



MetaSociety: A Simulation-Driven AI Policy Advisor for Data-Grounded Smart Governance

Priyanka Jadhav¹, Payal Dongre², Priti Chaugule³, Mayuri Kulkarni⁴

Department of Computer Science & Engineering, SKN Sinhgad College of Engineering¹⁻⁴

Abstract: Policy decisions in local governance are frequently taken with limited empirical grounding, relying on precedent, intuition, or small-scale pilots that are slow and costly to run. This paper presents Meta Society, a smart-governance platform that lets a policymaker state a goal in natural language, receive structured policy recommendations, and evaluate candidate policies against a virtual model of a town before committing to a real-world deployment. The platform combines three components. First, an AI Policy Advisor executes a multi-stage large-language-model (LLM) reasoning pipeline-orchestrated with LangGraph and served through a FastAPI backend-that turns a natural-language goal and multi-domain town statistics into structured candidate policies. Second, a Simulation Lab reports its outcomes from a deterministic engine rather than from an LLM: one agent per citizen is generated for a real town (Pandharpur, Solapur district, Maharashtra) so that published Census 2011 totals are reproduced exactly; a 35-node Bayesian network-thirty core nodes plus five exogenous shock roots-is rolled forward one period at a time with conditional-probability tables counted from the population wherever the data can speak; a historical-grounding stage transfers characteristics from real predecessor programmes to the proposed policy and derives a grounded effect scale that the causal network consumes; an external-shock layer stress-tests the policy under hypothetical events; and a Differential Evolution search with NSGA-II non-dominated selection, reported alongside an equal-budget random-search control, explores the policy-parameter space. Third, a data and dashboard layer renders the same population together with a field-by-field provenance ledger that separates verified Census fields from modelled ones. Every headline metric is a stated summation over the agent population, and each run re-checks the accounting identities, the network's documented response directions, and reproducibility, reporting the outcome as an evidence pack. We describe the architecture, the modules, the workflow, and the verification the engine performs, report a seed-ensemble uncertainty protocol and a draw-count sensitivity analysis, and state the system's boundaries plainly: response directions are validated but magnitudes are not calibrated against an evaluated real programme, income and sector are modelled rather than measured, and the platform is a counterfactual comparison tool rather than a forecasting service.

Keywords: smart governance, policy simulation, large language models, LangGraph, Bayesian networks, differential evolution, NSGA-II, agent-based modeling, microsimulation, synthetic population, external shocks, historical grounding.

I. INTRODUCTION

Local and regional policymakers routinely face decisions-adjusting a subsidy, changing a tax rate, expanding a public service-whose downstream social and economic effects are difficult to anticipate without either historical precedent or a live pilot. Experimenting directly on a population carries real cost and risk when a policy turns out to be poorly designed. This motivates a simulation-first approach: test a policy on a virtual model of a town, populated by citizen agents, before applying it in the real world.

Meta Society is a smart-governance platform built around this premise. A user reviews interactive dashboards that summarise a town's current state, states a policy goal in natural language (for example, "reduce unemployment without increasing inflation"), and receives structured policy recommendations with expected benefits and risks. A selected policy is then submitted as a structured brief and run in the Simulation Lab, which models how the town's population responds and reports economic and social outcomes, including risk alerts, per-zone incidence, and comparisons against a no-policy counterfactual. This paper documents the system as built: the completed AI Policy Advisor, the completed Simulation Lab engine, the data foundation they share, and the reasoning behind replacing an earlier regression-based agent model with an explicit probabilistic and evolutionary one.

The remainder of this paper is organised as follows. Section II reviews related work. Section III describes the architecture and technology stack. Section IV details the Simulation Lab engine: its design principle, data foundation, Bayesian network, multi-period dynamics and shock layer, policy representation and governance constraints, channel dictionary, historical grounding, policy search, decision layer, accounting identities and dashboard layer. Section V describes the AI Policy Advisor. Section VI explains the move from an aggregate corpus to census-anchored microdata. Section VII covers



persistence, reproducibility and verification. Section VIII presents the verification, uncertainty and sensitivity results. Section IX states the limitations and threats to validity, and Section X concludes with future work.

