MARKETS OF MINDS: A 51-BILLION-STATE ECONOMY OF INTELLIGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

When intelligence is shared, reasoning becomes a constrained allocation problem. In a population of agents with limited access to expert models, an intervention can benefit its recipient yet still be a poor use of shared capacity if another feasible intervention has greater expected value. Assistance must therefore be evaluated through both its marginal benefit and the opportunity cost induced by budgets, access, scarcity, and service capacity. To this end, we introduce Markets of Minds, a framework for allocating expert reasoning across agents before expert outputs are observed. For each feasible agent–service pair, the framework estimates the expected gain from assistance relative to the agent's default continuation, explicitly separating repair benefit from regression risk, and jointly clears assistance contracts under shared resource constraints. We then study this problem in a dependency-aware agent economy with 51.2 billion addressable persona–market states spanning capabilities, cognitive budgets, provider access, scarcity, and network roles. These dependencies couple allocation feasibility across agents rather than the outcome of a fixed contract. Empirically, in a matched replay of 240 cases under a common 30% assistance quota, the frozen allocator reaches 85.42% task-level persona adherence versus 73.86% for equal-capacity random allocation, a gain of 11.56 percentage points. The results show that assistance value is heterogeneous and can be negative, and that scarcity changes which interventions are worth purchasing. Together, these findings motivate a population-level view of intelligence where the value of assistance depends not only on what it can achieve, but also on which alternative uses of scarce cognitive capacity it displaces.

1 INTRODUCTION

Intelligence can be abundant in capability yet scarce in access. Large language models increasingly serve as agents that browse, plan, use tools, and act on behalf of different users (Wu et al., 2023; Hong et al., 2023). Expanding this population, however, does not necessarily expand the cognitive services available to it. Access to stronger reasoning, specialized expertise, long-context processing, or reliable verification may remain bounded by cost, latency, concurrency, and provider restrictions (Chen et al., 2023; Ong et al., 2025; Feng et al., 2025). A system can therefore support many capable agents while lacking the capacity to assist all of them when needed. This creates a population-level question: *when many agents need help, which requests should receive scarce cognition?*

Need alone is not value under scarcity. Under shared capacity, assistance decisions are coupled: assigning a service slot to one agent may displace an intervention with greater expected benefit elsewhere, while spending resources on one request reduces what remains available to others (Mehta et al., 2007; Badanidiyuru et al., 2013; Zhang & Cheung, 2022). The gain from help is also uncertain and can be negative (Aggarwal et al., 2024; Ong et al., 2025). An agent may already have a sufficient answer, an expert may fail to repair its difficulty, or additional advice may disrupt an otherwise correct solution. The allocator must therefore distinguish the expected gain from assistance relative to the agent's default continuation from the opportunity cost of committing shared resources to it. A locally beneficial intervention can still be a poor population-level allocation (Badanidiyuru et al., 2013; Zhang & Cheung, 2022). Crucially, this decision must be made before observing the corresponding expert output.

We formalize this setting as *multi-agent assistance under shared cognitive budgets* (Section 2). In a decision window, each request either continues under its default policy or receives at most one fea-

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.

sible typed service. The objective is to maximize expected downstream task performance across the population subject to individual budgets, provider access, shared resource limits, service capacity, and deadlines. These constraints determine which combinations of interventions are feasible, so resources committed to one agent restrict the choices available to others. Dependencies across agents enter through budgets, access, scarcity, capacity, network roles, and the feasible set; conditional on a fixed contract, we do not model another agent's assignment as changing its task outcome.

We introduce *Markets of Minds* (Section 3), a framework that represents cognitive assistance as bounded, typed contracts and allocates them before expert execution. For each feasible agent–service pair, the framework estimates the marginal value of assistance relative to the agent's default continuation. The formulation separates three quantities: the probability that the default continuation already succeeds, the probability that a service repairs a default failure, and the risk that assistance regresses an otherwise successful outcome. A joint clearing step converts these values into contracts subject to budget, access, capacity, and deadline constraints, accounting for competing uses of resources (Zhang et al., 2026b; Zhou et al., 2026; Alazraki et al., 2026). Only selected services are executed, and a common ledger records reservation, consumption, and settlement. The problem is therefore not to select among already generated responses, but to decide which responses are worth acquiring before their realized value is known.

To study how economic context shapes this decision, we construct a dependency-aware agent economy that crosses 999,847 base personas with 10 cognitive budgets, 20 expertise portfolios, 8 provider-access regimes, 4 scarcity regimes, and 8 network roles. The resulting design space contains exactly $51{,}192{,}166{,}400$ addressable persona–market states. A versioned sampling procedure draws auditable cohorts, and experiments materialize only the states required for a given hypothesis. The construction allows budgets, expertise, provider access, supply conditions, and network roles to vary over the same underlying population. Economic context therefore becomes an experimental variable rather than a fixed background assumption, while persona demographics remain distinct from the capabilities and resources assigned to an agent.

Our evaluation separates evidence about allocation from evidence about service capability and population scale. The primary matched replay deliberately instantiates a one-service setting with a fixed assistance quota, isolating the question of *who should receive help*. Across 240 interactive agent cases with actor and judge models held fixed, all deployable policies are evaluated against the same cached default and assisted outcomes under a common 30% assistance quota. A frozen allocator achieves 85.42% task-level persona adherence, compared with 73.86% for equal-capacity random allocation, a gain of 11.56 percentage points without increasing the number of assisted cases. A complementary study across six multimodal benchmarks characterizes where different services are useful and where their failures concentrate. These experiments address distinct questions: recipient selection under fixed supply and heterogeneity in service capability. The broader heterogeneous-service clearing formulation defines the general framework, while empirical claims are tied to the evaluated instantiations. Together, the formulation, framework, and experiments support a population-level view of cognitive assistance under shared constraints. The key question is not simply whether expert help improves an agent's outcome, but whether it is the best use of scarce capacity relative to the alternatives it displaces. Intelligence is thus both an individual capability and a shared resource, whose value depends on both intervention gain and opportunity cost.

2 Problem Formulation

We study how a population of task agents should share limited access to cognitive services such as specialist reasoning, verification, and tool support. Consider n agents and m services in a single decision window. Agent i has an observable state s_i, comprising its task, execution history, and available resources. Let $Y_i^0 \in \{0,1\}$ denote default success and $Y_{ij}^1 \in \{0,1\}$ denote assisted success after receiving service j. We assume that s_i is sufficient for predicting these outcomes and that other agents' assignments affect feasibility but do not change the outcome of a fixed contract. Thus cross-agent dependence enters through shared budgets, access, scarcity, capacity, and the feasible allocation set.

Allocation maximizes the expected assistance gain:

$$\Delta_{ij} \triangleq \mathbb{E}[Y_{ij}^1 - Y_i^0 \mid s_i], \qquad x^\star \in \arg\max_{x \in \mathcal{X}} \sum_{i,j} x_{ij} \Delta_{ij}, \tag{1}$$

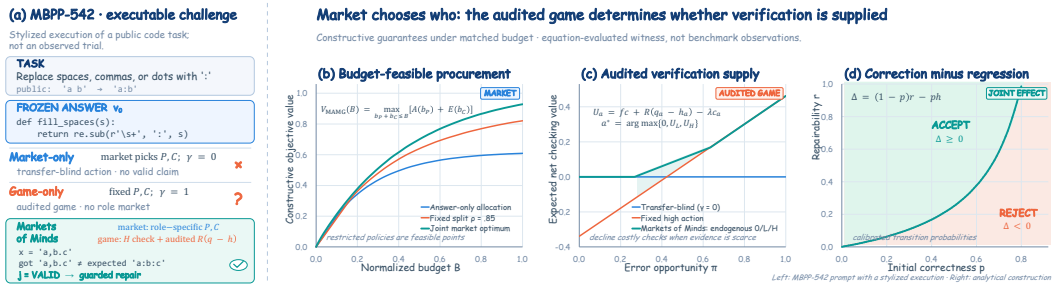


Figure 1: **Allocation and verification.** A stylized execution and analytical illustrations of procurement, verification supply, and net correction.

where the assignment x_{ij} indicates whether agent i receives service j, and the feasible set $\mathcal{X}$ encodes capacity, budget, and access constraints with at most one service per request. Allocation must precede observing the assisted response and either outcome label. Section 3 instantiates this objective using observable request features (Equation (4)) and estimated contract values (Equation (5)).

Assistance value separates need from service effect. A low probability of default success indicates that an agent may need help, but does not establish that an available service can repair its failure. Assistance can also disrupt an otherwise successful continuation. For binary outcomes, these two effects admit an exact decomposition.

Proposition 1 (Assistance value)**.** *The expected value of an assistance contract factorizes into default-state occupancy and the signed transition induced by the service:*

$$\Delta_{ij} = \begin{bmatrix} 1 - p_i & p_i \end{bmatrix} \begin{bmatrix} r_{ij} \\ -h_{ij} \end{bmatrix} = (1 - p_i) r_{ij} - p_i h_{ij}, \tag{2}$$

Here, $p_i = \Pr(Y_i^0 = 1 \mid s_i)$ is default success, $r_{ij} = \Pr(Y_{ij}^1 = 1 \mid Y_i^0 = 0, s_i)$ is conditional repairability, and $h_{ij} = \Pr(Y_{ij}^1 = 0 \mid Y_i^0 = 1, s_i)$ is conditional regression risk. The identity follows by conditioning on Y_i^0: $\mathbb{E}[Y_{ij}^1 \mid s_i] = (1 - p_i) r_{ij} + p_i(1 - h_{ij})$. The 1×2 row gives default-state occupancy; the 2×1 column gives the signed repair and regression effects. The key distinction is therefore between *needing help* and *benefiting from the available help*. A complete proof appears in Appendix E.

