MEDI-SIM: HEALTHCARE MECHANISM DESIGN UNDER STRATEGIC PROVIDER RESPONSE

Anonymous authors
Paper under double-blind review

ABSTRACT

How can AI help design effective hospital incentive and resource-allocation policies when providers respond strategically? Yet conventional research that evaluates these policies against fixed provider programs or a single performance measure can miss distortions that shift across behaviors or emerge through further adaptation. We introduce MEDI-SIM, a closed-loop multi-agent simulator that integrates hospital operations with five dimensions of strategic provider response: billing classification (coding), patient acceptance, treatment deferral, treatment effort, and bed requests. Administrator and provider policies are typed, inspectable programs, which an LLM edits using rollout feedback. We first map behavior under fixed response rules, then test whether gains from joint program design survive unilateral adaptation and responses to retained policy histories. Stronger audits suppress coding inflation while worsening access for complex patients; contract changes can restore effort while coding distortion persists; and further provider search can erode recovered health. We characterize finite-policy convergence under exact response oracles and separate search and payoff-estimation errors in an approximate stopping bound. These results identify *pressure migration* across response channels and adaptation stages, suggesting that AI-designed mechanisms should be evaluated through the behaviors they induce and the further adaptations they invite.

1 INTRODUCTION

Consider a hospital administrator trying to improve care with a limited budget. Paying more for complex cases may encourage their treatment, but it also makes reporting the complex diagnosis more profitable. Rewarding shorter waits may improve patient flow, but it can also change which patients are admitted or deferred. The administrator chooses the rule; providers choose how to respond. Consequently, the quality of a hospital mechanism depends on the behavior it induces as well as the objective it rewards. How, then, can we design and evaluate a rule when the agents governed by that rule can change their response?

Healthcare provides concrete evidence of this difficulty. A Medicare payment reform produced a substantial re-coding response without corresponding increases in treatment intensity (Dafny, 2005); waiting-time targets in the English National Health Service alter both reported waits and the organization of care (Bevan & Hood, 2006; Propper et al., 2010). These findings show why an improvement in a measured outcome need not imply an improvement in the underlying objective, a risk captured by *Goodhart-style drift* (Manheim & Garrabrant, 2018). They also point to a broader learning problem: administrative rules change the process that generates the observations used to evaluate them. This connects hospital mechanism design to strategic classification and performative prediction, where agents or data distributions respond to a deployed rule (Hardt et al., 2016; Perdomo et al., 2020).

A natural approach is to optimize hospital rules in simulation and select the highest-return candidate. For this evaluation to be informative, however, it must capture two coupled effects. First, a provider may discover a more profitable response after the rule changes, eroding gains obtained against the original behavior. Second, a response can improve the targeted metric while shifting costs to another part of care delivery. An audit may reduce inflated billing while leaving the incentive to avoid expensive patients intact. We use *pressure migration* to describe changes in where distortion appears: across response channels after an intervention, or across stages as providers adapt. Evaluating only a

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.

fixed provider or a single aggregate score can miss precisely the behavior that makes a mechanism vulnerable.

The difficulty grows when several rules operate together. A policy team can reason about a payment change or an audit in isolation, but their joint effects depend on patient mix, congestion, treatment effort, and the measurements that shape the next decision. A small set of hand-designed scenarios can leave profitable responses unexplored. Automated mechanism design offers a way to search this space systematically (Sandholm, 2003; Zheng et al., 2022), and recent LLM-based program search makes it possible to propose changes to explicit decision rules using execution feedback (Romera-Paredes et al., 2024; Lehman et al., 2024; Novikov et al., 2025). This creates an opportunity for AI-assisted policy design: the same search capability can help discover an administrative improvement and a provider response that undermines it.

Existing work supplies important components. Patient-level healthcare AI studies treatment within specified clinical environments, typically taking administrative rules as given (Komorowski et al., 2018; Yu et al., 2023; Gottesman et al., 2019). Hospital simulation models stochastic arrivals and constrained patient flow (Brailsford et al., 2009; Green, 2006). Automated mechanism design studies strategic responses in taxation, auctions, and allocation, but the cited settings do not jointly model hospital reimbursement, audits, queues, and measured quality. Moreover, the AI Economist's neural planner and RegretNet's allocation and payment networks leave individual decision rules difficult to inspect directly (Zheng et al., 2022; Dütting et al., 2024). We need both a coupled evaluation environment and a representation that reveals how a rule earns its return.

We introduce **MEDI-SIM**, which follows an Identify–Produce–Settle (IPS) loop: patients are classified for billing, providers deliver care under capacity constraints, and payments and performance rewards shape subsequent decisions. Five response channels track what is reported, who receives care, when care occurs, how much effort is supplied, and which resources are requested. Modeling them together lets us follow an intervention from its intended effect to consequences for access, health, and finances. Both roles use *policy-as-code*: typed, executable expressions that an LLM edits between episodes using rollout feedback. This turns mechanism design and strategic response into program-synthesis problems within a shared evaluation environment. The programs support inspection, version comparison, and replay (Rudin, 2019); the LLM acts as an outer-loop editor while executable policies govern each episode.

We use this environment to ask three questions, each allowing more adaptation than the last. L1 holds provider rules fixed and sweeps financial and quality incentives, mapping where coding, selection, delay, and effort distortions appear. L2 intervenes on that landscape one lever at a time, asking whether suppressing one distortion moves pressure into another channel. L3 lets the rules themselves change. Joint search first edits both programs under an administrative objective; each role then searches for its own utility, testing whether the initial improvement survives adaptation. Because a response to the latest opponent may reopen an earlier vulnerability, we also retain historical programs using population-based response-oracle methods (Lanctot et al., 2017; Marris et al., 2021; Hennes et al., 2026). We characterize termination under exact oracles over a finite policy universe and derive an approximate stopping bound; the experiments examine what finite LLM search budgets reveal within that framework.

Our contributions are a coupled environment for strategic healthcare policy evaluation, an inspectable program-search framework connecting joint design to current and historical responses, and evidence that initial reward gains can conceal strategic failure. With response rules fixed (L1–L2), distortions occupy different regions of the incentive landscape, and an audit that suppresses up-coding worsens access for complex patients. In L3, provider adaptation erodes a high-scoring pair, and the contract change that restores effort can leave coding inflation in place. These results separate three questions that a single rollout score conflates: how well a mechanism performs initially, whether that performance survives the responses it invites, and what those responses do to care in each channel.

2 RESPONSE-DEPENDENT EVALUATION IN MEDI-SIM

We model a hospital over T periods with an administrator and J care teams. The administrator commits to a program controlling incentives and resources; teams apply a shared provider program to their local queues, capacity signals, and fatigue, while coding staff apply its billing rule. Programs

are fixed within an episode, but actions depend on the evolving state. Between episodes, search can revise the programs. This commit–respond structure motivates a Stackelberg design objective, which we evaluate through simulated responses instead of solving for a dynamic equilibrium. Teams share one provider program, so they compete for beds and bonuses but do not strategize independently.

2.1 Roles, response channels, and objectives

The administrator sets reduced-form financial and quality incentive weights (α_t, β_t), total and flexible capacity, bonus-pool size and tournament sharpness, and performance-score weights. Audit intensity is an additional lever in the administrative sweeps (L2), held fixed during program search (L3). Providers respond through *coding*, *selection*, *delay*, *effort*, and *triage/resource allocation*. Selection and delay measure different consequences of related triage decisions. The channels share a team utility combining financial, quality, and workload trade-offs (Ma, 1994; Holmstrom & Milgrom, 1991; Eggleston, 2005):

$$U_{j,t} = \alpha_t(\mathrm{Rev}_{j,t} - C_{j,t}) + \beta_t H^{\mathrm{tot}}_{j,t} + \theta\,\mathrm{Bonus}_{j,t} - \nu\left[\max(0, \mathrm{Load}_{j,t} - \mathrm{Stam}_{j,t})\right]^2. \quad (1)$$

Here H^{tot} and Load are total health and effort, Stam is available stamina, and positive θ, ν weight bonuses and fatigue cost. The same utility links billing, access, and care intensity: making one response less attractive can change the relative appeal of the others. This coupling lets us study pressure migration. L1/L2 use specified behavioral rules motivated by this utility; L3 can rewrite their exposed expressions. Incentive sweeps thus change the conditions a response rule faces, while program search changes the rule itself; in both cases providers act through a rule rather than an exact dynamic optimum.

The administrator evaluates a mechanism through the rollout induced by the provider response. Its discounted return G combines funds with reputation, a smoothed health score penalized for rejection and waiting. The response-dependent design objective is

$$\pi_A^* \in \arg\max_{\pi_A \in \Pi_A} \left\{\mathbb{E}_{s,\widehat{\pi}_P}[G] - \lambda_{\mathrm{unsafe}}\mathbb{E}[V] - \lambda_{\mathrm{var}}\mathrm{Var}_s[G]\right\}, \quad (2)$$

where s is an environment seed and $\widehat{\pi}_P = \mathcal{R}(\pi_A)$ is the program selected by the specified response rule or search procedure. V softly penalizes unsafe waiting, high-complexity deferral, up-coding, rejection, and insolvency; the variance term regularizes seed reliability. Because soft penalties can be traded against return, we also report channel outcomes directly. Appendix B gives the state, actions, and returns. Joint design, alternating responses, and population games probe this commitment problem from different angles; the coarse correlated equilibrium (CCE) used in the last serves as a stability diagnostic over program pairs, separate from the nested optimization in Eq. (2).

The practical evaluation question is therefore conditional: which outcomes does a mechanism produce under the specified response procedure, and what changes when that procedure finds a new provider program? Answering it requires the initial pair and subsequent responses to face the same operational constraints.

The IPS loop executes these decisions under common operational constraints (Figure 1). It combines case classification, constrained patient flow, and payment incentives (Fetter et al., 1980; Green, 2006; Ma, 1994). The three stages order the computation; in practice, coding and billing can span admission, treatment, and discharge. The following stages explain how financial or measured gains can separate from care outcomes; Appendix C gives their functional forms and design evidence.

2.2 Identify: classification and the coding wedge

Patients arrive exogenously through a homogeneous Poisson process, with arrival intensity and case mix fixed across policy comparisons. Each has a true diagnosis-related group (DRG), complexity measured by a normalized case-mix index (CMI), urgency, and waiting tolerance. Case-based reimbursement follows the reported group (Fetter et al., 1980; Centers for Medicare & Medicaid Services & Office of the Secretary, Department of Health and Human Services, 2025), whereas clinical need follows true attributes. Keeping these separate creates a *coding wedge*: a provider can raise reimbursement without changing the patient's need.

The coding program scores an upward-only set containing the true group and bounded higher-weight alternatives. Scores use reimbursement, audit penalties, and inflation-related signals, reflecting the

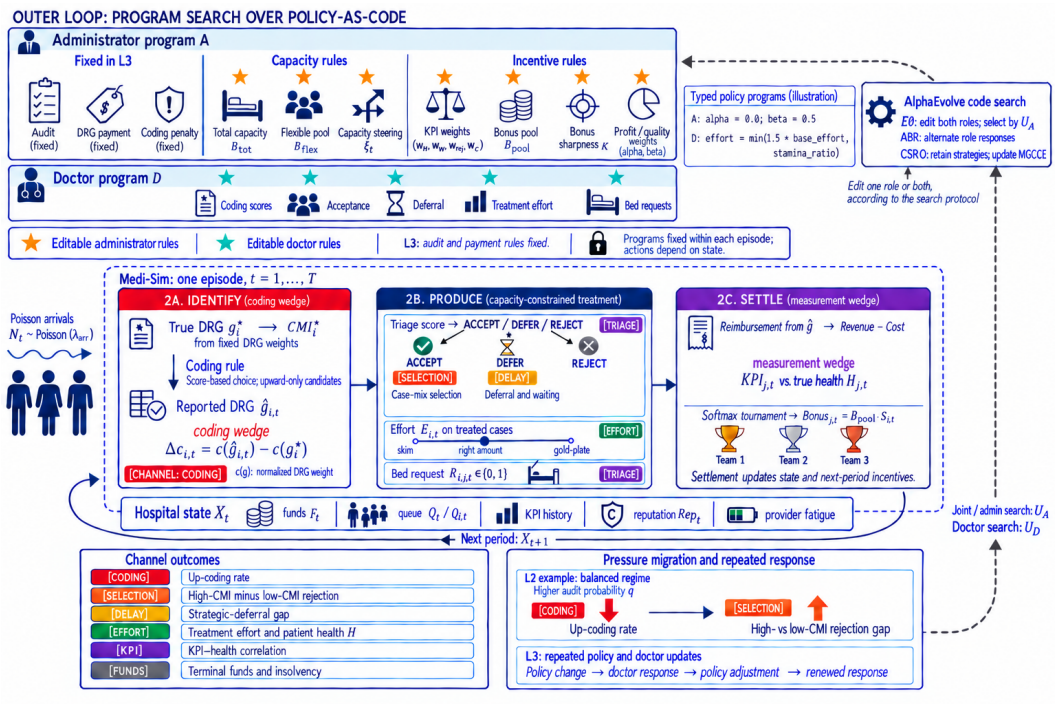


Figure 1: MEDI-SIM overview. Stars mark editable administrator and doctor rules. The outer loop searches programs through joint design, alternating responses or CSRO–MGCCE (§3); each episode executes a fixed pair through Identify–Produce–Settle. The L2 audit example shows lower up-coding alongside a wider high- versus low-CMI rejection gap. L3 keeps audit enforcement and payment rules fixed.

revenue–penalty trade-off in coding models (Dafny, 2005; Kuhn & Siciliani, 2008). The upward-only restriction isolates coding inflation. L3 can change the scoring logic, including thresholds and conditional structure, and selects its highest-scoring code; the host fixes admissible codes, audit enforcement, and settlement. Aggregate up-coding rates reflect the resulting choices across patients. The baseline score and stochastic alternative appear in Eq. (14).

2.3 PRODUCE: CAPACITY-CONSTRAINED TREATMENT

The reported group affects the financial value of a case, but care still requires time, effort, and a treatment resource. Patients enter team queues, where triage combines urgency, waiting, predicted margin, capacity, fatigue, and KPI signals to accept, reject, defer, or request a bed. Finite and flexible capacity connect these decisions to congestion (Green, 2002; Bekker et al., 2017). Effort determines treatment intensity, while selection determines who receives it (Ellis, 1998; Ma, 1994).

Health depends on skill and true complexity, has diminishing returns to effort, and decays with waiting; effort cost is convex (Eqs. (15)–(16)). Deferral adds a timing decision: postponing a costly case can relieve current pressure while reducing its eventual health benefit. It also changes subsequent waiting and congestion, so a response that lowers current treatment cost can affect later health and access through the same queues. Separating access from timing follows healthcare scheduling models (Cayirli & Veral, 2003; Gupta & Denton, 2008). The production and cost functions are smooth approximations chosen to expose these trade-offs rather than estimated from clinical data. L3 changes decision expressions while treatment, resource, and settlement mechanisms remain fixed.

2.4 SETTLE: BONUSES AND THE MEASUREMENT WEDGE

Settlement attaches reimbursement, audit penalties, and performance bonuses to the preceding choices. Team performance combines health with waiting, rejection, and cost penalties:

$$\mathrm{KPI}_{j,t} = w_H \bar{H}_{j,t} - w_W \bar{W}_{j,t} - w_{\mathrm{rej}} \mathrm{Reject}_{j,t} - w_C \bar{C}_{j,t}. \tag{3}$$

A softmax tournament allocates the period's bonus pool using these scores, following relative-performance incentives (Lazear & Rosen, 1981). The resulting bonus pressure feeds back into effort and triage (Eqs. (17)–(19)). Its consequences depend on the measured objective: treating a complex patient can raise health while worsening the score's cost or waiting components. This *measurement wedge* permits proxy improvement without better care, as in multitask and performance-payment models (Baker, 1992; Bevan & Hood, 2006; Eijkenaar et al., 2013). A stronger reward for the same score can therefore redirect patient selection instead of simply raising treatment effort. Tracking these choices alongside coding, queues, and access lets us evaluate where an intervention moves pressure.

Together, the coding and measurement wedges create two routes by which financial or measured performance can improve without a corresponding improvement in care. We next use IPS rollouts as feedback for searching policies and testing the responses they invite.

3 STRATEGIC POLICY-AS-CODE

The simulator evaluates a policy pair; program search discovers new pairs and responses. We use the same representation to move from joint design to unilateral adaptation, then to responses against historical opponents.

Policy representation and search. A pair (A, D) contains 11 administrative expressions and five doctor expressions for effort, acceptance, deferral, bed requests, and coding scores. Typed, assignment-only programs can change conditional logic within a fixed feature and action interface; the host checks and projects their outputs. We implement evolutionary search with OPENEVOLVE (Sharma, 2025), an open-source evolutionary coding agent modeled on ALPHAEVOLVE (Novikov et al., 2025) and FunSearch (Romera-Paredes et al., 2024). Training rollouts guide LLM edits; validation selects among retained references, shortlisted candidates, and prior winners. Inspection shows how a selected rule earns its return; whether it also delivers good care is measured separately through channel outcomes. Algorithms A.2–A.5 and Appendix L specify candidate selection and initialization.

Here an LLM proposal changes the logic of a policy, such as the conditions under which a provider exerts effort or accepts a case. The host then checks and executes that proposal within the existing action interface. This permits semantic search over decision rules while keeping their operational consequences comparable.

Joint design and unilateral adaptation. Joint search (E0) edits both programs and selects by administrative utility U_A. Strategic evaluation then gives each role control of its own program under its own objective. A unilateral search edits that role against a frozen opponent distribution:

$$J_A(A; q_D) = \sum_j q_D(j) U_A(A, D_j), \qquad J_D(D; q_A) = \sum_i q_A(i) U_D(A_i, D). \tag{4}$$

A fixed opponent is a point mass. U_D averages team utility in Eq. (1); U_A is the fixed administrative score with soft penalties (Appendix L.1). Role checks reject edits to the other program, and evaluations use the actual opponent. Alternating best-response search (ABR) selects each update by validation payoff, retaining the incumbent among eligible candidates. Both orders start from E0: administrator-first ABR-A tests whether an improvement survives provider response; doctor-first ABR-D tests whether revision repairs an observed failure. "Best response" here means the program selected by budget-limited search, which need not be an exact optimum.

Responses to historical strategies. A response to the latest opponent can be vulnerable to a strategy encountered earlier. We retain all earlier programs, including those currently assigned zero weight, using Code-Space Response Oracles (CSRO) (Hennes et al., 2026) and the Joint Policy-Space

Response Oracles framework (Marris et al., 2021). Starting from E0, each round freezes opponent distributions, searches both roles, admits validation-improving programs, and completes all new cross-play pairs. A maximum-Gini CCE (MGCCE) solver assigns joint weights using training payoffs and retained-library deviation constraints. Searches face its opponent marginals; payoff and patient-outcome evaluation uses the full joint distribution (Appendix L.4). These constraints cover retained programs under the specified utilities, so a stable mixture can still deliver poor care; we therefore evaluate stability and care separately.

Conditional convergence and approximate certificates. History retention lets the game grow even when individual responses cycle. With a finite policy universe, exact response oracles, and the same exact payoffs for admission and the meta-solver, strictly improving additions occur only finitely often; the first round without an addition yields a CCE over that universe (Proposition 2). With approximate search and separate training and validation, Proposition 3 bounds the remaining target deviation gain by

$$\epsilon_{\text{target}} \;\leq\; \epsilon_{\text{solver}} + \eta + \xi + 2\kappa_{\text{tv}} + 2\delta, \tag{5}$$

where η is the admission threshold, ξ the oracle error, κ_{tv} the uniform training–validation payoff discrepancy, and δ the uniform training–target error. Appendix N gives the assumptions and proofs. The bound separates numerical, search, and payoff-estimation error. Full cross-play lets us measure deviation gains within the retained library, and frozen opponents keep candidate and incumbent comparisons on the same response task. The oracle error ξ, and with it robustness to programs outside the library, remains unmeasured.

4 Experiments

We evaluate Medi-Sim through three increasingly demanding questions. **L1:** how do incentives distribute behavior across response channels? **L2:** can a targeted intervention improve one outcome without worsening another? **L3:** do gains from AI-guided program search survive subsequent strategic adaptation?

Literature consistency and setup. Nine comparisons check the intended incentive and operational mechanisms (Appendix Table I.1). Because the same literature informed the design, agreement in direction serves as a behavioral sanity check rather than independent validation. The table distinguishes literature-supported mechanisms from simulator findings such as cross-channel substitution and adverse KPI-steered allocation. L1/L2 report 30-seed means with maximum horizon $T = 200$; Appendix H specifies the grid and lever settings.

We read outcomes jointly. H is mean simulated health among treated patients, so it must be assessed alongside access. Up-coding is the period-averaged share of monitored patients assigned a higher-weight DRG than the true group, with queued patients included until they leave; rejection and deferral gaps subtract low-true-CMI rates from high-true-CMI rates among considered patients. F_T is terminal funds, and U_A includes distortion and seed-variance penalties. Appendix P gives the measurement conventions.

