound, including the comparisons needed to find uncovered pairs. □

D.3 MULTIPLICATIVE WEIGHTS OVER RIVALS

A committee chosen for fixed weights can perform poorly against a rival with little weight. The procedure below addresses this by increasing the weights of rivals that previous committees cover poorly. We prove its guarantee in two steps. First, we compare the weighted value of each chosen committee with the value of an optimal lottery. Second, the multiplicative-weights inequality relates these weighted values to the lottery's value against every fixed rival.

Proof of Theorem 4.5. Choose

$$R := \left\lceil \frac{64\log m}{\varepsilon^2} \right\rceil, \quad T_{\mathrm{eval}} := \left\lceil \frac{32}{\varepsilon^2}\log\frac{4mR}{\delta} \right\rceil, \quad \eta_{\mathrm{MW}} := \sqrt{\frac{\log m}{R}}.$$

Use these parameters in Algorithm 4; the update keeps all rival weights positive and normalizes them to sum to one.

We first show that every weighted selection call can be compared with the same optimal lottery. Let p^* maximize $\theta(p)$ over all lotteries over $\mathcal{S}_k$. A maximizer exists because the set of lotteries is compact and θ is continuous. For every probability vector λ over C,

$$\max_{S\in\mathcal{S}_k} \Phi_\lambda(S) \geq \mathbb{E}_{S\sim p^*}[\Phi_\lambda(S)] = \sum_x \lambda_x \mathbb{E}_{S\sim p^*}[g_x(S)] \geq \theta(p^*).$$

Algorithm 4 Selecting a lottery by reweighting rivals

1: Initialize $\lambda_1(x) = 1/m$ for every rival x.
2: **for** $t = 1, \dots, R$ **do**
3: Run Algorithm 3 at weights λ_t to choose S_t, with accuracy $\varepsilon/4$ and error probability $\delta/(2R)$.
4: On each of T_{eval} new tasks, find the best member of S_t and compare it with every omitted candidate x. Let $\widehat{g}_t(x)$ be the fraction of these tasks on which it beats x.
5: Set $\widehat{g}_t(x) = 1$ for $x \in S_t$ and update
$$\lambda_{t+1}(x) = \frac{\lambda_t(x) e^{\eta_{\mathrm{MW}}(1-\widehat{g}_t(x))}}{\sum_y \lambda_t(y) e^{\eta_{\mathrm{MW}}(1-\widehat{g}_t(y))}}.$$
6: **end for**
7: Return the uniform lottery over $S_1, \dots, S_R$.

The first inequality compares a maximum with an average; the last uses the guarantee of p^* against every rival. Conditioning on the history before a weighted greedy call fixes its weights, so Theorem 4.4 applies to that call. A union bound implies that, with probability at least $1-\delta/2$, all returned committees satisfy

$$\Phi_{\lambda_t}(S_t) \geq (1 - 1/e)\theta(p^*) - \varepsilon/4.$$

We next control the errors in the values used to update the weights. For each S_t, find its best member on each of T_{eval} fresh tasks and compare it with every outside candidate. These $m-1$ comparisons per task give the estimates $\widehat{g}_t(x)$; included candidates have known value $\widehat{g}_t(x) = 1$. Conditional Hoeffding bounds and a union bound over t and x give, with probability at least $1 - \delta/2$,

$$|\widehat{g}_t(x) - g_x(S_t)| \leq \varepsilon/8 \quad \text{for all } t \in [R],\ x \in C.$$

Write $\ell_t(x) := 1 - g_x(S_t)$ and $\widehat{\ell}_t(x) := 1 - \widehat{g}_t(x)$ for the true and estimated probabilities that rival x is uncovered. The update gives larger weight to rivals with larger $\widehat{\ell}_t(x)$. Since these estimates lie in $[0, 1]$, the multiplicative-weights inequality gives, for each fixed rival x,

$$\frac{1}{R}\sum_{t=1}^{R} \widehat{\ell}_t(x) \leq \frac{1}{R}\sum_{t=1}^{R} \langle \lambda_t, \widehat{\ell}_t \rangle + \frac{\log m}{\eta_{\mathrm{MW}} R} + \eta_{\mathrm{MW}} \leq \frac{1}{R}\sum_{t=1}^{R} \langle \lambda_t, \widehat{\ell}_t \rangle + \varepsilon/4.$$