II. RELATED WORK

A. AI-Assisted Policy and Governance Tools

Interest in applying large language models to public-sector decision support has grown alongside broader LLM adoption, with tools that summarise legislation, draft policy briefs, or answer citizen queries. Prior work has also coupled multi-agent simulation with e-government decision-making [9] and explored how LLM-enhanced agent-based modeling can support complex policy strategies [10], alongside broader analyses of AI-agent systems and their governance implications [11]. Meta Society differs from summarisation-oriented tools in that it couples policy generation with an explicit, inspectable outcome-simulation layer, so that a suggested policy is paired with a computed comparison rather than presented as text alone.

B. Agent-Based Modeling

Agent-based modeling (ABM) represents a system as a population of autonomous agents with individual attributes and simple behavioural rules, whose interactions produce emergent, population-level outcomes [1]. ABM has been applied extensively in epidemiology, traffic simulation, and the social sciences to represent heterogeneous populations more realistically than aggregate models can. This body of work motivates our shift away from a small number of “representative” agents toward a population in which every agent corresponds to one citizen.

C. Random Forests

Random Forests, introduced by Breiman [2], construct an ensemble of decision trees trained on labelled examples and aggregate their predictions by voting or averaging. They are effective for supervised classification and regression with sufficient labelled data, but they do not natively represent the causal or probabilistic dependency structure between variables, and their predictions are difficult to explain to a non-technical stakeholder—a significant limitation in a policymaking context, discussed further in Section IV-B.

D. Bayesian Networks

Bayesian networks, formalised by Pearl [3], represent a set of random variables and their conditional dependencies as a directed acyclic graph in which each node holds a conditional probability distribution given its parents. They support probabilistic inference and are valued for their interpretability, since the graph structure itself states which factors influence which outcomes. Koller and Friedman [4] provide a comprehensive treatment of probabilistic graphical models, including structure learning and inference algorithms relevant to this work.

E. Evolutionary Computation and Differential Evolution

Evolutionary algorithms optimise a population of candidate solutions through iterative selection, recombination, and mutation, guided by a fitness function, without requiring gradient information about the search space. Differential Evolution (DE), introduced by Storn and Price [5], is a population-based evolutionary algorithm that is particularly effective on continuous, nonlinear, and non-stationary optimisation problems. Das and Suganthan [6] survey DE variants and their applications, with a later updated survey by Das et al. [7], and Deb et al. [8] introduced the NSGA-II non-dominated sorting procedure used here for cases in which policy objectives conflict.

III. SYSTEM ARCHITECTURE

Meta Society is a single-page web application built with React and TypeScript on Vite, with a routed set of pages: a population dashboard, a data-intelligence view, the AI Policy Advisor, the Simulation Lab, an alerts view, saved reports, and a profile page. The application is organised so that the simulation engine is a self-contained TypeScript module tree that runs entirely in the browser; it requires no server and no database, and every number it reports is a deterministic function of the population, the policy vector, the engine version, and the random seed. The AI Policy Advisor is a separate, optional Python service. Supabase provides an optional Postgres schema, scoped by authentication with row-level security, for persisting saved simulation runs.

**MetaSociety: system architecture**

Components and data flow as implemented in the repository (origin/main, commit 5739512). The simulation engine is deterministic and runs entirely in the browser.