4.1 L1: How Do Incentives Redistribute Provider Behavior?

L1 sweeps financial sensitivity α and quality sensitivity β over an 11×11 grid, holding provider response rules fixed. The resulting landscape lets us follow coding, access, effort, and timing as incentives change.

Classical failures occupy different regions of the same landscape. The response surface is structured but non-monotone (Figure 2). Zero profit/quality weights, with bonuses retained, accompany broad rejection. The profit-driven regime combines coding and selection (up-coding 0.226, high-CMI rejection gap 0.182); the quality-driven regime has lower coding and higher effort and budget pressure; the balanced regime combines lower rejection with selective delay and KPI targeting (Appendix E). These regimes connect familiar incentive-theory predictions within one environment, making their trade-offs directly comparable.

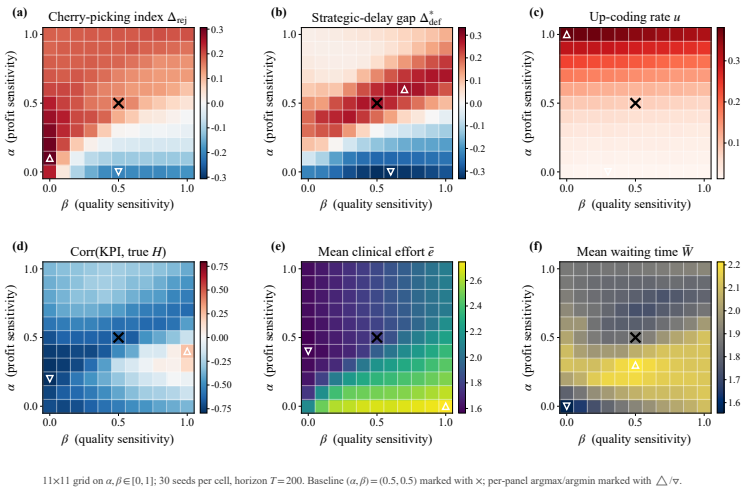


Figure 2: L1 incentive phase diagram. Panels report 30-seed means over the 11×11 (α, β) grid. Regions exhibit low-incentive access rationing, profit-driven coding and selection, quality-driven effort and budget pressure, and balanced-interior delay and KPI targeting.

Balanced incentives leave less-visible distortions. In the balanced regime, coding and rejection are lower than in the profit-driven regime, but high-CMI patients face strategic deferral (postponement while local beds are estimated to be free) at a rate of 0.290, versus 0.010 for low-CMI patients. Measured KPI has correlation -0.659 with team mean health. Case mix, waiting decisions, and bonus responses supply the behavioral context (Appendix E). A coding-only or rejection-only assessment would miss this interior failure. Supplementary response-form comparisons retain similar grid-wide coding and effort rankings, while stronger saturation changes delay and waiting (Appendix J).

4.2 L2: Do Targeted Interventions Reduce Distortion or Move It?

Can an administrator address one failure without worsening another? L2 varies one lever at a time in balanced, profit-driven, and quality-driven regimes. Provider programs remain fixed, isolating responses within the specified behavioral family.

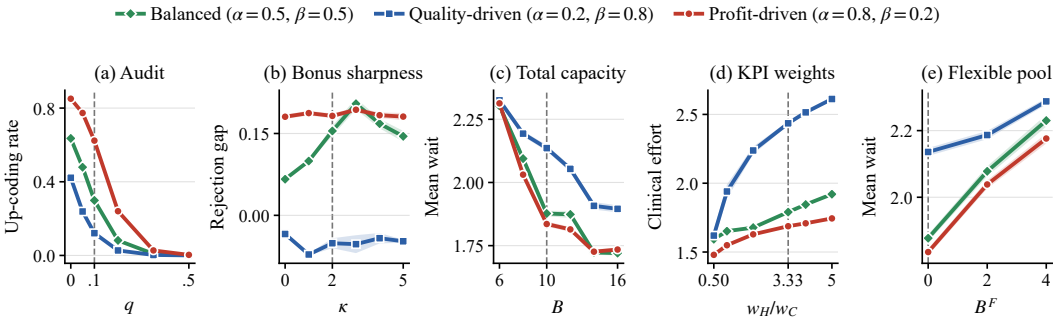


Figure 3: L2 policy ablations. Curves report 30-seed means at $T = 200$; shading denotes pointwise 95% confidence intervals (Appendix H) and dashed lines mark baselines. Audits suppress up-coding, total capacity generally lowers waiting, and KPI/bonus/flex levers induce nonlinear responses.

Audits improve billing accuracy while access becomes more selective. In the balanced regime, raising audit probability from 0 to 0.5 reduces up-coding from 0.636 to 0.001 while the high-minus-low-CMI rejection gap rises from 0.100 to 0.233 (Appendix Table F.1; Figure 3a shows coding). Improved billing accuracy thus coincides with worse access for complex patients. The two outcomes

illustrate pressure migration: suppressing the billing channel leaves incentives expressed through selection.

Stronger proxy incentives can amplify the mismatch. Raising the bonus pool from 0 to 15 makes KPI–health correlation more negative (-0.447 to -0.839) and widens the rejection gap (0.066 to 0.479; Appendix Table F.1). The widening rejection gap is what ties the weaker KPI–health alignment to behavior; the correlation alone would not. Appendix Figure F.1 gives the budget and correlation curves. Likewise, total-capacity expansion generally lowers waiting, but a larger KPI-steered flexible pool at fixed total capacity can increase it. Steering controls and further sweeps appear in Appendices F–G. These interventions move pressure even with fixed response rules. We next let providers search for new ones.

4.3 L3: Do Program-Search Gains Survive Strategic Adaptation?

L3 asks whether gains also erode across adaptation stages, and which channel each new response uses. It starts with joint design, then follows unilateral adaptation and responses to history. DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) and Z.ai GLM-5.3 (Z.ai, 2026; GLM-5-Team, 2026) edit code, with 200 proposals for E0 and each ABR update. Selection uses five validation environments; reported outcomes average 20 held-out environment seeds at $T = 200$. These seeds measure rollout variability for selected programs; targeted response tasks also use repeated searches (Appendix L).

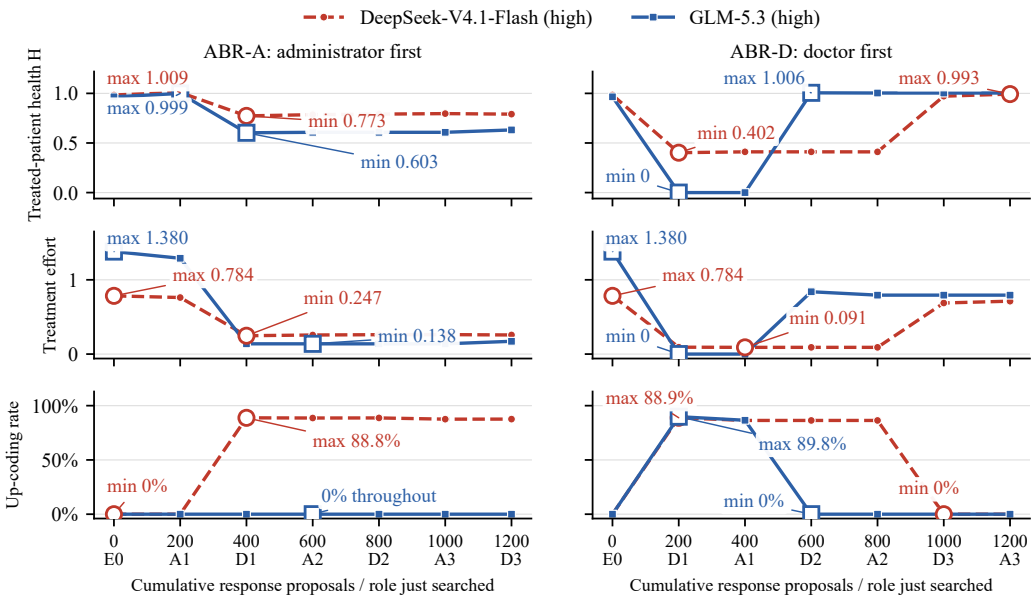


Figure 4: **Doctor adaptation and administrative repair.** Original-prompt/high DeepSeek and GLM paths start from their own E0. ABR-A/ABR-D update the administrator/doctor first; A/D mark 200-proposal role updates. Appendix L.7 gives utilities and transitions.

Joint-design gains invite profitable responses. E0 exceeds the best supplied start in administrative utility in all five editor configurations (Appendix Figure L.1). This establishes that program search can find useful changes within the interface. Its strategic significance depends on what happens when the jointly selected doctor program can change under the doctor objective.

In both administrator-first reference paths, doctors respond to an improved policy by raising their utility while reducing effort and health. At the first doctor update, H falls from 1.009 to 0.773 for DeepSeek and from 0.999 to 0.603 for GLM (Figure 4; Appendix Table L.4). The two responses use different channels: DeepSeek also raises up-coding from 0% to 88.8%, whereas GLM leaves it at zero, so a billing-only check would miss the second even though care deteriorates in both.

Revision changes the next response. Doctor-first ABR shows how administrators respond after doctors reduce effort. In both reference paths, removing the profit weight α precedes recovery at the next doctor update. The DeepSeek D2–A2 transition isolates the timing: the contract changes while the doctor and health remain unchanged. This suggests that revision changes which response is attractive, with the care benefit realized only after adaptation. We test this interpretation by repeating doctor search from that same low-effort program under contracts differing only in α. All three searches at $\alpha = 0.5$ retain the starting doctor; all three at $\alpha = 0$ raise effort and health (Appendix Figure L.4). The recovery is specific to effort and treated-patient health: in these searches up-coding stays near 86% and one raises waiting from 0.806 to 3.041 (Appendix L.8), although the reference path's own D3 update also removes up-coding (Figure 4). Restoring effort thus does not require closing the coding channel; as with audits in L2, repair on one channel can leave distortion in another. Code inspection shows why: GLM and revised-prompt DeepSeek both end ABR-D with $\alpha = 0$, but GLM scores codes with `upcode_pressure`, which contains α, whereas DeepSeek uses raw `upcode_gain`, so endpoint up-coding is 0% for GLM but 85.8% for DeepSeek (Appendix L.6).

Recovery remains vulnerable. We test further adaptation directly at the final DeepSeek ABR-D policy A3: three additional doctor searches all raise doctor utility and reduce health. In the first repeat, H falls from 0.994 to 0.878 while doctor utility rises by 0.474; in all three, up-coding returns from 0% to 87–89% (Appendix L.8). A pair that has recovered high health thus remains a starting point for further adaptation. Together with the matched-contract checks, this separates initial improvement, administrative repair, and vulnerability after repair.

History broadens evaluation without ensuring care quality. CSRO retains earlier administrative and doctor programs, letting each role search against a distribution of past opponents. MGCCE updates these distributions from cross-play payoffs. Earlier policies can regain relevance as doctors adapt: in the DeepSeek three-round reference run, adding a new doctor response restores positive weight to an administrative policy previously assigned zero weight. The GLM twelve-round run retains multiple generations of both roles in its final mixture (Appendix L.9).

The mixture analysis also distinguishes strategic stability from quality of care. On fixed retained libraries, MGCCE lowers validation deviation gains relative to uniform weighting for both roles in all four reference games (Appendix Table L.6), yet patient outcomes vary widely (Appendix Table L.3). At the extreme, revised-prompt DeepSeek (200×3) puts 99.94% of the doctor marginal on `effort = 0.0` ($H < 0.001$), a boundary case: settlement pays for coded cases regardless of effort. Administrator-first erosions occur at positive effort. Initial administrative utility, resistance to tested responses, and channel-level care outcomes remain separate evaluation questions.

5 Discussion, Conclusion, and Limitations

MEDI-SIM makes a central difficulty in AI-driven mechanism design observable: optimizing a rule changes the behavior against which it is evaluated. By connecting hospital operations to inspectable programs, it follows this feedback from fixed-rule responses through successive rounds of strategic search. An intervention can suppress coding while worsening access, restore effort while leaving coding distortion, or recover health before another response erodes it. The same search capability that discovers an administrative improvement can discover a response that undermines it. Strategic evaluation should therefore accompany policy generation, checking a mechanism's initial score, whether that score survives the responses it invites, and what those responses do to care in each channel. The released environment and program interface provide a common setting for these checks.

The environment uses stylized patient classes, arrivals, health production, and provider responses without calibration to hospital data. Search findings depend on the program interface, objectives, prompts, initialization, and finite budgets. History broadens the tested responses, but experimental certificates concern the retained game; stronger guarantees require the oracle and payoff assumptions in Appendix N. Stability does not establish care quality, and treated-patient health must be read with access. Literature checks and sensitivity analyses support interpretation within the tested settings. Empirical calibration, richer provider populations, repeated searches, and domain and fairness review remain necessary for operational relevance (Appendix M). MEDI-SIM currently serves as a controlled stress test for mechanism design.

Ethical Considerations

The simulator studies high-stakes healthcare operations. Its policies must not be interpreted as clinical recommendations or administrative guidance for real hospitals without domain validation, safety review, and fairness analysis. Modeling behaviors such as up-coding, cherry-picking, and strategic delay is intended for detection and stress testing, not for operationalizing manipulation. Any future deployment would require safeguards for patient access, equity across case complexity, clinical safety, privacy, and legal compliance.

AI Use Statement

AI assistants were used for code assistance, experiment organization, figure preparation and manuscript editing. The authors are responsible for checking all claims, numerical results, citations and submitted material.

Reproducibility Statement

Appendix A specifies execution and search algorithms; Appendices H and L document experimental settings, search budgets, and assessment protocols; Appendix O defines the editable interface and host constraints.

The anonymous implementation, released experiment records, and reproduction instructions are available at `https://anonymous.4open.science/r/Medi_Sim-CC85/`.

References

George P. Baker. Incentive contracts and performance measurement. *Journal of Political Economy*, 100(3):598–614, 1992. doi: 10.1086/261831.

René Bekker, Ger Koole, and Dennis Roubos. Flexible bed allocations for hospital wards. *Health Care Management Science*, 20(4):453–466, 2017. doi: 10.1007/s10729-016-9364-4.

Gwyn Bevan and Christopher Hood. What's measured is what matters: Targets and gaming in the english public health care system. *Public Administration*, 84(3):517–538, 2006. doi: 10.1111/j.1467-9299.2006.00600.x.

Sally C. Brailsford, Paul R. Harper, Bhakti Patel, and Martin Pitt. An analysis of the academic literature on simulation and modelling in health care. *Journal of Simulation*, 3(3):130–140, 2009. doi: 10.1057/jos.2009.10.

Donald T. Campbell. Assessing the impact of planned social change. *Evaluation and Program Planning*, 2(1):67–90, 1979. doi: 10.1016/0149-7189(79)90048-X. URL `https://doi.org/10.1016/0149-7189(79)90048-X`.

Tugba Cayirli and Emre Veral. Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, 12(4):519–549, 2003. doi: 10.1111/j.1937-5956.2003.tb00218.x.

Centers for Medicare & Medicaid Services and Office of the Secretary, Department of Health and Human Services. Medicare program; hospital inpatient prospective payment systems for acute care hospitals (ipps) and the long-term care hospital prospective payment system and policy changes and fiscal year (fy) 2026 rates; changes to the fy 2025 ipps rates due to court decision; requirements for quality programs; and other policy changes; health data, technology, and interoperability: Electronic prescribing, real-time prescription benefit and electronic prior authorization. Federal Register, final rule 2025-14681, published August 4, 2025, 2025. URL `https://www.federalregister.gov/d/2025-14681`. Fiscal year 2026 payment rates.

Leemore S. Dafny. How do hospitals respond to price changes? *American Economic Review*, 95(5):1525–1547, 2005. doi: 10.1257/000282805775014236. URL `https://www.aeaweb.org/articles?id=10.1257/000282805775014236`.

DeepSeek. DeepSeek-V4.1-Flash: Smarter, faster, more efficient, 2026. URL https://api-docs.deepseek.com/news/news260910/. Released September 10, 2026; accessed September 22, 2026.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026. URL https://arxiv.org/abs/2609.19969.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pp. 55–70, 2018. doi: 10.1145/3219166.3219193.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning: Advances in differentiable economics. *Journal of the ACM*, 71(1):1–53, 2024. doi: 10.1145/3630749. URL https://doi.org/10.1145/3630749.

Karen Eggleston. Multitasking and mixed systems for provider payment. *Journal of Health Economics*, 24(1):211–223, 2005. doi: 10.1016/j.jhealeco.2004.09.001.

Frank Eijkenaar, Martin Emmert, Manfred Scheppach, and Oliver Schöffski. Effects of pay for performance in health care: A systematic review of systematic reviews. *Health Policy*, 110(2–3): 115–130, 2013. doi: 10.1016/j.healthpol.2013.01.008.

Randall P. Ellis. Creaming, skimping and dumping: Provider competition on the intensive and extensive margins. *Journal of Health Economics*, 17(5):537–555, 1998. doi: 10.1016/S0167-6296(97)00042-8.

R. B. Fetter, Y. Shin, J. L. Freeman, R. F. Averill, and J. D. Thompson. Case mix definition by diagnosis-related groups. *Medical Care*, 18(2 Suppl):iii, 1–53, 1980. URL https://pubmed.ncbi.nlm.nih.gov/7188781/.

Michael Geruso and Timothy Layton. Upcoding: Evidence from Medicare on squishy risk adjustment. *Journal of Political Economy*, 128(3):984–1026, 2020. doi: 10.1086/704756.

GLM-5-Team. GLM-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1):16–18, 2019. doi: 10.1038/s41591-018-0310-5. URL https://pubmed.ncbi.nlm.nih.gov/30617332/.

Linda V. Green. How many hospital beds? *Inquiry*, 39(4):400–412, 2002. doi: 10.5034/inquiryjrnl_39.4.400.

Linda V. Green. Queueing analysis in healthcare. In Randolph W. Hall (ed.), *Patient Flow: Reducing Delay in Healthcare Delivery*, pp. 281–307. Springer, Boston, MA, 2006. doi: 10.1007/978-0-387-33636-7_10. URL https://doi.org/10.1007/978-0-387-33636-7_10.

Diwakar Gupta and Brian Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, 40(9):800–819, 2008. doi: 10.1080/07408170802165880.

Moritz Hardt, Nimrod Megiddo, Christos H. Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pp. 111–122, 2016. doi: 10.1145/2840728.2840730.

Daniel Hennes, Zun Li, John Schultz, and Marc Lanctot. Code-Space Response Oracles: Generating interpretable multi-agent policies with large language models. *arXiv preprint arXiv:2603.10098*, 2026. URL https://arxiv.org/abs/2603.10098.

Bengt Holmstrom and Paul Milgrom. Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design. *The Journal of Law, Economics, and Organization*, 7(Special Issue): 24–52, 1991. doi: 10.1093/jleo/7.special_issue.24.

Seth Karten, Wenzhe Li, Zihan Ding, Samuel Kleiner, Yu Bai, and Chi Jin. LLM Economist: Large population models and mechanism design in multi-agent generative simulacra, 2025. URL https://arxiv.org/abs/2507.15815.

Diwas S. Kc and Christian Terwiesch. Impact of workload on service time and patient safety: An econometric analysis of hospital operations. *Management Science*, 55(9):1486–1498, 2009. doi: 10.1287/mnsc.1090.1037.

Matthieu Komorowski, Leo A. Celi, Omar Badawi, Anthony C. Gordon, and A. Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716–1720, 2018. doi: 10.1038/s41591-018-0213-5. URL https://pubmed.ncbi.nlm.nih.gov/30349085/.

Richard Kronick and W. Pete Welch. Measuring coding intensity in the Medicare Advantage program. *Medicare & Medicaid Research Review*, 4(2):E1–E19, 2014. doi: 10.5600/mmrr.004.02.sa06. URL https://pubmed.ncbi.nlm.nih.gov/25068076/.

Michael Kuhn and Luigi Siciliani. Upcoding and optimal auditing in health care (or the economics of DRG creep). CEPR Discussion Paper 6689, Centre for Economic Policy Research, 2008. URL https://cepr.org/publications/dp6689.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3323fe11e9595c09af38fe67567a9394-Abstract.html.

Edward P. Lazear and Sherwin Rosen. Rank-order tournaments as optimum labor contracts. *Journal of Political Economy*, 89(5):841–864, 1981. doi: 10.1086/261010.

Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. Evolution through large models. In Wolfgang Banzhaf, Penousal Machado, and Mengjie Zhang (eds.), *Handbook of Evolutionary Machine Learning*, Genetic and Evolutionary Computation, pp. 331–366. Springer Nature Singapore, 2024. doi: 10.1007/978-981-99-3814-8_11. URL https://doi.org/10.1007/978-981-99-3814-8_11. First published online November 2, 2023; preprint arXiv:2206.08896, 2022.

