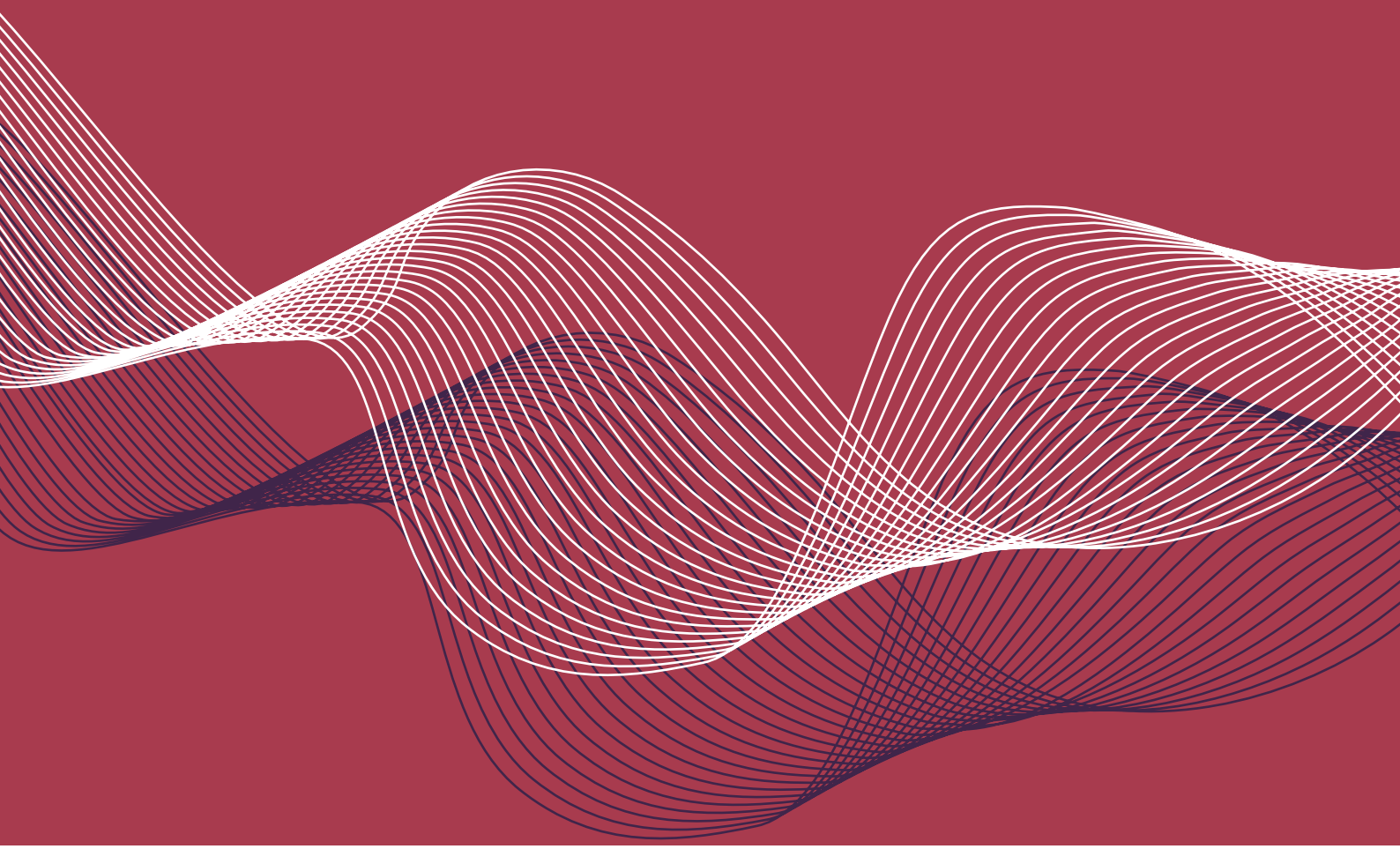
United Nations University
Institute in Macau

TECHNOLOGY REPORT

From Plausible Agents to Accountable Simulation

A Technical and Governance Framework for LLM-Enabled Agent-Based Modelling

Jia'an LIU



UNU Macau

macau.unu.edu

Author biography

Jia'an LIU is a computer scientist and AI research assistant at the United Nations University Institute in Macau. His research lies at the intersection of agentic AI, computational social science, and AI for sustainable development, with particular expertise in large language model-enabled agent-based modelling, multi-agent systems, urban and social simulation, and policy-oriented digital twin frameworks. His work develops spatially grounded, LLM-enhanced simulation systems to study complex societal challenges such as urban resilience, climate adaptation, healthcare accessibility, misinformation diffusion, inclusive education, and digital inclusion. His broader technical background spans machine learning, computer vision, remote sensing, and uncertainty-aware video motion analysis, alongside the design of AI workflows, intelligent knowledge pipelines, and research infrastructure for policy and scientific applications. His recent research also examines AI open-source readiness, and the evaluation of large language models for social policymaking, bridging technical innovation with questions of governance, openness, and public value.

Acknowledgements

The author thanks colleagues at UNU Macau for discussion of earlier versions and the maintainers of the open-source agent systems whose codebases informed the comparative analysis in this report. Any errors are the author's own.

Disclaimer

The views and opinions expressed in this technology report do not necessarily reflect the official policies or positions of the United Nations University.

Disclosure of AI use

Generative AI tools were used to assist with literature retrieval, source verification, code-base review, structural checking, textual refinement and figure rendering. All substantive claims were verified by the author against primary sources and source code, and final editorial responsibility rests with the author.

Citation

Jia'an LIU, "From Plausible Agents to Accountable Simulation: A Technical and Governance Framework for LLM-Enabled Agent-Based Modelling" UNU Technology Report: 2 (Macau: United Nations University Institute in Macau, 2026).

Layout & Visual Design: Zoey Jizi ZHENG

Table of Contents

Table of Figures

List of Tables

Recommended technical actions

- Document how language-model outputs are constrained, parsed, repaired or replaced, and report fallback rates as a first-class result.
- Disclose the full simulation stack — base model and version, prompts, memory and retrieval rules, scheduler and turn order, sampling temperature, random seeds, and parser and fallback logic — as the unit of documentation and reproducibility analysis, since these choices can materially affect outcomes.
- Define the reproducibility boundary. Fix and record random seeds across controllable components of the stack, record model and version information, and cache model responses for provenance and exact replay. Exact replay of a realised trajectory should be distinguished from independent stochastic replication.
- Validate against out-of-sample empirical patterns at more than one scale, and compare against a simpler non-generative baseline.
- Run robustness audits: vary base model, prompt, seed and scheduler, and report the resulting change in conclusions. Report findings that shift materially under re-specification as configuration-sensitive.

Recommended governance actions

- Require that any social simulation cited in a decision come with a stated purpose, target quantities, empirical anchors and an explicit claim boundary before its outputs are treated as evidence.
- Require participation: populations that a simulation stands in for — particularly marginalised or low-resource communities — should have a role in its design and a channel to contest its outputs.
- Establish disclosure and reproducibility standards for simulation-informed policy analysis, extending the documentation norms already used for traditional agent-based models to the language-model layer.
- Invest in shared, multilingual evaluation infrastructure and in the capacity of public institutions to interrogate simulation results independently.
- Locate accountability at the simulation stack and its operators: a believable output transfers no responsibility to the modelled population.

Executive Summary

Large language models (LLMs) have made agent-based social simulation dramatically more *plausible*. Agents in these systems hold memories, adopt personas, deliberate in natural language, and interact in ways that read as recognisably human.[1] This fluency has arrived at the moment such simulations are being proposed as instruments of public reasoning: emergency preparedness, urban and transport planning, public health, and economic and platform policy.

This report argues that plausibility and validity are being conflated, and that the conflation is consequential. A simulation can be entirely plausible — its agents can speak and act like people — and still be empirically unrealistic; recent evaluations against ground-truth human data confirm that today's systems frequently are.[2] The analysis treats the simulation stack — the parsers, defaults, schedulers and environment machinery — as a major source of observed plausibility and as the unit of validation and accountability.

Beyond the survey and validation literature, the analysis reviews fourteen public LLM-agent projects and nine traditional agent-based modelling platforms, including direct source-code inspection where available. The projects were reviewed in July 2026.

Model outputs enter the simulation through parsing, validation or constrained-action interfaces. Across the runnable systems examined, unusable outputs are handled through repair, retries, defaults or failure handling. In one widely cited macroeconomic simulator, each household agent decides how much labour to supply and how much income to consume; the model's answer is evaluated as a literal Python expression and, if that fails, replaced by the constant "supply all labour, consume half of income."[3]

Macro-level dynamics are jointly generated by LLM-mediated behaviour and the environment machinery that translates behaviour into simulated consequences. In the systems examined, gravity models, recommender systems, resolution loops and inherited economic components strongly structure mobility, congestion, virality and market outcomes. The relative contribution of the language model and the environment therefore has to be established empirically rather than inferred from the apparent autonomy of the agents.

