# Separating Diagnosis from Control: Auditable Policy Adaptation in Agent-Based Simulations with LLM-Based Diagnostics

Shaoxin Zhong, Yuchen Su, and Michael Witbrock

University of Auckland, Auckland, New Zealand
scho963@aucklanduni.ac.nz

**Abstract.** Mitigating elderly loneliness requires policy interventions that achieve both adaptability and auditability. Existing methods struggle to reconcile these objectives: traditional agent-based models suffer from static rigidity, while direct large language model (LLM) controllers lack essential traceability. This work proposes a three-layer framework that separates diagnosis from control to achieve both properties simultaneously. LLMs operate strictly as diagnostic instruments that assess population state and generate structured risk evaluations, while deterministic formulas with explicit bounds translate these assessments into traceable parameter updates. This separation ensures that every policy decision can be attributed to inspectable rules while maintaining adaptive response to emergent needs. We validate the framework through systematic ablation across five experimental conditions in an elderly-care simulation and through additional robustness checks across five LLM configurations. Under the primary LLM configuration, the closed-loop controller reduces final loneliness by 15.3% relative to baseline and by 11.7% relative to black-box LLM control. Across all tested LLM configurations, deterministic closed-loop control achieves lower final loneliness than direct black-box LLM parameter control while preserving explicit control-layer auditability.

**Keywords:** Agent-based modeling · Large language models · Policy adaptation · Auditable AI · Social simulation

## 1 Introduction

The mitigation of elderly loneliness constitutes a primary public health objective in aging societies, given its association with increased mortality risk and cognitive decline [20,15]. Effective policy interventions must be simultaneously *adaptive*, responding to evolving population needs, and *auditable*, offering transparent decision pathways for stakeholders.

Current approaches face a persistent trade-off between adaptivity and auditability. In policy domains such as elderly care, auditability is not merely desirable but often mandatory: stakeholders must understand why decisions were

made to ensure accountability and maintain trust [17]. While Agent-Based Models (ABMs) effectively capture social dynamics and facilitate theory-driven simulation, policy interventions within them are often operationalized as pre-specified scenario rules or parameterized levers evaluated through comparative simulation rather than continuously re-optimized online [14,5]. Conversely, reinforcement learning enables policy adaptation through feedback-driven optimization, but many high-performing RL implementations rely on complex function approximators whose decision logic is difficult to directly inspect, creating an auditability gap in high-stakes policy settings [12].

Large Language Models (LLMs) present an opportunity to bridge this gap through their capacity for reasoning and structured assessment. However, integrating LLMs into policy adaptation requires determining their optimal role. Should they function as direct controllers or as diagnostic support within a control framework? Existing LLM-augmented simulations predominantly adopt end-to-end architectures where LLMs act as direct decision-makers [16,3], a design that retains the opacity of black-box systems. Although recent closed-loop frameworks have investigated structured feedback [18,19], the specific requirement for auditability in policy contexts remains under-addressed.

This paper introduces a framework that separates diagnosis from control by positioning LLMs as diagnostic instruments rather than direct policy actuators. Implemented via a three-layer architecture (Figure 1), the system ensures every decision remains traceable to explicit, inspectable rules. We validate this design through simulation of an elderly care facility with 30 agents over 200 days, comparing five experimental conditions including fixed policies and black-box LLM baselines.

Overall, our key contributions are as follows:

- **Architecture.** We present a three-layer framework that achieves adaptive policy adjustment while maintaining control-layer auditability by separating LLM diagnosis from deterministic control.
- **Design principles.** We validate three principles through systematic ablation: separation of diagnosis and control, explicit bounded update rules, and integration of theory-grounded simulation with data-driven diagnosis.
- **Empirical validation.** Results on holdout seeds demonstrate 15.3% improvement over baseline and 11.7% over black-box LLM control under the primary LLM configuration. Additional model-robustness and output-validity audits show that the same diagnosis-control separation remains effective and inspectable across the tested local LLM families.

The remainder of this paper reviews related work (Section 2), details our methodology (Section 3), presents experimental results (Section 4), and discusses implications (Section 5) and limitations (Section 6).

## 2 Related Work

### 2.1 Agent-Based Models and LLM-Augmented Simulation

Agent-based models effectively capture social dynamics and facilitate theory-driven simulation [4]. In elderly care contexts, ABMs model loneliness dynamics, social network evolution, and intervention effects. The core advantage is interpretability: agent rules explicitly encode modeller assumptions, enabling stakeholders to understand decision rationales. Traditional ABM interventions, however, are static. Parameters remain fixed throughout simulation runs.

Recent work has integrated LLMs into agent-based simulations through two approaches. The first uses LLMs to generate agent behaviors directly. Generative agents [16], social interaction models [7], and evolutionary simulations [8] demonstrate that LLM-backed agents can exhibit emergent social dynamics. The second approach uses LLMs for higher-level reasoning. Land-system models [22] and decision-making frameworks [9] position LLMs as institutional agents that interpret complex scenarios and generate strategic responses. Comprehensive surveys [3] map this rapidly expanding intersection, identifying persistent challenges in evaluation methodology and behavioral fidelity.