To verify the first inequality, take the logarithm of each normalization factor in the update and bound it above by $\eta_{\mathrm{MW}} \langle \lambda_t, \widehat{\ell}_t \rangle + \eta_{\mathrm{MW}}^2$, using $e^{\eta z} \leq 1 + \eta z + \eta^2$ for $z \in [0, 1]$ and $\eta \leq 1$. Summing these logarithms gives a quantity at least $\eta_{\mathrm{MW}} \sum_t \widehat{\ell}_t(x) - \log m$ for every fixed rival x. Rearranging proves the first inequality; the second follows from $2\sqrt{\log m / R} \leq \varepsilon/4$.

The two confidence events hold together with probability at least $1 - \delta$. On this event, replacing the estimates on both sides of the preceding inequality by their true values costs at most $\varepsilon/4$. Combining it with the weighted selection guarantee gives, for every x,

$$\begin{aligned}\frac{1}{R}\sum_{t=1}^{R} \ell_t(x) \leq \frac{1}{R}\sum_{t=1}^{R} \langle \lambda_t, \ell_t \rangle + \varepsilon/2 &= 1 - \frac{1}{R}\sum_{t=1}^{R} \Phi_{\lambda_t}(S_t) + \varepsilon/2 \\ &\leq 1 - (1 - 1/e)\theta(p^*) + 3\varepsilon/4.\end{aligned}$$

Taking complements gives the required bound against every rival for the uniform lottery over the S_t. Evaluation uses $RT_{\mathrm{eval}}(m-1)$ comparisons in total, which is the stated cost outside the weighted greedy calls.

Finally, Theorem 3.2, with all miss probabilities bounded by one, all denominators bounded below by the accuracy cutoff, and $k < m$, bounds one weighted selection call by $O(mk^3\varepsilon^{-2}\log(emk^2R/(\delta\varepsilon)))$ comparisons on its confidence event. Multiplying by $R = O(\varepsilon^{-2}\log m)$ and adding the evaluation cost gives the total comparison bound in the theorem. These events are the same ones used for the value guarantee. □

E Experimental Methods and Additional Results

We give the methods and additional results for the three comparisons in Section 5. Tasks are drawn independently, answers already obtained on a task may be reused, and every new query is charged. Population values are used only to evaluate the returned committees, except in the explicitly full-information cyclic-ranking experiment.

E.1 Controlled Two-Candidate Comparisons

To study the cost in Theorem 3.1, we construct a task distribution in which the cost of testing the committee, the agreement between the two additions, and their mean gap can be varied separately. The current committee has i members: its first $i-1$ members are always incorrect, and its last member is correct with probability $1-\rho$. Testing the members in this order costs exactly i queries on every task. Conditional on a committee miss, the two possible additions have outcomes

$$\Pr(1,0) = (d_{ab} + g_{ab})/2, \quad \Pr(0,1) = (d_{ab} - g_{ab})/2, \quad \Pr(1,1) = \Pr(0,0) = (1 - d_{ab})/2.$$

Their individual success probabilities are $1/2 + g_{ab}/2$ and $1/2 - g_{ab}/2$, so varying d_{ab} changes how often they disagree without changing their mean gap. On tasks the committee solves, we may set both additions' outcomes to zero, since neither is queried. Tasks are drawn independently from this joint distribution, and the algorithms pay to find those missed by the committee.

Three statistical stopping rules. All three methods query both additions on each missed task. Let N count these tasks, let $X_j = u(d_j, a) - u(d_j, b) \in \{-1, 0, 1\}$ be the observed difference on the jth missed task d_j, and write $\overline{X}_N = N^{-1} \sum_j X_j$. The first two methods stop when a confidence interval for the mean difference excludes zero. The third stops when an interval for the probability that the first candidate wins a disagreement excludes $1/2$. In each case, the interval determines which candidate is selected.