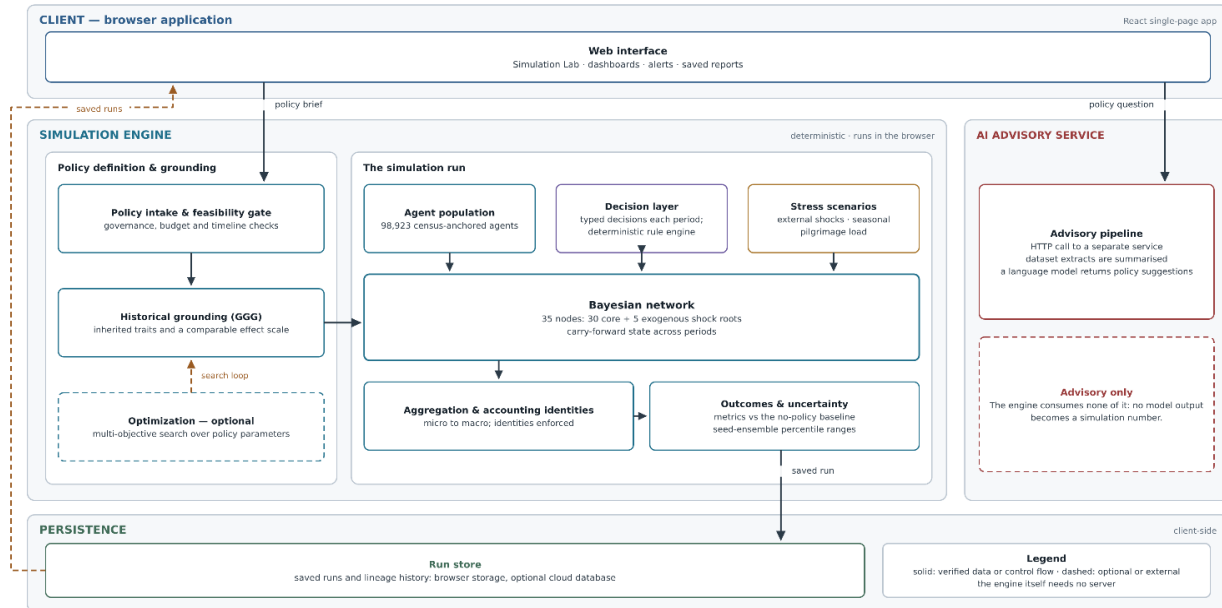


Fig. 1. Meta Society system architecture. The upper path is the AI Policy Advisor: the frontend sends a natural-language goal to the reasoning service, which returns structured candidate policies. The lower paths are the Simulation Lab: the census-anchored citizen population feeds a Bayesian network for outcome inference and a Differential Evolution search for policy-parameter optimisation, with results rendered in the Simulation Lab and persisted through the shared storage layer.

Two architectural decisions shape the rest of the paper. First, the AI Policy Advisor and the Simulation Lab are deliberately separated: the advisor proposes what to try, while the Simulation Lab computes what a policy would imply under the model. Second, the Simulation Lab never routes its outcome numbers through a language model. Section IV-A makes that constraint explicit, because a fluent but ungrounded number is, to a policymaker, indistinguishable from a computed one.

A. Technology Stack

Table I summarises the technology stack used at each layer of the system.

TABLE I.
TECHNOLOGY STACK OF THE META SOCIETY PLATFORM

Layer	Technology
Frontend	React 18, TypeScript, Vite, shadcn/ui (Radix), Tailwind CSS, Recharts, React Router
Simulation engine	Self-contained TypeScript modules executed in the browser (deterministic; no server dependency); a Web Worker path runs the same engine off the main thread
AI reasoning service	Python, FastAPI, LangGraph, Pandas; hosted LLM accessed through the Groq OpenAI-compatible API
Persistence	Browser local storage (verified path); optional Supabase Postgres mirror under row-level security
Testing	Vitest acceptance suites for the engine, the specification checks, historical grounding, shocks, seasonality, and page/persistence wiring

IV. THE SIMULATION LAB ENGINE

The Simulation Lab is the analytical core of Meta Society. It accepts a policy brief and a scenario configuration, rolls a probabilistic model forward over a synthetic citizen population, and reports outcomes together with a model-generated baseline. The engine is implemented in TypeScript and runs client-side; the same code path produces every reported number.



A. Design Principle: Decisive, Not Generative

The engine is decisive rather than generative. No language model sits anywhere in the path from “policy submitted” to “outcome shown”; an LLM may be used upstream (the AI Policy Advisor suggests what to try) or downstream (to phrase a caption around an already-computed number), but never to produce the number itself. The rationale is that an LLM asked to predict the effect of a policy will return a fluent, plausible figure that is not derived from any model, and such a figure is difficult for a non-specialist to distinguish from a computed result. Everything in the remainder of this section exists to give the Simulation Lab numbers that come from an actual, inspectable model.

B. Rationale: Replacing the Random Forest Baseline

An earlier prototype modelled agent reactions to a policy with a Random Forest classifier trained on town data clustered into ten groups, with a single representative agent standing in for each cluster. Three concrete limitations led the team to replace that approach. First, Random Forest requires labelled examples of the form (policy applied, outcome observed); the available data consists of town-level demographic and economic indicators, not a historical record of interventions and their measured effects, leaving too few labelled examples for a reliable classifier. Second, a Random Forest’s prediction is an aggregate of many trees and does not, by itself, indicate which inputs were responsible, which is a poor fit for a tool intended to build policymaker trust. Third, real policy effects propagate through a chain of intermediate variables—a subsidy changes household income, which changes spending, which changes demand, which changes inflation, which changes public sentiment—and a Random Forest treats each prediction as an independent function of its inputs rather than a sequential, compounding causal structure. A further limitation of the clustering step is that reducing many locations to ten representative agents discards within-group variation and provides no mechanism for representing variation among citizens within a single town—precisely the variation that determines how unevenly a policy is felt.