Sagi Levanon and Nir Rosenfeld. Strategic classification made practical. In *International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 6243–6253, 2021. URL https://proceedings.mlr.press/v139/levanon21a.html.

Ching-to Albert Ma. Health care payment systems: Cost and quality incentives. *Journal of Economics & Management Strategy*, 3(1):93–112, 1994. doi: 10.1111/j.1430-9134.1994.00093.x.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.12931.

Joshua C. Mandel, David A. Kreda, Kenneth D. Mandl, Isaac S. Kohane, and Rachel B. Ramoni. SMART on FHIR: A standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908, 2016. doi: 10.1093/jamia/ocv189. URL https://pubmed.ncbi.nlm.nih.gov/26911829/.

David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's law. *arXiv preprint arXiv:1803.04585*, 2018. URL https://arxiv.org/abs/1803.04585.

Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7480–7491, 2021. URL https://proceedings.mlr.press/v139/marris21a.html.

Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995. doi: 10.1006/game.1995.1023.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023. doi: 10.1145/3586183.3606763.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pp. 7599–7609, 2020. URL https://proceedings.mlr.press/v119/perdomo20a.html.

Carol Propper, Matt Sutton, Carolyn Whitnall, and Frank Windmeijer. Incentives and targets in hospital care: Evidence from a natural experiment. *Journal of Public Economics*, 94(3–4):318–335, 2010. doi: 10.1016/j.jpubeco.2010.01.002. URL https://doi.org/10.1016/j.jpubeco.2010.01.002.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6. URL https://www.nature.com/articles/s41586-023-06924-6.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x. URL https://doi.org/10.1038/s42256-019-0048-x.

Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *Principles and Practice of Constraint Programming – CP 2003*, volume 2833 of *Lecture Notes in Computer Science*, pp. 19–36. Springer, 2003. doi: 10.1007/978-3-540-45193-8_2. URL https://doi.org/10.1007/978-3-540-45193-8_2.

Asankhaya Sharma. OpenEvolve: An open-source evolutionary coding agent, 2025. URL https://github.com/algorithmicsuperintelligence/openevolve. GitHub repository.

Elaine Silverman and Jonathan Skinner. Medicare upcoding and hospital ownership. *Journal of Health Economics*, 23(2):369–389, 2004. doi: 10.1016/j.jhealeco.2003.09.007.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 9460–9471, 2022. doi: 10.52202/068431-0687. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/3d719fee332caa23d5038b8a90e81796-Abstract-Conference.html.

Howard R. Strasberg, Bryn Rhodes, Guilherme Del Fiol, Robert A. Jenders, Peter J. Haug, and Kensaku Kawamoto. Contemporary clinical decision support standards using Health Level Seven International Fast Healthcare Interoperability Resources. *Journal of the American Medical Informatics Association*, 28(8):1796–1806, 2021. doi: 10.1093/jamia/ocab070. URL https://pubmed.ncbi.nlm.nih.gov/34100949/.

U.S. Food and Drug Administration, Health Canada, and Medicines and Healthcare products Regulatory Agency. Good machine learning practice for medical device development: Guiding principles. Guidance document, October 2021. URL https://www.fda.gov/media/153486/download.

Pieter Van Herck, Delphine De Smedt, Lieven Annemans, Roy Remmen, Meredith B. Rosenthal, and Walter Sermeus. Systematic review: Effects, design choices, and context of pay-for-performance in health care. *BMC Health Services Research*, 10:247, 2010. doi: 10.1186/1472-6963-10-247.

Alexander Sasha Vezhnevets, John P. Agapiou, Avia Aharon, Ron Ziv, Jayd Matyas, Edgar A. Duéñez-Guzmán, William A. Cunningham, Simon Osindero, Danny Karmon, and Joel Z. Leibo. Generative agent-based modeling with actions grounded in physical, social, or digital space using Concordia, 2023. URL https://arxiv.org/abs/2312.03664.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2309.03409. URL https://arxiv.org/abs/2309.03409.

Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys*, 55(1):1–36, 2023. doi: 10.1145/3477600. First published online November 23, 2021; journal issue January 2023.

Z.ai. GLM-5.3: Model overview, 2026. URL https://docs.z.ai/guides/llm/glm-5.3. Accessed September 22, 2026.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The AI economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022. doi: 10.1126/sciadv.abk2607.

APPENDIX CONTENTS

The appendices follow the evaluation from simulator execution to strategic response. The guide below groups the material by its role in the argument.

Model and procedures (A–D). Execution and search algorithms appear in Appendix A, followed by the response-dependent objectives (B), environment equations (C) and related work (D).

Fixed-rule experiments (E–K). Appendix E develops the L1 access, delay, KPI and treatment results. Appendices F–G expand L2; Appendix H gives their settings. Literature checks (I), response-form sensitivity (J) and the severity-score extension (K) complete this evidence.

Program search and interpretation (L–P). Appendix L documents E0, ABR, repeated response checks and CSRO–MGCCE. Discussion (M), theory (N), the program interface (O) and measurement definitions (P) provide the interpretive and implementation details.

A EXECUTION AND SEARCH ALGORITHMS

The execution and search procedures operate on two time scales. During an episode, a fixed administrator–provider program pair acts on changing queues, capacity and performance signals. Between episodes, the search procedure uses rollout feedback to propose and select new programs. Algorithm A.1 specifies the shared IPS execution loop; Algorithms A.2–A.5 use that loop for joint design, unilateral response and population expansion.

Algorithm A.1 MEDI-SIM simulator execution loop

Require: Hospital capacity and queues, hospital provider teams, hospital administrator policy, reimbursement module, KPI/bonus rule, initial state X_1.

for $t = 1, \ldots, T$ **do**
  Hospital policy step: the hospital administrator sets period levers u_t, including incentives, capacity, bonus-pool, KPI weights, and steering parameters; the audit schedule is fixed in L3.
  Patient generation: sample arrivals $\mathcal{P}_t$ from the configured arrival process and assign latent clinical attributes such as true group, urgency, tolerance, and true complexity $\mathrm{CMI}_i^\star$.
  Identify / coding: hospital coding staff map each arriving patient's clinical presentation to a billable group $\hat{g}_{i,t}$; the gap between true and coded complexity realizes the coding wedge.
  Routing and queueing: register or route patients to hospital care-team queues, carry over backlog, and increment waiting time for patients not served in the current period.
  for each hospital care team j **do**
    Observe local queue $\mathcal{Q}_{j,t}$, capacity, fatigue, incentives, and prior KPI signals.
    Choose provider-side actions: accept/defer/reject, effort $E_{ij,t}$ for accepted cases, and resource requests under the response class Π_P.
  end for
  Capacity resolution and treatment: allocate resources $R_{ij,t}$ under hard capacity constraints, treat selected patients, keep unserved patients queued or rejected according to the triage decision, and realize health output $H_{ij,t}$ and cost $C_{ij,t}$.
  Settlement: compute reimbursement from the reported billing group $\hat{g}_{i,t}$, aggregate margin and KPI scores, apply audit penalties, and allocate bonuses.
  State update: update funds, reputation, queues, provider fatigue, KPI history, and the next hospital state X_{t+1}.
end for

Ensure: Rollout trajectory, hospital administrator return, and channel-level outcomes.

Each rollout links decisions to their later consequences: coding affects reimbursement, triage and effort affect care and congestion, and settlement updates the state seen in the next period. Search then uses the resulting utilities and channel outcomes to choose the programs evaluated in subsequent rollouts. Across E0, ABR and CSRO, the execution rules stay fixed while the editable role and opponent distribution change.

Algorithm A.2 Joint OPENEVOLVE policy search (E0)

Require: Human role pools $\mathcal{A}_h, \mathcal{D}_h$, neutral pair (A_n, D_n), budget B, training/validation seeds.
$\mathcal{L}_0 \leftarrow \mathrm{dedup}(\{(A, D_n) : A \in \mathcal{A}_h\} \cup \{(A_n, D) : D \in \mathcal{D}_h\})$
Evaluate $\mathcal{L}_0$ on training seeds; initialize the evolutionary database and a proposal journal.
for $b = 1, \ldots, B$ **do**
    Sample a parent and inspirations; request a joint DSL edit using training feedback.
    Check syntax, allowed features and typed outputs; log invalid/duplicate proposals.
    if candidate passes compilation and evaluation **then**
        Evaluate U_A and outcome measures; update the database and persist the complete proposal record.
    end if
end for
$\mathcal{C} \leftarrow \mathcal{L}_0 \cup \mathrm{Top5}_{\mathrm{train}}(\text{valid generated programs}) \cup$ prior selection winners
Re-evaluate $\mathcal{C}$ on validation seeds; select (A_0, D_0) by U_A.
Ensure: Selected pair (A_0, D_0) and its validation outcomes.

Algorithm A.3 Unilateral OPENEVOLVE response oracle $\mathcal{R}_r^B$

Require: Role r, frozen opponent distribution q_{-r}, retained own-role references $\mathcal{I}_r$, search-only warm starts $\mathcal{W}_r$, budget B.
Freeze opponent programs/probabilities and evaluation context; evaluate $\mathcal{I}_r \cup \mathcal{W}_r$ against q_{-r} on training seeds.
for $b = 1, \ldots, B$ **do**
    Sample a parent and inspirations; ask the editor to change only role r.
    Validate DSL and reject any change to the frozen opposite-role section.
    if valid **then**
        Compose the candidate with every actual opponent; evaluate the weighted role objective in Eq. (4).
        Update the evolutionary database and journal; retain training outcome summaries for editing feedback.
    end if
end for
$\mathcal{C}_r \leftarrow \mathcal{I}_r \cup \mathrm{Top5}_{\mathrm{train}}(\text{generated roles}) \cup$ prior permitted winners
Select r^+ by validation payoff against the same frozen q_{-r}.
$g \leftarrow J_r^{\mathrm{val}}(r^+; q_{-r}) - \max_{r' \in \mathcal{I}_r} J_r^{\mathrm{val}}(r'; q_{-r})$
Ensure: Selected role program r^+ and its validation gain g.

Algorithm A.4 Alternating best-response search (ABR) with finite-budget response oracles

Require: E0 pair (A_0, D_0), order $o \in \{(A, D), (D, A)\}$, cycles $C = 3$, per-role budget $B = 200$.
$(A, D) \leftarrow (A_0, D_0)$; publish stage zero.
for $c = 1, \ldots, C$ **do**
    for role r in order o **do**
        Freeze the current opposite program as a point-mass opponent.
        Run $\mathcal{R}_r^B$, keeping the current program eligible; update that role with its selected result.
        Record the complete pair, both utilities and channel outcomes, including no-change selections.
    end for
end for
Ensure: Initial pair and six selected pairs, with both utilities and channel outcomes at each stage.

Algorithm A.5 JPSRO(CCE)-style CSRO with an MGCCE meta-solver

Require: E0 pair, rounds R, per-role budget B, fixed train/validation seeds.
$\mathcal{A}_0 \leftarrow \{A_0\}, \mathcal{D}_0 \leftarrow \{D_0\}, \rho_0 \leftarrow [1]$.
for $t = 0, \ldots, R-1$ **do**
    Freeze $(\mathcal{A}_t, \mathcal{D}_t, \rho_t)$; compute marginals q_A^t, q_D^t.
    $(A^+, g_A) \leftarrow \mathcal{R}_A^B(q_D^t; \mathcal{A}_t)$.
    $(D^+, g_D) \leftarrow \mathcal{R}_D^B(q_A^t; \mathcal{D}_t)$, using the *same round-start* population.
    for $(r, r^+, g_r) \in \{(A, A^+, g_A), (D, D^+, g_D)\}$ **do**
        Admit r^+ if it is not already in the role's pool and $g_r > 10^{-8}$; otherwise retain the pool.
    end for
    Evaluate all newly formed cross-pairs on training and validation seeds.
    Solve Eq. (23) on the training matrices and verify feasibility/optimality.
    Publish ρ_{t+1} and joint-distribution outcomes after completing and checking the cross-play matrix.
end for
Ensure: Final pools, joint distribution, per-round response/admission records and train/validation deviation gains.

B Formal Definitions for Response-Dependent Evaluation

This appendix specifies the state, actions and rollout score underlying the response-dependent objective in §2. The state records the hospital's operational position, the two programs determine its decisions, and the administrative score combines the financial and care outcomes of their interaction. Channel-level measurements then explain how a program pair obtains that score.

Hospital state. We summarize the hospital state as

$$X_t = \big(F_t,\ Q_t,\ \{\mathcal{Q}_{j,t}, \mathrm{Stam}_{j,t}\}_{j=1}^{J},\ \mathrm{KPI}_{t-1},\ \mathrm{Rep}_t\big), \tag{6}$$

where F_t denotes funds, Q_t summarizes congestion, and $\mathcal{Q}_{j,t}$ contains team j's queued patients and their attributes. Available stamina $\mathrm{Stam}_{j,t}$ falls with delivered effort and recovers between periods. The remaining terms record previous-period KPI and reputation. Stamina and the overload penalty provide a stylized representation of workload pressure, motivated by evidence linking workload to service and patient safety (Kc & Terwiesch, 2009); the specific recovery rule and quadratic penalty are not clinically calibrated. Each program reads the features exposed by its interface (Appendix O).

Administrator action. The administrator chooses

$$u_t = \big(\alpha_t, \beta_t, B_t^{\mathrm{tot}}, B_t^{\mathrm{flex}}, B_t^{\mathrm{pool}}, w_H, w_W, w_{\mathrm{rej}}, w_C, \kappa, \xi_t\big), \tag{7}$$

where (α_t, β_t) set the financial and quality sensitivities entering provider utility. These are reduced-form incentive weights: editing them changes the relative contribution of profit and health to the modeled provider objective. Total capacity B_t^{tot} and its flexible component B_t^{flex} determine the resources available for treatment. The bonus pool B_t^{pool} and tournament sharpness κ determine the strength of relative-performance rewards. Weights $(w_H, w_W, w_{\mathrm{rej}}, w_C)$ define the measured KPI, and ξ_t controls performance-based steering of flexible capacity.

The incentive and capacity components follow the economic and operational abstractions in Ma (1994); Eggleston (2005); Green (2002); Bekker et al. (2017). Audit intensity is varied as an environment lever in L2 and held fixed within each L3 search experiment. Appendix O lists the editable fields.

Provider action and response class. For candidate patient i, team j's decisions expose five response channels:

$$a_{j,t} = \big(\ \underbrace{\hat{g}_{i,t}}_{\textit{coding}},\ \underbrace{d_{ij,t}^{\mathrm{acc}}}_{\textit{selection}},\ \underbrace{d_{ij,t}^{\mathrm{def}}}_{\textit{delay}},\ \underbrace{E_{ij,t}}_{\textit{effort}},\ \underbrace{R_{ij,t}}_{\textit{triage/resource}}\ \big). \tag{8}$$

The entries record the reported group, acceptance and deferral decisions, treatment effort, and allocation of the required resource. Acceptance determines access, while deferral changes the timing of a case that remains in the queue. In L1/L2, the provider follows a response class $\Pi_P = \{\pi_P^{\phi} : \phi \in \Phi\}$ with interpretable coefficients motivated by Eq. (1). Score-based stochastic choices provide a bounded, payoff-sensitive description of local decisions (McKelvey & Palfrey, 1995). L3 expands the search to executable expressions over the same action interface. All care teams apply the shared provider program to their local state.

Discounted administrative return. For administrator program π_A, provider response $\widehat{\pi}_P$ and environment seed s, a full-horizon rollout yields

$$G(\pi_A, \widehat{\pi}_P; s) = \sum_{t=1}^{T} \gamma_d^{t-1}\, r_{A,t}, \tag{9}$$

Here γ_d is the discount factor. The pair $(\pi_A, \widehat{\pi}_P)$ induces the state trajectory and its period rewards. Appendix L.1 gives the L3 estimator using each episode's realized length T_s when execution ends before the maximum horizon.

Safety and distortion penalties. The term V in Eq. (2) aggregates penalties for waiting, high-complexity deferral, up-coding, rejection and insolvency. Applying these penalties to rollout statistics makes each of these outcomes consequential for administrative selection. The L3 weights, thresholds and seed aggregation appear in Appendix L.1; the individual channel measures are also reported throughout the experiments.

Search fitness. The empirical objective combines return, distortion penalties and variation across seeds:

$$\mathrm{Fitness}(\pi) = \mathbb{E}_s[G(\pi; s)] - \lambda_{\mathrm{unsafe}}\mathbb{E}_s[V_s(\pi)] - \lambda_{\mathrm{var}}\mathrm{Var}_s(G(\pi; s)), \tag{10}$$

Here $G(\pi; s) \equiv G(\pi_A, \widehat{\pi}_P; s)$, and $\widehat{\pi}_P = \mathcal{R}(\pi_A)$ is the provider program selected by the specified response procedure. Thus a fitness value describes a mechanism together with the response used to evaluate it.

Administrative per-step reward. L3 combines log-scaled funds with reputation:

$$r_{A,t} = 0.5 \cdot \frac{\log(F_t + 10^{-6})}{10} + 0.5 \cdot \mathrm{Rep}_t. \tag{11}$$

Reputation smooths a care-quality signal that rewards treated-patient health and penalizes rejection and waiting. The reported configuration uses

$$\begin{aligned} q_t^{\mathrm{care}} &= \bar{H}_t^{\mathrm{treated}} - 0.30\,\frac{N_t^{\mathrm{rej}}}{\max(1, N_t^{\mathrm{dec}})} - 0.05\,\bar{W}_t^{\mathrm{dec}}, \\ \mathrm{Rep}_t &= 0.70\,\mathrm{Rep}_{t-1} + 0.30\,q_t^{\mathrm{care}}, \qquad \mathrm{Rep}_0 = 0. \end{aligned} \tag{12}$$

The treated-patient mean $\bar{H}_t^{\mathrm{treated}}$ and rejection count N_t^{rej} refer to period t. The queue count N_t^{dec} and mean wait $\bar{W}_t^{\mathrm{dec}}$ are measured after arrivals are registered and before treatment decisions. An empty treated set or queue contributes zero to its corresponding mean. Reputation is updated before the period reward is evaluated. This reward supplies a fixed administrative search criterion; its financial and care components are interpreted together with the channel outcomes. L1/L2 instead prescribe the administrative settings and measure the resulting responses.

C Environment Functional Forms

The IPS loop links four components. Arrivals supply the patient population; coding assigns payment categories; treatment converts resources, effort and waiting into health and cost; and settlement feeds measured performance into the next period's incentives. The expressions below define these components and the baseline responses used in L1/L2. In L3, the host retains the environment dynamics while search edits the exposed decision expressions.

Arrival process. All reported experiments use homogeneous Poisson arrivals,

$$N_t = |\mathcal{P}_t| \sim \mathrm{Poisson}(\lambda_{\mathrm{arr}}), \tag{13}$$

with true group $g_i^\star \in \mathcal{G}$, normalized complexity $\mathrm{CMI}_i^\star \in [0, 1]$, urgency $\mathrm{Urg}_i \in [0, 1]$, and waiting tolerance drawn before provider actions. The aggregated macro-DRG distribution draws on common inpatient categories and DRG relative weights (Fetter et al., 1980; Centers for Medicare & Medicaid Services & Office of the Secretary, Department of Health and Human Services, 2025). The homogeneous Poisson model gives a tractable queueing baseline (Green, 2002; 2006). Matched policy comparisons keep arrival intensity and case mix fixed, with demand exogenous to policy and reputation. Seasonal, over-dispersed or empirical demand could be introduced through the arrival sampler; demand-responsive extensions would additionally specify how intensity or case mix changes with policy.

Coding candidate score. The host constructs an upward-only candidate set containing the true DRG and a bounded number of higher-weight alternatives. For a candidate reported group g, let $\Delta R_i(g)$ denote incremental reimbursement and $\Delta c_i(g)$ coded-complexity inflation. The baseline score is

$$s_i(g) = \alpha_t\,\gamma_{\mathrm{code}}\,\Delta R_i(g) - p_{\mathrm{audit}}(\Delta c_i(g))\,\lambda_{\mathrm{pen}}\,[\Delta R_i(g)]_+ - r_0\,\eta_{\mathrm{eth}}\,[\Delta c_i(g)]_+, \tag{14}$$

where $[x]_+ = \max\{x, 0\}$ and $p_{\mathrm{audit}}(\cdot; q_t)$ is the host's audit-hit function at intensity q_t. The three terms capture the financial gain from a higher code, its expected audit penalty, and the ethics-related cost of inflation. This trade-off is motivated by empirical coding responses and models of DRG auditing (Dafny, 2005; Silverman & Skinner, 2004; Kuhn & Siciliani, 2008).

Stochastic response mode samples from a softmax over candidate scores. Deterministic L3 evaluation selects the highest-scoring candidate, the zero-temperature version of this rule. Search can rewrite the score using host-computed features, while the admissible codes, audit schedule and settlement routine stay fixed. Across patients, coding choices depend on both the expression and the available reimbursement gains. The reported up-coding measure also reflects the composition of the period's monitored population, including patients carried in the queue (Appendix P).