Researcher choices across the stack are rarely disclosed and can materially affect results: base model and version, prompt wording, memory and retrieval weights, scheduling and exposure rules, sampling temperature, and the parser and fallback logic itself. Most of the LLM-driven systems examined leave at least some controllable randomness unseeded, while provider-side model behaviour may also vary across runs. Reproducibility therefore depends on which layers of the stack are controlled, recorded or replayed.

These findings reframe the field's central problem. Validating an agent-based model has been the discipline's core methodological difficulty for fifty years, and language models inherit it intact. By making agents more expressive and more opaque at once, they can make it harder.[4] This report proposes making LLM-enabled simulation *accountable*: specifying, for a given use, what the model is for, which quantities must be validated against which evidence, how robust the conclusions are, and what claims they license. This extends to the language-model layer the validation practices (fitness-for-purpose, pattern-oriented modelling, and transparent documentation) that the agent-based modelling community developed long before LLMs.

For a policy audience the stakes are specific. A simulation that looks scientific can influence decisions about real populations, and the populations most often modelled (residents of a district, users of a platform, recipients of a health intervention) are frequently those least able to audit or contest the model. The report closes with technical and governance recommendations for ensuring that these tools support public decisions without lending them unearned authority.

Introduction

Agent-based modelling (ABM) is a method for studying a system from the bottom up: one specifies the behaviour of individual agents and their interactions, runs the system forward, and observes what aggregate patterns emerge.[5] Its founding results — Schelling's demonstration that mild individual preferences can produce sharp residential segregation, and the Sugarscape models of Epstein and Axtell — are valued because a transparent micro-rule generates a non-obvious macro-pattern.[6] For fifty years the method's persistent difficulty has been *validation*: many different micro-rules can produce the same macro-pattern, so a model that reproduces an observed curve has demonstrated that this mechanism *could* have produced it, not that it did.

Large language models let an agent carry a persona, maintain memory, plan, and explain itself in natural language. The demonstration that crystallised the field, Park and colleagues' *Generative Agents*, populated a small town with twenty-five such agents and evaluated them on their "believability" — whether human observers found their behaviour humanlike.[1] The three years since have produced a rapid expansion of surveys, frameworks and large-scale systems,[7] and, alongside them, a growing critical literature warning that believability is not validity.[4]

Sections 2–3 distinguish the roles language models can play in agent-based modelling and the evidentiary standard associated with each. Section 4 examines the simulation stack through source-code review. Section 5 relates those findings to established ABM validation practice. Sections 6–8 review application evidence, governance implications and recommendations.

1 Scope, method and definitions

Scope. The report concerns the use of large language models to drive or augment agent-based social simulation — the modelling of human populations and their interactions — and the conditions under which the resulting systems can responsibly inform public decisions. It does not treat task-completing or coding agents, which the companion report addresses,[8] except where their lineage clarifies the present subject.

Method. The analysis draws on three sources of evidence. The first is the peer-reviewed and preprint literature on LLM-based social simulation and on the older methodology of agent-based model validation. The second is a set of application and evaluation studies in specific domains. The third is a first-hand review of fourteen public LLM-agent projects and nine traditional ABM platforms, including direct source-code inspection where available.

Definitions. The report uses the following terms precisely:

- An *agent-based model* is a simulation in which individual agents, each with explicit state, follow specified rules of behaviour and interaction, and whose aggregate behaviour is the object of study.
- An *LLM-enabled agent-based model* is one in which a language model supplies some part of an agent's behaviour such as decision, utterance, or intention, in place of, or in addition to, a hand-written rule.
- A *generative agent* is an LLM-driven agent equipped with memory, retrieval, and often reflection and planning, after the architecture of Park and colleagues.[1]
- The *simulation stack* is the full apparatus that turns model outputs into simulated outcomes: the base model, the prompts, the memory and retrieval mechanism, the parser and its fallbacks, the scheduler and exposure rules, and the environment machinery (recommenders, mobility and market models, resolution loops).
- *Validation* is the assessment of whether a model is adequate *for a stated purpose*. This fitness-for-purpose sense is the one the ABM methodology literature has converged on, and the one this report adopts.[9]

Validation is fitness for purpose. "Is this simulation realistic?" is the wrong question. No model of a human population reproduces every individual and every contingency, nor need it. The methodological literature asks whether a model is adequate for a specific use, output quantity and error tolerance: whether, for the decision at hand, the things it abstracts away would change the conclusion.[9] "Realistic" is a threshold relative to a purpose, and much of the confusion around LLM-based simulation follows from treating a plausible surface as if it settled a fitness-for-purpose question it does not address.

2 Two agent traditions meet on the language-model stack

The phrase "LLM agent" fuses two lineages that developed largely independently, and keeping them distinct clarifies what LLM-enabled simulation inherits.

The first is the notion of an *agent* in artificial intelligence. The term enters mainstream AI in the mid-to-late 1980s through distributed AI and multi-agent systems, and is given its standard definition by Wooldridge and Jennings — a computer system situated in an environment and capable of flexible, autonomous action — and by Russell and Norvig's textbook framing of AI itself as the study of rational agents that perceive and act.[10] Reinforcement learning contributes one influential formalism within this line — the agent that acts in an environment to maximise a reward[11] — and appears as one stage in a longer sequence, from symbolic to reactive to reinforcement-learning to language-model agents.[12] This is the lineage that runs toward *task completion*, coding agents and the harness.[8]

The second is the *agent* of agent-based modelling, which descends from a different tradition: complex adaptive systems, cellular automata, and micro-simulation, from Schelling through Epstein and Axtell.[6] Here the agent is a deliberately simple element whose interactions, in aggregate, generate a macro-pattern to be explained. This is the lineage that runs toward *modelling and simulation*.

These two "agents" are cousins. Generative Agents traced its ancestry to believable non-player characters and symbolic cognitive architectures, and found reinforcement learning inadequate for open-world believability.[1] The subsequent survey literature framed generative agents as a new instrument for agent-based modelling.[7] The language model provides the substrate on which the two lineages meet: a single technical stack that can serve either as the "brain" of a task-completing agent or as the behavioural element of a social simulation (Figure 1).

That convergence has been uneven. The overwhelming share of engineering effort since 2023 has flowed toward the first lineage — general-purpose and coding agents and the harness machinery that makes them reliable[8] — while LLM-enabled ABM, though active, has attracted comparatively little of the scrutiny, tooling and engineering discipline that the task-agent world has developed. One consequence, documented in Section 4, is that the scaffolding which makes a social simulation work is often improvised and undocumented, even as the simulations themselves are proposed for consequential use.

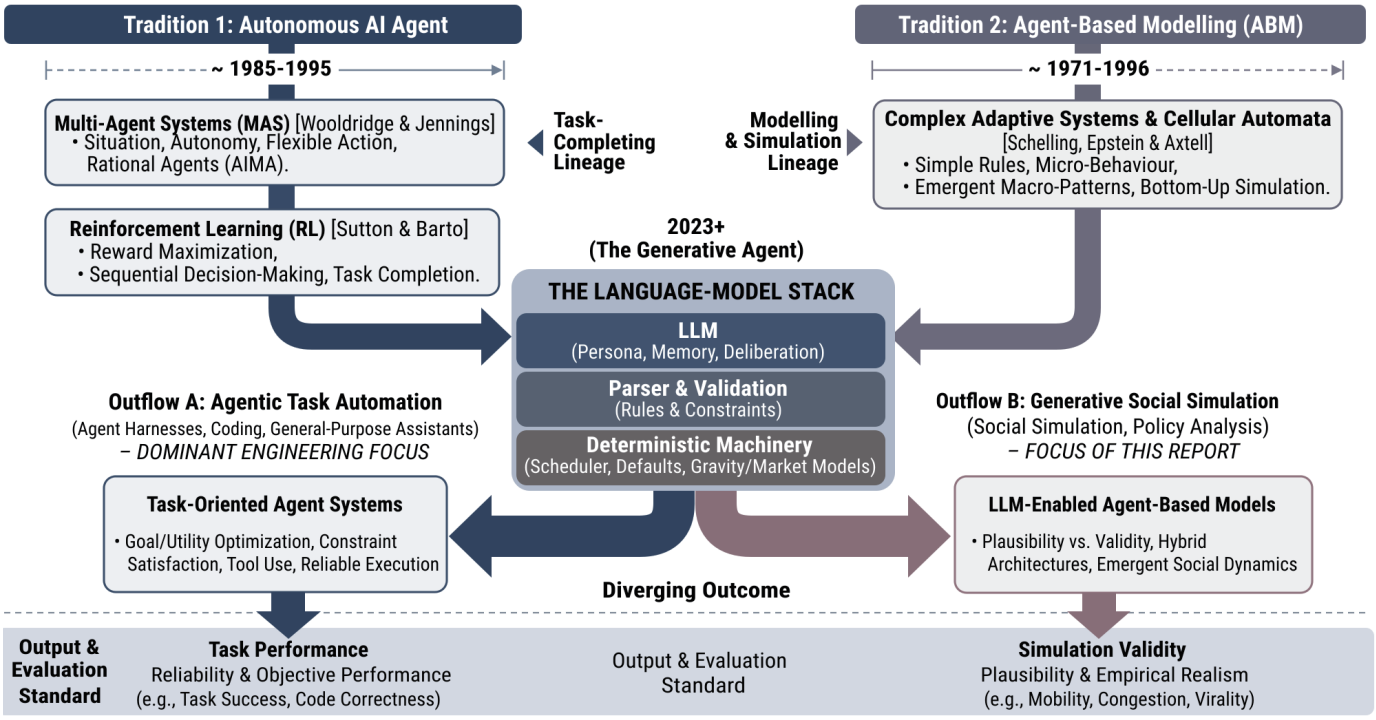


Figure 1. Two agent lineages converging on the language-model stack.