The *separate-means* method combines a Hoeffding interval for each candidate, giving an interval around $\overline{X}_N$ with radius $2\sqrt{\log(8N^2/\delta)/(2N)}$. The *paired Bernstein* method instead uses the variance of the observed differences, $V_N = (N-1)^{-1} \sum_j (X_j - \overline{X}_N)^2$, with radius

$$\sqrt{\frac{2V_N \log(8N^2/\delta)}{N}} + \frac{14 \log(8N^2/\delta)}{3(N-1)}.$$

This applies the empirical-Bernstein bound of Maurer & Pontil (2009) to a variable of range two. Finally, the *disagreement test* uses the sign interval from Appendix B.2: after H disagreements, its radius is $\sqrt{\log(4H^2/\delta)/(2H)}$ around the observed fraction won by the first candidate. For each method, assigning decreasing error probabilities to successive checks and taking a union bound makes its intervals valid simultaneously at all tested sample counts. We set $\delta = 0.05$.

Each method first checks its interval after eight relevant observations and then at counts $n \leftarrow \lceil 1.25n \rceil$. The relevant observations are missed tasks for the first two methods and disagreements for the third. To simulate large query counts efficiently, we draw multinomial or binomial counts and negative-binomial waiting times directly. These draws have the same distribution as generating the independent tasks one at a time, and we charge for every query they represent. If a run draws T tasks and examines both additions on N missed tasks, its total cost is exactly $iT + 2N$.

Parameters and repetitions. The main comparison fixes $(\rho, g_{ab}, i) = (0.1, 0.04, 4)$ and varies $d_{ab} \in \{0.08, 0.12, 0.2, 0.4, 0.8\}$. Three further comparisons vary one parameter at a time: $\rho \in \{0.025, 0.05, 0.1, 0.25, 0.5, 1\}$ at $(d_{ab}, g_{ab}, i) = (0.12, 0.04, 4)$; $g_{ab} \in \{0.01, 0.02, 0.04, 0.08\}$ at $(d_{ab}, \rho, i) = (0.2, 0.1, 4)$; and $i \in \{1, 2, 4, 8, 16\}$ at $(d_{ab}, g_{ab}, \rho) = (0.12, 0.04, 0.1)$. Each method uses 100 independent repetitions per setting, for 6,000 runs in total. Methods draw independently from the same distributions.

Because stopping costs vary across repetitions, the figures show medians and the middle half of the query counts. Every run stops by its statistical rule and selects the correct candidate. These observations agree with the probabilistic guarantee, but provide only a finite check. The savings from agreement are measured while keeping a positive mean gap fixed; they concern the cost of comparing additions, not the coverage obtained by selecting redundant models.

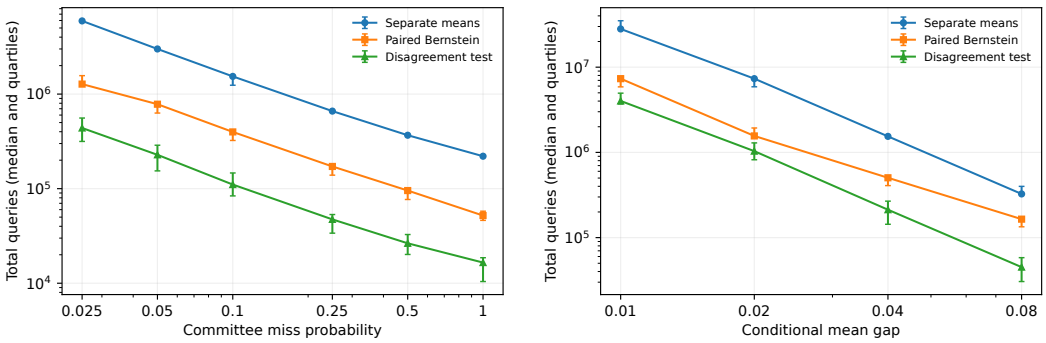


Figure 3: Two-candidate comparisons as the committee miss probability and mean gap vary. **Left:** missed tasks cost more to find when the committee solves a larger fraction of tasks. **Right:** a smaller mean gap requires more observations. All curves include committee and candidate queries; bars show the middle half of 100 repetitions.

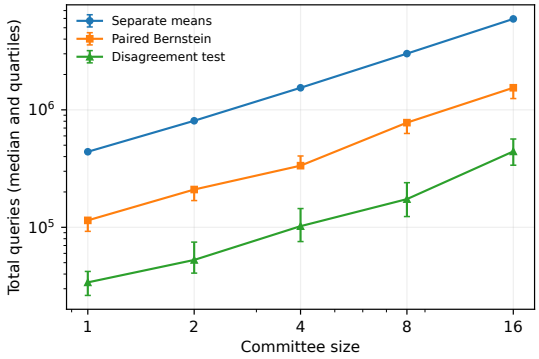


Figure 4: The cost of testing the current committee. In this construction, every member must be queried on every sampled task, so larger committees raise the cost per task. The disagreement test saves queries by examining fewer tasks.