Critical work questions whether LLM agents reliably replicate human patterns [11]. The concern is twofold. First, generative flexibility does not guarantee faithful simulation of social dynamics. Second, evaluation frameworks remain underdeveloped, making it difficult to validate LLM agent behavior against empirical data. Alternative approaches using structured knowledge representations [13] guarantee auditability through explicit schemas but sacrifice the contextual flexibility needed to interpret emergent social signals.

Closed-loop architectures feed simulation state back into LLM decision processes. In pandemic control [18], traffic management [19], and optimal control [2], iterative refinement enables adaptive policy. These frameworks demonstrate technical viability. But they do not address auditability requirements central to social policy contexts. When LLMs directly control parameters, decisions cannot be traced to explicit rules.

### 2.2 Positioning This Work

The persistent challenge is reconciling adaptability with transparency. LLM-driven control adapts to emergent states but produces opaque decisions. Control-theoretic methods provide transparent optimization but lack semantic reasoning over complex agent states. No prior work separates diagnosis from control in this way. Our framework separates LLM-based diagnosis from deterministic control: LLMs serve strictly as diagnostic instruments, while explicit, bounded formulas translate their assessments into parameter updates.

## 3 Methodology

The framework addresses a core challenge in social policy adaptation: maintaining auditability while enabling responsive intervention adjustment. We achieve

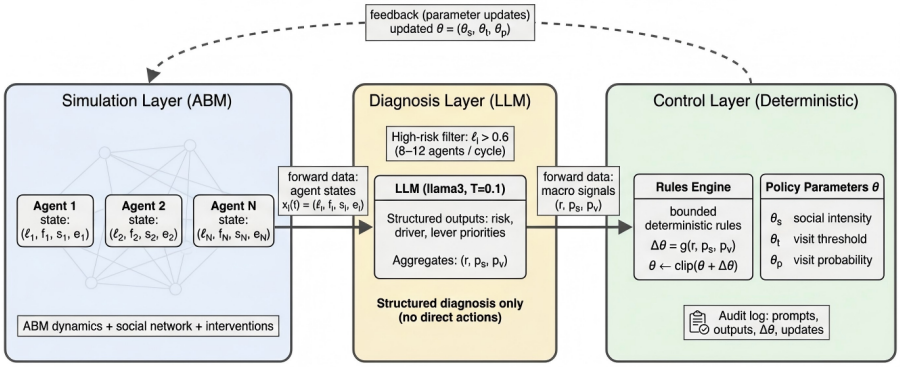


**Fig. 1.** Three-layer architecture for auditable policy adaptation. The simulation layer models agent dynamics, the diagnosis layer uses LLM-based diagnostics for risk assessment, and the control layer applies deterministic formulas. Solid arrows indicate data flow; the dashed arrow indicates the policy feedback loop.

this through a three-layer architecture (Figure 1) that strictly separates semantic diagnosis (where LLMs operate) from policy control (where transparency is enforced). The approach is validated through simulation of an elderly care facility with $N = 30$ agents over $T = 200$ days, where interventions must adapt to evolving loneliness patterns while remaining traceable to explicit decision rules.

### 3.1 Simulation Environment and Problem Formulation

We model an elderly care facility where each agent $i$ is characterized by loneliness $\ell_i(t) \in [0, 1]$, frailty $f_i(t)$, stress $s_i(t)$, and energy $e_i(t)$. All agents and trajectories are synthetic; no real resident records are used in the reported experiments. Agents are embedded in a dynamic social network $G(t)$ where edges represent social ties. Network evolution uses an opportunity-constrained similarity rule: potential ties depend on exposure within the simulated facility and existing network degree, with similarity in loneliness state modulating the probability of forming or maintaining a tie.

Two intervention types are available. Social events controlled by intensity $\theta_s \in [0.8, 1.5]$ reduce loneliness for all participants proportionally. Home visits provide targeted support for eligible agents (those with $\ell_i > \theta_t$), reducing both loneliness and stress with probability $\theta_p \in [0.15, 0.5]$. The eligibility threshold $\theta_t \in [0.4, 0.6]$ determines which agents receive visits. These policy levers are deliberately bounded because staff time, visit availability, resident burden, and intervention budgets are finite in realistic care settings. The objective combines adaptive reduction of population-level loneliness with bounded intervention intensity and full traceability of policy decisions.

### 3.2 Three-Layer Architecture for Auditable Adaptation

The framework partitions responsibilities across three layers, each with a distinct role in the adaptation process. This separation is motivated by the requirement that policy decisions must be both responsive to emergent population needs and traceable to explicit rules that stakeholders can verify.

**Simulation Layer: Theory-Grounded Agent Dynamics.** An agent-based model implements dynamics grounded in social science theories. Loneliness evolves toward individual baseline with rate $\alpha_\ell$ and decreases with social interactions at rate $\beta_\ell$. Frailty accumulates gradually with age and stress. Network formation uses the same opportunity-constrained similarity rule: potential ties depend on existing network degree and are modulated by similarity in loneliness state once agents are exposed within the simulated facility environment. Social events reduce loneliness for all participants, with effectiveness proportional to $\theta_s$. Home visits provide targeted support for eligible agents, reducing both loneliness and stress with probability $\theta_p$. This layer operates deterministically given fixed parameters, ensuring that variance across experimental runs stems only from network initialization.