3 Five roles for language models in agent-based modelling

"LLM-enabled ABM" is a family of methods, distinguished by what the language model is asked to do. The distinction matters because each role carries a different evidentiary burden, and because studies routinely borrow the credibility of one role while doing the work of another. Five roles recur across the systems examined; Table 1 summarises them.

1. **The model as an agent's brain (Role A).** The agent's actions and speech are generated by the model from the agent's persona, memory and current situation. This is the generative-agents paradigm, and the one behind the large social simulators.[1] Its proper evidentiary standard is whether the generated behaviour matches how real people behave, a demanding, population-level claim.

2. **The model as a behavioural submodule (Role B).** A conventional agent-based model retains its explicit state, network and transition rules, and calls the language model only at a specific decision point — whether a person isolates when ill, or how much a household consumes. The model supplies one input to an otherwise mechanistic system. Its standard is whether the submodule improves a calibrated baseline model on a defined target.

3. **The model as environment, referee or game master (Role C).** The model adjudicates the world: given the agents' attempted actions, it decides what actually happens and who observes it. Its standard is the coherence and consistency of its adjudication.

4. **The model as a synthetic respondent or digital twin (Role D).** The model stands in for human subjects, answering surveys or reproducing individual responses, often conditioned on demographic or interview data. Its standard is distributional: whether the synthetic answers reproduce the means, the variances and the heterogeneity of real ones.

5. **The model as an interactive policy-reasoning interface (Role E).** The system lets domain experts steer, branch and interrogate scenarios, surfacing assumptions and blind spots. Its standard is usefulness to expert deliberation, explicitly not predictive accuracy.

The roles are frequently blended in a single system and, more problematically, in a single claim. A system built and defensible as a deliberation aid may be reported as if it had met the bar of reproducing real behaviour; a social simulator may be presented as a sandbox whose only obligation is internal coherence. Section 4 shows why claims of correspondence to real human populations are especially demanding: population-level behaviour depends on interactions between LLM-mediated behaviour and the surrounding simulation stack.

Table 1: Five roles for language models in agent-based modelling, and the evidence each requires.

Role	Model output	Macro-outcome pathway	Appropriate evidence standard
A. Agent brain	Actions, speech, plans from a persona and memory	LLM-mediated actions routed and aggregated by the environment machinery	Correspondence of behaviour to real human populations (out-of-sample, multi-pattern)
B. Behavioural submodule	One decision fed into a conventional ABM	The surrounding mechanistic model (calibrated)	Improvement over a calibrated baseline on a defined target quantity
C. Game master / referee	Adjudication of what happens and who observes it	The resolution loop itself	Internal coherence and consistency of adjudication
D. Synthetic respondent	Survey or interview responses for an individual	Statistical pooling of individual responses	Distributional match: means, variance, heterogeneity vs real data
E. Policy interface	Steerable, branchable scenarios for experts	The experts' interpretation	Usefulness to deliberation; surfacing assumptions (not prediction)

4 Source-code analysis of the simulation stack

The claim that a social simulation is realistic is usually made for Role-A systems: large populations of generative agents whose collective behaviour is said to reproduce a social phenomenon. This section analyses the source code of those systems and the traditional platforms used for comparison. The findings identify implementation choices that shape behaviour and aggregate outcomes in the systems examined. Table 2 summarises the code-level observations; the subsections that follow develop them.

4.1 Parsing, Validation and Action Realisation

In the runnable LLM-driven systems examined, model outputs pass through parsing, validation and action-selection procedures before entering the simulation. These procedures include bounded action spaces, repair routines, retries, default substitutions and failure handling. Structured generation can reduce syntactic errors, while semantic, behavioural and system-level validity remain separate concerns.

EconAgent, a widely cited macroeconomic simulator, is the clearest case. The system models a population of households making two decisions each period: how much of their available labour to supply (a number from 0 to 1) and what fraction of their income to consume. The language model is asked for these two numbers as JSON; the code evaluates the returned string as a literal Python expression and, if a range check or the evaluation itself fails, substitutes the constant [1, 0.5] — supply all available labour, consume half of income — before quantising the consumption figure into one of fifty discrete bins.[3] The labour-supply number is then converted to a binary work-or-not decision by coin flip. The model's contribution, after all this processing, is a pair of clamped, defaulted, discretised scalars.

Related output-control patterns recur across the analysed materials:

- In Stanford's original *Generative Agents*, an agent's chosen destination is validated against the map's accessible sectors; if the model names an invalid one, the code silently overrides it to the agent's home area.[13]

Table 2: Code-level observations from public simulation systems inspected in July 2026. The lower block is the traditional-ABM contrast class.

System	Role	Model output handling	Macro-dynamic components	Seed / reproducibility
Generative Agents (Smallville)	A	JSON-slice; invalid destination → home area	Tile maze, A* pathfinding, fixed step clock	No seed; recorded trajectory replay only
AgentSociety	A	JSON-repair; fail → random category / "home" / 10 km	Gravity model, softmax market, traffic engine	No seed; temperature 1.0; replay only
OASIS	A	Function-calls only; fail → agent does nothing	Recommender "hot-score" ranking; feed capped at 2 posts	Model-assignment seed only; recsys unseeded
Concordia	C	Typed output dispatch; retries then error	Game-master resolves the event and its observers	Seed threaded through the model interface
Sotopia	A/D	Pydantic + JSON-repair; fail → second model call	Env step, action mask, LLM-as-judge scores	No seed; temperature 0.7
1,000-person study	D	Regex extraction; fail → constant / None	Statistical pooling (no environment)	No seed; runs one agent by default
EconAgent	B	eval(); fail → [1,0.5]; quantised to 50 bins	Inherited deterministic economy (*ai_economist*)	seed: null; unseeded sampling
AgentTorch	B	float(); fail → 0.0	Fixed epidemic equation; model off by default	Mechanistic; model optional
Covasim	—	Typed state transitions (no model)	Calibrated epidemiological rules	Seed a parameter; per-replicate offset; regression-tested
MATSim	—	Utility scoring (no model)	Co-evolutionary replanning to equilibrium	Re-seeded every iteration; per-thread streams
ABIDES	—	Message handlers (no model)	Discrete-event market kernel	Refuses to run without a seeded RNG

Note: "Role" refers to Table 1. File- and line-level references for every entry are given in the footnotes to Section 4.

- In AgentSociety, the model chooses a destination *category* that is repaired with a JSON-repair library; on failure the agent is sent to a random category, or simply "home," and an unparseable radius becomes a default ten kilometres.[14]

- In the 1,000-person study's released code, survey answers are extracted from free text by regular expression, with numeric importance scores defaulting to a constant and other fields to None.[15]

- In differentiable and framework systems the same holds: AgentTorch parses the model's output with float() and defaults to 0.0; Sotopia coerces output into a fixed set of action types with a JSON-repair pass and, failing that, a second model call to reformat it; OASIS admits only function-calls from a bounded action set and, on error, has the agent do nothing.[16]

Several of these systems also define an explicit "do nothing" action — DO_NOTHING, action_type="none", SKIP_THIS_STEP — for cases in which model output cannot be used. Failure handling ranges from silent drop (OASIS returns the error object and the agent acts not at all), through a hard default (EconAgent, AgentSociety), to a corrective second model call (Sotopia), to bounded retries that finally raise an error (Concordia retries a constrained choice up to twenty times before failing).[17] In these implementations, model output reaches the environment through an intermediate control layer.