E.2 Cyclic Ranking Populations

We next compare the best deterministic committee with the best lottery. Begin with candidates $1, \dots, m$ and the common order $1 \succ 2 \succ \cdots \succ m$. Two further orders rotate the first three candidates to $2 \succ 3 \succ 1$ and $3 \succ 1 \succ 2$, leaving the remaining candidates in place; together with the common order, they form three cyclic orders. For a cyclic weight $z \in [0, 1]$, put mass $1 - z$ on the common order and mass $z/3$ on each of the three cyclic orders. Increasing z thus moves the population from unanimous rankings toward the symmetric cycle.

We also consider a smooth version that replaces each order with a Plackett–Luce distribution. If candidate c is in position j of an order, with $j = 1$ at the top, assign it weight $w_c = \exp[-(j-1)/t]$. A ranking is generated by choosing the first candidate in proportion to these weights, removing it, and repeating. The positive parameter t controls variation: small values keep rankings close to the specified order, while larger values produce more varied rankings. We use $m \in \{3, 6, 9\}$, $k = 1$, $z \in \{0, 0.25, 0.5, 0.75, 1\}$, and $t \in \{0.2, 0.8, 1.6\}$, as well as the unsmoothed version. Figure 1 shows $m = 9$ and $t = 0.2$, connecting the smooth example to the three-cycle in the main text. The accompanying tables contain all 60 populations and all three round counts, giving 180 configurations.

Computing the objectives. For each committee, we compute $g_x(S)$ from the exact population probabilities and enumerate all committees to find the best deterministic value. To find the best

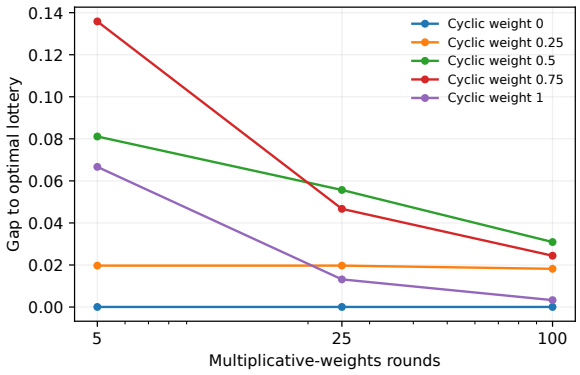


Figure 5: Full-information reweighting on the smooth nine-candidate ranking populations ($t = 0.2$). The gap to the best lottery generally decreases as the number of rounds increases. The theorem does not require monotonic improvement in every finite run.

lottery, a linear program chooses probabilities $p_S \geq 0$ and a value z_0. It maximizes z_0 subject to

$$\sum_{S \in \mathcal{S}_k} p_S = 1, \quad \sum_{S \in \mathcal{S}_k} p_S g_x(S) \geq z_0 \quad \text{for each } x \in C.$$

The first constraint makes p a probability distribution, and the remaining constraints require its expected value to be at least z_0 against every rival. The solver computes the optimum up to its numerical tolerance. We also run multiplicative weights for $R \in \{5, 25, 100\}$ rounds using exact population values. We start with uniform rival weights, use learning rate $\sqrt{\log(m)/R}$, choose the smallest-index maximizer of the current weighted value, and update rival log-weights by adding $\sqrt{\log(m)/R}(1 - g_x(S_t))$. The returned lottery is uniform over the R selected singletons, including repeated selections. Since those runs receive full information and use $k = 1$, they measure objective values and convergence, not the comparison cost of the general lottery algorithm.

E.3 Binary Selection under Fixed Query Budgets

We compare conditional greedy, screened greedy, screened Bernstein, and full evaluation on four populations. Each population has $m = 61$ candidates, and every method returns $k = 3$ distinct candidates. Tasks are independent draws. The population parameters are available to the simulator and evaluator but not to the selection routines, which receive only task identifiers and answers to charged queries.