**Diagnosis Layer: LLM-Based Risk Assessment.** Every 7 simulation days, an LLM assesses high-risk agents to inform control decisions. To manage computational costs, only agents with loneliness exceeding 0.6 are diagnosed (typically 8 to 12 agents per cycle). For each high-risk agent, a structured prompt is constructed including current state, recent interaction history, and network position. The primary configuration uses a locally served Llama3-8B model with temperature 0.1. All LLM-based calls use the same maximum generation budget of 500 tokens. The model produces structured JSON outputs containing risk level, primary driver identification, and intervention priorities for each policy lever. Outputs are parsed, schema-validated, clipped to admissible ranges, and replaced by deterministic fallback values if validation fails. The use of local LLM deployment improves reproducibility and avoids external API dependencies.

Individual assessments are aggregated to population-level statistics. The high-risk proportion $r$ represents the fraction of agents with assessed risk exceeding 0.6. Mean priority scores for social events ($p_s$) and home visits ($p_v$) are computed by averaging over diagnosed agents $H$:

$$p_s = \frac{1}{|H|} \sum_{i \in H} \text{priority}_i^{\text{social}}, \quad p_v = \frac{1}{|H|} \sum_{i \in H} \text{priority}_i^{\text{visits}} \tag{1}$$

This aggregation provides a privacy-preserving interface to the control layer. Individual diagnoses are not directly visible to control formulas, ensuring that policy decisions operate on population-level signals rather than individual-level data.

**Control Layer: Deterministic Parameter Updates.** Deterministic formulas convert population-level statistics into policy parameter updates. These update rules follow the logic of stepped and adaptive care: intervention intensity is adjusted using explicit decision rules at predefined stages based on monitored

risk indicators and their cutoffs [1,21]. Such cutoff-based rules are widely used to support reproducible, auditable escalation while balancing resource constraints and the risks of over- or under-treatment [21]. In our simulation, we instantiate this principle with the thresholds $r > 0.4$ (substantial high-risk prevalence) and $p_s > 0.75$ (high priority on a normalized $[0, 1]$ scale), which we treat as modeling choices. These values were selected through pilot experiments on train seeds to balance responsiveness and stability; their robustness is further validated via sensitivity analysis in Appendix A. Update magnitudes are deliberately small, reflecting incremental adjustment:

$$\Delta\theta_s = \begin{cases} \min(0.05,\ 0.1 \cdot p_s) & \text{if } r > 0.40 \text{ and } p_s > 0.75 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\Delta\theta_t = \begin{cases} -0.02 & \text{if } p_v > 0.75 \text{ and } \theta_t > 0.4 \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta\theta_p = \begin{cases} +0.05 & \text{if } p_v > 0.75 \text{ and } \theta_p < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

To promote stability, we bound each parameter update ($\|\Delta\theta\| \leq 0.05$) as a conservative step-size control mechanism, implementing gradual escalation instead of abrupt changes. This is consistent with adaptive-intervention practice that emphasizes staged, monitored adjustments [1] and with evidence that changes in stimuli and daily activity patterns in nursing-home environments can affect resident outcomes, motivating cautious, incremental modifications [10]. After computing updates, parameters are clipped to remain within valid ranges. The separation of diagnosis (where neural computation occurs) from control (where deterministic rules operate) ensures that every parameter change can be attributed to specific macro statistics and verified against explicit criteria.

## 4 Simulation Experiments

### 4.1 Implementation and Experimental Setup

To evaluate the proposed three-layer framework, we developed a simulation environment based on the theoretical dynamics described in Section 3.

We implemented the system in Python 3.8+ using NumPy for numerical computation and NetworkX for graph operations. Agent states update in discrete daily timesteps. Network edges are stored as adjacency lists and updated incrementally when opportunity-constrained attachment rules trigger new tie formation. We run the core simulation logic on the CPU, while offloading LLM inference tasks to the GPU for efficiency.

For diagnosis, the primary configuration uses a locally served Llama3-8B model. We construct structured prompts programmatically using JSON templates that inject agent state variables, interaction counts from the past 7 days, and network degree statistics. Responses are parsed using Python's `json` module

**Table 1.** Experimental conditions.

| Condition | LLM Diagnosis | Adaptation | Control Type |
|---|---|---|---|
| Baseline | – | – | None |
| Fixed Policy | – | – | Static $\theta$ |
| LLM Mapping | ✓ | – | Fixed response |
| Closed-loop (Ours) | ✓ | ✓ | Deterministic |
| Black-box LLM | ✓ | ✓ | LLM decides $\theta$ |

and validated against a predefined schema. We set the temperature to 0.1 to reduce stochastic variation in JSON-format outputs while preserving limited structured reasoning. All LLM-based conditions use the same maximum generation budget of 500 tokens per call. Holdout seeds refer to random initializations not used during pilot tuning or rule selection. Additional robustness runs evaluate Qwen3-8B, Qwen3-14B, Mistral-Small-24B, and DeepSeek-32B using the same prompt structure and decoding budget. For robustness comparisons, schema-valid diagnosis outputs are cached and replayed through the same deterministic controller after mapping model-specific surface forms, such as capitalization or synonymous risk labels, to the fixed schema fields. Caching keeps the control stage reproducible across model comparisons.