C. Census-Anchored Synthetic Population and Provenance

The engine operates on one agent per citizen for the pilot town of Pandharpur (Solapur district, Maharashtra). The population is generated deterministically from published Census 2011 totals [12] and stored as a structure of arrays so that the full population remains compact in memory. Population counts are allocated with the largest-remainder method and calibrated systematic sampling, rather than sampled independently, so that the verified totals are met exactly: total population 98,923 (male 50,645, female 48,278), 20,054 households, 11,151 children aged 0–6, effective literacy of 81.11% for males and 72.45% for females, 30,855 workers (25,162 male, 5,693 female), and 33 wards. A validation routine asserts each of these totals on every run.

Crucially, the system distinguishes what is measured from what is modelled. A field-level provenance ledger tags every generated field as verified Census data, estimated, modelled, or assumed. Population, sex split, households, children aged 0–6, literacy, the SC/ST shares, and worker counts are verified Census fields; income, sector, education, housing, and task exposure are modelled, because the Census does not record them. This ledger is rendered directly in the interface, so a modelled quantity is never presented to a user as measured data. Table II lists the verified anchors the generator is required to reproduce.

TABLE II.
VERIFIED CENSUS 2011 ANCHORS FOR PANDHARPUR REPRODUCED BY THE GENERATOR

Quantity (Census 2011)	Value
Total population	98,923
Male / female	50,645 / 48,278
Households	20,054
Children aged 0–6	11,151
Effective literacy (male / female)	81.11% / 72.45%
Workers (male / female / total)	25,162 / 5,693 / 30,855
Wards	33

D. Bayesian Network

Agent-level reactions are modelled with a discrete Bayesian network of **35 nodes: thirty core nodes plus five exogenous shock roots**. The five shock roots (ExternalHealthShock, ExternalEconomicShock, ExternalClimateShock, ExternalInfrastructureShock, ExternalSocialShock) are added by the external-shock system described in Section IV-E; each is a root with the domain none | mild | moderate | severe, defaults to none, and when it reads none every downstream shift is exactly zero, so a run that configures no scenario is unchanged. The registry is self-describing—each node spec carries its domain, parents, evaluation pass, table source and a machine-readable calibration tag—and buildBn validates the



graph (duplicate ids, missing parents, cycles, wrong-dimensional rows, probabilities outside $[0, 1]$, non-normalising rows) before use, so a malformed network fails loudly.

The network is evaluated in passes because town-level variables depend on population totals. A first pass maps policy inputs and agent attributes to per-agent outcomes; an aggregate pass converts population totals into discretised bands; a second pass produces town-level outcomes such as inflation, employment rate, GDP growth, and wages; and a third pass feeds those town outcomes back to the individual through sentiment, protest risk, and migration intent. This chain-income to spending to demand to inflation to sentiment to protest-is exactly the compounding causal structure the Random Forest could not represent. Table III summarises the passes.

TABLE III.
BAYESIAN-NETWORK EVALUATION PASSES (35 NODES: 30 CORE + 5 SHOCK ROOTS)

Pass	Content	Nodes
0 (input)	Policy roots and agent-attribute roots	PolicyType, PolicyIntensity, PolicyBudgetShare, PolicyDuration, IncomeClassPrior, AgeBand, EducationLevel, HousingQuality, TrustInGov
0.5 (aggregate)	Bands computed from population totals each period	AggregateDemand, AggregateSectorOutput, EmploymentAggregate, MigrationAggregate
1 (micro)	Per-agent causal outcomes	SectorDemand, IncomeClass, SkillRelevance, EmploymentStatus, SpendingCapacity, HouseholdStress, SectorOfWork, AgentSectorOutput, plus HealthInsurance/HealthBurden
2 (town)	Macro outcomes	Inflation, EmploymentRateBand, TownGDPGrowthBand, WageLevelBand
3 (feedback)	Town state $\rightarrow$ the individual	PublicSentiment, MigrationIntentBand, ProtestRiskBand
shock	Exogenous shock roots	ExternalHealthShock, ExternalEconomicShock, ExternalClimateShock, ExternalInfrastructureShock, ExternalSocialShock