Generalist, specialists, and decoys. Index candidates by $0, \ldots, 60$. Candidate 0 is the generalist, candidates 1 and 2 are specialists, and candidates $3, \ldots, 60$ are decoys. A common task type has mass $\alpha \in \{0.5, 0.8, 0.9\}$; two rare types each have mass $(1 - \alpha)/2$. On the common type, candidate 0 is always correct and all other candidates share a single Bernoulli outcome with mean 0.25. On rare type $j \in \{1, 2\}$, candidate 0 is always incorrect, specialist j succeeds with probability 0.9, the other specialist succeeds with probability 0.05, and each decoy succeeds with probability 0.25. Outcomes are independent conditional on a rare type. In particular, the common component is not a product distribution.

For a committee S, let p_{jc} denote candidate c's success probability on rare type j, and put $b(S) = 0$ when $0 \in S$, $b(S) = 0.75$ when S is nonempty and excludes 0, and $b(\varnothing) = 1$. Its coverage is

$$v(S) = 1 - \alpha b(S) - \frac{1 - \alpha}{2} \left(\prod_{c \in S} (1 - p_{1c}) + \prod_{c \in S} (1 - p_{2c}) \right).$$

Thus coverage can be evaluated exactly after selection, without spending evaluation queries or estimating it from a test sample.

Independent candidates. The fourth population has independent outcomes with success probabilities $p_c = 0.06 + 0.16c/60$, for $c = 0, \ldots, 60$. Here $v(S) = 1 - \prod_{c \in S}(1 - p_c)$. This population provides a different sampling tradeoff: full profiles can be reused across greedy additions, while no small set of rare task types identifies the useful candidates. The four populations are designed examples, not a representative sample of deployed ensembles. We report the complete population–budget grid and do not infer how often either regime occurs in practice.

Selection rules and query accounting. Budgets are $Q \in \{1{,}220, 2{,}440, 4{,}880, 9{,}760, 19{,}520\}$ correctness queries. Every previously unobserved candidate–task answer costs one query. Repeated access to an observed answer on the same task is free; an identical outcome on a different task draw must be queried again. Committee members are tested in their order of selection, stopping at the first success.

At each addition, an adaptive method allocates $\lfloor (Q - Q_{\text{used}})/(k - i) \rfloor$ queries, where i candidates have already been selected. Conditional greedy uses the preliminary miss-rate test of Algorithm 1, with $\varepsilon = 0.2$, $\delta = 0.05$,

$$r_0 = \left\lceil 6 \log \frac{8k}{\delta} \right\rceil, \quad M_0 = \left\lfloor \frac{2r_0 k}{\varepsilon} \right\rfloor.$$

It retains the first r_0 missed tasks and reuses them for conditional selection. If the test reaches M_0 draws first, it fills the remaining places uniformly without replacement. Otherwise, with N_i pilot draws, it uses $\widetilde{\rho}_i = \min\{1, 2r_0/N_i\}$ and conditional tolerance $\eta_i = \varepsilon/(k\widetilde{\rho}_i)$.

Screened greedy instead records a marginal observation on every sampled task. A task solved by the current committee gives every active candidate a known zero, requiring no candidate queries; on a missed task, all active candidates are queried. It uses tolerance $\eta_i = \varepsilon/k$ and does not run the miss-rate pilot. Both methods eliminate a candidate when its upper bound is more than η_i below the largest lower bound, with Hoeffding radius $\sqrt{L_n/(2n)}$, where

$$L_n = \log \frac{4(m - i)n^2}{\delta/(4k)}.$$

Here n counts complete conditional rounds for conditional greedy and complete marginal rounds for the screened methods. Screened Bernstein uses the same marginal observations and elimination rule, replacing the radius for candidate c by

$$\sqrt{\frac{2\widehat{V}_{n,c} L_n}{n}} + \frac{7L_n}{3(n-1)}, \quad \widehat{V}_{n,c} = \frac{n}{n-1}\widehat{\mu}_{n,c}(1 - \widehat{\mu}_{n,c}),$$

for $n > 1$, where $\widehat{\mu}_{n,c}$ is the empirical mean of candidate c's marginal observations, and infinite radius for $n = 1$. This is a variance estimate for individual marginal contributions, not the variance of a paired difference. Each adaptive rule stops statistically when one candidate remains or every surviving radius is at most $\eta_i/4$.