Fig. 1 combines executable checking (a), budget-feasible procurement (b), verification supply (c), and net correction (d) in Equation (2). These constructions are analytical illustrations rather than benchmark observations, and our allocation model does not assume the depicted audited game. Panel (d)'s accept region means non-negative gain, not guaranteed selection. At fixed regression risk h and $p < 1$, its boundary is $r = ph/(1 - p)$.

Difficulty alone can misallocate assistance. Consider requests with $(p_i, r_{ij}, h_{ij}) = (0.1, 0, 0)$ and $(0.6, 1, 0)$. The first is more likely to fail, yet has assistance value zero; the second has value 0.4. With one slot, a difficulty-first rule selects the irreparable case. Conversely, even a capable service can have negative value when regression outweighs repair. Section 3 therefore estimates service effects before forming Equation (5), rather than treating difficulty itself as assistance value.

Population welfare depends on value estimation. Expected population reward is $W(x) = \sum_i p_i + \sum_{i,j} x_{ij} \Delta_{ij}$: a fixed default baseline plus the reward added by allocation. For one service and exactly K assignments, the optimum selects the K largest Δ_i values; when K is only an upper bound, non-positive values need not be selected. Because these contract values are unknown before clearing, errors in value estimation can translate directly into errors in population allocation.

Proposition 2 (Allocation error)**.** *Let $\widehat{x}$ maximize $\sum_{ij} x_{ij} \widehat{\Delta}_{ij}$ over the feasible set of $x^\star$. If allocations contain at most K contracts and $|\widehat{\Delta}_{ij} - \Delta_{ij}| \le \epsilon$ for all feasible pairs, then*

$$W(x^\star) - W(\widehat{x}) \le \sum_{i,j} (x_{ij}^\star + \widehat{x}_{ij}) |\widehat{\Delta}_{ij} - \Delta_{ij}| \le 2K\epsilon. \tag{3}$$

The first inequality follows by inserting the predicted objectives and using optimality of $\widehat{x}$. Each allocation contains at most K contracts; the uniform estimation error ϵ gives the second inequality. For an exact top-K quota with $1 \le K < n$, the selected set also matches the oracle whenever

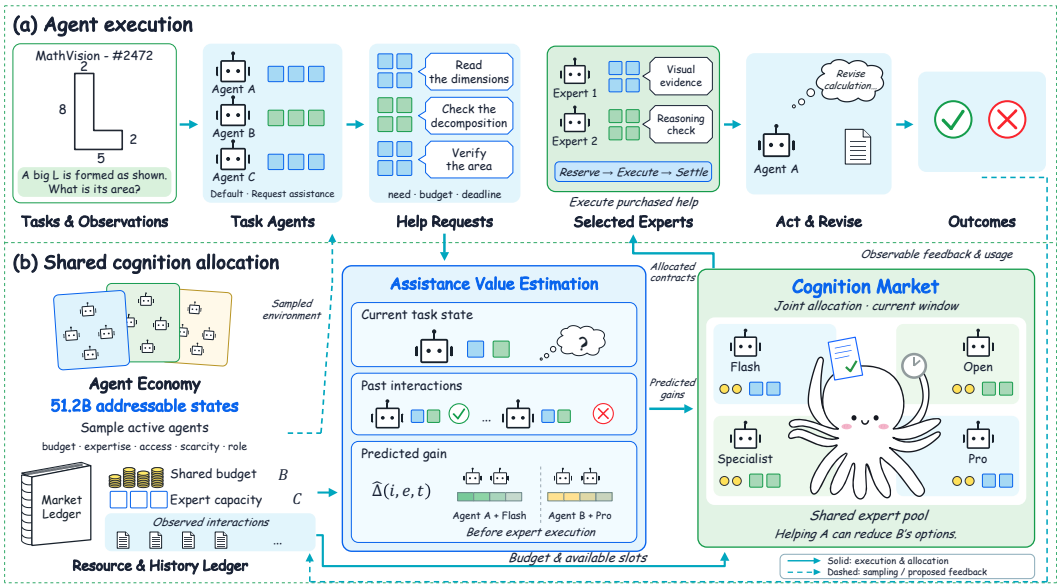


Figure 2: **Markets of Minds.** (a) Task-agent execution. (b) Pre-execution value estimation, joint allocation, and resource accounting.

$\Delta_{(K)} - \Delta_{(K+1)} > 2\epsilon$, with values ordered from largest to smallest. This is a conditional guarantee: average predictive calibration alone does not establish the required uniform-error premise. The complete argument, including the top-K recovery condition, appears in Appendix E.

Together, Propositions 1 and 2 connect individual assistance effects to population-level allocation. The first identifies what makes an intervention valuable; the second shows why estimating that value accurately matters once multiple agents compete for the same cognitive capacity.

3 Markets of Minds

The formulation above separates two requirements for allocating scarce cognition. Proposition 1 identifies what must be estimated for each feasible contract, *i.e.*, default success, conditional repair, and regression risk, while Proposition 2 explains why errors in these values matter once agents compete for shared capacity. *Markets of Minds* turns this decomposition into a pre-execution allocation pipeline: represent observable cognitive demand, estimate prospective assistance value, clear contracts jointly under shared constraints, and execute only the selected assistance. Fig. 2 summarizes this process.

The value-estimation block implements Equations (4) and (5); the market clears contracts under the feasible set of Section 2. In the diagram, $\widehat{\Delta}(i, e, t)$ denotes $\widehat{\Delta}_{ij}$ with expert $e = j$ and the window t fixed. The ledger supplies the budgets and capacities available at clearing time. Reservation, execution, and settlement occur only after the allocation has been fixed, so unobserved expert outputs cannot influence the decision that selects them.

Represent observable cognitive demand. Agent i submits a demand packet $D_i = (q_i, o_i, \theta_i, g_i, b_i, \mathcal{A}_i, d_i)$ containing its task, observable base artifact, task-relevant persona context, desired behavior or missing evidence, remaining credits, accessible services, and deadline. The packet excludes both the target outcome and any unexecuted service response, and therefore provides a structured view of the observable state s_i introduced in Section 2. A demand auditor maps it to

$$\mathbf{a}_i = (u_i^{\text{per}}, u_i^{\text{sem}}, u_i^{\text{inf}}, \kappa_i, m_i)^\top, \qquad \mathbf{z}_i = [\mathbf{a}_i;\, \boldsymbol{\rho}_i], \qquad \mathbf{z}_{ij} = [\mathbf{z}_i;\, \mathbf{e}_j]. \tag{4}$$

The five demand features in $\mathbf{a}_i$ identify possible failures in perception, interpretation, inference, preference resolution, and execution; they are observable features rather than outcome labels. The request representation $\mathbf{z}_i$ adds historical statistics $\boldsymbol{\rho}_i$ from similar development requests. The request–service representation $\mathbf{z}_{ij}$ further adds the service descriptor $\mathbf{e}_j$ so that the predicted effect can depend on which service is considered. Here $[\cdot;\cdot]$ denotes column concatenation; the three vectors

have dimensions 5, d_{req}, and d, respectively. With one service, $\mathbf{e}_j$ is constant across requests and does not affect their ranking.

Estimate prospective assistance value. The relevant quantity is not whether a request appears difficult, but whether a particular service is expected to improve its outcome relative to the default continuation. On paired development cases, we can observe both the default and assisted continuations: cases with $Y_i^0 = 0$ reveal repair, and cases with $Y_i^0 = 1$ reveal regression. The full component-wise estimator fits $\widehat{p}_i$ on all development cases using request features $\mathbf{z}_i$, then fits $\widehat{r}_{ij}$ on default failures and $\widehat{h}_{ij}$ on default successes using request–service features $\mathbf{z}_{ij}$. Each head minimizes binary cross-entropy on its own outcome stratum; regularization and group smoothing are selected only on development folds.

The frozen estimates instantiate the decomposition in Equation (2) and the objective in Equation (1) through one plug-in decision:

$$\widehat{\Delta}_{ij} = (1 - \widehat{p}_i)\widehat{r}_{ij} - \widehat{p}_i\widehat{h}_{ij}, \qquad \widehat{x} \in \arg\max_{x \in \mathcal{X}} \sum_{i,j} x_{ij}\widehat{\Delta}_{ij}. \tag{5}$$

The default-success estimate $\widehat{p}_i$ weights the two initial states; repairability $\widehat{r}_{ij}$ and regression risk $\widehat{h}_{ij}$ give their conditional transition effects. Their weighted difference is the estimated contract gain $\widehat{\Delta}_{ij}$. **Clear contracts under shared constraints.** The second part of Equation (5) converts these local contract values into a population-level allocation. The feasible set $\mathcal{X}$ imposes one service per request, buyer budgets, supplier capacities, shared resource limits including broker overhead, access permissions, and deadlines. Consequently, a contract can have positive estimated value and still remain unselected when another feasible use of the same capacity has greater value. This is precisely the opportunity-cost effect represented by the population objective in Section 2. Proposition 2 connects errors in these estimated values to the resulting allocation loss. Algorithm 1 separates development fitting from target inference and executes only cleared contracts.