The control layer functions as a deterministic controller. It accepts aggregated statistics ($r$, $p_s$, $p_v$) and current parameters ($\theta_s$, $\theta_t$, $\theta_p$), then returns updates according to Equations 2–3. This logic uses explicit conditional branches with no learned components, ensuring reproducibility and direct inspection of the decision logic.

We initialized the environment with a population of $N = 30$ agents over $T = 200$ days. Each run takes approximately 15 to 20 minutes, dominated by simulation stepping and diagnosis cycles. To ensure consistent network structures, we used fixed random seeds for all experiments. We used seeds 42, 100, and 200 for development, while the reported results are based on four holdout seeds (300, 400, 500, 600) to validate performance on unseen initializations.

Baseline runs with no intervention. Fixed Policy applies static parameters ($\theta_s = 1.0$, $\theta_t = 0.6$, $\theta_p = 0.3$) throughout. LLM Mapping activates diagnosis but uses a fixed switching rule: social intensity is 1.2 if high-risk proportion exceeds 0.4, otherwise 1.0. This isolates semantic assessment without adaptive updates. Closed-loop implements the full framework (Equations 2–3). Black-box LLM prompts the LLM to directly suggest parameter values given macro statistics ($r$, $p_s$, $p_v$). Parameter bounds are enforced, but decision logic is not specified by explicit rules.

Table 2 shows holdout performance across all conditions under the primary LLM configuration. Closed-loop achieves the lowest mean loneliness (0.607), corresponding to a 15.3% reduction over baseline (0.717). Fixed Policy reduces loneliness by 6.0%, while LLM Mapping and Black-box LLM achieve 5.2% and 4.2% reductions respectively. Pairwise comparisons are reported in Table 3. Be-

**Table 2.** Main holdout performance across four seeds.

| Method | Mean | SD | 95% CI | Reduction vs. Baseline |
|---|---|---|---|---|
| Baseline | 0.717 | 0.018 | [0.699, 0.734] | – |
| Fixed Policy | 0.674 | 0.012 | [0.662, 0.686] | 6.0% |
| LLM Mapping | 0.680 | 0.007 | [0.673, 0.687] | 5.2% |
| **Closed-loop (Ours)** | **0.607** | **0.020** | **[0.587, 0.626]** | **15.3%** |
| Black-box LLM | 0.687 | 0.025 | [0.663, 0.712] | 4.2% |

**Table 3.** Pairwise comparisons on the primary holdout setting. These comparisons are descriptive because each condition uses four holdout seeds.

| Comparison | $\Delta$ | % Change | Cohen's $d$ | $p$-value | Sig. |
|---|---|---|---|---|---|
| Closed-loop vs. LLM Mapping | −0.073 | +10.7% | −4.871 | 0.0005 | *** |
| Closed-loop vs. Black-box | −0.081 | +11.7% | −3.539 | 0.0024 | ** |
| Closed-loop vs. Fixed | −0.067 | +10.0% | −4.053 | 0.0012 | ** |
| Black-box vs. Fixed | +0.013 | −2.0% | +0.663 | 0.3848 | n.s. |

cause each condition uses four holdout initializations, these tests are reported as descriptive evidence for this controlled simulation setting.

Loneliness trajectories over 200 days (Figure 2) show how conditions diverge over time.

Parameter trajectories (Figure 3) reveal how the deterministic controller adapts. Social event intensity ($\theta_s$) remains at 1.0 throughout all runs; population-level social priority never exceeds the required threshold. In contrast, home visit parameters adapt systematically. Across all seeds, $\theta_t$ decreases from 0.6 to 0.4 and $\theta_p$ increases from 0.3 to 0.5. The controller reaches the upper bound for $\theta_p$ (0.50) by Day 134 and the lower bound for $\theta_t$ (0.40) by Day 176. Black-box LLM shows higher cross-seed variance and less systematic progression. Parameters oscillate rather than converge monotonically to bounds.

To test whether the result depends on a single diagnostic model, we reran the LLM-dependent conditions with four additional local models. Baseline and Fixed Policy are model-independent, and LLM Mapping is fixed by construction in this implementation, so the robustness comparison focuses on Closed-loop and Black-box LLM control. As shown in Table 4, Closed-loop outperforms black-box control across all five tested LLM configurations, with gains ranging from 9.4% to 14.1%. This comparison tests whether the diagnosis-control separation remains effective when the diagnostic model changes. The controller remains deterministic and auditable, while the LLM family affects the diagnostic signal but not the transparency of policy actuation.