At an allocation limit, the method returns an empirical maximizer among surviving candidates, breaking ties uniformly. Queries in a partially completed round are charged but omitted from the statistics, so all surviving candidates have the same sample count. If no complete round was obtained, selection is uniform among the surviving candidates. These budget-forced choices are recorded separately from statistical stopping. Full evaluation collects $\lfloor Q/m \rfloor$ complete profiles and reuses them for all three empirical greedy additions, also breaking ties uniformly. The chosen budgets correspond to 20, 40, 80, 160, and 320 complete profiles.

Repetitions and summaries. Every population–budget–method combination has 200 independent repetitions, for 16,000 trials. Methods use independent random streams, rather than common random numbers. We report the mean coverage and 1.96 times its sample standard error, with sample variance computed using denominator 199. These bars summarize Monte Carlo variation on each fixed population; they are not simultaneous confidence bands, and zero variation across the observed repetitions does not establish certainty about future runs. Population-greedy references use exact coverage and are not available to the query algorithms.

All saved trials spend exactly their specified budget. All 12,000 adaptive trials contain at least one budget-forced addition. Consequently, these experiments compare the coverage attained by budget-limited implementations; they do not show that the confidence-based algorithm of Theorem 3.2

Table 1: Mean population coverage at 4,880 queries, with 1.96 standard errors over 200 repetitions. CG denotes conditional greedy. The other columns are screened greedy, screened Bernstein, and full evaluation.

Population	CG	Screened	Bernstein	Full evaluation
Common mass 0.5	0.9281 ± 0.0080	0.9279 ± 0.0085	0.9247 ± 0.0085	0.9487 ± 0.0038
Common mass 0.8	0.9727 ± 0.0030	0.9713 ± 0.0033	0.9712 ± 0.0033	0.9396 ± 0.0054
Common mass 0.9	0.9833 ± 0.0020	0.9858 ± 0.0016	0.9842 ± 0.0018	0.9536 ± 0.0020
Independent candidates	0.4732 ± 0.0038	0.4714 ± 0.0034	0.4703 ± 0.0035	0.4787 ± 0.0030

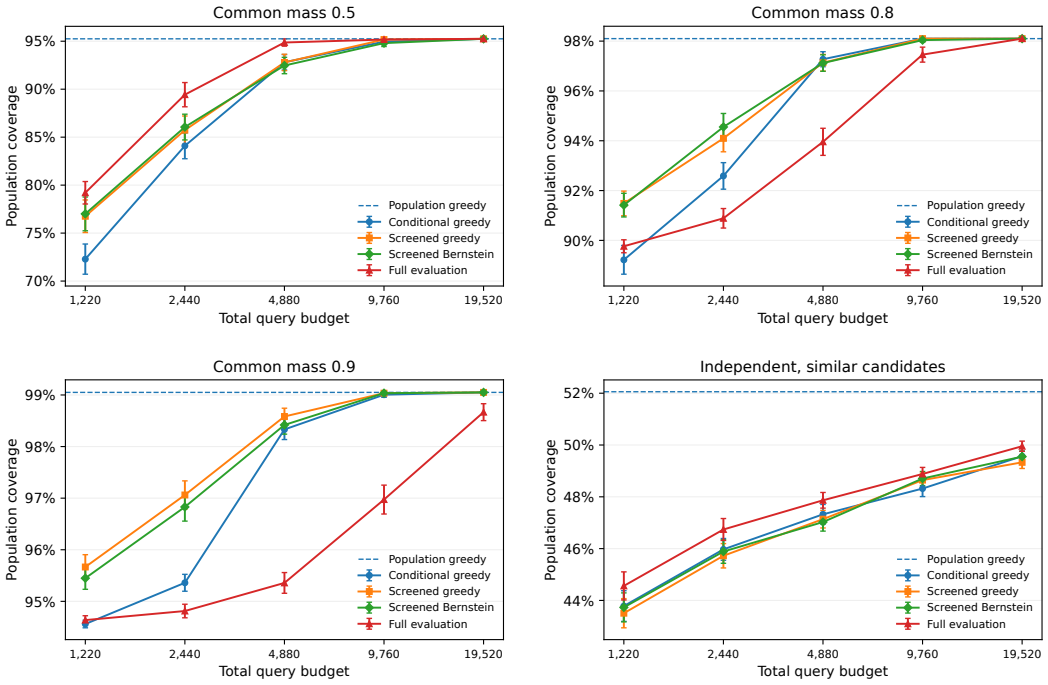


Figure 6: All four binary populations and all five budgets. Bars show 1.96 standard errors over 200 independent repetitions. Conditional sampling is not uniformly better than either full evaluation or the screened baselines.

finishes within those budgets. Table 1 gives the comparison at $Q = 4{,}880$, and Figure 6 gives every tested budget.