Treatment production and cost. For patient i assigned to team j, health combines effort-dependent production with waiting decay:

$$H_{ij,t} = \mathrm{Skill}_j \left(1 - \exp\left[-\frac{\lambda E_{ij,t}}{\max\{c_{\min}, \mathrm{CMI}_i^\star\}}\right]\right) e^{-\delta(1-\mathrm{Tol}_i)W_{i,t}} R_{ij,t}. \tag{15}$$

Delivered effort is $E_{ij,t} \geq 0$, waiting before treatment is $W_{i,t}$, and tolerance is $\mathrm{Tol}_i \in [0,1]$. The complexity floor is $c_{\min} = 0.10$, and waiting decay uses $\delta = 0.12$. The indicator $R_{ij,t} \in \{0,1\}$ records allocation of a treatment bed. Health increases toward a skill-dependent ceiling as effort rises, while additional waiting reduces the realized output according to patient tolerance. Treatment cost is

$$C_{ij,t} = C_{\mathrm{fixed}}\, R_{ij,t} + \omega\, E_{ij,t}^{\phi}, \qquad \phi > 1. \tag{16}$$

The combination of saturating health and convex effort cost makes the marginal return to effort depend on complexity, waiting and the financial–quality weights. Zero delivered effort yields zero health under Eq. (15); Appendix L.6 discusses how searched programs use this boundary together with the settlement rule.

Bonus tournament and marginal pressure. For sharpness κ and pool B_t^{pool}, each team's KPI determines its bonus share:

$$s_{j,t} = \frac{\exp(\kappa\, \mathrm{KPI}_{j,t})}{\sum_{\ell=1}^{J} \exp(\kappa\, \mathrm{KPI}_{\ell,t})}, \tag{17}$$

$$\mathrm{Bonus}_{j,t} = B_t^{\mathrm{pool}}\, s_{j,t}, \tag{18}$$

The local marginal bonus pressure is

$$\frac{\partial \mathrm{Bonus}_{j,t}}{\partial \mathrm{KPI}_{j,t}} = B_t^{\mathrm{pool}}\, \kappa\, s_{j,t}(1 - s_{j,t}), \tag{19}$$

This term feeds back into effort, triage and KPI-targeting behavior. It depends on the bonus pool, tournament sharpness and the team's share, so a change in sharpness can alter both the scale of incentives and their distribution across teams.

C.1 Design choices and literature anchors

Table C.1 connects each component to its economic or operational motivation and records the specific abstraction used in MEDI-SIM. The cited work motivates the modeled relationships; the functional forms and parameter settings define the controlled environment evaluated here.

Table C.1: Literature anchors and modeling choices for the environment.

Component	Literature anchor	Choice made in MEDI-SIM
Roles and objectives	Payment-system and multitask agency models (Ma, 1994; Holmstrom & Milgrom, 1991; Eggleston, 2005)	One administrator and a shared provider program; reduced-form weights set financial, quality and workload incentives.
Classification and settlement	DRG case-mix classification (Fetter et al., 1980); coding responses (Dafny, 2005; Silverman & Skinner, 2004)	True complexity is separate from reported billing weight; the candidate set is upward-only to isolate inflation.
Audits	Optimal auditing of DRG upcoding (Kuhn & Siciliani, 2008)	A fixed host schedule supplies expected penalties; precise audit functions and penalty coefficients are model parameters.

Component	Literature anchor	Choice made in MEDI-SIM
Arrivals and queues	Hospital queueing and simulation (Green, 2002; 2006; Brailsford et al., 2009)	Homogeneous Poisson arrivals and fixed case mix simplify demand; comparisons share the same arrival conditions.
Capacity and pooling	Flexible ward allocations (Bekker et al., 2017)	Dedicated and flexible capacity obey hard limits; integer rounding and KPI steering are implementation choices. A binary resource indicator represents allocation for treatment.
Admission and timing	Intensive/extensive provider margins (Ellis, 1998); scheduling and access (Cayirli & Veral, 2003; Gupta & Denton, 2008)	Local scores distinguish rejection, deferral, and resource requests. Their thresholds and cost-relief terms are specified responses.
Effort, health, and cost	Cost–quality incentives (Ma, 1994; Ellis, 1998)	Saturating effort-to-health, waiting decay and convex effort cost define health in simulator units; simulated health is neither survival nor a QALY.
Workload and recovery	Workload affects service time and safety (Kc & Terwiesch, 2009)	Stamina dynamics and a quadratic overload penalty provide a stylized workload mechanism.
Performance rewards	Tournament contracts (Lazear & Rosen, 1981); imperfect measurement (Baker, 1992); healthcare pay-for-performance (Eijkenaar et al., 2013; Van Herck et al., 2010)	A linear KPI and smooth softmax tournament specify relative-performance incentives.
Bounded responses	Quantal response (McKelvey & Palfrey, 1995)	Softmax/logit scores specify noisy local choices; deterministic L3 coding uses the zero-temperature variant.
Program interface	Interoperable EHR applications and clinical decision support standards (Mandel et al., 2016; Strasberg et al., 2021)	Typed assignments separate rule logic from host services; clinical vocabulary mapping and production-system integration remain future work.

D RELATED WORK

MEDI-SIM connects treatment decisions, hospital incentives and operations, strategic responses to deployed rules, and program-space policy search. These literatures supply the clinical, economic and computational components of response-dependent mechanism evaluation.

Clinical decision-making. Patient-level healthcare AI commonly formulates treatment as a sequential decision problem using electronic health records. The AI Clinician learns vasopressor and fluid policies for sepsis from MIMIC-III (Komorowski et al., 2018); broader surveys describe applications to chronic disease management, anesthesia and ventilation (Yu et al., 2023). Evaluation guidelines emphasize off-policy assessment, distributional shift and reward specification (Gottesman et al., 2019). These formulations typically take the surrounding administrative rules as given. MEDI-SIM studies those rules directly: the administrator changes incentives, capacity and performance measurement, and a shared provider program determines the response of care teams. Patient-level treatment and mechanism-level design therefore address complementary parts of the care-delivery process.

Healthcare incentives and operations. Agency models connect payment to creaming, skimping, dumping and treatment intensity (Ellis, 1998; Ma, 1994; Eggleston, 2005). Empirical work on DRG payment documents coding responses to prices and differences associated with hospital ownership (Dafny, 2005; Silverman & Skinner, 2004). Medicare Advantage studies likewise distinguish reported risk from underlying clinical need through differences in coding intensity (Kronick & Welch, 2014; Geruso & Layton, 2020). These findings motivate the separation of true patient complexity from the group used for reimbursement.

Operational models supply the second part of the interaction. Queueing and scheduling analyses connect capacity and utilization to patient waiting (Green, 2002; 2006; Cayirli & Veral, 2003; Gupta &

Denton, 2008), while flexible-bed models study allocation across hospital wards (Bekker et al., 2017). Healthcare simulation integrates such patient-flow processes through several modeling approaches (Brailsford et al., 2009). MEDI-SIM couples these flows to incentive-sensitive coding, triage and effort. A provider decision changes both its immediate payoff and the queues and resources facing later decisions.

Measurement, auditing and strategic gaming. DRG audit models explain how monitoring and penalties affect the return to coding inflation (Kuhn & Siciliani, 2008). Performance measurement creates a related incentive problem when a rewarded score combines imperfect signals of the underlying objective. Multitask models, incentive-contract theory and evidence on healthcare targets describe how providers can redirect activity toward the measured outcome (Holmstrom & Milgrom, 1991; Baker, 1992; Bevan & Hood, 2006; Propper et al., 2010). Reviews of healthcare pay-for-performance emphasize heterogeneous effects and the importance of contract design and local context (Eijkenaar et al., 2013; Van Herck et al., 2010). Goodhart-style taxonomies further distinguish routes through which proxy optimization and the intended objective can diverge (Manheim & Garrabrant, 2018).

The coding and measurement wedges in MEDI-SIM place these relationships inside the same operational loop. Coding changes payment categories, triage changes access and timing, and effort changes treatment output. Their separate measurements allow an intervention's targeted effect to be read alongside its consequences elsewhere in the system.

Strategic learning and mechanism design. Strategic classification studies deployed rules whose subjects can alter their response (Hardt et al., 2016), including learning from revealed preferences (Dong et al., 2018) and differentiable training formulations (Levanon & Rosenfeld, 2021). Performative prediction studies the distributional changes induced by deployment (Perdomo et al., 2020). Reward-gaming theory analyzes the relationship between proxy rewards and target objectives over specified policy classes (Skalse et al., 2022). Automated mechanism design treats the rule itself as a search object (Sandholm, 2003), and differentiable economics learns allocation and payment mechanisms using neural architectures (Dütting et al., 2024).

MEDI-SIM adopts this response-dependent view while representing both sides with typed executable programs. The representation exposes the conditions, coefficients and feature dependencies behind individual decisions. It also supports a common sequence of evaluations: joint design under the administrative objective, unilateral revision under each role's utility, and response to a distribution of historical opponents.

Multi-agent simulation and program-space search. The AI Economist uses two-level reinforcement learning to study co-adaptation between a tax-setting planner and economic agents (Zheng et al., 2022). Generative agents (Park et al., 2023), Concordia (Vezhnevets et al., 2023) and LLM-based economic simulacra (Karten et al., 2025) extend agent-based modeling to social and economic interactions. MEDI-SIM uses a hospital's constrained care-delivery process as the environment and places the LLM in the outer search loop: executable policies act during each episode, and the editor revises their logic between evaluations.

This search procedure builds on LLM-guided program evolution. FunSearch combines code generation and evaluation for mathematical constructions and heuristics (Romera-Paredes et al., 2024); Evolution through Large Models uses language-model mutation in code-defined domains (Lehman et al., 2024); AlphaEvolve extends this approach to scientific and algorithmic discovery (Novikov et al., 2025), and OpenEvolve provides an open-source implementation of this design (Sharma, 2025); and Eureka applies code generation to reward design (Ma et al., 2024). LLMs have also been studied as black-box optimizers (Yang et al., 2024). For strategic evaluation, PSRO and JPSRO retain policies and update the opponent distributions faced by response oracles (Lanctot et al., 2017; Marris et al., 2021), while CSRO generates these responses in code space (Hennes et al., 2026). Our implementation combines evolutionary program search with these population-based evaluations, using hospital rollouts to assess utilities, access, treatment and coding for each policy pair.

E L1 RESULTS BY RESPONSE CHANNEL

L1 varies (α, β) over an 11×11 grid while keeping the native provider response rules fixed. The four regimes in Table E.1 show why the resulting outcomes need to be read together. Profit-driven incentives combine relatively low rejection of easy cases with coding inflation and selective rejection of complex cases. Quality-driven incentives produce higher effort and lower terminal funds. The balanced regime reduces coding and rejection relative to the profit-driven regime, while its timing and team-level results reveal additional disparities. We follow this comparison from access to deferral, then to performance measurement and treatment costs.

Table E.1: Representative L1 incentive regimes (30-seed means, maximum horizon $T = 200$). Rejection and up-coding are rates; funds are in simulator units. Zero profit and quality weights leave bonus and workload incentives in place. Δ rej. is high-CMI minus low-CMI rejection, with entries and gaps rounded separately.

Regime	α	β	Δ rej.	High rej.	Low rej.	Upcode	Effort	Funds
Zero α, β	0.0	0.0	0.241	0.542	0.301	0.014	2.128	23.4
Profit-driven	0.8	0.2	0.182	0.182	0.000	0.226	1.688	5451.7
Quality-driven	0.2	0.8	-0.051	0.026	0.078	0.027	2.435	15.2
Balanced	0.5	0.5	0.155	0.157	0.002	0.080	1.790	3486.6

E.1 BROAD REJECTION AND COMPLEXITY-SELECTIVE ACCESS

The first comparison distinguishes the level of rejection from its distribution across patient complexity. At $(\alpha, \beta) = (0, 0)$, high- and low-CMI rejection rates are 0.542 and 0.301. This setting removes the profit and health terms from provider utility while retaining the bonus and workload terms; the L1 bonus pool remains 5.0 with sharpness $\kappa = 2.0$ (Appendix H). Its effort level of 2.128 should therefore be read in the context of those remaining incentives.

At the profit-driven setting $(0.8, 0.2)$, rejection falls in both groups, to 0.182 for high-CMI patients and approximately zero for low-CMI patients. Access improves relative to the zero-profit/quality setting, but the remaining rejection is concentrated among complex cases. The quality-driven setting $(0.2, 0.8)$ changes this distribution: high-CMI rejection is 0.026 and low-CMI rejection is 0.078, giving a separately rounded gap of -0.051. These group-specific rates explain more than the gap alone, because its sign and magnitude depend on both components.

Across the full grid, the rejection gap reaches 0.31 at $(0.1, 0.0)$ and -0.20 at $(0.0, 0.5)$. Up-coding increases with α at each fixed β, spanning approximately 0.01–0.38. Figure E.1 extends the access comparison to deferral and waiting. Their different spatial patterns motivate a separate examination of treatment timing.

E.2 LOWER REJECTION CAN COEXIST WITH UNEQUAL DEFERRAL

Relative to the profit-driven regime, the balanced setting reduces up-coding from 0.226 to 0.080 and the rejection gap from 0.182 to 0.155. Yet its high-CMI strategic-deferral rate is 0.290, compared with 0.010 for low-CMI patients. These outcomes locate an additional access burden in the timing of treatment.

Here *strategic delay* has an operational definition: the provider defers a patient when its local bed-availability estimate is positive. The denominator is all considered patients in the relevant group. This diagnostic records the conjunction of a defer decision and an available-capacity signal. The native response rule includes a cost-relief term that can make such deferral attractive, providing a mechanism consistent with the observed pattern.

Figure E.2 separates total deferral from this subset. The overall deferral gap reaches 0.316 at $(0.6, 0.7)$; the strategic-delay gap has a maximum of about 0.29 at the same setting. At low α, the strategic gap can reverse, reaching -0.33 at $(0.0, 0.6)$. Thus different incentive regions place the timing burden on different complexity groups. The locations also differ from the rejection extrema,

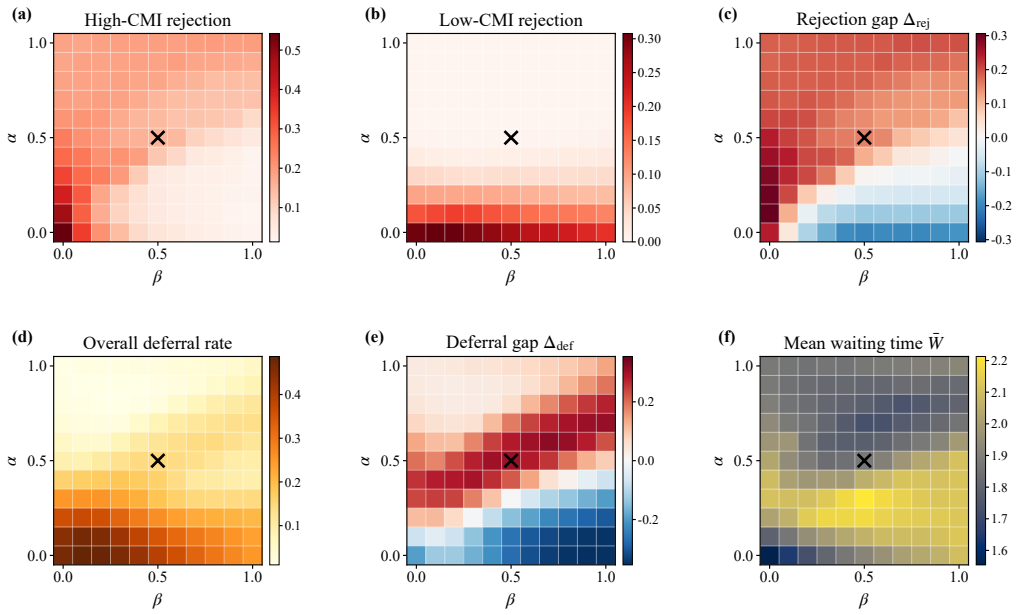


Figure E.1: Access decomposition on the L1 grid. Panels (a) and (b) split rejection by true complexity into high-CMI and low-CMI components, panel (c) is the resulting cherry-picking gap, panel (d) reports overall deferral, panel (e) reports the high-minus-low CMI deferral gap, and panel (f) reports mean waiting time. The components distinguish broad access rationing from case-mix selection and unequal deferral across incentive regimes.

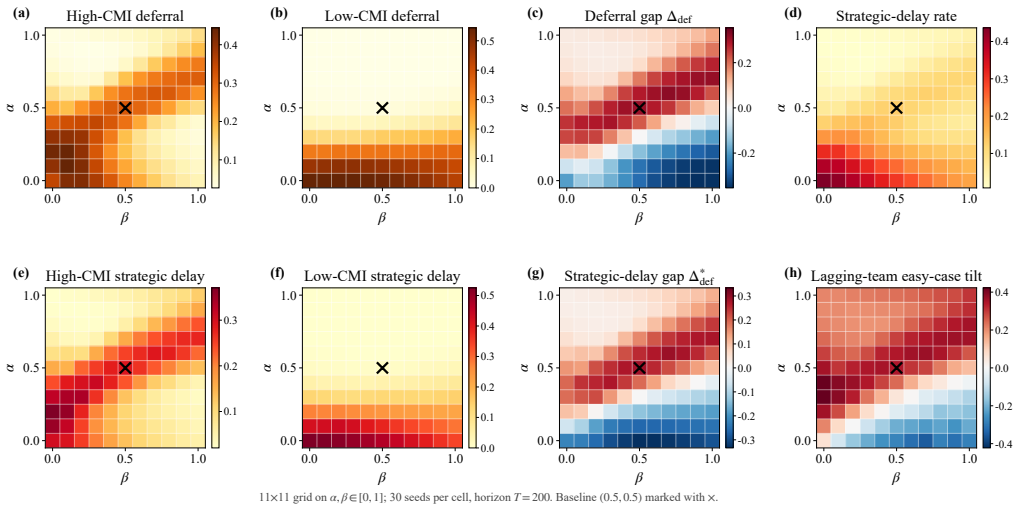


Figure E.2: Deferral and strategic-delay decomposition on the L1 grid. Strategic delay is a deferral made with a positive local bed-availability estimate; rates use considered patients as the denominator. Panels (a) and (b) report deferral rates separately for high-CMI and low-CMI patients; panel (c) is the deferral gap. Panel (d) reports the overall strategic-delay rate, panels (e)–(f) split it by true CMI, and panel (g) reports their gap; panel (h) is the easy-case CMI tilt of KPI-lagging teams.

while mean waiting peaks near $(0.3, 0.5)$. Together, these patterns show how similar rejection outcomes can conceal different queueing and deferral behavior.

Panel (h) adds the easy-case acceptance tilt of KPI-lagging teams. Its role is examined next, linking the patient-level access patterns to the performance incentives facing each team.

E.3 KPI–HEALTH ALIGNMENT AND CASE SELECTION

The balanced regime combines its deferral disparity with a KPI–health correlation of -0.659 and an easy-case tilt of 0.341 among lagging teams. These quantities describe two related aspects of the response: which patients teams accept, and how their measured performance co-varies with treated-patient health.

KPI combines health with waiting, rejection and cost penalties (Eq. (3)). A negative correlation associates higher measured scores with lower team mean health in the pooled team-period observations. The other KPI components and the selected case mix contribute to this association. We interpret the correlation together with the acceptance diagnostics and the specified KPI-sensitive response rule.

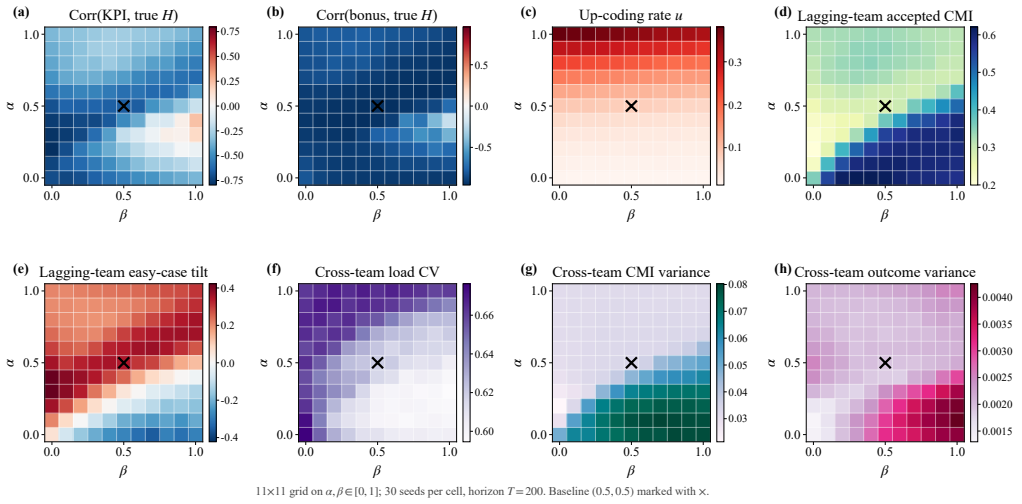


Figure E.3: Team-level outcomes and KPI alignment on the L1 grid. Panels (a) and (b) report Pearson correlations of KPI and bonus against team mean health, pooled over team-periods with considered patients within each rollout and then averaged across seeds; team mean health is zero when no patient is treated. Panels (c)–(e) report the up-coding rate, mean accepted CMI by lagging teams, and their easy-case acceptance tilt. Panel (f) reports the cross-team coefficient of variation of considered load; panels (g)–(h) report variances of mean treated CMI and health among treating teams. These period-level dispersion statistics are averaged within a rollout and then across seeds.