We also audit the validity of LLM outputs. Table 5 reports that all diagnosis calls were schema-valid, no fallback was required, and no truncation occurred under the 500-token generation budget. The black-box controller outputs were also valid and within admissible parameter bounds in all cycles. These checks

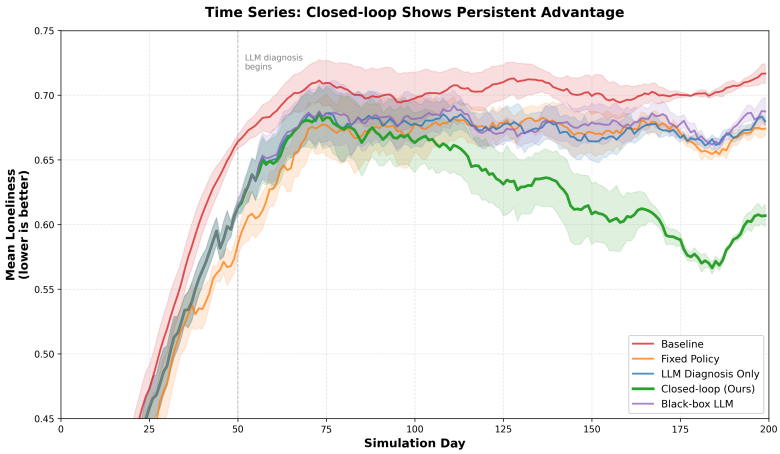


**Fig. 2.** Loneliness trajectories across experimental conditions over 200 days ($N = 30$). Shaded regions indicate $\pm 1$ SD across four holdout seeds. The vertical dashed line marks the start of LLM diagnosis.

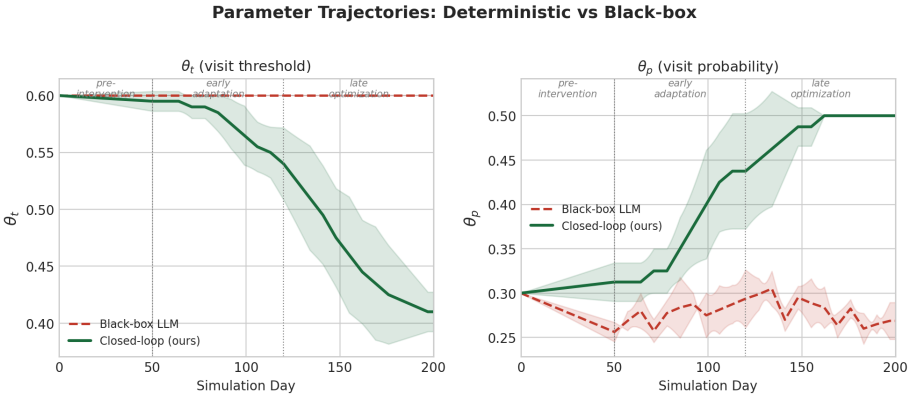


**Fig. 3.** Parameter trajectories under the primary LLM configuration. Shaded regions indicate $\pm 1$ SD across four holdout seeds. (Left) Visit threshold $\theta_t$. (Right) Visit probability $\theta_p$.

show that the architecture uses LLM outputs as constrained structured signals before they enter the control layer. Outputs are accepted only after parsing and validation, then passed to the deterministic controller through bounded aggregate statistics.

Table 6 summarizes the resource implications of the main policies. The saturated Max Policy is included as a resource-heavy upper bound: it obtains the lowest final loneliness but uses far more visits than the bounded closed-loop controller. This comparison frames the task as bounded adaptive policy selection

**Table 4.** Multi-model robustness across LLM-dependent conditions. Values are mean final loneliness ± SD across four holdout seeds; lower is better.

| LLM | Closed-loop | Black-box LLM | Closed-loop gain over Black-box |
|---|---|---|---|
| Llama3-8B | 0.607 ± 0.020 | 0.687 ± 0.025 | 11.7% |
| Qwen3-8B | 0.586 ± 0.012 | 0.682 ± 0.021 | 14.1% |
| Qwen3-14B | 0.602 ± 0.022 | 0.664 ± 0.021 | 9.4% |
| Mistral-Small-24B | 0.608 ± 0.019 | 0.674 ± 0.032 | 9.8% |
| DeepSeek-32B | 0.602 ± 0.017 | 0.671 ± 0.013 | 10.2% |

**Table 5.** LLM output validity audit. Diagnosis calls count individual agent-level diagnosis calls. Black-box counts are valid and bounded control cycles. Llama3-8B includes the primary five-condition experiments; additional models are evaluated on the LLM-dependent robustness conditions.

| LLM | Diag. calls | Schema valid | Fallback | Trunc. | Black-box valid/bounded | Tokens | Latency (s) |
|---|---|---|---|---|---|---|---|
| Llama3-8B | 13072 | 100.0% | 0.0% | 0.0% | 88/88 | 19.5 | 1.34 |
| Qwen3-8B | 7474 | 100.0% | 0.0% | 0.0% | 88/88 | 20.2 | 1.32 |
| Qwen3-14B | 7488 | 100.0% | 0.0% | 0.0% | 88/88 | 20.0 | 2.27 |
| Mistral-Small-24B | 7460 | 100.0% | 0.0% | 0.0% | 88/88 | 17.5 | 2.67 |
| DeepSeek-32B | 7488 | 100.0% | 0.0% | 0.0% | 88/88 | 18.9 | 3.61 |

under explicit policy levers, with the saturated policy serving as a resource-heavy reference point.