The systems do not track how often these defaults fire. Of the eight LLM-driven frameworks whose source code was inspected, none records the fallback rate as a persistent, reportable quantity. EconAgent maintains a cumulative error counter that is printed to the terminal every three timesteps but never written to disk and never included in saved results.[18] AgentSociety increments an llm_error counter in its economy block that no other line of code ever reads.[19] The 1,000-person study's retry mechanism contains a string-comparison bug that prevents it from ever executing a retry: the code checks whether the model's response equals the literal string "GENERATION ERROR", but the error path appends the exception text after a colon, so the check never matches and every call has at most one attempt.[15] Concordia ships an optional profiler that tracks call-level success and failure rates, but it wraps the twenty-attempt retry loop as a single call and does not cover the 'nan' float fallback.[17] Among all fourteen LLM-driven repositories, only OdysSim, a training pipeline, maintains named parse-failure counters and computes an aggregate rate.[20]

The single directly reported fallback rate in the reviewed literature comes from a generative-agent epidemic model: fewer than 0.33% of 68,000 binary stay-or-go decisions failed to comply with the expected format, and all were silently defaulted to "no" (stay home).[21] That figure, however, covers the simplest possible output: a single binary token. Agent decisions in the systems above often require valid values across multiple fields at once, and requiring every field to be valid can sharply reduce the share of complete outputs as the number of required fields increases. In a study that used a language model to simulate responses to the American National Election Studies survey — conditioning it with demographic backstories to produce synthetic respondents — Argyle and colleagues found that individual items achieved compliance rates above 75%, with three items exceeding 99%, but requiring all variables for a single respondent to be valid simultaneously left only 1,782 of 4,270 cases — 41.7% — complete.[22] The defaults themselves are directional: "stay home," "do nothing," "supply all labour and consume half of income" encode behavioural assumptions that pull aggregate statistics in one direction whose magnitude is never measured. In the epidemic model, every defaulted response reduced population mobility and therefore the infection rate the model was designed to study. And because fallback frequency varies with the base model, the prompt, the temperature, and the output complexity, a rate observed under one configuration cannot be assumed to hold under another. No reviewed system provides the means to answer the question of what fraction of a run's behaviour came from the model and what fraction from the defaults.

4.2 Environment machinery and macro-level outcomes

The model proposes actions or intentions; the environment maps them into simulated consequences. Macro-level outcomes are therefore jointly generated by the LLM-mediated behavioural policy and the environment transition function. The code-level analysis shows that conventional, non-generative components often impose strong structure on those outcomes, but their contribution relative to the language model must ultimately be established through intervention, ablation or sensitivity analysis.

AgentSociety is the clearest case, because it is explicitly a model of *urban* behaviour. The language model selects a category of place to visit; the specific destination is then drawn by a *gravity model* — a classical spatial-interaction formula weighting locations by density and inverse-square distance — and the agent is routed there by a separate traffic and mobility engine that advances the world clock.

Consumption and prices are set by a deterministic softmax market.[23] The trip-length distribution, origin–destination flows and congestion patterns are therefore strongly shaped by the gravity model and traffic engine, conditional on the distribution of intentions supplied upstream by the agents. An independent evaluation against real mobility data from Paris and Shanghai found these systems plausible but unrealistic.[2]

OASIS provides a second example. In OASIS, a social-media simulator, the environment rebuilds its recommender table *before* the agents act each step, so a deterministic engagement-ranking formula shapes who sees what; a default configuration shows each agent only two recommended posts.[24] Cascades, virality and polarisation in such a system are sensitive to the ranking and exposure layer, which can be changed without modifying the language model. In Concordia, a general-purpose social-simulation framework by DeepMind, the architecture makes the point explicit: an agent's action is only a *putative* event, and a separate game-master model runs its own chain-of-thought to decide what actually happens and who observes it. An agent "attempting" something submits it to the referee, whose resolution produces the actual outcome.[25] In Sotopia, a platform for evaluating social interactions between LLM agents, an environment step, an action mask and a language-model "judge" assign the outcome scores that are the study's dependent variable.[26]

The most direct demonstration comes from a system that lets the model be switched off. AgentTorch's differentiable epidemic model computes its force of infection from a fixed epidemiological equation that is *identical* whether or not a language model is in the loop; the model, when present, supplies only a single behavioural scalar — a propensity to isolate — and is off by default.[27] In AgentTorch, the deterministic epidemiological core remains unchanged when the optional LLM input is disabled, matching the Role-B structure described above.

4.3 Researcher degrees of freedom across the stack

Between the language model and the reported outcome sits a large number of choices, almost all set by hand and rarely reported. They include the base model and its version; the exact wording of prompts; the sampling temperature (fixed inline at 0.7 in one system, 1.0 in another, 0 in a third); the memory-retrieval weights (hard-coded magic numbers such as [0.5, 3, 2] or [0, 1, 0.5] over recency, relevance and importance); the fallback constants and quantisation steps documented above; and the scheduling and exposure rules that decide who acts and what they see.[28] A recent line of work has begun to treat this design space as a first-class object, showing that the choice of base model alone can move a simulation's "social" outcomes substantially, indicating that some reported social findings may be sensitive to configuration.[29] Scheduling is especially consequential: in Concordia the game master chooses the next actor; in Sotopia an action mask does; in OASIS the feed is a random subsample of a ranked table. Turn order and exposure can materially affect the collective result.

4.4 Reproducibility boundaries are poorly specified

Reproducibility in an LLM-enabled simulation depends on several layers of the stack. Most of the LLM-driven systems examined do not fix random seeds across their controllable stochastic components, and several do not cache model responses. Seeding improves control over local randomness but does not, by itself, define reproducibility of the full simulation stack. Saved trajectories or cached responses can support exact replay of a realised run; independent reruns are still required to assess the stability of results under stochastic sampling.

The traditional agent-based platforms reviewed here show how extensively local sources of randomness can be controlled. ABIDES, a simulator of financial markets and trading strategies, refuses to construct its kernel or any agent without a seeded random-number generator, raising an error otherwise. MATSim, a transport microsimulation used by planning agencies to model daily travel demand in metropolitan areas, re-seeds deterministically at every iteration and derives per-thread streams for reproducibility across cores. Covasim, an epidemic simulator used by governments during the COVID-19 pandemic, exposes the seed as a first-class parameter, offsets it deterministically across parallel replicates, and ships versioned baseline results with a regression test that fails on any numerical drift. Mesa, the standard Python framework for building agent-based models, gives each model its own seeded generator and centralises all randomness through it.[30] These platforms also make agent state an explicit, closed vocabulary — Covasim enumerates exactly the disease states a person may occupy — and treat calibration to data as a named subsystem, such as Covasim's optimisation-based fitting of parameters to observed case data.[31] Compared with these platforms, the LLM-driven systems reviewed here apply explicit state, calibration and reproducibility controls less consistently.

```
for each scheduled agent (order chosen by scheduler / game master):
  observation = environment.render(agent)  // exposure set by env, not agent
  prompt = template(persona, retrieve(memory, weights=[...]), observation)
  for attempt in 1..N:
    text = model.sample(prompt, temperature=T)  // usually unseeded
    action = parse(text)  // json_repair / regex / eval / function-call
    if valid(action, allowed_set):  break
  if not valid:  action = DEFAULT  // "home" / [1, 0.5] / do_nothing
  outcome = environment.step(action)  // gravity model / recsys / resolver
  log(agent, action, outcome)  // recorded trajectory supports exact replay
aggregate_result = f(environment state)  // macro result reflects model and environment dynamics
```

Figure 2. Illustrative step of a generative-agent simulator.

Published descriptions and default implementations sometimes differ in scale and scope. Two of the fourteen "LLM-agent" repositories are not running social simulations at all: one is a model-training and evaluation pipeline, and one is distributed as a technical report with no source code. Among those that do run, the shipped and default scales fall well short of the figures that give the field its sense of magnitude: a study headlined as "1,000 people" runs a single agent by default, a "twenty-five agent" town ships its demonstrations as a three-agent scenario, and a "large-scale urban society" runs one hundred agents for a single simulated day in every shipped example.[32] These differences matter when interpreting claims about scale and generality. Code-level evidence should be tied to the specific stack and configuration reviewed.

The simulation stack, so understood (Figures 2 and 3), plays the same structural role as the "agent harness" analysed in the companion report[8]: it is the intermediating runtime infrastructure between model and environment, and it is where reliability and governability are jointly determined. The difference is that task-agent harnesses have attracted intensive engineering scrutiny — typed state, explicit tool schemas and fallback/error handling — while the simulation stacks examined here remain largely improvised. A *simulation harness* that adopted the discipline of its task-agent counterpart would make most of the disclosures this report's recommendations call for.