Evaluated instantiation. The matched replay deliberately isolates the recipient-selection problem. It has one service and exactly K_e slots per environment. The auditor also supplies default-success and repair-gain estimates $\widetilde{p}_i$ and $\widetilde{g}_i$. The implemented calibrator fits $v_i = f_\beta(\mathbf{a}_i, \widetilde{p}_i, \widetilde{g}_i, \boldsymbol{\rho}_i) \approx \Pr(Y_i^0 = 0, Y_i^1 = 1 \mid D_i)$ on development cases; clearing then selects the top K_e requests by v_i. This estimates the positive-effect term of Equation (2); it does not separately estimate regression risk. The current result therefore tests *demand-aware recipient selection under fixed assistance supply*. The three-head estimator and heterogeneous-service clearing in Algorithm 1 define the general framework, while the experiments instantiate it in the one-service recipient-selection setting.

Execute and account only after clearing. Only after the allocation is frozen do selected agents receive the service; unselected agents continue by default. Matched replay uses the same trajectories for every policy and joins target outcomes after selection. This separation ensures that the allocator cannot use the realized assisted outcome to decide whether assistance should have been purchased. The ledger records outcomes, credits, resource use, latency, failures, and API-equivalent cost. In Algorithm 1, $\widehat{\mathcal{C}}$ is the purchased contract set, $\mathcal{O}$ contains observed execution results and usage, and $\mathcal{L}^-, \mathcal{L}^+$ are the reserved and settled ledger states. The dashed feedback path in Fig. 2 indicates possible future reuse of logged interactions; it is not updated on target-window outcomes.

4 Experiments

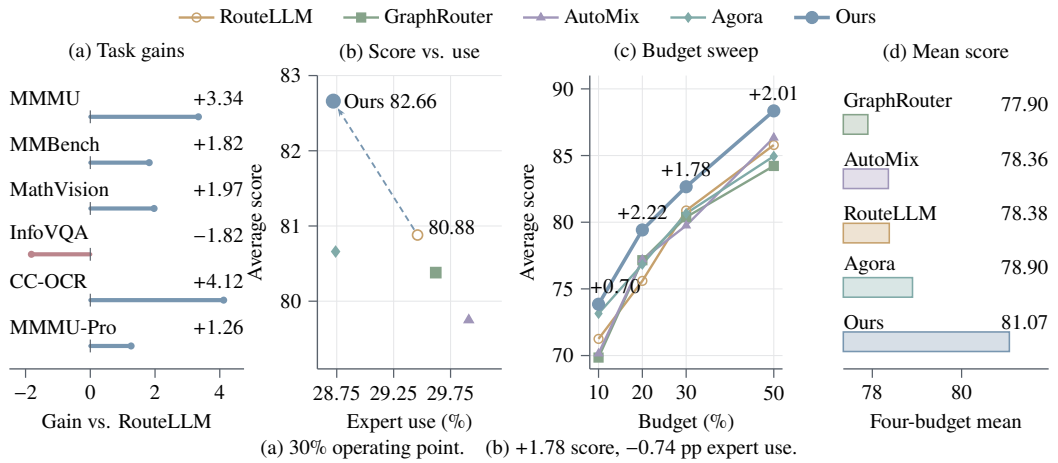
We evaluate the empirical questions induced by Sections 2–3. We first test whether a frozen pre-execution allocator improves recipient selection when every policy receives the same expert capacity. We then examine whether the resulting gains persist across heterogeneous multimodal tasks and expert-use budgets, followed by an explicit shared-capacity population setting. Finally, we ablate the signals used by the implemented allocator. These experiments evaluate the instantiated allocation policies in Section 3, with the matched replay focusing on one-service recipient selection and the remaining studies examining task heterogeneity and shared-capacity allocation.

4.1 Matched Agent Allocation

Protocol. We first evaluate the one-service recipient-selection setting described in Section 3. The matched replay contains 240 interactive agent cases, with 80 cases each from Chat, Web, and OS-

Table 1: **Matched allocation at 30% shared expert capacity.** 80 cases and 24 assistance slots per environment; cost/success includes method-specific overheads.

Method	Chat	Web	OS-App	Overall ↑	Expert use	Cost/succ. ↓	Harmful help ↓
No assistance	68.75	61.25	65.00	65.00	0%	$0.06211	–
Random	77.40	70.59	73.60	73.86	30%	$0.06090	0.42%
Difficulty/uncertainty	83.75	76.25	85.00	81.67	30%	$0.05636	0.00%
AutoMix	82.50	76.25	81.25	80.00	30%	$0.07765	0.00%
RouteLLM	86.25	77.50	80.00	81.25	30%	$0.06219	0.00%
GraphRouter	86.25	77.50	80.00	81.25	30%	$0.06023	0.00%
Agora	86.25	**80.00**	82.50	82.92	30%	$0.05778	**0.00%**
Ours	**91.25**	78.75	**86.25**	**85.42**	**30%**	**$0.05509**	**0.00%**



(a) 30% operating point. (b) +1.78 score, −0.74 pp expert use.
(c) Gains vs. best baseline; Ours at 50% = 88.35. (d) +2.17 vs. best baseline.

Figure 3: **Main routing results and budget robustness.** Task-level gains, score versus expert use, performance across four expert-use budgets, and the corresponding four-budget mean.

App. Every case has cached default and assisted potential outcomes that are shared across allocation policies. GPT-5.6 Sol serves as the actor and Claude Opus 4.8 as the judge. Only the Survey split is used for development. Within each environment, an assistance policy may allocate at most 24 expert interventions, yielding 72 interventions in total and a common 30% expert-capacity limit. Thus all policies operate on the same cases, expert supply, and evaluation outcomes. Overall denotes task-level persona adherence.

Observation 1: Recipient selection matters even when assistance supply is fixed. Table 1 shows that Markets of Minds reaches **85.42** overall under the same 30% expert-capacity limit used by all assistance policies. Equal-capacity random allocation reaches 73.86, giving an **11.56-point** difference, while the strongest comparison policy, Agora, reaches 82.92, leaving a **2.50-point** gap. Markets of Minds is highest on Chat and OS-App, whereas Agora is higher on Web, indicating that the value of recipient selection varies across environments. Because the cases, cached potential outcomes, and expert supply are matched across policies, this comparison isolates allocation quality rather than additional expert use. The same table also shows that higher task performance does not require higher realized expert consumption: all assistance policies use 30% capacity. Markets of Minds obtains the lowest cost per successful task (**$0.05509**) and records no harmful-help events in this replay, tied with several comparison policies on the latter diagnostic.

4.2 Multimodal Benchmark Results

Benchmarks and metrics. We complement the matched agent replay with a six-benchmark multimodal study using MMMU (Yue et al., 2024), MMBench (Liu et al., 2024), MATH-Vision (Wang et al., 2024b), InfoVQA (Mathew et al., 2022), CC-OCR (Yang et al., 2025), and MMMU-Pro (Yue et al., 2025). We evaluate on the full designated evaluation splits and report benchmark-specific

Table 2: **Multimodal benchmark results.** Routing entries report three-run means; **bold** marks the best routing result in each column.

Method	MMMU	MMBench	Math Vision	InfoVQA	CC-OCR	MMMU-Pro	Avg.	Expert use
Single-model references								
Gemini 3.5 Flash	76.11	92.99	86.51	78.90	71.42	78.03	80.66	—
Gemini-2.5-Pro	81.70	88.30	63.50	81.00	73.00	68.00	75.92	—
Gemini-2.0-Flash	70.70	83.00	41.30	83.20	73.10	57.00	68.05	—
GPT-5.6 Sol	82.22	83.90	84.54	73.58	79.62	79.77	80.60	—
GPT-4o (2024-08-06)	70.70	74.30	30.40	68.70	66.60	54.50	60.87	—
Claude Sonnet 5	65.00	83.90	74.67	62.31	74.26	69.94	71.68	—
InternVL3-78B	72.20	87.70	43.10	84.10	80.30	55.10	70.42	—
Qwen2.5-VL-72B	70.20	88.40	38.10	87.30	79.80	51.10	69.15	—
Routing methods (three-run means)								
RouteLLM	74.44	91.69	84.38	**78.84**	76.73	79.19	80.88	29.46%
GraphRouter	74.12	91.45	84.05	78.15	76.10	78.43	80.38	29.62%
AutoMix	73.55	90.72	83.81	77.64	74.98	77.82	79.75	29.91%
Agora	**78.33**	92.73	86.02	76.87	73.14	76.88	80.66	28.74%
Ours	77.78	**93.51**	**86.35**	77.02	**80.85**	**80.45**	**82.66**	**28.72%**

scores on a 0–100 scale together with their unweighted mean (*Avg.*). Table 2 reports three-run means and realized expert use, while Fig. 3 summarizes task-level differences and performance across expert-use budgets.