Conditional-probability tables follow one rule: learn from the population wherever the data can speak, and use a documented prior only where it cannot. Variables that are observable in the generated population, such as income class, employment status, education, housing quality, sentiment, and skill relevance, have their tables counted directly from that population with Laplace smoothing. Only genuinely unobservable variables-latent market demand, sector output, the aggregate bands-and the policy parameters themselves are supplied by documented priors and additive log-weight shifts. Every table records its provenance, so the interface can state which parts of the model are estimated and which are assumed. Interventions are applied as interventions rather than as observations: applying a policy mutilates the graph by cutting the target node's incoming edges and fixing its value (a *do*-operator), so a policy effect is causal rather than a conditional read. Inference is ancestral forward sampling, which is exact for this query pattern because the evidence sits on root or near-root exogenous variables; a runtime guard enforces that precondition so the engine cannot silently produce biased samples. On every run the network's documented response directions are re-checked-for example, that a subsidy raises the probability of formal employment and that a high budget share raises the probability of high inflation-and the results are reported as part of the run's evidence pack.

E. Multi-Period Dynamics and External Shocks

The network is rolled forward one period at a time (three months per period), carrying agent state forward-employment, a savings stock, sentiment, trust, migration intent and skills-and each policy instrument has its own channel lag (for example, subsidy-employment 5 months, regulation 9, housing 14, education 30), so the trajectory does not move in lock-step. The no-policy path is produced by the same engine on the same population, so a result always ships with its own counterfactual baseline rather than a hardcoded series. Zone incidence is computed after the run: East, West, North and South are a partition of all 33 wards, so an output can distinguish a town-wide movement from a locally concentrated one.

Because a policy must survive more than a quiet decade, the Lab can also run external-shock stress tests. The user chooses normal conditions, a specific hypothetical event (start period, duration, severity), or a stochastic scenario in which a Poisson arrival model generates hypothetical event *timing*. A generated event is queued and dequeued in period order; on dequeue it temporarily intervenes on its exogenous network node ($\text{do}(\text{ExternalEconomicShock} = \text{moderate})$) for its active window,



none when the window ends). The consequences then propagate causally through the existing edges; no unsupported edge is added. Critically, the shock schedule is generated once from the scenario seed and replayed identically for both the no-policy baseline and the policy, so the comparison stays causally fair and a policy cannot look bad merely because it drew a worse event. Default arrival rates are labelled scenario assumptions, not sourced hazard rates, and the system describes these as hypothetical stress tests, not predictions.

F. Policy Representation: Structured Brief, Governance and Budget Constraints

The authoritative policy input is a structured brief-objective, problem statement, governing authorities, targeted population, budget line items and implementation timeline-not three sliders. The brief is validated before the engine will run it, and the gate is enforced *inside* the engine: runSimulation re-runs the brief and governance validation and throws a PolicyFeasibilityError if the policy is not feasible, so a blocked policy cannot be simulated even if the interface is bypassed. Three checks matter. First, governance competence: the registry names seven authorities (the Pandharpur Municipal Council, the Vitthal-Rukmini Temples Committee, the Pandharpur Development Authority, the Solapur District Administration, the Solapur Zilla Parishad, the Government of Maharashtra and the Government of India), and a policy's declared domain must be one a named proposing authority is competent to decide. Second, the budget line items must sum exactly to the stated total. Third, the timeline must be internally consistent.

The engine also bounds the total programme cost at ₹200,000,000. This is explicitly a **model assumption**, not a statutory ceiling: the governance registry records no verified statutory monetary ceiling for any authority, so the limit is documented as bounding the search space and is never presented as a legal limit. Where a stated total exceeds it, the brief reports that the engine will clamp it, and labels the limit as an assumption rather than a law.

G. Channel Dictionary: Implemented and Declared Channels

Policies are expressed over a channel dictionary that makes the gap between *declared* and *implemented* mechanisms visible. Eight channels are implemented-they map onto a mechanism the engine actually models-and five are declared: named in the dictionary, but with no engine mechanism yet, so a policy built only from declared channels grounds its characteristics through historical inheritance yet applies no causal effect, and the interface says so. Table IV lists both sets.