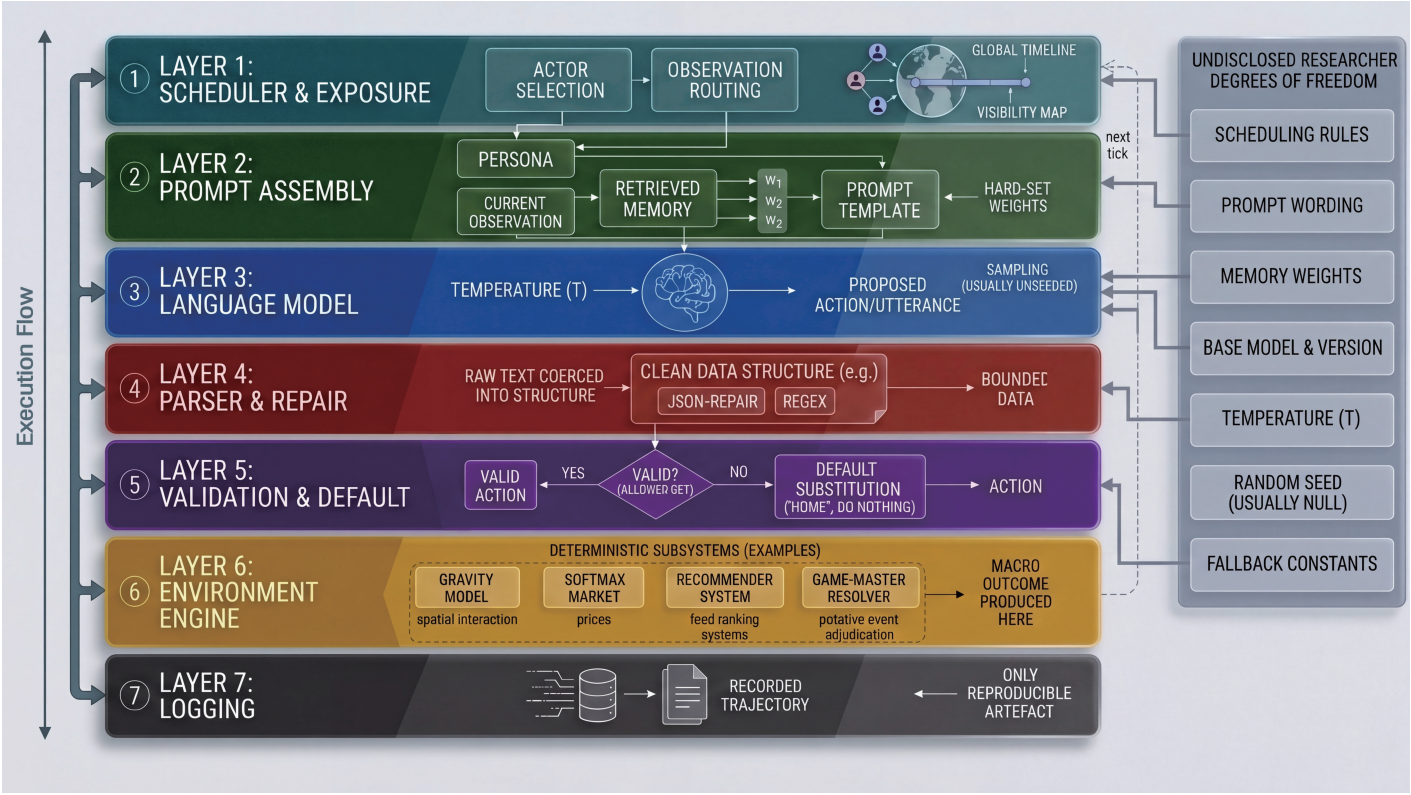


Figure 3. Anatomy of an LLM-enabled simulation stack.

5 The validation problem predates language models

Validation in LLM-enabled ABM applies to the complete simulation stack because each layer can affect the quantities being evaluated. The underlying methodological problem predates language models and has been studied in agent-based modelling for decades.

ABM methodology frames validation in terms of adequacy for a stated purpose, output quantity and error tolerance.[9] Established practices include pattern-oriented modelling, sensitivity and uncertainty analysis, docking and structured documentation.

Pattern-oriented modelling constrains a model by requiring it to reproduce several independent patterns at different scales at once. A model that fits one observed curve proves little, since many mechanisms can match a single pattern; requiring a simultaneous fit to several independent patterns helps rule out many of them.[33]

Sensitivity and uncertainty analysis asks how far a conclusion moves as assumptions vary, and which assumptions it is fragile to. If a small change in one parameter overturns the result, the conclusion depends on that parameter more than on the theory.

Docking compares independent re-implementations of the same model. If two teams build the same model from the same specification and get different results, the specification is underspecified.

Documentation protocols — ODD for model description, TRACE for the evidence trail from purpose through calibration to conclusion — make a model inspectable and its claims traceable.[34]

LLM-enabled simulation can use this toolkit. The LLM layer adds four further validation difficulties.

Believability and face validity. Believability provides evidence of face validity and supports a limited claim: that the simulated behaviour appears recognisably human. Whether that evidence is sufficient depends on the intended use. Plausibility may be useful for exploration and deliberation, but it does not by itself establish empirical, predictive or causal validity.

A simulation that generates a fluent account of why a person stayed home in the rain has demonstrated fluency; whether real people actually change that behaviour is a separate, empirical question that only comparison to data can settle.[4]

Population heterogeneity. Language models tend toward an average persona. Trained for helpfulness and safety, they under-produce the heterogeneity, conflict and refusal that characterise real populations, and heterogeneity is exactly what agent-based modelling exists to represent. Empirical work finds that persona conditioning (instructing the model through its prompt to respond as a person of a specified age, gender, income, political leaning and so on) yields limited and unstable gains, that model "populations" under-represent demographic margins, and that outputs collapse toward a narrow band of dominant views.[35]

Configuration sensitivity. Changes in prompt, sampling temperature or model version can move outcomes, and a finding may turn out to reflect the configuration.[29] **Opacity and possible memorisation.** A language model trained on much of the public record may reproduce a target pattern from memorisation rather than mechanism, and its black-box character limits the transparency that the ODD/TRACE norms were designed to secure.[4]

When tested against ground truth, LLM simulations frequently fail in ways that matter. In a controlled behavioural game, essentially all advanced prompting and persona methods failed to reproduce the human response *distribution*.[36] Synthetic survey respondents match some population means but compress variance and disagree with real regression relationships often enough to be unsafe for inference.[37] A large replication of scenario experiments reproduced the direction of most main effects but found that the models consistently produced larger effect sizes than the original human studies and often generated significant results where the originals were null.[38] And LLM opinion-dynamics models tend to converge to consensus unless a confirmation bias is deliberately injected, failing to reproduce real polarisation, a reminder that such systems often reveal the model's own tendencies rather than a social mechanism.[39]

Figure 4 organises these evidence requirements into five tiers. Lower tiers cover plausibility and exploratory use. Higher tiers require stronger empirical validation and monitoring.

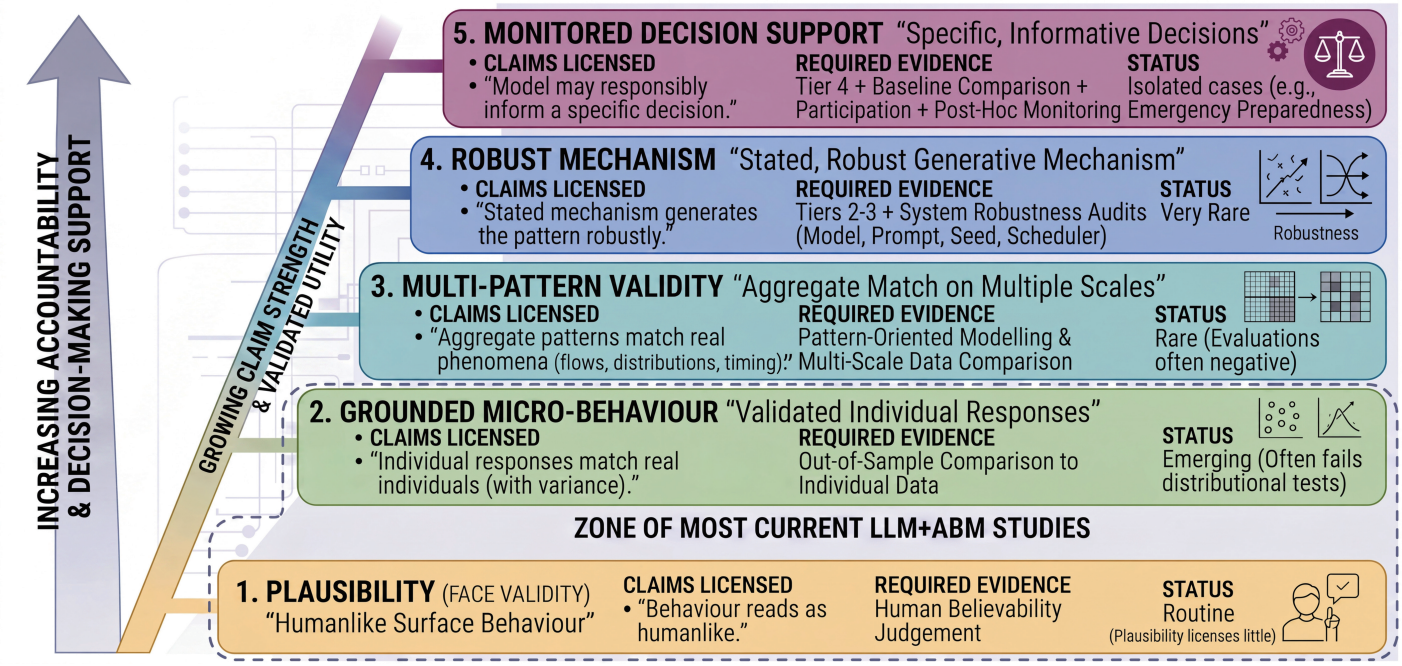


Figure 4. From plausible to accountable: an evidence hierarchy.