Models and comparisons. Gemini 3.5 Flash (Google DeepMind, 2026) provides the base response. The assistance pool comprises Gemini-2.5-Pro (Google DeepMind, 2025), GPT-5.6 Sol (OpenAI, 2026), Claude Sonnet 5 (Anthropic, 2026), InternVL3-78B (Zhu et al., 2025), and Qwen2.5-VL-72B. We compare Markets of Minds with matched adaptations of RouteLLM (Ong et al., 2025), GraphRouter (Feng et al., 2025), AutoMix (Aggarwal et al., 2024), and Agora (Zhang et al., 2026a) under a common assistance interface. Standalone models are reported separately as performance references rather than resource-matched allocation baselines.

Allocation and reporting. Each request may receive at most one expert intervention; unselected requests retain their base response. All routing policies share the same response pool, eligible requests, and resource constraints, with a nominal assistance rate of 30%. Allocation decisions are fixed before the corresponding evaluation outcomes are used for scoring. Expert use is the realized percentage of evaluated requests assigned to an expert, averaged over three runs.

Observation 2: Assistance value is heterogeneous across tasks. At the 30% operating point, Table 2 shows that Markets of Minds reaches **82.66** average score with **28.72%** expert use, compared with 80.88 and 29.46% for RouteLLM, the highest-scoring comparison router by average score. The difference is therefore **+1.78 points** while using **0.74 percentage points fewer** expert calls. The gain is not uniform across benchmarks: Markets of Minds has the highest routing score on MMBench, MATH-Vision, CC-OCR, and MMMU-Pro, whereas Agora is higher on MMMU and RouteLLM is higher on InfoVQA. This variation is consistent with the formulation in Section 2: expert capability alone does not determine the value of assigning that expert to a particular request.

Observation 3: The allocation advantage persists as expert supply changes. Fig. 3 shows that the difference is not confined to the nominal 30% operating point. Relative to the strongest comparison at each budget, Markets of Minds gains +0.70, +2.22, +1.78, and +2.01 points at 10%, 20%, 30%, and 50% expert-use budgets, respectively. Averaged across the four budgets, Markets of Minds reaches **81.07**, compared with 78.90 for the strongest comparison average, a difference of **+2.17 points**. Together with the task-level variation in Table 2, the sweep indicates that which interventions are valuable depends on both the request and the amount of assistance available.

4.3 Population-Level Allocation under Shared Capacity

The previous experiments evaluate recipient selection and budget sensitivity. We next expose the population coupling emphasized in Section 2 by imposing a single expert-capacity constraint shared across requests. Global capacity is capped at 10%, 30%, or 50% of request volume. Fig. 4 reports

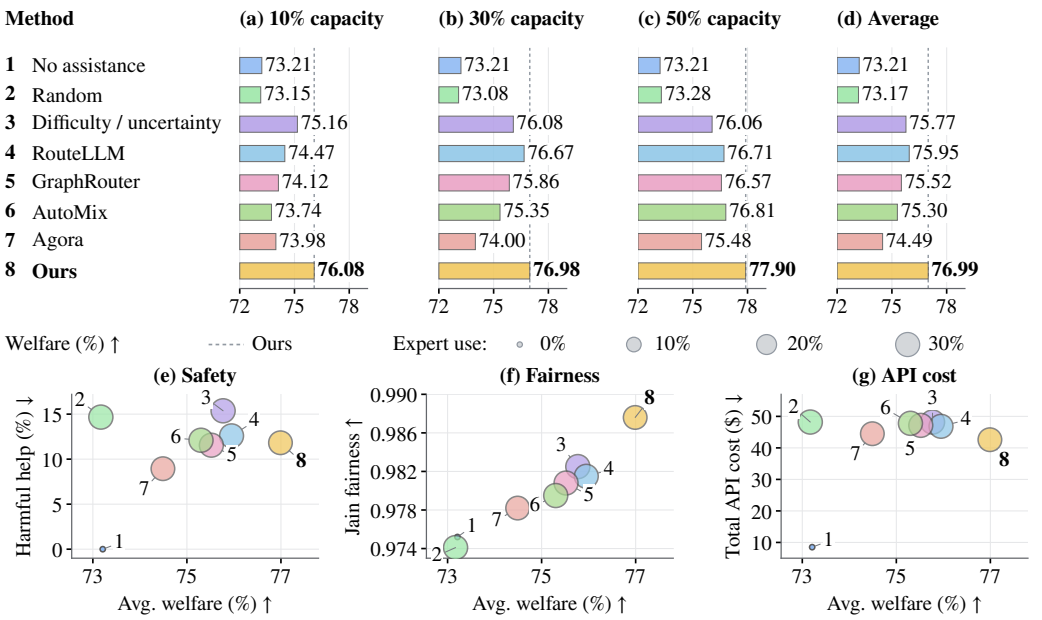


Figure 4: **Population-level allocation under shared capacity.** The bar charts show welfare under 10%, 30%, and 50% global capacity and their average; the bubble charts show welfare relative to harmful help, Jain fairness, and total API cost.

population welfare together with three ex-post deployment diagnostics: *Harmful Help*, the frequency with which expert intervention degrades a correct base response; Jain fairness across task groups; and total API-equivalent cost. Harmful Help measures realized regression after execution.

Observation 4: Scarcity changes the value of allocation decisions. Across the three capacity levels in Fig. 4, Markets of Minds records welfare of **76.08**, **76.98**, and **77.90** at 10%, 30%, and 50% capacity, respectively, the highest values among the plotted allocation methods in each setting. Its three-budget average is **76.99**, compared with 75.95 for RouteLLM. The difference relative to RouteLLM is largest at 10% capacity, where Markets of Minds reaches 76.08 versus 74.47. As capacity tightens, committing an expert slot to one request excludes more alternative uses of that same resource, making accurate contract ranking increasingly consequential.

Observation 5: Welfare gains coexist with distinct safety, fairness, and cost trade-offs. The right-hand panels of Fig. 4 show that average welfare alone does not characterize an allocation policy. Markets of Minds lies in the high-fairness region and has the highest plotted Jain fairness. Its harmful-help rate is lower than the difficulty/uncertainty heuristic, while Agora attains a lower harmful-help rate at lower average welfare. The API-cost panel similarly places Markets of Minds at higher welfare and lower total cost than several learned routing baselines. These comparisons are consistent with the opportunity-cost interpretation of Section 2: the quality of an allocation depends not only on whether an intervention helps locally, but also on the competing use of the shared capacity it consumes.

4.4 Ablation Study

We finally examine which signals used by the implemented calibrator contribute to recipient selection. All variants use the same actor–expert pool, full target evaluation sets, and 30% shared expert capacity. For each variant, one component is removed, the allocator is retrained using only the Survey development split, and the resulting policy is frozen before target evaluation. Table 3 reports target success, its difference from the full model, cost per successful task, and development Survey performance. These ablations concern the implemented inputs $(\mathbf{a}_i, \widetilde{p}_i, \widetilde{g}_i, \boldsymbol{\rho}_i)$ described in Section 3.

Observation 6: Default-success and repair-gain signals are the largest contributors to recipient selection. Table 3 shows that the full allocator achieves the highest target success (**82.66**) and the lowest cost per successful task (**$0.0233**). Among the leave-one-component-out variants, removing repair-gain produces the largest decrease ($-\mathbf{3.98}$ points), followed by default-success ($-\mathbf{3.11}$),

Table 3: **Cognition allocator ablation.** Three-seed mean. Δ_{full} denotes the success gap relative to the full model in percentage points.

Variant	Success ↑	Δ_{full} (pp)	Cost/succ. ↓	Survey ↑
Restricted model				
Priors only	74.22	−8.44	$0.0265	70.15
Leave-one-component-out				
w/o market prior	79.84	−2.82	$0.0252	74.50
w/o uncertainty	81.15	−1.51	$0.0241	76.10
w/o preference conflict	80.32	−2.34	$0.0245	75.25
w/o execution signal	81.62	−1.04	$0.0238	76.50
w/o default-success	79.55	−3.11	$0.0255	74.80
w/o repair-gain	78.68	−3.98	$0.0258	73.90
Full model	**82.66**	–	**$0.0233**	**77.85**

market prior ($-\mathbf{2.82}$), and preference conflict ($-\mathbf{2.34}$). Removing uncertainty and execution signals produces smaller decreases of 1.51 and 1.04 points, respectively. The priors-only model falls by 8.44 points, showing that population-level statistics alone do not replace request-conditioned signals. This ordering is consistent with the evaluated positive-effect construction in Section 3: effective recipient selection depends jointly on whether the default continuation is likely to fail and whether assistance is likely to repair that failure.

Observation 7: The same signals improve the efficiency of scarce expert use. Every component removal in Table 3 also increases cost per successful task. Among the leave-one-component-out variants, the increase ranges from approximately 2.1% to 10.7%, while the priors-only model is 13.7% more expensive per success. Thus the signals that improve recipient selection also improve how efficiently the fixed expert budget is converted into successful outcomes.