Figure E.3(a) reports within-rollout Pearson correlations pooled over team-periods with considered patients, then averaged across seeds. The correlation ranges from -0.80 at $(0.2, 0.0)$ to $+0.24$ at $(0.4, 1.0)$, with positive values concentrated at high β. Team-periods with considered patients and an empty treated set contribute zero mean health, so the statistic also reflects variation in whether treatment occurs.

Panel (b) follows the reward that teams actually receive. Bonus–health correlation remains negative across the grid, ranging from approximately -0.96 to -0.22. The softmax tournament normalizes each team's KPI against its contemporaneous peers. This period-specific, nonlinear transformation gives bonus–health association a different pattern from pooled KPI–health association.

Panels (d)–(e) describe selection by teams below their peers' previous-period mean KPI. Their mean accepted CMI spans about 0.20–0.62, and their low-minus-high-CMI acceptance gap spans approximately -0.35 to $+0.42$. Positive tilt means higher acceptance of easy cases within this group. The native triage rule contains a bonus-pressure-dependent easy-case preference for lagging teams; these panels show its expression within the full response system. Appendix F then varies the bonus pool to examine how the salience of the measured score changes these outcomes.

The remaining panels describe variation across teams. Considered-load dispersion, treated-CMI variance and health variance capture different aspects of the allocation. Treated-CMI variance peaks near $(0.1, 0.7)$ and health variance near $(0.1, 1.0)$, whereas easy-case tilt is strongest near $(0.3, 0.0)$. These locations distinguish selective acceptance within lagging teams from differences among the teams delivering treatment.

E.4 Effort, treated-patient health and financial outcomes

The access and timing results also shape the population on which treatment outcomes are measured. Across the L1 grid, mean treated-patient health remains within $[0.95, 1.02]$, while mean effort spans 1.56–2.74 and mean treatment cost spans 8.23–16.41. Terminal funds range from 0.4 to 6490 simulator units. Health therefore varies little while resource use and financial outcomes change substantially.

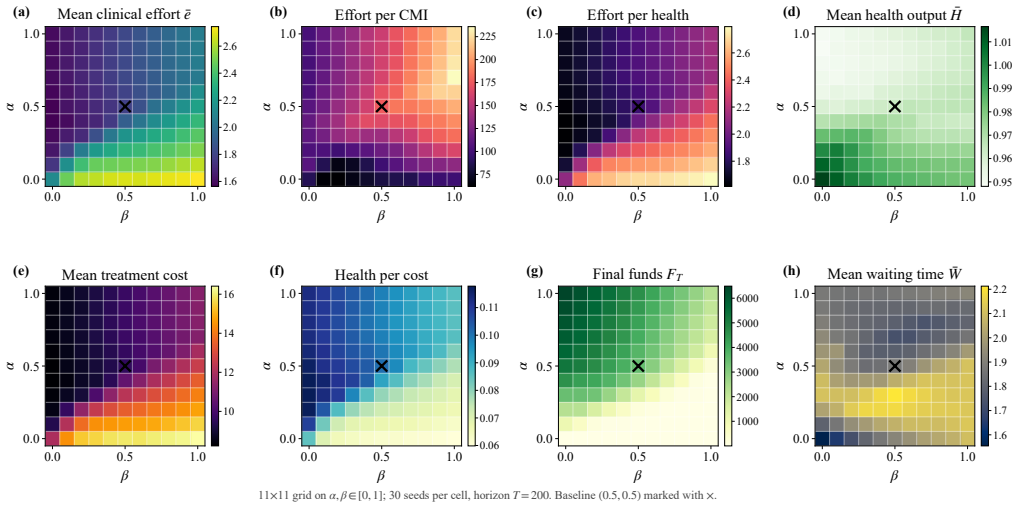


Figure E.4: Clinical and budgetary outcomes on the L1 grid. Top-row panels (a)–(d) report effort, effort per true CMI, effort per realized health and mean health. Bottom-row panels (e)–(h) report treatment cost, health per cost, terminal funds F_T and waiting time. Mean health spans 0.95–1.02, while terminal funds span 0.4–6490 simulator units.

The production function helps interpret this comparison. Additional effort has diminishing health returns, whereas its cost grows convexly. As β rises, health per cost falls from about 0.12 to 0.06. The representative quality-driven regime illustrates the same trade-off: effort is 2.435 and terminal funds are 15.2, compared with 1.688 and 5451.7 in the profit-driven regime. These are joint outcomes of the specified production function, accepted case mix and waiting experienced before treatment.

The effort-per-CMI and effort-per-health diagnostics in Figure E.4(b)–(c) relate treatment input to complexity and realized output. Both move with effort and the composition of treated patients. Waiting adds a further link: its maximum of about 2.21 near $(0.3, 0.5)$ occurs in the moderate-incentive region, where delay can reduce the health obtained from a given treatment input.

Taken together, L1 describes a structured response landscape. Rejection identifies who exits without treatment, deferral identifies differences in treatment timing, team-level diagnostics connect selection to performance incentives, and the production outcomes reveal the associated resource and financial trade-offs. This joint view explains why the balanced regime's lower coding and rejection rates coexist with unresolved timing and measurement disparities.

F L2 Additional Lever Experiments

L2 varies one administrative lever at a time within the balanced, profit-driven and quality-driven regimes. The comparisons distinguish changes to a targeted response channel from changes to the resources and performance incentives governing the rest of care delivery. Appendix H gives the settings shared by these sweeps.

Audits and the distribution of access. The main audit comparison links billing to selection. In the balanced regime, raising q from 0 to 0.5 lowers up-coding from 0.636 to 0.001, while the high-minus-low-CMI rejection gap rises from 0.100 to 0.233. Table F.1 places these outcomes together. Figure 3(a) traces the coding response across the sweep; the rejection comparison shows how the same intervention changes access by complexity.

Table F.1: Joint outcome summaries from the balanced-regime L2 sweeps. Values are 30-seed endpoint means, with maximum horizon $T = 200$. Rejection gaps are high-minus-low-CMI rates; the correlation is dimensionless. Each pair of rows refers to the same lever comparison.

Lever comparison	Outcome	Lower endpoint	Upper endpoint
Audit $q : 0 \to 0.5$	Up-coding rate	0.636	0.001
	Rejection gap	0.100	0.233
Bonus $B^{\text{pool}} : 0 \to 15$	KPI–health correlation	-0.447	-0.839
	Rejection gap	0.066	0.479

Bonus size and score alignment. The bonus-pool sweep changes how strongly the measured KPI affects provider rewards. Between $B^{\text{pool}} = 0$ and 15, balanced-regime KPI–health correlation becomes more negative, from -0.447 to -0.839, and the rejection gap widens from 0.066 to 0.479. The combination links weaker health alignment to a change in patient selection. Figure F.1 shows the accompanying budget and correlation curves: terminal funds are lower at the larger-pool endpoint in all three regimes. Intermediate values include reversals in quality-driven funds and in the correlation curves, so the endpoint contrast summarizes the overall change across the tested range.

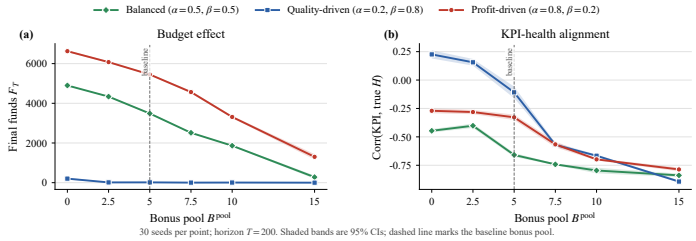


Figure F.1: L2 bonus-pool ablation. Points are 30-seed means over horizon $T = 200$; shaded bands are pointwise 95% confidence intervals (Appendix H) and the dashed line marks the baseline pool. The panels show terminal funds and KPI–health correlation; Table F.1 reports the accompanying balanced-regime rejection comparison.

Capacity and KPI composition. Increasing total capacity from 6 to 16 lowers endpoint mean waiting in all three regimes: from 2.313 to 1.735 in the profit-driven regime, 2.325 to 1.896 in the quality-driven regime, and 2.303 to 1.720 in the balanced regime. The profit-driven curve has a small reversal between capacities 14 and 16. Raising the KPI health-to-cost weight ratio increases effort and lowers terminal funds across the tested values, consistent with the treatment-cost trade-off described in Appendix E.4.

Tournament sharpness. Rejection gaps vary non-monotonically with bonus sharpness. Equation (19) provides a useful interpretation: local pressure is $B^{\text{pool}}\kappa s_j(1 - s_j)$, and changing κ also changes the shares s_j. Sharpening the tournament therefore redistributes incentives across teams as well as changing their scale. The observed curve combines this effect with the resulting triage and queue feedback. The flexible-capacity comparison in Appendix G examines a second use of KPI, as an allocation signal.

G L2 Flexible Capacity: Steering-on and Steering-off Comparisons

The flexible-pool experiment holds total capacity fixed and varies the share allocated across teams. This separates adding resources to the hospital from reallocating its existing resources. The main L2

sweep uses previous-period KPI to steer flexible capacity. Here we repeat the sweep with $\xi = 0$ and `kpi_steering_mode=none`, so the flexible subpool follows static base-capacity shares.

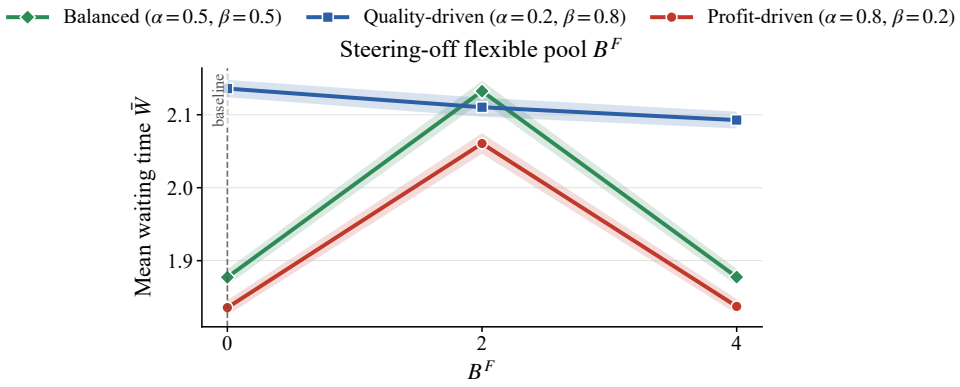


Figure G.1: L2 steering-off flexible-pool comparison. Mean waiting (30 seeds, $T = 200$) as a function of the flexible-pool size B^{flex}, holding routing, team specialization, and total capacity fixed while disabling KPI capacity steering. Shading shows pointwise 95% confidence intervals (Appendix H); the dashed line marks the L2 baseline.

Figure G.1 reports the steering-off comparison at $B^{\text{flex}} \in \{0, 2, 4\}$. Balanced-regime waiting is approximately 1.88 at both endpoints and 2.13 at the midpoint. Profit-driven waiting follows a similar pattern, with endpoints near 1.84 and a midpoint of 2.06. Quality-driven waiting decreases from 2.14 to 2.09 across the sweep.

The intermediate peak occurs with discrete capacity allocation. Dedicated and flexible beds are allocated separately using integer rounding, so changing the split can change the number assigned to a team even when allocation shares stay fixed. Turning on KPI steering adds another source of reallocation, because flexible shares then respond to the previous period's measured performance.

The balanced-regime endpoint comparison makes the distinction clear. With KPI steering, waiting rises from 1.88 to 2.23 as the flexible pool grows from zero to four. With static shares, those endpoints are nearly equal. The intermediate steering-off peak shows the additional influence of discrete allocation. The value of flexible capacity in these runs thus depends jointly on pool size, rounding and the information used to distribute beds.

H L1/L2 Experimental Setup

L1 maps the response to profit and quality weights, while L2 changes one administrative lever at a time around three of those regimes. Table H.1 records the settings needed to interpret both layers, including the distinction between KPI capacity steering and active hospital routing. Both layers use 30 common environment seeds and a maximum horizon of $T = 200$. Channel statistics are computed within each rollout and then averaged across seeds.

L2 confidence intervals. At each fixed regime and lever value, let x_s be the rollout-level metric for environment seed s, with $n = 30$. Shaded bands in Figures 3, F.1 and G.1 are pointwise 95% normal-approximation confidence intervals:

$$\bar{x} \pm 1.96 \frac{s_x}{\sqrt{n}}, \qquad \bar{x} = \frac{1}{n}\sum_{s=1}^{n} x_s, \qquad s_x^2 = \frac{1}{n-1}\sum_{s=1}^{n}(x_s - \bar{x})^2.$$

Each interval describes uncertainty in the mean for one fixed setting. A comparison between settings has a different target: the mean paired change across their shared seeds. Appendix L.8 specifies the paired Student t intervals used for the L3 response checks over 20 held-out environments.

Table H.1: L1/L2 experimental setup and hyperparameters. B^{tot} is total capacity, B^{flex} is the flexible-capacity pool, and B^{pool} is the KPI bonus pool.

Layer	Scope	Values	Notes
L1/L2	Shared rollout protocol	Horizon $T = 200$; 30 common environment seeds; reported values are seed means unless a confidence interval is shown.	Both layers use prescribed administrative settings and the native provider-response rules of §2.
L1	Incentive phase diagram	$\alpha, \beta \in \{0.0, 0.1, \ldots, 1.0\}$, giving an 11×11 grid.	The sweep varies only provider financial sensitivity α and quality sensitivity β; all other administrative settings stay fixed.
L1	Default administrative and environment settings	$B^{\text{tot}} = 10$, $B^{\text{flex}} = 0$, $B^{\text{pool}} = 5.0$, $\kappa = 2.0$, $(w_H, w_W, w_{\text{rej}}, w_C) = (1.0, 0.5, 0.5, 0.3)$, $\xi = 0$.	Strategic routing is disabled; `kpi_steering_mode=none`; probabilistic up-coding, KPI-gaming tilt, and KPI-weighted effort feedback are enabled.
L2	Representative regimes	Profit-driven $(\alpha, \beta) = (0.8, 0.2)$; quality-driven $(0.2, 0.8)$; balanced $(0.5, 0.5)$.	Each L2 axis is swept one at a time around these three L1 anchor regimes.
L2	Main one-at-a-time lever sweeps	$q \in \{0, 0.05, 0.10, 0.20, 0.35, 0.50\}$; $B^{\text{pool}} \in \{0, 2.5, 5, 7.5, 10, 15\}$; $\kappa \in \{0, 1, 2, 3, 4, 5\}$; $B^{\text{tot}} \in \{6, 8, 10, 12, 14, 16\}$; $w_H/w_C \in \{0.5, 1, 2, 3.333, 4, 5\}$; $B^{\text{flex}} \in \{0, 2, 4\}$.	The audit sweep sets `audit_base_prob`$= q$ and `audit_slope`$= 0$; the w_H/w_C sweep holds $w_H + w_C$ fixed at the L1 baseline.
L2	Main baseline and steering settings	$B^{\text{tot}} = 10$, $B^{\text{flex}} = 0$, $B^{\text{pool}} = 5.0$, $\kappa = 2.0$, $(w_H, w_W, w_{\text{rej}}, w_C) = (1.0, 0.5, 0.5, 0.3)$, audit baseline $q = 0.10$, $\xi = 1.0$.	Strategic routing is disabled; `kpi_steering_mode=capacity_only`; probabilistic up-coding, KPI-targeting tilt, and KPI-weighted effort feedback are enabled.
L2	Steering-off flexible-capacity comparison	$B^{\text{flex}} \in \{0, 2, 4\}$ under the same three regimes, with $B^{\text{tot}} = 10$, $\xi = 0$, and `kpi_steering_mode=none`.	The static-share comparison exposes the contributions of the allocation rule and integer rounding at fixed total capacity.

I Literature Consistency and Qualitative Checks

Table I.1 connects the literature anchors to the interventions and observations in Medi-Sim. The first role of these comparisons is to check that the specified response rules reproduce the directions motivating their design. Their second role is to identify coupled consequences, such as a wider rejection gap during audit enforcement or increased waiting under KPI-steered allocation. The final column states which of these roles each comparison serves. Effect sizes refer to the reported simulator configurations. These comparisons support interpretation of the model without providing independent external validation or quantitative calibration.

Table I.1: Literature anchors, simulator interventions and observed responses. The final column distinguishes intended-direction checks, partial observables and findings from coupled mechanisms. Directional matches are checks under specified simulator rules; they do not independently validate the joint response or calibrate effect sizes to a real hospital.

Mechanism or hypothesis	Anchor	Simulation intervention	Expected qualitative signature	Medi-Sim observed signature	Interpretation
Diagnosis-based payment creates coding rent	Dafny (2005); Kronick & Welch (2014)	Increase profit sensitivity or payment spread; high-α/low-β region	Reported complexity or up-coding rises while true patient attributes remain fixed	Profit-driven representative regime: up-coding $= 0.226$; funds $= 5451.7$	Intended-direction check for the coding-rent mechanism under the specified response rules.
Risk-adjusted reimbursement raises reported risk or coded complexity	Kronick & Welch (2014); Geruso & Layton (2020)	Strengthen reimbursement value of the coded group while the coding wedge is active	Coded risk rises relative to true patient complexity	High-profit cells show elevated up-coding; the reported observable is the share assigned a higher-weight group	Partial observable: the rate records the occurrence of up-coding; coded-CMI inflation would also describe its magnitude.
Profit or ownership-like incentives shift case mix	Silverman & Skinner (2004); Ellis (1998); Ma (1994)	Move toward high α at low β; compare profit-driven cells with quality and balanced cells	More profitable or easier patients are favored; high-cost/high-CMI patients face worse access	Profit-driven representative regime: high-CMI rejection gap $= 0.182$ alongside elevated coding and funds	Intended-direction check for provider triage; active hospital routing is disabled in L1/L2.

Mechanism or hypothesis	Anchor	Simulation intervention	Expected qualitative signature	MEDI-SIM observed signature	Interpretation
Audit suppresses coding but can shift pressure elsewhere	Kuhn & Siciliani (2008)	Increase audit probability q from 0 to 0.5	Up-coding falls; selection or delay pressure may rise when the billing channel closes	Profit-driven up-coding $0.851 \rightarrow 0.003$; balanced up-coding $0.636 \rightarrow 0.001$; balanced cherry-picking $0.100 \rightarrow 0.233$	Audit suppression checks the intended coding response. The simultaneous selection change is a coupled-simulator finding.
Measured performance targets induce gaming	Bevan & Hood (2006); Propper et al. (2010); Campbell (1979); Manheim & Garrabrant (2018)	Increase KPI salience through balanced-interior incentives, bonus pressure, or KPI steering	Measured score becomes behaviorally salient while true-health alignment can weaken	Balanced interior: lagging teams tilt accepted case mix toward easier cases (tilt $= 0.341$); $\mathrm{corr}(\mathrm{KPI}, \mathrm{health}) = -0.659$	The accepted-case tilt and health correlation describe selection and score alignment under the specified KPI response.
Stronger pay-for-performance can worsen true alignment when the proxy is misaligned	Holmstrom & Milgrom (1991); Baker (1992); Eijkenaar et al. (2013); Van Herck et al. (2010)	Increase the bonus pool B^{pool}	The measured proxy becomes more consequential; KPI–health alignment may worsen	Endpoint comparison: balanced $\mathrm{corr}(\mathrm{KPI}, \mathrm{health})$ is -0.447 at $B^{\mathrm{pool}} = 0$ and -0.839 at $B^{\mathrm{pool}} = 15$; the smallest positive pool gives a small non-monotone uptick	A configuration-specific score-design outcome, interpreted in the context of heterogeneous pay-for-performance effects.
Quality-oriented incentives suppress coding/selection but raise effort and budget pressure	Ma (1994); Ellis (1998); Eggleston (2005)	Low α, high β regime	Lower up-coding and less cream-skimming; higher effort and weaker solvency	Quality-driven regime: up-coding $= 0.027$; rejection gap $= -0.051$; effort $= 2.435$; funds $= 15.2$	A joint access, effort and financial trade-off under the specified treatment-production function.
Total capacity expansion reduces waiting under queueing pressure	Green (2002; 2006)	Increase total capacity from 6 to 16	Mean waiting should fall across regimes	Mean waiting falls in all three L2 regimes: profit $2.313 \rightarrow 1.735$, quality $2.325 \rightarrow 1.896$, balanced $2.303 \rightarrow 1.720$	Intended-direction check for the queueing response to total capacity.
Flexible-capacity outcomes depend on the allocation rule	Bekker et al. (2017) and hospital-flow allocation logic	Compare KPI-steered flexible pool with steering-off flexible pool	Changing the allocation signal changes how a flexible pool affects waiting	With KPI steering, balanced waiting rises $1.88 \rightarrow 2.23$ as B^{flex} grows $0 \rightarrow 4$. With steering off, balanced endpoints both round to 1.88 (midpoint: 2.13); profit-driven endpoints both round to 1.84, while quality falls $2.14 \rightarrow 2.09$	Coupled allocation finding at fixed total capacity; integer rounding and KPI steering both affect team-level resources.