Relation to existing ABM standards. ODD and TRACE already provide important foundations for model description and evidence tracking.[34] Those protocols, designed for hand-written models, do not anticipate the disclosures that the code-level reading shows to be important here: the parser and its default behaviour as a reported statistic, the prompt and base-model version as parameters, the reproducibility boundary and its associated metadata, and the robustness audit over model and prompt. The first documentation and configuration standards designed for LLM-based simulation are only now appearing,[a] and the specification above is complementary to them: they standardise how a stack is recorded; it states what must be disclosed and validated before the stack informs a decision. **Model improvement.** Better behavioural models may reduce some of the current gaps. The expectation that model quality can close these gaps is implicit in the original "algorithmic fidelity" argument for silicon sampling[22] and explicit in current efforts to train dedicated behavioural foundation models.[b] Even a perfectly behaviour-matching model would still need to state a purpose, validate its target quantities, compare against a baseline and disclose its stack. These are properties of responsible modelling, independent of model quality. Current evidence still documents homogenisation in helpfulness-tuned models, including reduced representation of the heterogeneity required for social simulation.[35] The dedicated models are unproven, and they too would be used inside a stack whose disclosure remains necessary.

[a] Sarangi, S., Puelma Touzel, M., Bück-Kaeffer, A., Yang, Z., Godbout, J.-F., & Rabbany, R. (2026). *EASE configuration facilitates a reproducible science of LLM social simulations*. arXiv. https://arxiv.org/abs/2605.30258

[b] Zhou, X., Sun, W., Du, W., Liu, J., Sun, H., Ma, Q., Wu, T., Yang, Y., & Sap, M. (2026). *OdysSim: Building foundation models for human behavior simulation*. arXiv. https://arxiv.org/abs/2606.14199. The paper argues that helpfulness-tuned assistants are ill-suited to behaviour simulation and trains a dedicated model instead.

6 Application domains and the evidence gap

The gap between plausibility and validity varies by use. It is widest where the target is a broad social prediction and narrowest where the task is bounded and the model serves human deliberation. This section reviews the main application areas covered by the studies above and rates them against the evidence hierarchy (Table 3).

Policy simulation and governance sandboxes. The most defensible policy use is deliberation. Systems such as *WhatIf* let emergency and public-health experts steer, branch and interrogate LLM-powered scenarios, and their evaluated value is in surfacing tacit assumptions and unnoticed vulnerabilities.[40] Platform-governance sandboxes such as PolicySim, which tune agents to a specific platform's data to test moderation interventions before deployment, are a bounded and promising variant.[41] The risk in this domain is the slide from deliberation aid to predictive authority. A recent position paper sets out three preconditions: simulations of marginalised groups must be treated as politically loaded, since the choice of who to simulate and how encodes assumptions that can entrench existing inequalities; the populations being simulated should participate in the design and have a channel to challenge the outputs; and a named party must be responsible for how the simulation's results are used.[42]

Emergency preparedness provides a credible case of changed practice. In a year-long collaboration with a university's preparedness team, an LLM-agent simulation modelled crowd behaviour at a large campus gathering, letting the team run evacuation scenarios, test volunteer-placement strategies and identify infrastructure bottlenecks such as narrow exits and poorly positioned barriers. Over iterative rounds, the preparedness team steered the simulation to refine their plans for volunteer training, evacuation routing and barrier placement. The authors are explicit that its value was as a reasoning and design tool.[43] The study's practical value came from a bounded operational problem, continued involvement of domain experts, and an explicitly deliberative use of the simulation.

Smart cities and urban mobility provides a well-documented case of the plausibility–validity gap. Large simulators generate fluent accounts of urban life, but when their output is checked against real mobility data — trip-length distributions, origin–destination flows, dwell times — from real cities, the fit is poor.[2] As Section 4 shows, these spatial statistics reflect both LLM-generated intentions and the gravity and traffic components that transform them. A narrower, data-anchored example in this domain calibrates an agent-based model of a specific bus-route reduction to real fare-card data[44], and should be compared with mature, calibrated transport microsimulations such as MATSim as a baseline.

Economics and central banking remains exploratory. LLM agents reproduce qualitative regularities of behavioural economics, and macroeconomic simulators such as EconAgent recover textbook relationships. As Section 4 showed, those relationships arise within an inherited deterministic economy to which the model contributes two clamped scalars.[3] The "homo silicus" proposal remains a research approach, while central banks use classical agent-based models as analytical tools within policy institutions.[45] This established tradition provides a demanding validation baseline for LLM-based systems.

Synthetic respondents deserve specific caution. Using a language model in place of survey respondents is attractive, and it has a defensible exploratory use for piloting instruments and generating hypotheses.[22] But as noted above it cannot substitute for real data for inference, because it compresses variance and misstates relationships.[37] A related, genuinely new line of work studies the social structures that emerge when AI agents interact on networks built for them, a distinct question from whether such agents resemble humans, and one that should be treated on its own terms.[46]

Table 3: Application domains rated against the evidence hierarchy (Figure 4). "Highest defensible tier" is the strongest claim supported by the studies reviewed in this report.

Domain	Highest defensible tier	Assessment
Emergency preparedness	5 (deliberation)	Strongest real-practice case; changed procedures; disclaims prediction
Policy / governance sandboxes	4–5 (deliberation)	Value as an interactive reasoning aid; predictive claims not supported
Public health behaviour	2–3	Bounded, data-anchored studies show promise; not yet a decision tool
Synthetic respondents	2	Useful for piloting and hypotheses; cannot replace survey data for inference
Urban mobility / smart cities	1–2	Plausible narratives; fails spatial statistics against real data
Economics / central banking	1–2	Reproduces stylised facts on an inherited economy; not used to set policy
Social media / opinion dynamics	1–2	Often studies the model's tendencies rather than a social mechanism

7 Implications: sustainable development, the digital divide and AI governance

The technical argument has a direct bearing on the concerns of a development-oriented institution.[47] The distance between plausibility and validity becomes a political matter when a simulation touches public decisions.

Representation and contestability. The populations these systems model — the residents of a district, the users of a platform, the recipients of a health or cash-transfer intervention — are frequently those with the least capacity to audit or challenge a model that claims to represent them. A simulation that looks scientific can lend a policy the appearance of evidence while concealing that its "population" is an average persona shaped by undisclosed prompts and defaults. The risk is that a contestable choice gets laundered as a technical output. For policy-facing simulations of identifiable communities, participation contributes to fitness for purpose by giving the represented population a role in specification and a channel to contest the results.[42]

Access and representational asymmetries. Building and, crucially, *interrogating* these systems requires access to capable models, compute and scarce expertise, all concentrated in a few institutions and a few languages. These constraints affect both system development and the populations represented. Communities in the Global South are underserved as *builders*, dependent on systems and base models they cannot readily adapt or audit. They are also disadvantaged as *subjects*: the same homogenisation that flattens minority views within a population is worse for languages and cultures underrepresented in training data, so a simulation is likely to represent least well exactly those it most claims to include.[35] Without appropriate safeguards, these systems can reproduce existing inequalities in access to policy analysis.

Influence and correctness. The history of model-driven policy is a caution: models have shaped consequential decisions in past public-health and animal-disease crises while being, in hindsight, poorly validated, and reviews of pandemic agent-based modelling have found wide variation in quality and a persistent gap between models and the needs of decision-makers.[48] That a simulation entered a decision process says nothing about whether it improved the decision. The persuasive quality of LLM-generated simulations makes validation especially important when their outputs enter policy deliberation.

Position in the governance landscape. Current AI-governance instruments (the EU AI Act, the NIST AI Risk Management Framework, the OECD AI Principles, and, in the multilateral sphere, the Global Digital Compact and the report of the UN Secretary-General's High-level Advisory Body on AI) address AI systems and their risks in general terms, but they do not provide a dedicated framework for the

use of *AI-generated social simulation as an input to public decisions*.[49] Multilateral institutions could address this gap through reference architectures and disclosure standards for simulation-informed policy analysis, pooled multilingual evaluation infrastructure, and stronger public-sector capacity — especially in under-resourced settings — to interrogate simulation results independently.

8 Recommendations

The following recommendations are addressed to two audiences — those who build and use these systems, and those who govern or procure their use — and they are graduated: a deliberation aid used in a workshop need satisfy the specification lightly, whereas a simulation whose output is cited in a public decision must satisfy it fully. The specification's function is to make the evidentiary tier of Figure 4 explicit, so that a tier-1 plausibility demonstration cannot be presented as if it were a tier-3 validated model.

Recommended technical actions

The recommendations below specify the information and tests that should accompany an LLM-enabled simulation before it informs a decision. They extend existing fitness-for-purpose validation and ODD/TRACE documentation to the language-model layer,[34] with additional disclosures identified through the source-code analysis in Section 4.

1. Purpose and role. State which of the five roles (Table 1) the model plays, and which decision, if any, the result is to inform.

2. Target quantities. Name the specific outputs that must be correct: the quantities against which the simulation will be judged.

3. Empirical anchors. Identify which real data validate which target quantity, out of sample. Follow pattern-oriented practice: require the model to match several independent empirical patterns at different scales, not one headline curve, and hold out the validation data from any tuning. Report distributional fit — variance and heterogeneity — not only means.