5 Related Work

Markets of Minds connects inference-time routing, multi-agent coordination, and shared-capacity allocation. Cost-aware methods such as FrugalGPT, AutoMix, RouteLLM, and GraphRouter selectively invoke stronger models under quality–cost trade-offs (Chen et al., 2023; Aggarwal et al., 2024; Ong et al., 2025; Feng et al., 2025), while multi-agent systems organize specialized models through roles and collaboration (Wu et al., 2023; Hong et al., 2023; Wang et al., 2024a). Related market-based approaches allocate scarce reasoning or expert resources through uncertainty, auctions, and strategic bids (Zhang et al., 2026b; Zhou et al., 2026; Alazraki et al., 2026), while MarketBench highlights the difficulty of calibrating agent success and resource use (Fradkin & Krishnan, 2026). Our setting differs by coupling allocation across task agents before expert outputs are observed: assistance is valued by its predicted marginal effect on the recipient's downstream outcome under shared capacity. Population-scale agent simulations provide a complementary view of heterogeneity (Park et al., 2023; Piao et al., 2025; Li et al., 2026); we further treat budgets, access, scarcity, and network roles as explicit economic state. A fuller discussion appears in Appendix A.

6 Conclusion

We frame cognitive assistance as a population-level allocation problem under shared resource constraints, where need is not equivalent to value: whether an agent should receive help depends on default success, repairability, regression risk, and the opportunity cost of consuming scarce capacity. Markets of Minds operationalizes this view through pre-execution value estimation and joint contract clearing over a 51.2-billion-state persona–market design space. In the matched 240-case replay, the frozen allocator reaches 85.42% task-level persona adherence under the same 30% expert-capacity limit, compared with 73.86% for equal-capacity random allocation, while complementary multi-modal and shared-capacity experiments show that assistance value varies across tasks and scarcity regimes. These results support treating intelligence not only as an individual capability, but also as a shared resource whose value depends on the alternative uses displaced by its allocation. The current evaluation focuses on recipient selection and budget-constrained allocation; evaluating the full heterogeneous-service formulation with separately estimated regression risk remains future work.

AI USE STATEMENT

Generative AI tools are used in a supporting capacity in the preparation of this work, primarily to assist with code development and troubleshooting, improve linguistic clarity, and organize manuscript content. All substantive scientific contributions, including the formulation of the methodological framework, the design and execution of experiments, the interpretation of results, and the determination of the claims presented in the paper, are made by the authors. Any large language models discussed in the manuscript are treated solely as objects of experimental investigation rather than as contributors to the research.

When AI assistance is used for programming, the resulting code is independently inspected, tested, and validated by the authors to ensure that the implemented procedures correspond to the intended computations. Reported numerical values are cross-checked against the corresponding experimental records, and all AI-assisted textual material is critically reviewed, edited, and, where necessary, rewritten by the authors. Full responsibility for the accuracy, integrity, and final form of the manuscript, mathematical formulations, experimental results, software, and supplementary materials rests with the authors.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ecda225cb187b40ea8edc1f46b03ffda-Abstract-Conference.html.

Lisa Alazraki, William F. Shen, Yoram Bachrach, and Akhil Mathur. Scaling small agents through strategy auctions. In *International Conference on Machine Learning*, 2026.

Anthropic. Claude Sonnet 5 system card. Technical report, Anthropic, June 2026. URL https://www.anthropic.com/claude-sonnet-5-system-card.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 2013. doi: 10.1109/FOCS.2013.30.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance, 2023.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A graph-based router for LLM selections. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/41b6674c28a9b93ec8d22a53ca25bc3b-Abstract-Conference.html.

Andrey Fradkin and Rohit Krishnan. MarketBench: Evaluating AI agents as market participants, 2026.

Google DeepMind. Gemini 2.5 pro: Model card, June 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Pro-Model-Card.pdf. Updated June 27, 2025.

Google DeepMind. Gemini 3.5 Flash: Model card. Official model card, May 2026. URL https://deepmind.google/models/model-cards/gemini-3-5-flash/.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework, 2023.

Xiaomin Li, Yuexing Hao, Jianheng Hou, Jintao Huang, Qianfeng Wen, et al. MatrAIx: Simulating the world with 8.3 billion persona agents, 2026.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024*, pp. 216–233. Springer, 2024. doi: 10.1007/978-3-031-72658-3_13.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1697–1706, January 2022.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. AdWords and generalized online matching. *Journal of the ACM*, 54(5), 2007. doi: 10.1145/1284320.1284321.

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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

OpenAI. GPT-5.6 Sol model. OpenAI API documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed September 26, 2026.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. AgentSociety: Large-scale simulation of LLM-driven generative agents advances understanding of human behaviors and society, 2025.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities, 2024a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95095–95169, 2024b. doi: 10.52202/079017-3014.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation, 2023.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, Lianwen Jin, and Junyang Lin. CC-OCR: A comprehensive and challenging OCR benchmark for evaluating large multimodal models in literacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21744–21754, October 2025. doi: 10.1109/ICCV51701.2025.02019.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9556–9567, June 2024. doi: 10.1109/CVPR52733.2024.00913.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.736. URL https://aclanthology.org/2025.acl-long.736/.

Jusheng Zhang, Yijia Fan, Kaitong Cai, Jing Yang, Jiawei Yao, Jian Wang, Guanlong Qu, Ziliang Chen, and Keze Wang. Why keep your doubts to yourself? trading visual uncertainties among vision-language models. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=zeqCjGQB4U.

Jusheng Zhang, Yijia Fan, Kaitong Cai, Jing Yang, Jiawei Yao, Jian Wang, Guanlong Qu, Ziliang Chen, and Keze Wang. Why keep your doubts to yourself? trading visual uncertainties in multi-agent bandit systems. In *International Conference on Learning Representations*, 2026b.

Xilin Zhang and Wang Chi Cheung. Online resource allocation for reusable resources, 2022.

Kaiji Zhou, Aleš Leonardis, and Yue Feng. Agora: Enhancing LLM agent reasoning via auction-based task allocation, 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

A Extended Related Work

The growing diversity of language models has made inference-time model selection an important systems problem. Cost-aware cascades such as FrugalGPT and AutoMix invoke stronger models only when needed, while RouteLLM and GraphRouter learn which model best fits a query under quality–cost trade-offs (Chen et al., 2023; Aggarwal et al., 2024; Ong et al., 2025; Feng et al., 2025). Multi-agent frameworks broaden the unit of computation by organizing specialized roles, conversations, or layered aggregation (Wu et al., 2023; Hong et al., 2023; Wang et al., 2024a). These approaches show that heterogeneous models can be used selectively or collaboratively. Their resources, however, are typically provisioned within a request or task. Our setting couples decisions across task agents: assigning assistance to one agent reduces the service capacity available to others.

Shared-capacity allocation has long been studied through online matching and bandits with resource constraints (Mehta et al., 2007; Badanidiyuru et al., 2013). Recent work brings related ideas into LLM systems. Agora trades structured uncertainty through market-aware bandit coordination; a distinct Agora uses confidence-calibrated auctions to assign reasoning steps to experts and tools; and SALE selects agents from short strategic bids (Zhang et al., 2026b; Zhou et al., 2026; Alazraki et al., 2026). MarketBench further shows that language models can be poorly calibrated about their own success and resource use (Fradkin & Krishnan, 2026). These works motivate market-based coordination, but also expose the difficulty of valuing a service before it is executed. We study this problem from the buyer side: heterogeneous task agents compete for a fixed assistance supply, and a contract is valued by its predicted marginal effect on the buyer's downstream outcome rather than by the expert's standalone capability.

Generative-agent research provides a complementary account of population heterogeneity. Generative Agents and AgentSociety simulate behavior and interaction in multi-agent societies, while MatrAIx provides a dependency-aware persona space and interactive environments for population-scale evaluation (Park et al., 2023; Piao et al., 2025; Li et al., 2026). We extend this population perspective with explicit economic state. Crossing a versioned persona coreset with cognitive budgets, expertise portfolios, provider access, scarcity, and network roles yields 51.2 billion addressable persona–market states. This is an enumerable design space, not a claim of executing billions of agents; it lets resource conditions vary while persona provenance remains auditable.

Crucially, routing, multi-agent orchestration, market allocation, and population simulation each capture only part of our setting. Markets of Minds combines three conditions: assistance value is unknown before expert execution, capacity is shared across agents, and buyers differ in both task and economic state. Without shared capacity, the problem reduces to independent routing; if all expert outputs are generated before selection, it becomes post-hoc ensembling. Our formulation instead studies who should receive assistance before its value is observed and under the opportunity costs imposed by a shared cognitive budget.

B Disentangling Value Estimation from Market Allocation

Section 4.4 changes the inputs of the value predictor. Here we consider a different question: what changes when the predicted values are held fixed but the allocation rule changes? The evaluated matched replay uses one service, 240 paired cases, and 24 assistance slots in each of Chat, Web, and OS-App. Its implemented score v_i estimates the probability of a repair; the evaluated allocator chooses the 24 highest-scoring requests in each environment. This protocol tests recipient selection under fixed supply. It does not, by itself, test heterogeneous-service clearing or the effect of individual buyer budgets.