## 5 Discussion

The primary closed-loop configuration achieves a 15.3% loneliness reduction over baseline and an 11.7% reduction relative to black-box LLM control. The multi-model runs strengthen this result: across all five tested LLM configurations, closed-loop control achieves lower final loneliness than direct black-box LLM parameter control. Together, these results support the paper's central design principle: LLMs can be used for bounded, structured diagnosis while policy actuation remains explicit, deterministic, and inspectable.

Two mechanisms explain the main gap between deterministic control and black-box LLM control. First, bounded updates prevent instability. The deterministic controller (Equations 2–3) caps each parameter change at 0.05 per update cycle. The system then responds over 7 days before the next update. Black-box LLM control receives the same admissible parameter ranges, but its decision logic is not fixed by explicit update rules. This produces less systematic parameter movement and occasional reversals. Parameter trajectories (Figure 3) visualize the difference: closed-loop visit parameters progress gradually toward interpretable bounds, while black-box control oscillates.

Second, explicit rules encode domain constraints reproducibly. Equation 2 implements a conjunction: increase social intensity only if $r > 0.4$ and $p_s > 0.75$. Given identical aggregate inputs, the controller always produces identical

**Table 6.** Intervention usage and resource summary. Values are means across four holdout seeds. Max Policy is a resource-heavy upper bound rather than a deployable policy baseline.

| Method | Final loneliness | Total visits | Mean visit prob. | LLM calls | Social events | Notes |
|---|---|---|---|---|---|---|
| Fixed Policy | 0.674 ± 0.012 | 795.5 | 0.300 | 0.0 | 47.5 | static |
| LLM Mapping | 0.680 ± 0.007 | 820.0 | 0.300 | 624.2 | 26.2 | diagnosis only; fixed response |
| Closed-loop | 0.607 ± 0.020 | 1342.8 | 0.393 | 615.8 | 32.8 | bounded adaptive |
| Black-box LLM | 0.687 ± 0.025 | 786.5 | 0.284 | 621.0 | 26.0 | direct LLM parameter control |
| Max Policy | 0.595 ± 0.029 | 2014.5 | 0.500 | 0.0 | 47.0 | resource-heavy upper bound |

outputs. By contrast, black-box control must reconstruct the policy logic from a natural-language prompt at each cycle. In policy contexts where decisions must be reproducible and auditable, this consistency is part of the objective rather than merely an implementation detail.

The intervention-usage analysis in Table 6 highlights the resource profile of each policy. A saturated Max Policy achieves the lowest final loneliness while using substantially more visits than the closed-loop controller, making it a resource-heavy upper bound. The closed-loop controller provides bounded adaptive escalation: it reduces loneliness relative to the main baselines while keeping every adjustment traceable to explicit statistics and rules.

Finally, the validity audit (Table 5) shows that the LLM components operate through structured, schema-validated outputs. Every output is parsed, bounded, and aggregated before it can affect the controller. This design keeps the LLM's role at the diagnostic level and reserves policy actuation for explicit rules.

## 6 Limitations and Future Directions

The experiments use a small population ($N = 30$) with simplified loneliness dynamics. Real facilities house 50 to 200 residents, and loneliness is affected by factors not modeled here: family contact frequency, acute health events, and staffing constraints. Direct validation against real facility data was not feasible due to privacy restrictions. Testing larger populations and calibrating dynamics against anonymized measurements would strengthen external validity.

We evaluate five local LLM configurations, which is broader than a single-model case study but still limited. The closed-loop controller outperforms black-box control across these tested configurations, but larger proprietary models, smaller edge-deployable models, and alternative prompting strategies may yield different diagnosis quality and black-box control behavior. The core design choice—using LLMs for diagnosis while keeping control explicit—does not depend on one model, but broader validation would strengthen generality.

The update rules in Equations 2–3 are hand-specified. Future work could learn rule parameters from data while preserving interpretability, for example via symbolic regression constrained to simple functions [2] or rule learning from historical intervention data. This would reduce manual specification while maintaining auditability.

We report intervention usage (LLM calls, visit counts, and social-event counts) but do not optimize cost directly. Practical deployment requires balancing loneliness reduction against staffing, budget, and resident-burden constraints. Extending the framework to explicit multi-objective optimization is important for real-world adoption, particularly in settings where resource limitations are strict.

Although the case study focuses on elderly care, the design principles apply to other policy domains requiring both adaptation and transparency. Healthcare resource allocation, educational interventions, and urban planning all face similar trade-offs between flexible response and stakeholder accountability. Testing the architecture in these domains would validate generalizability and identify domain-specific challenges.