J RESPONSE-FORM SENSITIVITY

The sensitivity experiments examine two properties of the native response family: the ordering of outcomes across incentive settings, and the change in coding and delay during an audit intervention. Both comparisons use ten seeds per condition and the native compressed-DRG configuration with active hospital routing. The main L1/L2 experiments use thirty seeds and patient-choice registration with active routing disabled. Each supplementary comparison is therefore evaluated against its own matched baseline.

Alternative response forms. We vary two aspects of the response rules while keeping the action channels and settlement mechanism fixed. Quantal-response smoothing replaces routing argmax choices with a softmax and discrete triage and bed-request thresholds with binary logits, using temperatures $\tau \in \{0.1, 0.3\}$. This introduces noisy payoff-sensitive choice (McKelvey & Palfrey, 1995); as $\tau \rightarrow 0$, decisions approach the baseline away from threshold ties. Saturating responses replace a linear incentive contribution z with $s \tanh(z/s)$. The transformation preserves the slope at zero and bounds the contribution's magnitude. We use $s \in \{1, 0.5\}$ with the corresponding unit scaling for monetary channels; the linear response is recovered as $s \rightarrow \infty$.

Each alternative and its baseline are evaluated on the 5×5 grid $\alpha, \beta \in \{0, 0.25, 0.5, 0.75, 1\}$, using the same ten environment seeds and $T = 200$. Table J.1 reports Spearman correlations between the 25 cell means. This statistic measures how similarly the two response forms order the incentive settings for a given outcome.

Table J.1: Response-form sensitivity: rank correlation with the matched baseline across the incentive grid. Rejection gap is high-minus-low CMI rejection; delay is the strategic-delay rate.

Response form	Coding	Rej. gap	Delay	Effort	Wait	KPI–H
Quantal, $\tau = 0.1$	0.985	0.988	0.995	0.982	0.970	0.957
Quantal, $\tau = 0.3$	0.981	0.981	0.933	0.986	0.572	0.729
Saturating, $s = 1$	0.992	0.938	0.815	0.990	0.761	0.888
Saturating, $s = 0.5$	0.986	0.784	0.265	0.969	-0.335	0.848

Coding and effort retain rank correlations above 0.96 in all four comparisons. At both quantal temperatures, coding, rejection gap, strategic delay and effort each remain above 0.93. Waiting is more sensitive: its correlation falls to 0.572 at $\tau = 0.3$, even as the direct behavioral measures retain their ordering. Saturation at $s = 1$ preserves correlations above 0.81 for the four direct measures, while $s = 0.5$ changes delay and waiting more substantially. The latter correlations are 0.265 and -0.335. The sensitivity is thus concentrated in how local responses combine into timing and congestion, while coding and effort orderings remain comparatively persistent.

Audit-induced coding and delay changes. We next sweep $q \in \{0, 0.05, 0.1, 0.2, 0.35, 0.5\}$ under baseline, quantal ($\tau = 0.1$) and saturating ($s = 0.5$) responses in each of the three representative regimes. These runs share a separate group of ten environment seeds, with $T = 200$, total capacity 10, zero flexible capacity and capacity-only KPI steering.

Table J.2: Audit increase from $q = 0$ to $q = 0.2$. Ranges span the three incentive regimes within each response form. Coding reductions are relative percentages; delay changes are percentage points.

Response form	Coding reduction (%)	Strategic-delay increase (pp)
Baseline	71.9–93.0	0.79–1.75
Quantal, $\tau = 0.1$	72.3–92.9	1.12–1.50
Saturating, $s = 0.5$	70.9–93.9	0.79–1.55

Across the nine response-form/regime combinations, increasing q from 0 to 0.2 reduces up-coding by 70.9–93.9% and increases strategic delay by 0.79–1.75 percentage points. This coding–delay pattern persists under the stronger saturation that changes the grid-wide timing order. The comparison complements the main L2 coding–rejection result: it traces a timing consequence of audit enforcement within the supplementary active-routing configuration.

K Severity-Weighted Health-Score Extension

A score that gives more weight to health delivered to complex patients could change both treatment incentives and the case mix contributing to the score. We examine this interaction through an optional severity term, evaluated in a supplementary configuration separate from the main experiments.

The extension introduces coefficient w_{sev} and team-period observable

$$S_{j,t} = \frac{1}{|\mathcal{T}_{j,t}|} \sum_{i \in \mathcal{T}_{j,t}} \text{CMI}_i^* H_{i,j,t}, \qquad \text{KPI}'_{j,t} = \text{KPI}_{j,t} + w_{\text{sev}} S_{j,t},$$

where $\mathcal{T}_{j,t}$ is the treated set and $S_{j,t} = 0$ for an empty set. True patient complexity supplies the severity weight. Setting $w_{\text{sev}} = 0$ recovers the unextended KPI within the same configuration. The term measures single-period simulated health weighted by complexity; the L3 program interface retains the original KPI fields.

Experimental setup. We vary $w_{\text{sev}} \in \{0, 0.5, 1, 2\}$ in the profit-driven, balanced and quality-driven regimes. Each setting uses $T = 200$ and ten common environment seeds, shared with the supplementary audit sweeps. The configuration uses native linear responses, active routing, $\kappa = 1$, ten beds and zero flexible capacity. KPI gaming and KPI-weighted effort feedback are disabled. The resulting comparison measures the response of this supplementary configuration to the added score term.

Table K.1: Effects of increasing the severity weight ($w_{\text{sev}} : 0 \to 2$). Entries are mean changes over ten paired runs, in percentage points. Rejection, deferral and strategic-delay gaps are high-minus-low true-CMI rates; easy-case tilt is the low-minus-high acceptance-rate gap among KPI-lagging teams.

Regime	Rejection gap	Easy-case tilt	Deferral gap	Strategic delay gap	High-CMI rejection
Profit-driven	+1.23	+7.11	+6.27	+4.71	+1.23
Balanced	+2.63	-4.19	-6.10	-5.50	+2.69
Quality-driven	-0.45	-0.04	+0.05	-0.27	+0.21

Access and timing respond differently across regimes. In the profit-driven regime, raising w_{sev} from 0 to 2 increases the easy-case tilt among lagging teams by 7.11 percentage points and the strategic-delay gap by 4.71 points. Both increases occur in all ten paired runs. The rejection gap also increases by 1.23 points. Thus the greater score weight on complex-patient health coexists with more selective access and timing under these incentives.

The balanced regime illustrates a different trade-off. Easy-case tilt falls by 4.19 points, the deferral gap by 6.10 points and the strategic-delay gap by 5.50 points, while high-CMI rejection rises by 2.69 points. An improvement in the timing gaps therefore accompanies increased rejection of complex patients. In the quality-driven regime, high-CMI rejection rises by 0.21 points and low-CMI rejection by 0.66 points; the rejection gap consequently becomes more negative, with a reported mean change of -0.45 points.

These outcomes follow the two components of the added score: realized health and the complexity of patients who receive treatment. Changing its weight alters the measured value of that treated population, while the complete response system also changes admission and deferral. The comparison therefore evaluates score design through group-specific access and timing outcomes together.

L L3: Joint Design, Role Responses and Empirical Games

L3 follows a policy pair through three stages: joint design under the administrative objective, unilateral adaptation under each role's own objective, and responses to retained policy histories. The protocol below keeps track of the program being edited, the opponent it faces and the data used for selection. We then relate changes in utility to the selected code and to treatment, coding and waiting outcomes.

Search and assessment protocol. Training uses three environments and selection uses five validation environments, with a maximum horizon of $T = 200$. After selection, programs and joint weights are frozen for assessment on groups of $n = 20$ held-out environments. Main endpoints and reference ABR stages share one group. R1 response checks and initialization comparisons share a second group; R2/R3 response checks use the main-assessment group. This grouping accounts for small differences when the same program is reported in different assessment tables. Mixture outcomes include every positive-probability program pair. Figure L.3 reports validation trajectories; final outcomes and reference-path comparisons use held-out assessments.

L.1 Programs and payoffs

Each episode executes an administrative program A and a provider program D for up to $T = 200$ hospital periods. Care teams apply the shared provider program to their own queues and state features. The two roles have separate objectives: joint design selects both programs using U_A, whereas provider response search selects D using U_D. The term *doctor program* refers to the provider-side implementation, including the coding expression used by coding staff.

For environment seed s, the realized episode length is T_s and the administrative return is $G_s(A, D) = \sum_{t=1}^{T_s} 0.95^{t-1} r_{A,t}$. Its payoff is

$$U_A(A, D; \mathcal{S}) = \text{mean}_{s\in\mathcal{S}}\, G_s - 0.06\, \text{mean}_{s\in\mathcal{S}}\, V_s - 0.25\, \text{Var}_{s\in\mathcal{S}}\, G_s, \tag{20}$$

$$V_s = 0.08 N_s^{\text{urgent rejection}} + 2 I_s^{\text{insolvency}} + 3[u_s - 0.20]_+ + [\bar{W}_s - 2]_+ + 2[\Delta_s^{\text{def}} - 0.25]_+. \tag{21}$$

Here $[x]_+ = \max\{x, 0\}$; $N_s^{\text{urgent rejection}}$ counts rejected urgent patients, and $I_s^{\text{insolvency}}$ indicates insolvency. The other terms penalize the rollout's up-coding rate u_s, mean queue wait $\bar{W}_s$ and high-minus-low true-CMI deferral gap Δ_s^{def} above their specified thresholds. Penalties are applied to each rollout before averaging, and variance uses the population convention over $\mathcal{S}$.

The provider payoff averages team utility over periods and environment seeds:

$$U_D(A, D; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{1}{JT_s} \sum_{j=1}^{J} \sum_{t=1}^{T_s} U_{j,t}, \tag{22}$$

Equation (1) supplies $U_{j,t}$, with equal weight across periods. Administrative changes to α, β or bonuses can therefore change the scale and composition of U_D as well as behavior. We assess a provider's gain by comparing its starting and selected programs under the same administrative contract. Throughout L3, H denotes mean simulated health among treated patients.

L.2 Initialization and joint OpenEvolve search

E0 establishes the pair from which each editor configuration begins strategic evaluation. The supplied library contains nine administrative and nine doctor programs, including neutral, clinical, profit-sensitive and audit-sensitive rules. Pairing each program with the neutral opposite role gives $9 + 9 - 1 = 17$ starting pairs.

Joint search edits both roles under U_A. Validation selection considers the supplied pairs, the five highest-scoring generated training candidates and prior selection winners (Algorithm A.2). All five editor configurations use this library; the original-prompt/high DeepSeek condition retains its previously completed E0. Each configuration's selected pair is then shared by its ABR and CSRO experiments. Appendix L.10.2 compares the full library with neutral-only and model-generated initialization.

L.3 Single-role response search and complete ABR

A unilateral update edits one role against a frozen opponent. The response oracle initializes from retained own-role programs and a neutral warm start, and receives the code, weight and training feedback of every positive-probability opponent (Algorithm A.3). In ABR, the opponent is the current opposite-role program and the incumbent remains eligible for selection. In CSRO, the opponent is a distribution over retained programs and the best retained own-role program supplies the validation reference.

ABR-A applies $A \to D \to A \to D \to A \to D$; ABR-D reverses this order. Each update receives 200 proposals, giving 1,200 proposals per path. A1–A3 and D1–D3 denote the successive administrative and doctor updates. Invalid and duplicate proposals consume slots, while API retries preserve the current slot. We record the selected pair and both roles' outcomes after every update, including selections that retain the incumbent. This record ties each transition to the role whose objective guided it.

L.4 JPSRO, CSRO and maximum-Gini CCE

Retaining historical programs creates a finite empirical game whose cross-play outcomes determine the next response task. CSRO supplies code-generation response oracles (Hennes et al., 2026) within Joint Policy-Space Response Oracles (JPSRO) (Marris et al., 2021). Each round expands the program libraries and updates a joint distribution over their interactions.

For m administrative and n doctor programs, let $M_A(i, j)$ and $M_D(i, j)$ be the training payoffs of (A_i, D_j), and let ρ_{ij} be its probability. The maximum-Gini coarse correlated equilibrium (MGCCE)

solves

$$\begin{aligned} \max_{\rho \in \mathbb{R}^{m \times n}} \quad & 1 - \sum_{i,j} \rho_{ij}^2 \\ \text{subject to} \quad & \rho_{ij} \geq 0, \qquad \sum_{i,j} \rho_{ij} = 1, \\ & \sum_{i,j} \rho_{ij}[M_A(k,j) - M_A(i,j)] \leq 0, \quad k = 1, \ldots, m, \\ & \sum_{i,j} \rho_{ij}[M_D(i,\ell) - M_D(i,j)] \leq 0, \quad \ell = 1, \ldots, n. \end{aligned} \tag{23}$$

Each deviation constraint compares the joint-distribution payoff with the payoff from choosing one fixed program against the opponent marginal. Every retained program remains a possible deviation, including those with zero current probability. Within the feasible set, the Gini objective favors a dispersed distribution. We solve the equivalent quadratic program minimizing $\frac{1}{2}\| \operatorname{vec}(\rho)\|_2^2$; Appendix O gives the numerical checks.

At the start of a round, the administrator searches against $q_D(j) = \sum_i \rho_{ij}$ and the doctor searches against $q_A(i) = \sum_j \rho_{ij}$. Both distributions remain fixed for the entire round. A novel program is admitted when its validation payoff exceeds the best retained own-role payoff by more than 10^{-8}. The procedure admits at most one program per role, completes all new cross-play pairs, and recomputes ρ. From an E0 1×1 pool, three and twelve rounds yield libraries of at most 4×4 and 13×13.

Mixture evaluation uses the full joint distribution. The expected payoff and maximum nonnegative administrative deviation gain are

$$\begin{aligned} v_r(\rho) &= \sum_{i,j} \rho_{ij} M_r(i,j), \qquad r \in \{A, D\}, \\ \epsilon_A(\rho) &= \max\left(0, \max_k \sum_{i,j} \rho_{ij}[M_A(k,j) - M_A(i,j)]\right). \end{aligned} \tag{24}$$

The provider gain ϵ_D is defined analogously. For validation, we apply the training-selected weights to the validation payoff matrices. Health and up-coding use the same joint weights over pairwise outcomes. Administrative mixture payoff therefore averages the risk-adjusted values of the individual program pairs, with each pair's seed-variance penalty already included. These calculations describe deviation incentives and patient outcomes in the retained game; Appendix N relates them to a larger policy universe.

L.5 Models and search budgets

The text-only code editors are DeepSeek-V4.1-Flash and Z.ai GLM-5.3 (Table L.1). The DeepSeek release notice and technical report document that model (DeepSeek-AI, 2026; DeepSeek, 2026); the GLM-5.3 overview documents the version used here (Z.ai, 2026), and the GLM-5 report provides background on the model family (GLM-5-Team, 2026).

Table L.1: Model identities and access channels for the completed L3 comparison. Short labels DeepSeek/DS and GLM elsewhere refer to these exact models.

	DeepSeek-V4.1-Flash	Z.ai GLM-5.3
API model identifier	`deepseek-flash`	`glm-5.3`
Service	DeepSeek official API	Huawei Cloud ModelArts MaaS
Thinking request	Disabled; or enabled/high	Enabled/high
Prompt conditions	Old and new packages	Old package
Input modality	Text	Text

DeepSeek uses `https://api.deepseek.com`; GLM uses Huawei Cloud MaaS at `https://api.modelarts-maas.com/v2`. Both use the chat-completions interface. The *off* condition disables thinking; *high* enables thinking with `reasoning_effort=high`. Figures and tables retain the labels

old and *new* for the original and revised prompt packages, respectively. Appendix L.10.1 describes their contents.

Table L.2: L3 proposal budgets. ABR and CSRO budgets cover response search after E0. Supplemental totals count all repeated searches listed in the middle column.

Condition	Initialization and update	Total proposals
E0	17 joint programs; both roles editable; select by U_A	200
ABR-A / ABR-D	Same E0; six alternating updates of 200 proposals each	1,200 each
MGCCE 200×3	E0 1×1 pool; 200 per role per frozen round	1,200
MGCCE 50×12	Same E0; 50 per role per frozen round	1,200
Terminal response	Three doctor searches at the fixed DeepSeek A3 contract	3×200
Matched α response	Same starting doctor; three searches per contract	$2 \times 3 \times 200$
Initialization checks	One search each: neutral-only and model-initialized	2×200

The main comparison crosses the two prompt packages with thinking off/high for DeepSeek and uses original-prompt/high for GLM. Each of the five configurations has its own E0, two ABR paths and both CSRO schedules, with one complete search per condition. The three targeted response tasks each have three independent searches, and the two initialization variants each have one. Held-out environment seeds characterize rollout variation for the selected programs; the targeted repetitions additionally show variation across searches for those response tasks.

Search uses an adapted OPENEVOLVE runtime (Sharma, 2025) with MEDI-SIM prompt templates and evaluator hooks; the released code records the upstream base version and local modifications. Joint search uses three islands, 20 population slots per island and an archive of 30 programs. Migration is triggered when the maximum per-island generation counter has advanced by 200 since the previous migration; the configured migration rate is 0.05. With three islands and at most 200 proposal slots per response-search block, this migration threshold is not reached. Generation temperature is 0.2. A 20-period preliminary rollout precedes full-horizon evaluation. Requests are capped at 250,000 prompt bytes and 65,536 output tokens. Prompt conditions share objectives, dynamics and proposal budgets. Population responses evaluate each candidate against multiple opponents, so a shared proposal budget fixes edit opportunities while rollout costs depend on the opponent distribution.

L.6 JOINT-DESIGN AND TERMINAL OUTCOMES

Joint search improves validation administrative utility over the best supplied pair in all five editor configurations (Figure L.1). Table L.3 follows the selected E0 pairs to their ABR and CSRO endpoints on the common held-out group. These endpoints reveal two recurring distinctions: provider utility depends on the current contract, and coding accuracy can move separately from treatment provision.

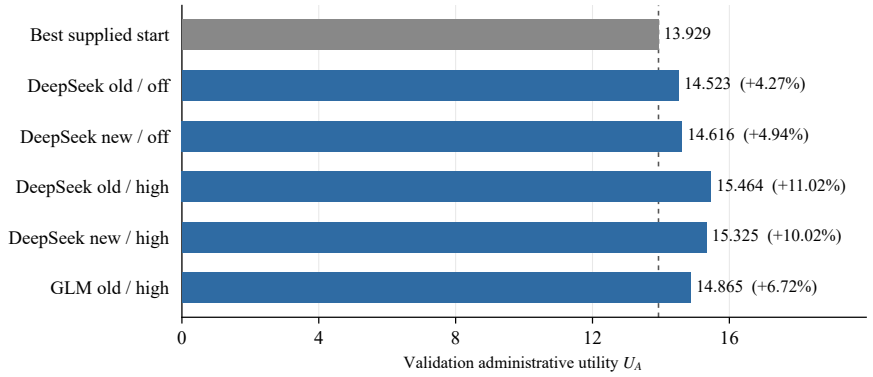


Figure L.1: **Joint-search improvement over the supplied starts.** Validation U_A for the best of 17 initial programs and the selected E0 from each 200-proposal search. All bars use the same validation group; labels give utility and percentage improvement over the initial reference. DeepSeek settings cross old/new prompts with thinking off/high; GLM uses old/high.

Table L.3: Frozen E0 and final ABR/CSRO outcomes across the five editor configurations, using the common assessment protocol.