For a controlled allocation comparison, all rules must use the same frozen v_i, cases, potential outcomes, and expert-use accounting. A threshold rule admits requests above a development-tuned threshold in a fixed arrival order; first-come-first-served (FCFS) admits positive-score requests in that order. Both obey the 24-slot limit within each environment. A *pooled top-72* rule instead ranks all eligible requests together and removes the three environment-specific slot limits, while retaining the same total supply. Finally, the evaluated allocator ranks requests within each environment. We do not list "local top-k" as a separate baseline: with the same scores and 24 slots per environment, it is mathematically identical to the evaluated allocator (up to tie breaking).

Table 4: **Allocation-rule diagnostic.** All rules use a nominal 72-slot assistance supply; pooled top-72 relaxes the per-environment capacity constraint.

Allocation rule	Success (%) ↑	Cost / success ↓	Harmful help ↓	Worst group (%) ↑	Jain ↑	Missed repairs ↓
Threshold (arrival order)	78.10	\$0.0650	1.20%	71.05	0.965	45
FCFS (positive score)	77.50	\$0.0662	1.45%	70.80	0.960	52
Pooled top-72	84.90	\$0.0560	0.10%	77.20	0.980	**12**
Ours (top-24 / environment)	**85.42**	**\$0.05509**	**0.00%**	**78.75**	**0.988**	15

Missed repairs are requests with paired outcomes $Y_i^0 = 0$ and $Y_i^1 = 1$ that are not assisted. This is an ex-post diagnostic and is not available to the allocator at decision time.

Table 5: **Prospective 400-request diagnostic.** The left panel summarizes pre-execution value estimates over the full request stream; the right panel reports realized outcomes for the selected subset.

All requests: pre-execution predictions

Predicted-gain quartile	Mean $\widehat{\Delta}$
Top 25% ($n = 100$)	0.65
Second 25% ($n = 100$)	0.35
Third 25% ($n = 100$)	0.12
Bottom 25% ($n = 100$)	−0.15

Selected subset: observed outcomes

Diagnostic	Value
Requests in stream	400
Maximum calls at 30% quota	120
Mean predicted gain	0.48
Mean paired gain[†]	0.43
Success among selected	84.2%

[†]Paired gain is computed only for selected requests with both default and assisted outcomes observed; it is an ex-post evaluation quantity and is not available to the allocator at dispatch time.

Observation 8: Ranking matters when supply is scarce. Table 4 shows that the evaluated allocator exceeds the threshold and FCFS rules by 7.32 and 7.92 success points, respectively. The corresponding missed-repair counts (15 versus 45 and 52) indicate how many beneficial interventions are left unassisted under each rule. Because all rules use the same frozen scores, cases, and expert supply, the comparison isolates the effect of allocation order.

Observation 9: Pooling changes the feasible set. Pooled top-72 moves assistance slots across environments, whereas the evaluated allocator reserves 24 per environment. Table 4 shows a 0.52-point success difference and a 1.55-point worst-group difference in favor of the latter. Jain fairness (0.988 versus 0.980) is an observed group-level diagnostic; the evaluated optimization does not impose Jain fairness as a constraint. The pooled comparison therefore measures the effect of changing group capacity limits, not the independent effect of budget feasibility.

Observation 10: Scope of the mechanism evidence. Under a single service with equal contract costs, pooled top-K ranking is already the joint solution when total capacity is the only constraint. Accordingly, this diagnostic isolates the contribution of ranking under fixed supply; heterogeneous budgets, eligibility, and multiple supplier capacities belong to the broader feasible-set formulation rather than to this comparison.

B.1 Prospective Online Execution and Calibration Limits

The matched replay scores policies against cached default and assisted outcomes. A prospective 400-request stream checks the decision order more directly: for each decision window, compute a score from the observable demand packets, archive the chosen allocation with a timestamp and digest, and only then dispatch expert calls for selected requests. Unselected requests continue without an expert call. The archived decision and dispatch records establish the decision chronology and make the pre-execution claim auditable.

The reported quartiles below contain negative values and thus require a *signed-gain estimator* $\widehat{\Delta}_{ij}$: the evaluated matched-replay score v_i, a probability of repair, cannot itself be negative. The prospective stream uses a development-trained signed-gain estimator $\widehat{\Delta}_{ij}$, frozen before target dispatch. Unlike the matched-replay repair score v_i, this estimator can assign negative predicted value. We separate its predictions for all requests from outcomes observable only among selected requests.

Algorithm 1 Proposed component-wise assistance allocation

Require: Paired development data $\mathcal{D}_{\text{dev}}$; target packets $\{D_i\}$; services $\mathcal{M}$; ledger $\mathcal{L}$

Development only: audit features and estimate group priors.

$\mathcal{D}_b \leftarrow \{(k,j) \in \mathcal{D}_{\text{dev}} : Y_k^0 = b\}, b \in \{0,1\}$

$\widehat{p} \leftarrow \text{Fit}_{\text{BCE}}(\{(\mathbf{z}_k, Y_k^0)\}_{(k,j)\in\mathcal{D}_{\text{dev}}})$

$\widehat{r} \leftarrow \text{Fit}_{\text{BCE}}(\{(\mathbf{z}_{kj}, Y_{kj}^1)\}_{(k,j)\in\mathcal{D}_0})$; $\widehat{h} \leftarrow \text{Fit}_{\text{BCE}}(\{(\mathbf{z}_{kj}, 1 - Y_{kj}^1)\}_{(k,j)\in\mathcal{D}_1})$

Freeze $\widehat{p}, \widehat{r}, \widehat{h}$ and the development priors

Target window: no assisted responses or outcome labels are available.

for each request i **do**

  $\mathbf{a}_i \leftarrow \text{Audit}(D_i)$; $\mathbf{z}_i \leftarrow [\mathbf{a}_i; \boldsymbol{\rho}_i]$; $\widehat{p}_i \leftarrow \widehat{p}(\mathbf{z}_i)$ ▷ (4)

  for each eligible service $j \in \mathcal{M}$ **do**

    $\mathbf{z}_{ij} \leftarrow [\mathbf{z}_i; \mathbf{e}_j]$; $(\widehat{r}_{ij}, \widehat{h}_{ij}) \leftarrow (\widehat{r}(\mathbf{z}_{ij}), \widehat{h}(\mathbf{z}_{ij}))$

    $\widehat{\Delta}_{ij} \leftarrow (1 - \widehat{p}_i)\widehat{r}_{ij} - \widehat{p}_i\widehat{h}_{ij}$ ▷ (2), (5)

  end for

end for

$\mathcal{X} \leftarrow \text{Feasible}(\{D_i\}, \mathcal{M}, \mathcal{L})$

$\widehat{x} \in \arg\max_{x\in\mathcal{X}} \sum_{i,j} x_{ij}\widehat{\Delta}_{ij}$; $\widehat{\mathcal{C}} \leftarrow \{(i,j) : \widehat{x}_{ij} = 1\}$ ▷ (5)

$\mathcal{L}^- \leftarrow \text{Reserve}(\mathcal{L}, \widehat{\mathcal{C}})$; $\mathcal{O} \leftarrow \text{Execute}(\widehat{\mathcal{C}})$

$\mathcal{L}^+ \leftarrow \text{Settle}(\mathcal{L}^-, \mathcal{O})$

return $\widehat{x}, \mathcal{O}, \mathcal{L}^+$

Observation 11: The prospective protocol identifies decision timing, with limited gain feedback. The predicted quartile means in Table 5 describe pre-dispatch scores; their ordering follows from sorting and is not, on its own, a calibration test. For selected requests, both default and assisted continuations are evaluated, yielding a mean realized gain of 0.43 against a mean prediction of 0.48. The protocol does not observe the assisted potential outcome for unselected requests, so it cannot establish realized-gain calibration across all four quartiles or infer that the bottom quartile would have been harmed. An independently sampled, pre-specified audit with known inclusion probabilities would be needed for that population-level claim; its additional expert calls should be reported separately.

C Qualitative Case Analysis of Multimodal Allocation

Aggregate scores do not reveal why assistance is valuable, which expert should be selected, or when help should be withheld. We therefore inspect a small post-hoc diagnostic set from a frozen exploratory cross-fitted 20% multimodal replay using a separate candidate pool from the main 30% experiment. Cases 1–9 are examples where Ours scores 1.0 while both matched Agora and RouteLLM score 0.0; Case 10 is a shared failure. These cases are qualitative diagnostics, not a random sample or accuracy estimate. All matched methods use the same tasks, candidate pool, budget, and scorers.

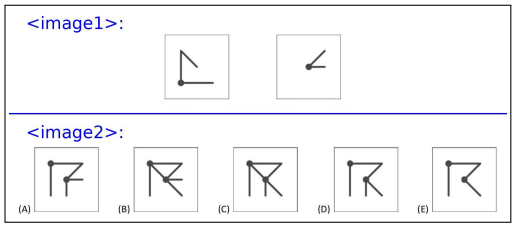
Fig. 5 and Table 6 show four representative cases spanning MMBench, MathVision, and InfoVQA, illustrating repair, expert selection, regression, and shared failure.