## 7 Conclusion

We presented an auditable closed-loop framework for policy adaptation in agent-based simulations. The key design choice is positioning LLMs as diagnostic tools and implementing parameter adaptation through explicit deterministic rules.

The ablation study yields three main findings. First, under the primary LLM configuration, the closed-loop condition achieves 15.3% reduction in final loneliness relative to baseline and 11.7% relative to black-box LLM control. Second, deterministic control improves stability and decision consistency by translating diagnostic signals into bounded parameter updates rather than allowing the LLM to directly choose policy parameters. Third, multi-model results show that this advantage over black-box control holds across all five tested local LLM configurations while preserving control-layer auditability. Diagnosis alone offers little benefit when not coupled with adaptive control: LLM Mapping performs no better than fixed policies, indicating that diagnostic signals must be translated into structured parameter updates to produce operational gains.

Overall, the results support a practical principle for high-stakes applications: LLMs are better suited to structured assessment, while policy control should remain explicit and inspectable. This separation offers a viable path to deploy adaptive AI components in policy domains where accountability is non-negotiable.

## A Parameter Sensitivity Analysis

Outcomes are insensitive to $r$ and cap in the tested ranges; only $p = 0.85$ shifts results (+6.4%), so $p = 0.75$ is retained as the default priority threshold. The sensitivity analysis serves as a robustness check for the transparent modeling choices used by the controller.

## B LLM Diagnosis Examples

The primary Llama3-8B configuration ($T = 0.1$, maximum generation budget 500 tokens) assesses agents every 7 days, outputting JSON risk labels and inter-

**Table 7.** Sensitivity on holdout seeds 300 and 400 (default: $r = 0.4$, $p = 0.75$, cap= 0.05). These values are transparent modeling choices used by the deterministic controller.

| Parameter | Value | Mean | SD | Min | Max | $\Delta$% |
|---|---|---|---|---|---|---|
| **Default** | – | **0.636** | 0.016 | 0.623 | 0.648 | – |
| Risk threshold ($r$) | 0.30 | 0.636 | 0.016 | 0.623 | 0.648 | +0.0% |
| | 0.50 | 0.636 | 0.016 | 0.623 | 0.648 | +0.0% |
| Priority ($p$) | 0.65 | 0.636 | 0.016 | 0.623 | 0.648 | +0.0% |
| | 0.85 | 0.680 | 0.007 | 0.677 | 0.682 | +6.4% |
| Update cap ($\Delta\theta$) | 0.03 | 0.636 | 0.016 | 0.623 | 0.648 | +0.0% |
| | 0.08 | 0.636 | 0.016 | 0.623 | 0.648 | +0.0% |

vention priorities. The following examples are schema-valid and did not require fallback.

**Table 8.** Example diagnosis outputs (Day 120, Seed 300; 30 agents diagnosed).

| Agent | Risk (L) | Risk (F) | Primary Driver | Priority |
|---|---|---|---|---|
| Agent 8 | High | Low | Social isolation | Network: 0.80 |
| Agent 13 | High | Medium | Health & isolation | Network: 0.80 |
| Agent 0 | Medium | Low | Minor mood dip | Visit: 0.70 |
| *7/30 classified as High risk ($r = 0.23$)* | | | | |

Diagnosis outputs feed into Eqs. 2–3; here $r = 0.23 < 0.4$ (Appendix A), so social intensity remains unchanged.

## C Prompt Templates, Output Schema, and Fallback Rules

The diagnosis prompt is generated from a fixed template. The template inserts the current agent state, recent interaction counts, and network position, and asks the model to return only JSON.

```
You are a diagnostic module for an elderly-care simulation.
Assess the following agent state and return only valid JSON.
Agent state: {loneliness, frailty, stress, energy}
Recent interactions: {interaction_count_7d, social_event_count_7d}
Network position: {degree, relative_degree, isolation_flag}
Required fields:
  risk_loneliness: low|medium|high
```

```
  risk_frailty: low|medium|high
  primary_driver: short string
  priority_social: number in [0,1]
  priority_visit: number in [0,1]
  confidence: number in [0,1]
```

The black-box control prompt receives only aggregate statistics and asks the model to suggest bounded policy parameters directly.

```
You control policy parameters in an elderly-care simulation.
Given aggregate statistics {r, p_s, p_v} and current parameters
{theta_s, theta_t, theta_p}, return only valid JSON with:
  theta_s: number in [0.8,1.5]
  theta_t: number in [0.4,0.6]
  theta_p: number in [0.15,0.5]
Do not include any additional text.
```

All outputs are parsed with Python's `json` module and validated against the required fields and numeric bounds. Missing fields, invalid JSON, or out-of-range values trigger a deterministic fallback: diagnosis priorities are replaced by neutral values and black-box parameter suggestions are replaced by the previous valid parameters. In the reported runs, no diagnosis fallback or truncation was required, and all black-box outputs were valid and bounded (Table 5).