4. Baseline comparison. Compare the LLM-enabled system with a rule-based, statistical or classical microsimulation baseline on the target quantity. Prefer the simpler model when the LLM provides no measurable improvement.

5. Stack disclosure. Publish the full simulation stack: base model and version, prompts, memory and retrieval rules, scheduler and exposure logic, sampling temperature, random seeds, and parser and fallback code. Record seeds for every controllable stochastic component, together with model, provider and version information. Retain model responses when exact replay is required, and distinguish trajectory replay from independent stochastic replication. Report the rate at which model outputs are repaired, defaulted or dropped as a first-class behavioural statistic.

6. Robustness audit. Vary the base model, prompt, memory configuration, seed, scheduler and, where feasible, key environment assumptions. Use factorial ablations to identify components and interactions that materially change the target outcomes. Report substantial sensitivity as configuration dependence.

7. Participation and accountability. Where a simulation stands in for an identifiable population — especially a marginalised or low-resource community — require that the population have a role in its design and a channel to contest its outputs. Name the party responsible for how the simulation's results are used. A simulation of a group conducted in the group's absence should not be treated as evidence about it.

8. Claim boundary. State explicitly which claims each result licenses, including exploration, prediction, causal inference and generalisation across time, culture or platform, and tie those claims to the corresponding evidence tier.

Recommended governance actions

9. Do not procure or cite on plausibility. Require, before a simulation's output informs a decision, a completed accountability specification: a stated purpose and role, named target quantities, empirical anchors, a baseline comparison, stack disclosure, a robustness audit, a participation record and a claim boundary. Absent these, treat the output as illustration.

10. Establish disclosure and reproducibility standards. Extend the documentation norms already used for agent-based models (ODD, TRACE) to the language-model layer, adding the stack disclosures this report identifies. Make them a condition of publication and of procurement for policy-facing work.

11. Build shared, multilingual evaluation infrastructure. Support independent, pooled benchmarks — covering under-resourced languages and settings — against which claims of realism can be tested by parties other than the systems' authors.

12. Invest in interrogative capacity. Fund the capacity of public institutions, particularly in the Global South, to scrutinise simulation results, to demand and read an accountability specification.

13. Locate accountability at the stack and its operators. Ensure that responsibility for a simulation-informed decision rests with the humans who built, configured and used the stack, and cannot be diffused onto the "agents" or the modelled population. A believable output transfers no responsibility.

Conclusion

Large language models have given agent-based social simulation a new and genuine capability: agents that can be read, questioned and made to explain themselves, and simulations that can be steered and interrogated in natural language. That capability is real, and in bounded, expert-guided settings — the emergency-preparedness work is the clearest case — it is already useful. The same fluency can also encourage over-interpretation of simulation outputs. The code reviewed here shows that parsers, defaults, schedulers, environment models and other implementation choices materially affect simulated outcomes. The validation problem familiar from traditional ABM therefore remains, with additional configuration and reproducibility issues at the LLM layer.

For policy use, evaluation should cover the complete simulation stack, the empirical quantities the model is expected to reproduce, and the stability of results under reasonable re-specification. Plausibility can support exploratory and deliberative uses. Empirical, predictive and causal claims require corresponding validation, disclosure and accountability.

References

1 Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. ACM. https://doi.org/10.1145/3586183.3606763

2 Santos, G. H., Viana, A. C., & Silva, T. H. (2026). *When plausible is not realistic: Evaluating human mobility in LLM-based urban simulation*. arXiv. https://arxiv.org/abs/2606.13835

3 Li, N., Gao, C., Li, M., Li, Y., & Liao, Q. (2024). EconAgent: Large language model-empowered agents for simulating macroeconomic activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.829. Source at https://github.com/tsinghua-fib-lab/ACL24-EconAgent; the parse-and-default logic is in simulate.py, lines 109–118.

4 Larooij, M., & Törnberg, P. (2026). Validation is the central challenge for generative social simulation: A critical review of LLMs in agent-based modeling. *Artificial Intelligence Review, 59*, Article 15. https://doi.org/10.1007/s10462-025-11412-6

5 Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences, 99*(suppl. 3), 7280–7287. https://doi.org/10.1073/pnas.082080899

6 Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology, 1*(2), 143–186. https://doi.org/10.1080/0022250X.1971.9989794; Epstein, J. M., & Axtell, R. (1996). *Growing artificial societies: Social science from the bottom up*. Brookings Institution Press / MIT Press.

7 Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., & Li, Y. (2024). Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications, 11*, Article 1259. https://doi.org/10.1057/s41599-024-03611-3; Mou, X., Ding, X., He, Q., Wang, L., Liang, J., Zhang, X., Sun, L., Lin, J., Zhou, J., Huang, X., & Wei, Z. (2024). *From individual to society: A survey on social simulation driven by large language model-based agents*. arXiv. https://arxiv.org/abs/2412.03563

8 Liu, J. A. (2026). *Engineering and governing the agent harness: A technology and policy framework for the runtime layer of agentic AI*. United Nations University Institute in Macau.

9 Windrum, P., Fagiolo, G., & Moneta, A. (2007). Empirical validation of agent-based models: Alternatives and prospects. *Journal of Artificial Societies and Social Simulation, 10*(2), Article 8. https://www.jasss.org/10/2/8.html

10 Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review, 10*(2), 115–152. https://doi.org/10.1017/S0269888900008122; Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

11 Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.

12 Xi, Z., et al. (2023). *The rise and potential of large language model based agents: A survey*. arXiv. https://arxiv.org/abs/2309.07864; its §2.1 traces the agent concept to philosophy, its entry into AI through multi-agent systems, and reinforcement learning as one later stage.

13 Stanford *Generative Agents* (reverie backend), https://github.com/joonspk-research/generative_agents: the constrained-then-defaulted action is at reverie/backend_server/persona/prompt_template/run_gpt_prompt.py, lines 617–619; the JSON-slicing parser and retry-with-fail-safe loop are in gpt_structure.py, lines 105–120 and 265–273; hard-coded fail-safe values (e.g. a default wake hour of 8) recur throughout run_gpt_prompt.py.

14 AgentSociety, https://github.com/tsinghua-fib-lab/AgentSociety: parse-and-default logic in packages/agentsociety/agentsociety/cityagent/blocks/mobility_block.py, lines 209–214 (random-category fallback), 241–244 (radius = 10000) and 309–314 (destination defaults to "home"); retry loops e.g. cognition_block.py line 170.

15 Generative Agent Simulations of 1,000 People, https://github.com/StanfordHCI/genagents: brace-slice-and-regex parser in simulation_engine/llm_json_parser.py (lines 14–45); default importance score of 25 in genagents/modules/memory_stream.py line 35; None fail-safes e.g. modules/interaction.py line 67.

16 AgentTorch, https://github.com/AgentTorch/AgentTorch: agent_torch/core/llm/archetype.py lines 166–174. Sotopia, https://github.com/sotopia-lab/sotopia: sotopia/generation_utils/output_parsers.py lines 29–38 and the reformat retry at generate.py lines 295–305; bounded ActionType at messages/message_classes.py line 10. OASIS, https://github.com/camel-ai/oasis: function-call handling and the “return the exception” fallback at oasis/social_agent/agent.py lines 140–155; bounded ActionType enum incl. DO_NOTHING at social_platform/typing.py lines 17–49.

17 Concordia, https://github.com/google-deepmind/concordia: bounded-choice retry limit and error at contrib/language_models/openai/base_gpt_model.py lines 155–176; typed output-type dispatch with a ’nan’ float fallback at components/game_master/switch_act.py lines 287–338.

18 EconAgent: the gpt_error counter is printed at simulate.py line 235 but absent from the pickle checkpoints saved at lines 237–241; it does not distinguish API failures from parse failures from range-check failures.

19 AgentSociety: self.llm_error is incremented at economy_block.py line 409 but never read or saved by any other line of the codebase.

20 OdysSim: named counters pairwise_parse_failure_count, rating_parse_failure_count and the computed parse_failure_rate_overall at agents/rolermbench/agent.py lines 319–413.

21 Williams, R., Hosseinichimeh, N., Majumdar, A., & Ghaffarzadegan, N. (2023). *Epidemic modeling with generative agents*. arXiv. https://arxiv.org/abs/2307.04986; the compliance figure is reported in the supplementary materials.

22 Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J. R., Rytting, C., & Wingate, D. (2023). Out of one, many: Using language models to simulate human samples. *Political Analysis, 31*(3), 337–351. https://doi.org/10.1017/pan.2023.2; the per-item and compound compliance figures are in Appendix D.2.1.

23 AgentSociety: gravity model at mobility_block.py line 94 (weight = density / (distance**2), sampling at lines 147 and 259); deterministic market at cityagent/blocks/economy_block.py line 225; traffic/time engine step() at environment/environment.py lines 695–699.