Method	U_A	U_D	H	Effort	Up-code (%)
DeepSeek-V4.1-Flash / old / off					
E0	14.536	6.919	0.946	1.248	0.00
ABR-A	12.725	10.730	0.677	0.227	0.00
ABR-D	12.556	10.335	0.668	0.371	0.00
MGCCE 200×3	14.606	8.912	0.910	0.937	0.00
MGCCE 50×12	12.669	10.229	0.684	0.381	0.00
DeepSeek-V4.1-Flash / new / off					
E0	14.630	6.682	0.995	1.269	0.00
ABR-A	13.768	8.336	0.816	0.448	0.00
ABR-D	12.246	8.893	0.666	0.371	0.00
MGCCE 200×3	14.873	7.217	0.987	0.941	0.00
MGCCE 50×12	13.680	9.319	0.807	0.297	0.00
DeepSeek-V4.1-Flash / old / high					
E0	15.445	12.015	0.984	0.784	0.00
ABR-A	13.894	16.708	0.791	0.257	87.61
ABR-D	15.429	4.448	0.993	0.712	0.00
MGCCE 200×3	12.816	12.459	0.703	0.243	90.10
MGCCE 50×12	14.998	5.487	0.926	0.531	85.90
DeepSeek-V4.1-Flash / new / high					
E0	15.309	11.716	1.009	0.918	0.00
ABR-A	8.288	15.565	0.148	0.027	85.83
ABR-D	15.474	2.409	1.005	1.066	85.80
MGCCE 200×3	6.992	22.291	0.000427	0.000116	85.61
MGCCE 50×12	11.453	15.728	0.508	0.102	85.76
GLM-5.3 / old / high					
E0	14.888	4.128	0.965	1.380	0.00
ABR-A	12.296	14.379	0.631	0.172	0.00
ABR-D	15.588	0.362	1.004	0.794	0.00
MGCCE 200×3	12.442	18.957	0.624	0.153	85.63
MGCCE 50×12	10.911	21.073	0.449	0.103	87.71

Contract incentives and provider utility. The weights in provider utility determine which outcomes contribute to its value. At the ABR-D endpoints, GLM uses $(\alpha, \beta) = (0, 0.15)$ and revised-prompt/high DeepSeek uses $(0, 1)$, both with zero bonuses. Their utilities reduce to weighted total health per team-period minus fatigue cost, giving $U_D = 0.362$ and 2.409. In the GLM path, the last administrative update that changes the program lowers β from 0.5 to 0.15 and raises the KPI health weight from 1 to 1.5, while keeping the doctor program fixed. Health remains near its recovered level as the contract changes the value assigned to those outcomes.

The GLM 50×12 endpoint illustrates the financial side of the same objective. Its active administrative programs use $\alpha \in \{0.65, 0.75\}$, $\beta = 0.5$ and a bonus pool of 0.10. Four active doctor programs cap effort at 2%–15% of `base_effort` and rank coding candidates by `upcode_gain`. The assessed utility consists of 20.519 from profit, 0.534 from health and 0.020 from bonuses, with zero fatigue penalty. Profit accounts for 97.4% of $U_D = 21.073$, explaining the high utility despite low treatment effort.

Revised-prompt/high DeepSeek 200×3 reaches the zero-effort boundary. Its final doctor search faces a contract with $\beta = 0$ and admits `effort = 0.0`. This program receives 99.94% of the final doctor marginal, while the remaining positive-effort programs account for the small positive mixture means. The simulator pays for completed coded cases and charges for delivered effort, allowing a completed case to generate revenue at zero effort and zero health. This settlement and action boundary explains the near-zero endpoint and is important when interpreting its magnitude.

Coding accuracy and treatment provision. Accurate coding can coexist with a substantial reduction in care. In original-prompt/off DeepSeek, E0 and both ABR endpoints have zero up-coding, while effort falls from 1.248 to 0.227 in ABR-A and 0.371 in ABR-D; H falls from 0.946 to 0.677 and 0.668. GLM ABR-A similarly keeps coding at zero while effort falls from 1.380 to 0.172 and

health from 0.965 to 0.631. Following effort and health reveals these changes even when the coding measure stays constant.

Similar positive coding rates also accompany different levels of treatment. In revised-prompt/high DeepSeek, ABR-A, ABR-D and MGCCE 200×3 have up-coding rates of 85.83%, 85.80% and 85.61%, with health 0.148, 1.005 and 0.000427, respectively. Their corresponding effort levels are 0.027, 1.066 and 0.000116. The ABR-D endpoint combines high treated-patient health with persistent coding inflation, while the population-search endpoint combines similar coding with near-zero effort.

The selected code shows how the two channels can separate. Both GLM and revised-prompt/high DeepSeek end ABR-D with $\alpha = 0$. GLM scores codes using `upcode_pressure - audit_penalty - ethics_pressure`, whose positive term includes α. Setting $\alpha = 0$ therefore removes that gain term. DeepSeek retains `upcode_gain - audit_penalty`, which evaluates raw billing gains independently of α. Its effort and health recover while the coding expression continues to favor billing gains. Inspecting the expressions traces the different outcomes to the incentive features each program actually uses.

L.7 Response trajectories and key updates

Figure L.2 follows the reference ABR paths by the role just updated. A doctor update changes behavior under a fixed administrative contract; an administrative update can change both operational levers and the weights entering doctor utility. Reading the trajectories in this order links a same-contract utility gain to the subsequent change in care.

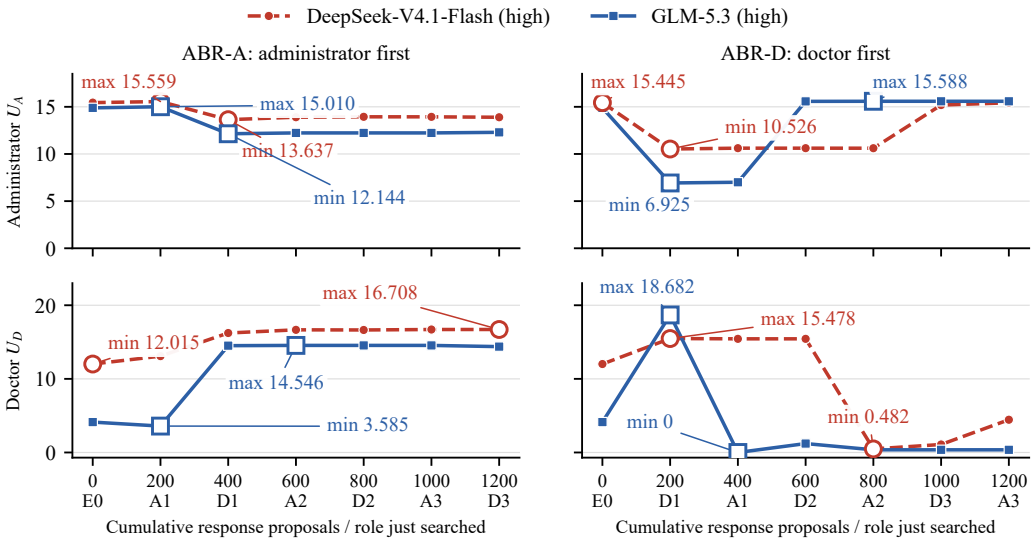


Figure L.2: **Full utility trajectories for ABR-A and ABR-D.** Administrative utility U_A and doctor utility U_D for the same programs and stages as Figure 4.

Table L.4 locates the main behavioral transitions. The first doctor response in administrator-first GLM reduces effort while leaving coding unchanged, whereas DeepSeek changes both channels. Doctor-first paths begin with substantial effort reductions, followed by administrative revisions and later doctor programs that restore treatment. The matched searches in Appendix L.8 examine the incentive change behind that recovery. Figure L.3 gives the validation trajectories for all four DeepSeek configurations. With thinking off, the revised-prompt configuration ends ABR-A at higher health; with high thinking, it ends at lower health. High-thinking ABR-D recovers health under both prompt packages, with different coding outcomes. Each path begins from its own E0 and encounters its own opponents. The figure therefore describes how the complete search configurations develop, complementing the proposal-level summaries in Appendix L.10.1.

Table L.4: Selected doctor updates in the original-prompt/high reference paths. Arrows show outcomes before and after each update.

Model	Path	Proposals	H	Effort	Up-code (%)
DeepSeek	ABR-A	200 → 400	1.009 → 0.773	0.762 → 0.247	0.00 → 88.79
GLM	ABR-A	200 → 400	0.999 → 0.603	1.290 → 0.138	0.00 → 0.00
DeepSeek	ABR-D	0 → 200	0.984 → 0.402	0.784 → 0.093	0.00 → 88.85
GLM	ABR-D	0 → 200	0.965 → 0.000	1.380 → 0.000	0.00 → 89.84
DeepSeek	ABR-D	800 → 1000	0.411 → 0.972	0.091 → 0.688	86.37 → 0.00
GLM	ABR-D	400 → 600	0.000 → 1.006	0.000 → 0.839	86.53 → 0.00

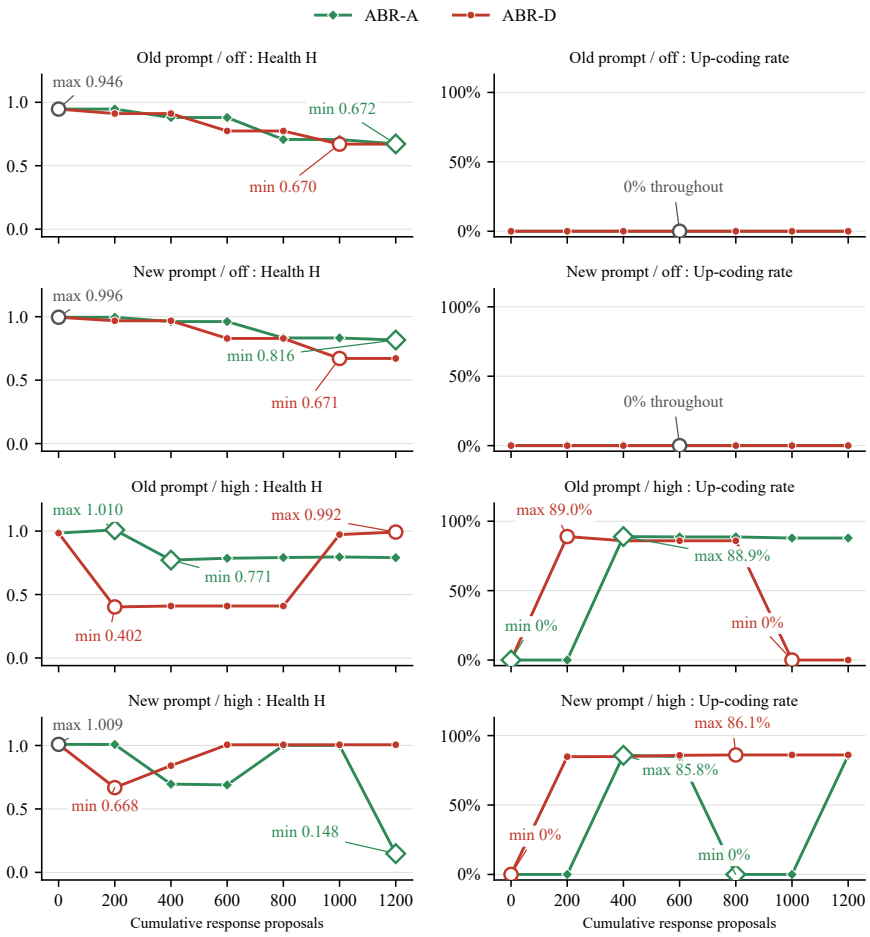


Figure L.3: **Validation trajectories across DeepSeek configurations.** Four rows show old/new prompts with thinking off/high. Each configuration has its own E0 and one search per ABR path. The third row uses the DeepSeek programs assessed in Figure 4.

L.8 Matched contract and terminal-response checks

Two repeated-search experiments examine the recovery observed along DeepSeek ABR-D. The first changes a single incentive weight while holding the starting doctor fixed. The second holds the final administrative program fixed and searches for another provider response from its selected doctor. Together they distinguish a contract's effect on the next response from its exposure to further adaptation.

Changing the profit weight from a common starting doctor. The matched branches use the D2 and A2 snapshots at 600 and 800 proposals. They share the same low-effort doctor, and the

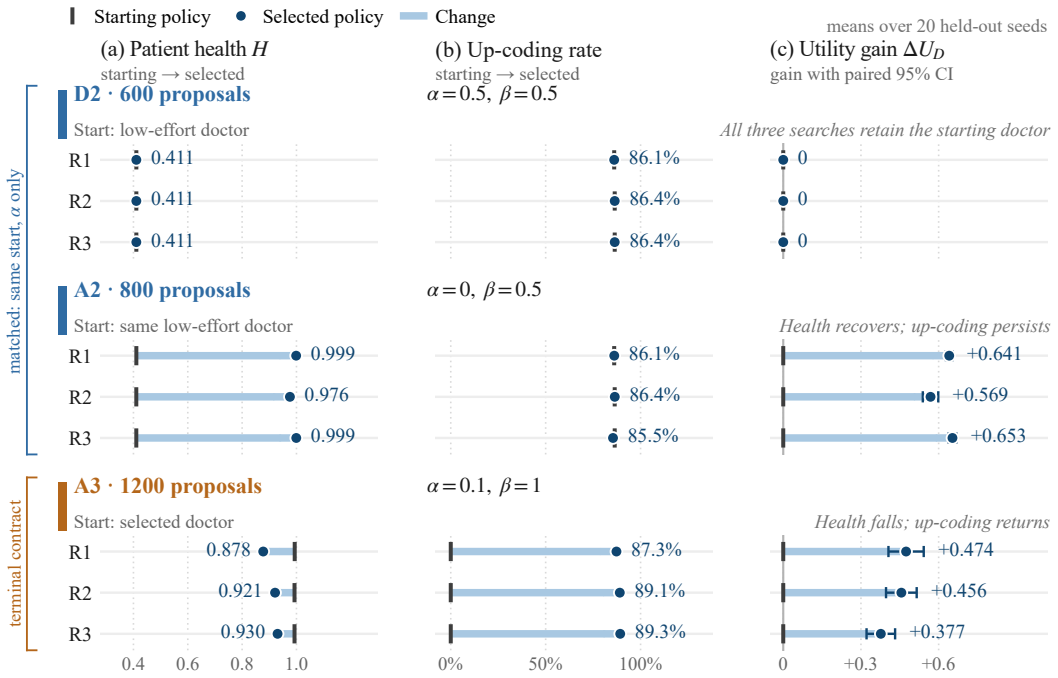


Figure L.4: **Doctor responses at DeepSeek ABR-D checkpoints (old prompt/high).** D2/A2 compare contracts differing only in α from the same doctor; A3 tests further adaptation at the terminal contract. R1–R3 are independent 200-proposal searches. Ticks/circles mark starting/selected outcomes; connectors show changes. Panel (c) gives same-contract utility gains with paired 95% t intervals (Eq. 26).

administrative programs differ only in $\alpha = 0.5$ versus 0. Both use $\beta = 0.5$, the same training and validation groups, the original-prompt/high editor and 200 proposals per search. All three $\alpha = 0.5$ searches retain the starting doctor; all three $\alpha = 0$ searches select a higher-effort program (Table L.5). In R1, the selected expression is `min(1.5 * base_effort, stamina_ratio)`, and the program accepts all cases subject to the host's resource constraints.

The recovery concerns effort and treated-patient health. Up-coding remains around 86%, and the R2 $\alpha = 0$ response raises mean waiting from 0.806 to 3.041. Thus the same incentive change supports a higher-effort response while coding distortion persists and, in one repetition, waiting increases. The contract comparison therefore needs to be read across multiple response channels.

Further response to the terminal contract. The terminal experiment fixes the full A3 program at 1,200 proposals, including $\alpha = 0.1$, $\beta = 1$, capacity, bonuses and KPI rules. Each search starts from A3's selected doctor. All three searches increase doctor utility and reduce treated-patient health. For R1, $\Delta U_D = 0.47429$ has paired 95% t interval $[0.40620, 0.54238]$, and $\Delta H = -0.11590$ has interval $[-0.11767, -0.11414]$. All 20 assessment environments exhibit these respective signs. The selected effort expression is `base_effort * 0.2 * stamina_ratio`, with coding score `upcode_gain - audit_penalty`. The new response earns more under the same contract through lower effort and renewed coding inflation.

Paired assessment. For fixed contract A, let D_{start} and D_{sel} be the starting and selected doctor programs. For each $X \in \{U_D, H\}$, the paired change is

$$d_s^X = X(A, D_{\text{sel}}; s) - X(A, D_{\text{start}}; s), \qquad \bar{d}^X = \frac{1}{n} \sum_s d_s^X. \tag{25}$$

The 95% Student t interval uses the sample standard deviation of these changes:

$$\bar{d}^X \pm t_{0.975,n-1} \frac{s_d}{\sqrt{n}}, \qquad s_d^2 = \frac{1}{n-1} \sum_s (d_s^X - \bar{d}^X)^2. \tag{26}$$

Each interval summarizes rollout uncertainty for one selected program comparison. R1–R3 report the independent searches separately, using the assessment groups specified at the start of this appendix. A retained starting program gives zero paired changes.

Table L.5: Three independent 200-proposal searches per response task. Health and coding arrows compare starting and selected programs. Brackets give paired 95% t intervals for mean same-contract utility gains (Eq. 26).

Task	R	Starting U_D	H: start → selected	ΔU_D [95% CI]	Up-code (%)
$\alpha = 0.5$	1	15.399	0.411 → 0.411	+0.000 [+0.000, +0.000]	86.11 → 86.11
$\alpha = 0.5$	2	15.439	0.411 → 0.411	+0.000 [+0.000, +0.000]	86.38 → 86.38
$\alpha = 0.5$	3	15.439	0.411 → 0.411	+0.000 [+0.000, +0.000]	86.38 → 86.38
$\alpha = 0$	1	0.483	0.411 → 0.999	+0.641 [+0.628, +0.654]	86.11 → 86.14
$\alpha = 0$	2	0.482	0.411 → 0.976	+0.569 [+0.539, +0.598]	86.37 → 86.41
$\alpha = 0$	3	0.482	0.411 → 0.999	+0.653 [+0.638, +0.668]	86.37 → 85.48
Final ABR-D	1	4.409	0.994 → 0.878	+0.474 [+0.406, +0.542]	0.00 → 87.30
Final ABR-D	2	4.448	0.993 → 0.921	+0.456 [+0.397, +0.515]	0.00 → 89.06
Final ABR-D	3	4.448	0.993 → 0.930	+0.377 [+0.322, +0.432]	0.00 → 89.28

Across the three repetitions, the matched-contract searches recover the same direction of effort and health change, while the terminal searches recover the same direction of utility gain and health decline. Their magnitudes vary with the selected programs. The terminal contract remains a productive starting point for provider search even after the earlier ABR recovery.

L.9 Adaptation against retained policies

Retained histories allow an earlier program to become relevant again when its opponents change. In original-prompt/high DeepSeek 200×3, the initial administrative policy has zero marginal weight after round two and weight 0.296 after round three. The administrative library and its existing payoff entries stay fixed while one doctor program is added. In GLM 50×12, the final distribution uses six administrative and four doctor programs from several rounds, yielding 18 positive-probability pairs in an 8×8 library; the largest pair weight is 0.12977. The former shows renewed participation of an earlier policy, while the latter combines several generations of policies and responses.

The two schedules distribute the same 1,200 response proposals differently across population updates. Revised-prompt/high DeepSeek reaches held-out health 0.000427 under 200×3 and 0.508 under 50×12. GLM changes from 0.624 to 0.449, and both thinking-off DeepSeek configurations have higher health under 200×3. These differing directions show how search depth per response and the frequency of opponent updates interact with the programs discovered in each configuration.

Weighting a fixed historical library. To examine the weighting rule separately, Table L.6 holds the completed libraries and payoff matrices fixed and compares training-selected MGCCE weights with uniform weights over all pairs. MGCCE yields smaller validation deviation gains for both roles in all four reference games. In GLM 50×12, the doctor gain decreases from 2.678 to 0.132. The comparison measures the incentive to switch among known programs. Table L.3 and the code analysis describe the treatment and coding outcomes delivered by those same programs, providing a separate account of care quality.

Table L.6: Final-library weighting comparison for original-prompt/high reference runs. Entries are maximum nonnegative unilateral deviation gains on the validation matrices. Each row pair uses the same programs; boldface marks the smaller gain for each role.

Reference run	Joint weights	Administrator ϵ_A	Doctor ϵ_D
DeepSeek, 200×3	Uniform	0.327	1.396
DeepSeek, 200×3	MGCCE	**0.038**	**0.018**
DeepSeek, 50×12	Uniform	0.533	0.489
DeepSeek, 50×12	MGCCE	**0.014**	**0.030**
GLM, 200×3	Uniform	1.249	3.505
GLM, 200×3	MGCCE	**0.000**	**0.024**
GLM, 50×12	Uniform	0.055	2.678
GLM, 50×12	MGCCE	**0.016**	**0.132**

L.10 Search sensitivity

The final comparisons examine the information supplied to the editor and the programs available at initialization. They characterize the completed search configurations, including proposal statistics and selected outcomes, rather than averaging over a broad distribution of search runs.

L.10.1 Prompt and thinking conditions

The original prompts supply the task, DSL and payoff feedback. The revised prompts additionally request opponent inspection, predictions about edit effects and a different supported expression after repeated unchanged payoffs. Both full-rewrite and diff templates clarify editable incentives and feature activation.

We examine six administrative response blocks per DeepSeek configuration, totaling 1,200 proposals across the two ABR orders. Each valid candidate is compared with its parent on the same training task; equality and distinct-payoff counts use tolerance 10^{-10}. Table L.7 summarizes the configurations under their realized sequences of opponents, starting from their respective E0 pairs.