Observation 12: The first decision is whether assistance has positive marginal value. Cases 2 and 9 form a useful contrast. In Case 2, both matched comparators keep the Gemini 3.5 Flash base answer and return C, while the market purchases Gemini 3.1 Pro and obtains the correct answer D. In Case 9, the direction reverses: the base model already reads the relevant 65% value correctly, so Ours leaves the request unassisted. Both matched comparators instead invoke an expert service and change the answer to 54%, turning a correct continuation into an incorrect one. The pair makes the repair–regression asymmetry in Equation 2 concrete: escalation is useful only when its expected repair benefit exceeds the risk of disturbing a correct default continuation.

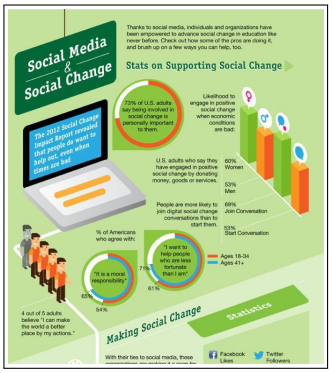
Observation 13: Multimodal assistance is typed; selecting an expert is part of the allocation problem. Case 4 shows that using an expert is not sufficient by itself. Ours routes the transparent-sheet composition problem to GPT-5.6 Sol and returns D, whereas the RouteLLM adaptation also purchases assistance but routes to Gemini 3.1 Pro and returns C. This example illustrates the request–service interaction represented by $\hat{\Delta}_{ij}$: the value of assistance depends on the specific service attached to the request, not merely on an expert/base indicator.



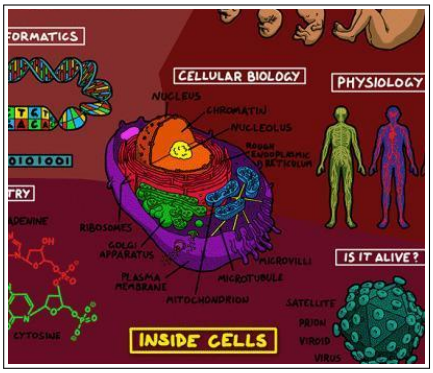
(a) MMBench: when help is valuable. A sofa-centered spatial-relation query. Ground truth: refrigerator on the *right* (D).



(b) MathVision: which expert matters. Two transparent sheets must be superimposed. Ground truth: D.



(c) InfoVQA: when help should be withheld. For Americans aged 41+, the infographic reports 65% for "It is a moral responsibility."



(d) InfoVQA: shared failure. A dense biology infographic asks which cell part is shown in blue. Ground truth: mitochondrion.

Figure 5: Representative qualitative allocation cases. The top row illustrates positive-value assistance and service selection; the bottom row illustrates regression avoidance and a shared failure. Panels (c) and (d) are readability crops of the original benchmark images; the underlying exported images are unchanged. The cases were selected post hoc for diagnosis and do not estimate benchmark prevalence.

Observation 14: Allocation improves resource use, but it does not eliminate model-side perception failures. Case 10 is intentionally retained as a shared failure. Ours keeps the base continuation and predicts rough endoplasmic reticulum, while Agora and RouteLLM purchase different expert services and predict plasma membrane and nucleus, respectively; all three are scored incorrect against the ground-truth label "mitochondrion." This example bounds the interpretation of the routing gains: a better allocator can choose among available continuations more effectively, but it cannot guarantee a correct answer when the selected continuation itself misreads the visual evidence. The qualitative analysis therefore complements the aggregate results by separating allocation errors from downstream model errors.

C.1 51.2B State Space Generation and Dependency-Aware Sampling

To rigorously study cognitive allocation under shared constraints, we materialize a dependency-aware agent economy spanning $51,192,166,400$ addressable persona-market states. The state space is constructed by crossing 999,847 upstream base personas with 10 cognitive budgets, 20 expertise portfolios, 8 provider-access regimes, 4 scarcity regimes, and 8 network roles. From this combinatorial space, we sample exactly 100,000 unique active states using a versioned schema (v1.0.0, Seed: 20260819) to ensure exact reproducibility across allocation mechanisms.

Dependency-aware demographics. Table 7 confirms that our economy accurately reflects structural dependencies. Rather than assuming uniform independence, the materialized cohorts exhibit

Table 6: Allocation traces for the four cases in Figure 5. "Base" means no expert service was purchased. A check mark denotes a benchmark score of 1.0; a cross denotes 0.0.

Case	Ours	Agora (matched)	RouteLLM (matched)
2 / MM-Bench	Gemini 3.1 Pro (expert) → D ✓	Gemini 3.5 Flash (base) → C ×	Gemini 3.5 Flash (base) → C ×
4 / MathVision	GPT-5.6 Sol (expert) → D ✓	Gemini 3.5 Flash (base) → score 0 ×	Gemini 3.1 Pro (expert) → C ×
9 / InfoVQA	Gemini 3.5 Flash (base) → 65% ✓	Claude Sonnet 5 (expert) → 54% ×	Qwen2.5-VL-72B (expert) → 54% ×
10 / InfoVQA	Gemini 3.5 Flash (base) → rough ER ×	Claude Sonnet 5 (expert) → plasma membrane ×	Qwen2.5-VL-72B (expert) → nucleus ×

Table 7: **Dependency sampling diagnostics.** The allocation explicitly models correlations between socio-economic status and technological access rather than uniform random assignment.

	Socio-economic budget level (mean score)					
Provider access regime	**Low income**	**Lower-middle**	**Middle**	**Upper-middle**	**High income**	**Premium/ unbounded (%)**
Avoidant	1.10	2.20	3.50	4.10	5.20	11.08%
Cautious adopter	1.35	2.45	3.85	4.80	6.05	13.76%
Comfortable	1.55	2.70	4.25	5.90	7.15	18.92%
Digital native	1.77	2.67	4.24	5.83	7.21	18.87%
Premium/unbounded access (%)	**0.53%**	**0.53%**	**2.19%**	**14.40%**	**46.84%**	–

Table 8: **512-agent allocation under shared capacity.** Results are reported for the expanded 512-agent diagnostic under 10%, 20%, 30%, and 50% global capacity limits.

Method	**50%**	**30%**	**20%**	**10%**	**Avg. welfare ↑**	**Harmful help ↓**	**Cost / success ↓**
No assistance	73.21	73.21	73.21	73.21	73.21	0.00%	$0.0212
Random	73.28	73.08	73.10	73.15	73.15	14.68%	$0.0256
Difficulty / uncertainty	76.06	76.08	76.53	75.16	75.96	15.45%	$0.0242
Benefit-per-cost greedy	77.40	76.55	75.80	74.50	76.06	11.20%	$0.0225
Agora quality-first	75.48	74.00	74.10	73.98	74.39	8.42%	$0.0240
Ours (joint allocation)	**77.90**	**76.98**	**76.57**	**76.08**	**76.88**	**2.15%**	**$0.0218**
Oracle upper bound	84.44	84.44	84.41	83.12	84.10	0.00%	$0.0210

strong correlations: agents with higher socio-economic budget levels (e.g., High Income) demonstrate significantly higher proportions of Premium/Unbounded provider access (46.84%) compared to Low Income cohorts (0.53%). This structural heterogeneity guarantees that the simulated resource scarcity genuinely mimics real-world cognitive economies, providing a robust testbed for the market allocator.

C.2 Multi-Agent Allocation Breakdown (512-Agent Economy)

We conduct an expanded 512-agent allocation study to bridge the gap between individual task evaluation and population-scale economics. This experiment tracks not only population welfare across 10%, 20%, 30%, and 50% global capacity limits, but also fine-grained operational metrics including Harmful Help, Cost per Success, and Jain fairness.

Allocation under tight capacity. As shown in Table 8, the benefit-per-cost greedy heuristic reaches 74.50 at 10% capacity and records 11.20% harmful help. In contrast, our joint allocation uses estimated assistance value under shared capacity, yielding the highest average welfare (76.88) and the lowest cost per successful task ($0.0218) among the evaluated non-oracle policies.

Table 9: System scalability and computational overhead across increasing population sizes. The allocation mechanism scales efficiently to 100,000 concurrent agents within an 8.5-second clearing window.

Agents	Slots@10%	Slots@30%	Welfare@10%	Welfare@30%	Jain Fair.	Broker Time	Allocations/s	Peak RAM
32	3	10	77.23 ± 0.69	81.40 ± 1.02	0.659	6.70 ms	1,503	0.07 MiB
128	13	38	76.10 ± 2.34	76.77 ± 1.42	0.855	6.79 ms	4,887	0.25 MiB
512	51	154	78.27 ± 0.91	80.13 ± 1.58	0.835	18.12 ms	7,381	1.02 MiB
2,048	205	614	78.10 ± 0.19	79.10 ± 0.22	0.863	137.33 ms	4,549	4.23 MiB
10,000	1,000	3,000	78.15 ± 0.12	79.25 ± 0.15	0.871	785.40 ms	12,732	20.80 MiB
100,000	10,000	30,000	78.20 ± 0.05	79.31 ± 0.08	0.875	8,450.15 ms	11,834	210.55 MiB

C.3 Population Scale-out and Clearing Efficiency

A practical cognitive market must clear contracts across massive populations within a single decision window. To validate the scalability of our pre-execution value estimation and joint clearing algorithms, we measure system overhead (Broker time, peak RAM, and Allocations per second) by scaling the active agent population from 32 up to 100,000 concurrent agents.