E.4 Reproducing and Checking the Results

The `supplementary/` folder contains one implementation of each retained experiment in its `experiments/` subdirectory, with no external data dependency. Paths below are relative to `supplementary/`. The binary runner implements the four budget-limited methods; the controlled runner implements the three paired tests and the exact cyclic games. The ordinal experiments do not implement or benchmark the general comparison-based lottery algorithm. The theoretical claims are supported by the proofs in the preceding appendices, not by the finite numerical checks.

Environment and seeds. The reconciled experiments were run with Python 3.13.5, NumPy 2.3.5, SciPy 1.17.0, pandas 2.2.3, and Matplotlib 3.10.8 on a Linux CPU, without a GPU. The direct dependencies are pinned in `experiments/requirements.txt`, and the supplied lock file also records the versions of their dependencies. We set `OPENBLAS_NUM_THREADS=1` and

`OMP_NUM_THREADS=1`. Environment records and measured running times accompany the results; they are not performance benchmarks.

For binary trials, the seed tuple is $(20260925, p, b, r, a)$, where p indexes the populations in the order $0.5, 0.8, 0.9$, independent; b indexes increasing budgets; $r = 0, \ldots, 199$; and a indexes conditional greedy, screened greedy, full evaluation, and screened Bernstein in that order. NumPy's `SeedSequence` spawns separate streams for task outcomes and tie-breaking. Paired trials use $(20260926, s, a, r)$, with settings $s = 0, \ldots, 19$ ordered by the disagreement, miss-rate, gap, and committee-size sweeps above, increasing within each sweep; methods are separate means, paired Bernstein, and the disagreement test; and $r = 0, \ldots, 99$. The cyclic experiment is deterministic and resolves ties by candidate index.

Commands and generated outputs. Download the supplementary folder, install the pinned packages, check the retained records, and regenerate the manuscript's figures and numbers with

```
cd supplementary
python -m pip install -r experiments/requirements-lock.txt
python -m unittest discover -s experiments/tests -v
python -m experiments.verify_results
python -m experiments.make_figures
python build_paper.py
```

The last command requires a LaTeX installation with pdfLaTeX and BibTeX. A full independent rerun is

```
python reproduce.py --output reproduced
```

This command creates a fresh output directory, reruns every trial, verifies the result tables, compares them with the retained records, and regenerates the figures and numerical values there. It does not overwrite the retained results. A smaller execution check is available through `-smoke`; it is not a reproduction of the reported statistical comparisons. The lower-level runners accept explicit output directories, repetition counts, and master seeds.

Checks and limits. For binary trials, verification checks committee size and uniqueness, seed assignment, query totals and budgets, recorded forced selections, and all 80 summary rows. It recomputes every committee's coverage using an independent implementation of the population formulas. For paired trials, an independent routine replays all 6,000 stopping decisions and their represented waiting times, and recomputes all 60 median and quartile summaries. For each of the 60 cyclic populations, verification independently constructs the payoff matrix and solves primal and dual linear programs; it also replays all 180 multiplicative-weights configurations. Unit tests cover cached answers, distinct task draws, budget exhaustion, partial rounds, and degenerate populations.

The full reconciled rerun matches all 16,000 binary trial records, all 6,000 paired trial records, and all 180 cyclic rows, with numerical comparisons at tolerance 10^{-12} for the stored tables. The default verification command replays 16 representative binary trials in addition to checking every binary record; `-replay full` replays all of them. Cyclic linear programs have feasibility and primal–dual checks at tolerance 10^{-8}. We do not require byte-identical PDF files or identical linear-program solutions on other platforms: numerical values are the comparison target, and an optimal lottery need not be unique. Reproducing these controlled examples does not establish performance on model outputs or human preferences.