Table L.7: Administrative search under old and new prompts. Percentages use valid proposals; Scores/block is the mean number of distinct training payoffs per response block. Boldface highlights the old/new comparison with thinking off.

Thinking	Prompt	Valid	Unchanged (%)	Improved (%)	Scores/block
Off	Old	1195	**94.1**	**2.7**	**2.0**
Off	New	1180	**48.5**	**30.8**	**11.0**
High	Old	1188	7.0	72.8	45.2
High	New	1196	6.1	75.4	40.5

With thinking off, the revised-prompt configuration has fewer unchanged valid proposals (48.5% versus 94.1%), more improving proposals (30.8% versus 2.7%), and more distinct scores per block (11.0 versus 2.0). The two high-thinking configurations have closer unchanged and improving fractions. These statistics describe how broadly each search explores payoff-changing edits in its own response tasks. Figure L.3 shows how those searches translate into patient outcomes along the complete trajectories.

L.10.2 Initialization sensitivity

Neutral-only search starts from one human default program and receives 200 proposals. Model-initialized search generates its start from the task, objective and DSL constraints, taking the first complete program that passes training checks. This initialization uses one proposal slot, leaving 199 for evolutionary search. Both variants and the historical full-library E0 are assessed on a common held-out group.

Table L.8: Initialization sensitivity. The full-library row re-evaluates the reference E0; the other rows show the start and result of each additional search. All up-coding rates are zero.

Initialization	Stage	U_A	U_D	H	Effort
Full 17-program library	Historical selected E0	15.446	12.236	0.984	0.780
Neutral-only	Initial	13.614	-0.828	0.953	1.807
Neutral-only	Selected	14.814	7.460	0.942	0.833
Model-initialized	Initial	12.805	2.112	1.028	1.776
Model-initialized	Selected	14.282	-12.024	1.016	1.706

On the common validation group, neutral-only and model-initialized search reach $U_A = 14.856$ and 14.457, exceeding the best supplied pair's 13.929. On the paired held-out assessment, both improve administrative utility over their own starts. The historical full-library E0 scores higher by 0.633 and 1.164, respectively (Table L.8). Search can therefore improve from each tested initialization, while the available starting programs affect the selected endpoint.

M Discussion and Limits of Interpretation

Following an intervention through the coupled system. Coding, case selection and target-oriented behavior share the same queues, resources and financial incentives. The L2 audit comparison traces a change in billing incentives to concurrent changes in access. The L3 matched-contract searches extend this analysis to revised response logic, and the terminal searches show how providers can subsequently improve their own utility while reducing health. Together, these experiments follow a mechanism from its immediate effects to the responses it invites. The observed directions and magnitudes describe the tested simulator settings; estimating their prevalence in hospitals requires empirical study.

Inspecting executable policies. Typed programs support source-level inspection, version comparison and replay. The coding expressions in Appendix L.6, for example, expose whether an incentive weight enters a billing decision through `upcode_pressure` or whether the program uses raw `upcode_gain`. Interpreting more complex expressions requires the same attention to feature definitions and host execution. Integration with operational systems would involve validated data and terminology mappings, review of permitted actions, change control and prospective assessment. The separation of rule logic, data and invocation is related to healthcare interoperability standards (Mandel et al., 2016; Strasberg et al., 2021); implementing those mappings remains a development task. Guidance on clear information for an intended use provides a further governance consideration (U.S. Food and Drug Administration et al., 2021).

Environment assumptions. The environment uses stylized patient categories, exogenous arrivals, health production and provider responses. All teams apply a shared program to local state, coding precedes treatment in the computational loop, and α and β encode reduced-form incentives. The action space includes zero effort, which can generate completed-case payment under the current settlement rule. These choices define the mechanisms available to search and help explain extreme endpoints. Literature checks examine intended directions, and alternative response forms examine a limited family of behavioral variations. Empirical calibration and extensions for heterogeneous provider programs, referrals, demand responses, bargaining and staffing would test the conclusions under richer operational assumptions; calibration could alter the dominance, magnitude, or even direction of some channel effects.

Search and assessment coverage. Each main editor configuration has one complete search per condition. The targeted response tasks have three repetitions, while twenty held-out environment seeds estimate outcomes for selected programs. These sources of variation address different questions: search repetitions concern which programs are discovered, and environment seeds concern the rollout performance of a fixed program pair. Adaptive validation reuse and response-oracle quality also affect evaluation coverage. Proposal budgets equalize edit opportunities, while compute costs depend on the number of opponents and rollouts evaluated. Further restarts, fresh assessment groups and

broader response searches would extend this coverage. No reported endpoint certifies the absence of profitable unseen responses.

Objectives, access and care. The administrative payoff combines finances, reputation and soft penalties; it is not a complete social-welfare function. Provider utility assigns its own weights to profit, health, bonuses and workload, and can prefer actions that lower patient health. CCE constraints characterize incentives to make unconditional program deviations in the corresponding empirical game, so strategic stability does not imply fairness, high health, or a Stackelberg-optimal committed rule. Assessing a committed administrative rule requires the response-dependent objective in Eq. (2), and assessing care requires the channel outcomes themselves. In particular, treated-patient health varies with effort, waiting and accepted case mix. Reading it together with rejection, deferral and resource use reveals the access consequences of a policy. Clinical safety and equitable access would require additional domain constraints and evaluation for the intended setting.

N Conditional Convergence and Response Certificates

The analysis covers three levels of strategic evaluation. A fixed library gives a finite game whose deviation constraints can be checked directly. Population expansion adds new programs and, under exact response oracles over a finite universe, reaches a full-game CCE at a no-admission round. With approximate search and separate payoff estimates, an error bound identifies the contributions of numerical tolerance, response quality and payoff estimation. The argument specializes PSRO and JPSRO population expansion to the admission rule used here (Lanctot et al., 2017; Marris et al., 2021).

N.1 Games, distributions, and deviations

Let $\mathcal{P}_A$ and $\mathcal{P}_D$ be policy universes and $\mathcal{L}_A \subseteq \mathcal{P}_A$, $\mathcal{L}_D \subseteq \mathcal{P}_D$ the retained libraries. All payoffs in this appendix are fixed scalar functions of a program pair; for the administrator, this includes the pairwise seed-variance penalty before mixing, as in Appendix L.4. For a joint distribution ρ supported on the libraries and its opposite-role marginal q_{-r}, define

$$v_r^M(\rho) = \mathbb{E}_{(A,D)\sim\rho} M_r(A, D), \qquad b_r^M(p; q_{-r}) = \mathbb{E}_{p_{-r}\sim q_{-r}} M_r(p, p_{-r}), \tag{27}$$

with the role order understood when $r = D$. The full-universe deviation gap is

$$\mathcal{E}_r^M(\rho) = \left[\sup_{p\in\mathcal{P}_r} b_r^M(p; q_{-r}) - v_r^M(\rho)\right]_+ . \tag{28}$$

Replacing $\mathcal{P}_r$ by $\mathcal{L}_r$ gives the restricted-library gap. A CCE has zero full-universe gaps. The deviation uses the opponent marginal because it is chosen unconditionally, before observing a recommended program. Its reference payoff is the expected payoff under the full joint distribution, retaining the dependence between the two recommended programs; sampling the roles independently from their marginals would generally change this reference payoff and invalidate the certificate.

Proposition 1 (Fixed-library meta-solver)**.** *For any finite nonempty libraries and finite payoff entries, the feasible set in Eq.* (23) *is nonempty and compact. The maximum-Gini objective has a unique maximizer. A returned probability distribution with maximum unilateral-deviation residual at most* $\epsilon_{\text{solver}} \geq 0$ *is an* ϵ_{solver}*-CCE of the retained game.*

Proof. A finite game has a mixed Nash equilibrium, whose product distribution is a CCE, so the feasible set is nonempty. It is the intersection of a probability simplex and finitely many closed half-spaces, hence compact and convex. The objective $1 - \|\rho\|_2^2$ is continuous and strictly concave, yielding existence and uniqueness. The final statement follows directly from the unilateral-deviation inequalities. The unique solution is defined for a fixed payoff matrix; expanding the library changes the optimization problem. □

Proposition 2 (Finite-universe termination)**.** *Suppose* $\mathcal{P}_A$ *and* $\mathcal{P}_D$ *are finite; each round solves a CCE of the current retained game; both response oracles return exact best responses over their full policy universes to the frozen round-start opponent marginals; and admission and the meta-solver use the same exact payoff functions. Admit a novel program whenever its payoff strictly exceeds the*

best retained own-role payoff against that marginal, and stop when neither role admits a program. Starting from nonempty libraries, at most

$$(|\mathcal{P}_A| - |\mathcal{L}_A^0|) + (|\mathcal{P}_D| - |\mathcal{L}_D^0|)$$

rounds admit new programs. At the first no-admission round, the current distribution is a CCE of the full finite policy game.

Proof. Every round that continues admits at least one previously absent program. Library sizes increase and are bounded by the finite universe sizes, proving the bound. At a no-admission round, exactness of the response oracle implies, for each role,

$$\max_{p\in\mathcal{P}_r} b_r^M(p; q_{-r}) \le \max_{p\in\mathcal{L}_r} b_r^M(p; q_{-r}) \le v_r^M(\rho).$$

The second inequality follows from the retained-game CCE constraints. Thus every full-universe deviation gap is zero. The no-admission condition is sufficient; admission may still occur at a full-game CCE if a new program beats the best fixed retained program while remaining below the correlated baseline payoff. □

The two baselines play different roles. Admission compares a candidate with the *best fixed retained program*, while CCE compares a fixed deviation with the *joint-distribution payoff*. At a retained-game CCE, the latter is at least as large as the former. A new program can lie between them, producing a positive admission gain while satisfying the full-game CCE inequality. Proposition 2 uses this ordering at a no-admission round. It also assumes a finite program universe explicitly; action bounds alone constrain outputs, while finiteness additionally requires a finite set of admissible programs.

Proposition 3 (Approximate certificate with separate payoff estimates)**.** *Let $\widehat{M}$ be the training payoff function, $\widetilde{M}$ the validation payoff function, and M a target payoff function. Suppose the current distribution has restricted training gap at most $\epsilon_{\mathrm{solver}}$. For each role, assume*

1. *a response oracle returns p_r^+ with validation payoff within ξ_r of the supremum over $\mathcal{P}_r$;*
2. *no response is admitted, so $b_r^{\widetilde{M}}(p_r^+; q_{-r}) \le \max_{p\in\mathcal{L}_r} b_r^{\widetilde{M}}(p; q_{-r}) + \eta$;*
3. *uniformly over retained pairs and all unilateral deviations against retained opponents, $|\widehat{M}_r - \widetilde{M}_r| \le \kappa_{\mathrm{tv},r}$ and $|\widehat{M}_r - M_r| \le \delta_r$.*

Then

$$\mathcal{E}_r^M(\rho) \le \epsilon_{\mathrm{solver}} + \eta + \xi_r + 2\kappa_{\mathrm{tv},r} + 2\delta_r. \tag{29}$$

The same conclusion holds when a duplicate candidate is rejected, since its validation payoff cannot exceed the best retained payoff.

Proof. Uniform entrywise errors also bound expectations under any fixed opponent marginal. Suppressing the role index, the assumptions give

$$\begin{aligned}
\sup_{p\in\mathcal{P}_r} b^{\widehat{M}}(p; q) &\le \sup_{p\in\mathcal{P}_r} b^{\widetilde{M}}(p; q) + \kappa_{\mathrm{tv}} \\
&\le b^{\widetilde{M}}(p^+; q) + \xi + \kappa_{\mathrm{tv}} \\
&\le \max_{p\in\mathcal{L}_r} b^{\widetilde{M}}(p; q) + \eta + \xi + \kappa_{\mathrm{tv}} \\
&\le \max_{p\in\mathcal{L}_r} b^{\widehat{M}}(p; q) + \eta + \xi + 2\kappa_{\mathrm{tv}} \\
&\le v^{\widehat{M}}(\rho) + \epsilon_{\mathrm{solver}} + \eta + \xi + 2\kappa_{\mathrm{tv}}.
\end{aligned}$$

Changing both the deviation payoff and the baseline payoff from $\widehat{M}$ to M adds at most 2δ. Taking the nonnegative part yields Eq. (29). □

The bound is deterministic conditional on its assumptions. If the uniform error bounds hold jointly with probability at least $1 - \zeta$, the certificate inherits that probability. Obtaining such bounds requires control over the policy class, adaptive selection and the administrative variance penalty. This separates a confidence statement about a fixed program comparison from a uniform guarantee covering all allowed deviations.

N.2 Implications for the implemented search

The implementation uses admission threshold $\eta = 10^{-8}$, retained-game deviation tolerance 10^{-7}, training-based mixture weights and validation-based candidate selection. These thresholds control numerical comparisons; they are not statistical significance thresholds. Complete cross-play supplies every payoff entry required by the constraints; frozen opponent distributions keep candidate and incumbent evaluations on the same response task. Historical programs remain available as deviations even at zero current probability, and role checks keep unilateral edits within the permitted program.

The experiments stop after three or twelve rounds because their proposal budgets are exhausted, not because Proposition 2 was verified. Their measured certificates concern the retained training or validation matrices, as in Table L.6. Extending these results to the full program class through Proposition 3 additionally requires a bound on the response-oracle error ξ_r and the uniform payoff discrepancies. Those quantities remain unmeasured in the current experiments, and failure to find a new response within a finite budget is not evidence that an improving one does not exist. Repeated searches and fresh evaluations provide practical tests of additional responses; the terminal-response experiments illustrate the information such tests can add.

Finally, the CCE evaluates a simultaneous empirical game over complete programs under the specified utilities. The committed-leader objective in Eq. (2) evaluates an administrative rule through its selected provider response. These are distinct evaluation problems, and neither guarantees patient health or equity; the paper therefore reports health, access and coding directly.

O Implementation Details and DSL Guardrails

The program interface separates editable decisions from the host routines that execute and evaluate them. L1/L2 use native response rules; L3 searches the typed fields in Listing 1. Across L3 candidates, the host supplies patient generation, state transitions, action projection, physical feasibility, audits, settlement and outcome measurement. The selected expressions determine how each role acts within this environment.

Editable rules and evaluation context. Administrative expressions set incentive weights, capacity, bonus design and KPI weights. Provider expressions determine effort, acceptance, deferral, bed requests and coding scores. Programs read the features exposed for each expression, and syntax and role checks enforce the permitted assignments. During a unilateral response, the opposite-role program and its evaluation distribution are fixed. Appendix L.1 defines the role payoffs, and Appendix L gives the training, validation and assessment groups.

Coding execution. For each admissible DRG, the host provides incremental reimbursement, coding gap, expected audit penalty and ethics pressure. The searched `candidate_score` ranks these candidates; the highest-scoring group becomes the reported DRG. Audit enforcement and settlement then follow the host rules. This separation allows search to change which supplied features determine a coding decision while preserving the candidate set and the financial consequences of that decision. The system, full-rewrite and diff prompts use the same interface.

Listing 1: L3 editable interface and host constraints.

```
Assignment-only module: kind, version, admin_policy, doctor_policy.
Admin: alpha, beta, total_capacity, bonus_pool, flex_pool,
       xi, wH, wW, wrej, wC, kappa.
Doctor: effort, triage_accept, triage_defer, request_bed,
        candidate_score.
Safe calls: I, clip, min, max, abs, round, soft_gt, soft_lt.
Expressions read exposed features; the host enforces the schema,
clipping, physical feasibility, audit application and metrics.
```

MGCCE numerical checks. Equation (23) is solved in CVXPY using OSQP. A rejected OSQP solution triggers one CLARABEL attempt on the same zero-slack quadratic program. Absolute and relative solver tolerances are 10^{-10}, probability tolerance is 10^{-8}, and deviation acceptance tolerance is 10^{-7}. Acceptance requires a complete cross-play matrix, optimal solver status, normalized non-negative probabilities within tolerance, and satisfied unilateral-deviation inequalities. CLARABEL

solutions additionally require an explicit optimality certificate. These checks establish the numerical validity of the returned distribution for the supplied payoff matrices.

P TERMINOLOGY AND MEASUREMENT

The outcome measures describe different populations and stages of the IPS loop. Coding is measured over the patients currently monitored by the hospital; rejection and deferral describe considered patients; health and effort describe delivered treatment; and KPI alignment summarizes team-period observations. Keeping these populations explicit ties each result to the decision it measures.

P.1 CLINICAL ATTRIBUTES AND RESPONSE CHANNELS

A diagnosis-related group (DRG) is a billable case category. Each arrival has a true group, normalized clinical complexity (CMI), urgency and waiting tolerance. The reported group determines case-based reimbursement, while true attributes determine clinical need and the high-/low-CMI comparisons. This separation defines the coding wedge. The measurement wedge arises because the KPI combines health with waiting, rejection and cost and then governs bonuses or capacity steering.

The administrator sets hospital levers, and care teams execute the shared provider program on local state. Coding staff use its coding expression. Coding changes the reported group; selection changes who is accepted or rejected; deferral leaves a patient waiting; effort sets treatment intensity; and resource requests interact with host capacity allocation. The classical terms *cream-skimming*, *skimping* and *gold-plating* refer, respectively, to favoring easier or more profitable cases, reducing care inputs and increasing treatment intensity.

P.2 PATIENT AND QUEUE OUTCOMES

Table P.1: Patient-level and queue-level outcome definitions.

Measure	Population and interpretation
Treated-patient health H	Mean simulated health among treated cases, using Eq. (15). The team-skill multiplier permits values above one. Effort, true complexity and waiting enter production; the accepted case mix also affects the treated-patient mean.
Treatment effort	Effort delivered to treated cases. The effort and health outcomes are read together with rejection and deferral, which describe access to treatment.
Rejection and deferral gaps	High-true-CMI rate minus low-true-CMI rate among considered patients in the corresponding group. A positive gap places the greater rejection or deferral burden on high-CMI patients. Overall deferral and its complexity gap describe all defer decisions.
Strategic-delay rate and gap	Deferrals made with a positive local bed-availability estimate, expressed relative to all considered patients in the relevant population. The gap subtracts the low-CMI rate from the high-CMI rate. The term identifies this operational subset of deferrals using the capacity signal available at the decision.
Waiting	Queue waiting accumulated before treatment. The reputation calculation uses the mean queue wait measured after arrivals are registered and before treatment decisions, with zero for an empty queue (Appendix B). Rollout waiting is reported with access and treatment outcomes.
Up-coding rate	In each period, the monitored population contains patients treated, rejected or abandoning during that period and those still queued. Its up-coding fraction is averaged over periods, with zero for an empty period. A queued patient remains in the denominator until leaving. Every changed reported group is an up-code because admissible alternatives are upward-only.
Terminal funds F_T	Funds at episode termination, expressed in simulator units. Reported horizons are maximum episode lengths; Appendix L.1 uses realized length T_s in the L3 payoffs.
Severity-weighted score	Mean of true CMI times realized health over treated cases; zero for an empty treated set. Appendix K adds this observable to the KPI through coefficient w_{sev}.

The up-coding statistic measures the prevalence of inflated coding in the monitored population. It depends on initial coding choices and on how long differently coded patients remain in the system. Changes in selection, deferral and queue residence can therefore affect its period-averaged value. This population interpretation is used consistently in the incentive grids, intervention sweeps and program-search results.

P.3 TEAM-LEVEL SCORES AND AGGREGATION

KPI combines team health with waiting, rejection and cost according to Eq. (3). The bonus tournament maps all teams' scores to relative shares, and KPI steering uses measured performance to allocate flexible capacity. The following conventions apply to the team-level figures in Appendix E.3.

KPI–health and bonus–health correlations. Within each rollout, Pearson correlations pool team-periods with considered patients. Team mean health is zero in a period with no treated patients. The reported grid value then averages the rollout correlations across seeds. These summaries combine variation across teams and across time under that observation convention.

Lagging-team case selection. A team is KPI-lagging when its previous-period score is below the mean of its peers. Mean accepted CMI describes the case mix accepted by these teams. Easy-case tilt is their low-minus-high CMI acceptance-rate gap: a positive value indicates more frequent acceptance of low-CMI patients, while a negative value indicates more frequent acceptance of high-CMI patients.

Cross-team dispersion. The load coefficient of variation uses considered patient counts across teams. CMI and health variances use team means among teams delivering treatment. These dispersion statistics are computed by period, averaged within the rollout and then across seeds. They complement the lagging-team measures by describing variation across the hospital's teams.

Rollout and mixture assessment. Main L1/L2 results report means over 30 environment seeds. The supplementary response-form and severity-score experiments use their stated ten-seed configurations. L3 separates training and validation from twenty-seed held-out assessments of selected programs. For a retained-policy mixture, outcomes are averages over program pairs using the full joint weights. Administrative payoff is likewise an average of pairwise risk-adjusted payoffs, with the seed-variance penalty evaluated within each pair. Appendix L.4 gives the mixture and deviation formulas; Appendix L.8 defines paired changes for the repeated doctor-response tasks.