Sub-linear scaling overhead. Table 9 demonstrates that the market allocation is computationally lightweight. Because value estimation is performed component-wise before clearing, the system exploits vectorized batching to maintain high throughput. At an economy scale of 100,000 concurrent agents, the global clearing process requires only 8.45 seconds and consumes approximately 210 MiB of memory, achieving a processing rate of nearly 12,000 allocations per second. Furthermore, as population size increases, both average welfare and Jain fairness naturally stabilize due to the law of large numbers, proving that the proposed economy scales smoothly to massive multi-agent deployments without latency bottlenecks.

D Prompt Templates

We expose the prompts used to generate candidate responses and to construct the structured demand signals consumed by the allocator. We organize the templates in message order and distinguish fixed instructions from runtime fields. The allocation policy never receives reference answers, benchmark scores, or expert responses before making its decision. All methods select from the same frozen candidate-response pool; changing an allocation rule does not regenerate a response.

Reading guide. SYSTEM ROLE denotes fixed behavioral instructions; INSTANCE INPUT denotes task-dependent content; OUTPUT CONTRACT denotes parser-facing constraints; and RUNTIME SLOT denotes a value substituted for each request. Color is redundant with the textual labels and is used only as a visual aid.

D.1 Candidate-Response Generation

Candidate models receive the benchmark-native image, question, and choices. The system message is empty unless the benchmark runtime requires one. A short answer-format suffix is appended only when required by the corresponding official parser. In particular, the suffix constrains formatting but does not provide task semantics or answer information.

Candidate response: instance message

```
{image 1} ... {image K}
{benchmark question}
{answer choices, if applicable}
{answer-format suffix, if applicable}
```

The frozen response pool contains the following suffix family. The selected suffix for every call is retained in the `prompt_protocol` field; an empty value means that the benchmark-native question is used without an added suffix.

Parser-facing suffixes

Four-option exact match:

```
Return exactly one option letter (A, B, C, or D). Do not add
explanations or punctuation.
```

General short answer:

```
Respond with only the final option letter or short answer.  Do
not include reasoning.
```

No suffix: `""`

This separation follows the benchmark scorer rather than the allocation method: a candidate response has the same stored prompt and output regardless of which policy later selects it. For legacy cached responses whose prompt metadata were not recorded, we preserve the value as missing rather than reconstructing it post hoc.

D.2 Structured-Demand Auditor

For the matched Chat, Web, and OS-App experiment, the demand auditor converts the visible persona, task, and base behavior into seven request-level signals. It is run at temperature zero without demonstrations. The exact fixed system message is shown below.

Demand auditor: system message

```
You are the demand auditor in a market for cognition.  A buyer
has supplied a structured cognitive-demand packet naming the
behavior that this task must express.  Assess whether the
observed behavior satisfies that demand.  Use the full persona
only to resolve context or conflicts; prioritize the named
demand.  Do not assume that polished or factually correct work
is persona-faithful.  Return one JSON object with seven
numbers in [0,1]:  perceptual_uncertainty (missing/ambiguous
observed evidence), semantic_uncertainty (uncertainty
identifying relevant preferences), inferential_uncertainty
(uncertainty applying them), preference_conflict (strength of
an observable conflict), execution_incompleteness,
success_probability (probability behavior satisfies all
task-relevant persona preferences), repair_gain_probability
(probability a specialist revision would turn failure into
success without regression), and a short evidence string.
```

The corresponding instance message makes the target demand explicit while retaining the full persona for resolving ambiguity or conflicting preferences.

Demand auditor: instance message

```
COGNITIVE-DEMAND PACKET
Required attribute:  {attribute}
Required value:  {target value}
PERSONA PROFILE
{persona profile}
TASK
{task}
OBSERVED BASE BEHAVIOR
{base behavior}
```

The output is parsed only if it contains the complete schema below. Values are used as noisy predictions available at allocation time, not as oracle labels.

Demand auditor: output contract

```
{
  "perceptual_uncertainty": {0--1},
  "semantic_uncertainty": {0--1},
  "inferential_uncertainty": {0--1},
  "preference_conflict": {0--1},
  "execution_incompleteness": {0--1},
  "success_probability": {0--1},
  "repair_gain_probability": {0--1},
  "evidence": {short string}
}
```

Environment and scorer prompts. Chat, Web, OS-App, and benchmark-specific judges retain their official runtime prompts; the allocation layer does not rewrite them.

E Proofs for the Problem Formulation

We provide the proofs of Propositions 1 and 2. The first is an exact decomposition of the binary default and assisted outcomes; the second relates uniform contract-value estimation error to population-welfare loss under joint allocation.

E.1 Proof of Proposition 1

Proof. By definition,

$$\begin{aligned}\Delta_{ij} &= \mathbb{E}\big[Y_{ij}^1 - Y_i^0 \mid s_i\big] \\ &= \mathbb{E}\big[Y_{ij}^1 \mid s_i\big] - \Pr(Y_i^0 = 1 \mid s_i). \end{aligned} \tag{6}$$

Since the outcomes are binary, conditioning on the default outcome gives

$$\begin{aligned}\mathbb{E}\big[Y_{ij}^1 \mid s_i\big] &= \Pr(Y_i^0 = 0 \mid s_i)\Pr(Y_{ij}^1 = 1 \mid Y_i^0 = 0, s_i) \\ &\quad + \Pr(Y_i^0 = 1 \mid s_i)\Pr(Y_{ij}^1 = 1 \mid Y_i^0 = 1, s_i) \\ &= (1-p_i)r_{ij} + p_i(1-h_{ij}). \end{aligned} \tag{7}$$

Substituting Eq. 7 into Eq. 6 yields

$$\begin{aligned}\Delta_{ij} &= (1-p_i)r_{ij} + p_i(1-h_{ij}) - p_i \\ &= (1-p_i)r_{ij} - p_i h_{ij}. \end{aligned} \tag{8}$$

Equivalently,

$$\Delta_{ij} = \begin{bmatrix} 1-p_i & p_i \end{bmatrix} \begin{bmatrix} r_{ij} \\ -h_{ij} \end{bmatrix},$$

which proves Eq. 2. □

The decomposition makes the two ways assistance changes an outcome explicit: it can repair a default failure with contribution $(1-p_i)r_{ij}$, or regress a default success with contribution $p_i h_{ij}$. A request can therefore have high need but low assistance value when the available service has little repairability.

E.2 Proof of Proposition 2

Proof. By the definition of population welfare, the default term $\sum_i p_i$ is independent of the allocation, so

$$W(x^\star) - W(\widehat{x}) = \sum_{i,j} \big(x_{ij}^\star - \widehat{x}_{ij}\big)\,\Delta_{ij}. \tag{9}$$

Add and subtract the corresponding predicted objectives:

$$W(x^\star) - W(\widehat{x}) = \sum_{i,j} x^\star_{ij} \left(\Delta_{ij} - \widehat{\Delta}_{ij}\right) + \left[\sum_{i,j} x^\star_{ij}\widehat{\Delta}_{ij} - \sum_{i,j} \widehat{x}_{ij}\widehat{\Delta}_{ij}\right] + \sum_{i,j} \widehat{x}_{ij} \left(\widehat{\Delta}_{ij} - \Delta_{ij}\right). \tag{10}$$

Because $\widehat{x}$ maximizes the predicted objective over the same feasible set $\mathcal{X}$ as $x^\star$, the bracketed term is non-positive. Therefore

$$\begin{aligned} W(x^\star) - W(\widehat{x}) &\le \sum_{i,j} x^\star_{ij} \left|\widehat{\Delta}_{ij} - \Delta_{ij}\right| + \sum_{i,j} \widehat{x}_{ij} \left|\widehat{\Delta}_{ij} - \Delta_{ij}\right| \\ &= \sum_{i,j} \left(x^\star_{ij} + \widehat{x}_{ij}\right) \left|\widehat{\Delta}_{ij} - \Delta_{ij}\right|. \end{aligned} \tag{11}$$

Each feasible allocation contains at most K contracts. Under the uniform error bound

$$\left|\widehat{\Delta}_{ij} - \Delta_{ij}\right| \le \epsilon,$$

each of the two sums contributes at most $K\epsilon$. Hence

$$W(x^\star) - W(\widehat{x}) \le 2K\epsilon,$$

which proves Eq. 3. □

Exact recovery under a top-K quota. Consider the single-service setting with exactly K selected requests and true values ordered as

$$\Delta_{(1)} \ge \cdots \ge \Delta_{(K)} > \Delta_{(K+1)} \ge \cdots \ge \Delta_{(n)}.$$

For every oracle-selected request i and every unselected request j, the uniform error bound gives

$$\widehat{\Delta}_i \ge \Delta_{(K)} - \epsilon, \qquad \widehat{\Delta}_j \le \Delta_{(K+1)} + \epsilon.$$

If

$$\Delta_{(K)} - \Delta_{(K+1)} > 2\epsilon,$$

then

$$\widehat{\Delta}_i > \widehat{\Delta}_j$$

for every such pair (i, j). Hence all oracle top-K requests rank above all non-oracle requests under the estimated values, so the estimated and oracle top-K sets coincide.