24 OASIS: recommender refresh before agent action at oasis/environment/env.py lines 151–152; the deterministic hot-score ranker at oasis/social_platform/recsys.py lines 168–196 and 213–259; the default max_rec_post_len of 2 at social_platform/platform.py lines 65–66.

25 Concordia: putative-event tag and game-master resolution at concordia/components/ game_master/ event_resolution.py lines 36 and 209–211; the game master also selects who observes the event (lines 224–234) and, in the sequential engine, who acts next (environment/engines/sequential.py lines 117–122).

26 Sotopia: environment step, evaluator call and action-mask recomputation at sotopia/envs/parallel.py lines 496–528; the LLM-as-judge evaluator at envs/evaluators.py lines 162–233.

27 AgentTorch: the force-of-infection computation, identical in the "llm" and non-LLM branches, at agent_torch/models/covid/substeps/new_transmission/transition.py lines 78–82; the language model contributes only the will_isolate gate (line 75); the default execution mode is heuristic (LLM off) with 37,518 agents at models/covid/yamls/config.yaml lines 3 and 42.

28 Hard-coded retrieval weights: Stanford persona/cognitive_modules/retrieve.py line 244 (gw = [0.5, 3, 2]); 1,000-person study memory_stream.py line 347 (hp=[0, 1, 0.5]). Inline temperatures: 0.7 (genagents .../gpt_structure.py line 74), 1.0 (AgentSociety llm/llm.py line 114), 0.0 (OASIS experiment YAMLs). The exposure cap and scheduler choices are cited above.

29 On the influence of design choices and the base model, see Bück-Kaeffer, A., Sarangi, S., Puelma Touzel, M., Rabbany, R., Yang, Z., & Godbout, J.-F. (2026). *The silicon society cookbook: Design space of LLM-based social simulations*. arXiv. https://arxiv.org/abs/2605.00197; and Ye, J., Cao, L., Chen, D., & Ferrara, E. (2026). *Stop drawing scientific claims from LLM social simulations without robustness audits*. arXiv. https://arxiv.org/abs/2605.18890

30 ABIDES, https://github.com/abides-sim/abides: the seeded-RNG requirement (raising ValueError otherwise) at Kernel.py lines 17–20 and agent/Agent.py lines 14–24. MATSim, https://github.com/matsim-org/matsim-libs: deterministic re-seeding each iteration at .../core/controler/AbstractController.java lines 63–65. Covasim, https://github.com/InstituteforDiseaseModeling/covasim: set_seed at utils.py lines 276–301, per-replicate offset at run.py lines 1363–1365, and the baseline regression test at tests/test_baselines.py lines 81–93. Mesa, https://github.com/projectmesa/mesa: model-owned seeded generators at mesa/model.py lines 128–130.

31 Covasim: the closed set of boolean disease states at covasim/defaults.py lines 58–75; the optimisation-based Calibration class at covasim/analysis.py lines 1358–1362 and 1471–1493, with parameters annotated as calibrated to specific published sources at parameters.py lines 60–63.

32 The training-pipeline repository is OdysSim (https://github.com/sunnweiwei/OdysSim), built on a reinforcement-learning trainer; the source-free repository is Project Sid / Altera's PIANO (https://github.com/altera-al/project-sid), which ships a PDF and figures only. Default scales: the 1,000-person study's entry point runs one agent (main.py line 31); every shipped AgentSociety example sets number = 100 with a one-day workflow; Stanford's precomputed replays are all the three-agent scenario. Headline scientific claims of these systems should be read against the scale and determinism their public code actually exercises.

33 Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W. M., Railsback, S. F., Thulke, H.-H., Weiner, J., Wiegand, T., & DeAngelis, D. L. (2005). Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science, 310*(5750), 987–991. https://doi.org/10.1126/science.1116681

34 Grimm, V., et al. (2020). The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation, 23*(2), Article 7. https://doi.org/10.18564/jasss.4259. On TRACE, see Grimm, V., Augusiak, J., Focks, A., Frank, B. M., Gabsi, F., Johnston, A. S. A., Liu, C., Martin, B. T., Meli, M., Radchuk, V., Thorbek, P., & Railsback, S. F. (2014). Towards better modelling and decision support: Documenting model development, testing, and analysis using TRACE. *Ecological Modelling, 280*, 129–139. https://doi.org/10.1016/j.ecolmodel.2014.01.018. On the fitness-for-purpose orientation of validation as an integrated process, see Troost, C., et al. (2023). How to keep it adequate: A protocol for ensuring validity in agent-based simulation. *Environmental Modelling & Software, 159*, Article 105559. https://doi.org/10.1016/j.envsoft.2022.105559

35 Wu, Z., Peng, R., Ito, T., Onizuka, M., & Xiao, C. (2025). *LLM-based social simulations require a boundary*. arXiv. https://arxiv.org/abs/2506.19806. On representativeness gaps and homogenisation, see Santurkar, S., Durmus, E., Ladhak, F., Lee, C., Liang, P., & Hashimoto, T. (2023). Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*. https://arxiv.org/abs/2303.17548; and Sen, I., et al. (2025). Missing the margins: A systematic literature review on the demographic representativeness of LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 24263–24289). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-acl.1246

36 Gao, Y., Lee, D., Burtch, G., & Fazelpour, S. (2025). Take caution in using LLMs as human surrogates. *Proceedings of the National Academy of Sciences, 122*(24), e2501660122. https://doi.org/10.1073/pnas.2501660122

37 Bisbee, J., Clinton, J. D., Dorff, C., Kenkel, B., & Larson, J. M. (2024). Synthetic replacements for human survey data? The perils of large language models. *Political Analysis, 32*(4), 401–416. https://doi.org/10.1017/pan.2024.5

38 Cui, Z., Li, N., & Zhou, H. (2025). A large-scale replication of scenario-based experiments in psychology and management using large language models. *Nature Computational Science, 5*, 627–634. https://doi.org/10.1038/s43588-025-00840-7

39 Chuang, Y.-S., et al. (2024). Simulating opinion dynamics with networks of LLM-based agents. In *Findings of the Association for Computational Linguistics: NAACL 2024* (pp. 3326–3346). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.211

40 Li, Y., Monteiro, K., Shirado, H., & Das, S. (2026). *WhatIf: Interactive exploration of LLM-powered social simulations for policy reasoning*. arXiv. https://arxiv.org/abs/2604.17615

41 Huang, R., Tang, N., Xu, J., Cao, Y., Tu, Q., Guo, S., Zheng, B., Liu, H., & Yang, Y. (2026). PolicySim: An LLM-based agent social simulation sandbox for proactive policy optimization. In *Proceedings of the ACM Web Conference 2026 (WWW '26)* (pp. 4781–4792). ACM. https://doi.org/10.1145/3774904.3792555

42 Luo, S., Arora, S., & Guirado, C. (2026). *We need strong preconditions for using simulations in policy*. arXiv. https://arxiv.org/abs/2604.07838

43 Li, Y., Das, S., & Shirado, H. (2025). *What makes LLM agent simulations useful for policy? Insights from an iterative design engagement in emergency preparedness*. arXiv. https://arxiv.org/abs/2509.21868v1

44 Li, Q., Yang, X., & Zhou, S. (2025). *Exploring dissatisfaction in bus route reduction through LLM-calibrated agent-based modeling*. arXiv. https://arxiv.org/abs/2510.26163

45 On the original "homo silicus" proposal, see Horton, J. J. (2023). *Large language models as simulated economic agents: What can we learn from homo silicus?* arXiv. https://arxiv.org/abs/2301.07543v1. On the classical use of agent-based models in central banking, see Borsos, A., Carro, A., Glielmo, A., Hinterschweiger, M., Kaszowska-Mojsa, J., & Uluc, A. (2025). *Agent-based modeling at central banks: Recent developments and new challenges* (Staff Working Paper No. 1122). Bank of England.

46 See, e.g., Price, H. C. W., AlMuhanna, H., Bassani, P. M., Ho, M., & Evans, T. S. (2026). *Let there be claws: An early social network analysis of AI agents on Moltbook*. arXiv. https://arxiv.org/abs/2602.20044, which documents extreme attention concentration and core–periphery structure among interacting agents.

47 Liu, J., Chu, C., Zhao, Y., Aoki, G., & Xiao, Z. (2025). Agentic AI for sustainable development: Leveraging large language model-enhanced agent-based modeling for complex policy strategies. *Emerging Media, 3*(3), 401–413. https://doi.org/10.1177/27523543251365678

48 Lorig, F., Johansson, E., & Davidsson, P. (2021). Agent-based social simulation of the COVID-19 pandemic: A systematic review. *Journal of Artificial Societies and Social Simulation, 24*(3), Article 5. https://www.jasss.org/24/3/5.html

49 European Union. (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council (Artificial Intelligence Act). National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). https://doi.org/10.6028/NIST.AI.100-1. OECD. (2024). *Recommendation of the Council on artificial intelligence* (as amended). United Nations. (2024). *Global digital compact*. United Nations Secretary-General's High-level Advisory Body on Artificial Intelligence. (2024). *Governing AI for humanity: Final report*. United Nations.