## References


1. Almirall, D., Nahum-Shani, I., Sherwood, N.E., Murphy, S.A.: Introduction to SMART designs for the development of adaptive interventions: with application to weight loss research. Transl. Behav. Med. **4**(3), 260–274 (2014). https://doi.org/10.1007/s13142-014-0265-0
2. Fonseca, L.L., Böttcher, L., Mehrad, B., Laubenbacher, R.C.: Optimal control of agent-based models via surrogate modeling. PLOS Comput. Biol. **21**(1), e1012138 (2025). https://doi.org/10.1371/journal.pcbi.1012138
3. Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., Li, Y.: Large language models empowered agent-based modeling and simulation: a survey and perspectives. Humanit. Soc. Sci. Commun. **11**, 1259 (2024). https://doi.org/10.1057/s41599-024-03611-3
4. Grimm, V., et al.: The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. J. Artif. Soc. Social Simul. **23**(2), 7 (2020). https://doi.org/10.18564/jasss.4259
5. Hammond, R.A.: Considerations and best practices in agent-based modeling to inform policy. In: Wallace, R., et al. (eds.) Assessing the Use of Agent-Based Models for Tobacco Regulation. National Academies Press, Washington (DC) (2015). https://www.ncbi.nlm.nih.gov/books/NBK305917/
6. Heuillet, A., Couthouis, F., Díaz-Rodríguez, N.: Explainability in deep reinforcement learning. arXiv:2008.06693 (2020)

7. Holt, S., Ruiz Luyten, M., Berthon, A., van der Schaar, M.: G-Sim: Generative simulations with large language models and gradient-free calibration. arXiv:2506.09272 (2025)
8. Jiang, Z., Shi, Y., Li, M., Xiao, H., Qin, Y., Wei, Q., Wang, Y., Zhang, Y.: Casevo: A cognitive agents and social evolution simulator. arXiv:2412.19498 (2024)
9. Kleiman, J., Frank, K., Voyles, J., Campagna, S.: Simulation agent: A framework for integrating simulation and large language models for enhanced decision-making. arXiv:2505.13761 (2025)
10. Knippenberg, I.A.H., Leontjevas, R., Nijsten, J.M.H., Bakker, C., Koopmans, R.T.C.M., Gerritsen, D.L.: Stimuli changes and challenging behavior in nursing homes during the COVID-19 pandemic. BMC Geriatr. **22**(1), 142 (2022). https://doi.org/10.1186/s12877-022-02824-y
11. Larooij, M., Törnberg, P.: Validation is the central challenge for generative social simulation: a critical review of LLMs in agent-based modeling. Artif. Intell. Rev. **59**, 15 (2026). https://doi.org/10.1007/s10462-025-11412-6
12. Milani, S., Topin, N., Veloso, M., Fang, F.: Explainable reinforcement learning: A survey and comparative review. ACM Comput. Surv. **56**(7), 168 (2024). https://doi.org/10.1145/3616864
13. Muller, G., Maneerat, S., Adamiak, A.: KEMASS: Knowledge-enhanced multi-agent simulation for energy scheduling support. In: Thompson, J., Stankov, I. (eds.) Multi-Agent-Based Simulation XXV (MABS 2024). LNCS, vol. 15583, Springer, Cham (2025). https://doi.org/10.1007/978-3-031-88017-9_5
14. Nespeca, V., Comes, T., Brazier, F.: A methodology to develop agent-based models for policy support via qualitative inquiry. J. Artif. Soc. Social Simul. **26**(1), 10 (2023). https://doi.org/10.18564/jasss.5014
15. OECD: Social connections and loneliness in OECD countries. OECD Publishing, Paris (2025). https://doi.org/10.1787/6df2d6a0-en
16. Park, J.S., O'Brien, J.C., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative agents: Interactive simulacra of human behavior. arXiv:2304.03442 (2023)
17. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. **1**(5), 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x
18. Shi, Z., Guo, X., Lu, H., Peng, M., Wang, H., Zhu, Z., Li, Z., Liang, Y., Zheng, X., Yang, H.: Coordinated pandemic control with large language model agents as policymaking assistants. arXiv:2601.09264 (2026)
19. Tan, S., Ivanovic, B., Chen, Y., Li, B., Weng, X., Cao, Y., Krähenbühl, P., Pavone, M.: Promptable closed-loop traffic simulation. arXiv:2409.05863 (2024)
20. WHO: From loneliness to social connection: Charting a path to healthier societies. Report of the WHO Commission on Social Connection, Geneva (2025)
21. Wilhelm, M., Moessner, M., Jost, S., Okon, E., Malinowski, V., Schinke, K., Sommerfeld, S., Bauer, S.: Development of decision rules for an adaptive aftercare intervention based on individual symptom courses for agoraphobia patients. Sci. Rep. **14**(1), 3056 (2024). https://doi.org/10.1038/s41598-024-52803-z
22. Zeng, Y., Brown, C., Raymond, J., Byari, M., Hotz, R., Rounsevell, M.: Exploring the opportunities and challenges of using large language models to represent institutional agency in land system modelling. EGUsphere (2024). https://doi.org/10.5194/egusphere-2024-449