TABLE IV.
CHANNEL DICTIONARY: IMPLEMENTED VERSUS DECLARED CHANNELS

Status	Channels
Implemented (causal mechanism wired)	INCOME_SUPPORT, LABOR_MARKET, HOUSING, EDUCATION_SKILL, TAX_FISCAL, REGULATION, HEALTHCARE_ACCESS, PILGRIMAGE_FACILITIES
Declared (no engine mechanism yet)	DIGITAL_ACCESS, ENVIRONMENT_CLIMATE, FOOD_SECURITY, INFRASTRUCTURE, FINANCIAL_INCLUSION

A multi-channel policy selects several channels, but the network conditions each agent on exactly one policy family: PolicyType is a single categorical node. When a policy carries more than one implemented channel, the applied family is chosen by a fixed, documented precedence order (most-direct labour/employment mechanism first), so the result is order-invariant-reordering the channels cannot change the outcome-but the other implemented channels are **not independently composed**. This is a stated limitation, not a hidden one: runSimulation emits a warning naming the applied family and listing the declared families that are not independently composed. True additive composition of multiple instrument pathways remains future work.

H. Historical Grounding: Predecessor-Policy Inheritance (GGG)

Before a policy is simulated, a historical-grounding stage transfers characteristics from real predecessor programmes into the proposed Pandharpur policy. The registry is deliberately small and sourced: it contains national and state predecessors whose mechanism maps onto a channel the engine already models-for example MGNREGA (2005) and DAY-NULM (2013) for labour-market channels, PMAY-U for housing, PM-JAY (2018) for healthcare access, PM POSHAN (2021) for education and skills, PM-KISAN (2019) and a 2024 state income-support scheme, the contested Pandharpur pilgrimage corridor and Shaktipeeth Expressway, and the Wari toll exemption. Each entry records its authority, jurisdiction, period, mechanism, target population, context, seasonality, scale, delivery route, the network nodes its pathway plausibly touches, its documented outcomes with sources, its side effects, and an evidence-strength and comparability note. No effect size is invented: an outcome carries a magnitude only where a source reports one.



Matching is a transparent, deterministic weighted similarity over eight dimensions (mechanism 0.24, channel 0.22, causal nodes 0.18, target population 0.12, context 0.08, scale 0.06, duration 0.06, seasonality 0.04). A candidate must reach a minimum score and share the channel or show real mechanism, target or pathway overlap, so a housing policy does not inherit MGNREGA's employment effect merely because both are government programmes. Inherited traits are aggregated conservatively—a trait the child did not already carry needs stronger support to be added than one it did—and adapted to Pandharpur's context: a national programme's *mechanism* transfers, but its *scale* does not.

The stage then derives a single grounded effect scale, $\text{effectScale} = \text{clamp}(\text{comparability} \times \text{evidenceFactor} \times \text{scaleFactor}, 0.1, 1)$, where *comparability* is the best parent score (or a fixed floor lower than the weakest admitted parent when nothing matches, so absence of evidence is never rewarded), *evidenceFactor* encodes the strongest parent evidence, and *scaleFactor* penalises the number of administrative levels by which the widest predecessor exceeds a local application. This scale is passed to the network as a **continuous** multiplier on the policy log-shifts, so two materially different grounded policies no longer collapse onto the same coarse low/medium/high bands. It is a structural grounding of a causal parameter, not an output cap; there is no code path that clamps a headline number. The stage is deterministic, is not a language model, is not the evolutionary search, and is not backtesting, and it never copies a historical outcome into a result. Every run retains the full lineage from historical evidence, through matched predecessors and inherited traits, to the grounded parameters and the simulation.

I. Policy Search

Beyond evaluating a single policy, the engine searches the policy-parameter space. A candidate is encoded as a real-valued vector over intensity, total budget, duration, and a three-way allocation across housing, education, and employment; the ranges are listed in Table V. The search uses Differential Evolution (DE/rand1bin) [7] with dithering and Latin-hypercube initialisation, scored by the Bayesian network. When objectives conflict—for example, reducing unemployment while containing inflation—the search uses NSGA-II-style selection, ranking candidates by non-domination and then by crowding distance, and returns a Pareto front rather than silently selecting a single trade-off. To make the search's value visible rather than assumed, an equal-budget random-search control is reported alongside the front. Because a full-population evaluation is expensive, the search tier uses a reduced agent sample and coarser periods; this is the only approximation in the system, and the interface labels it as such. All randomness is seeded, so the same seed reproduces the same front.

TABLE V.
POLICY-SEARCH GENES AND RANGES

Gene	Range	Unit
intensity	0.05 – 1.00	share of instrument strength
budget	₹2,000,000 – ₹200,000,000	total programme cost (INR)
durationMonths	3 – 60 (multiple of 3)	months
allocation: housing, education, employment	0.02 – 1.00 (normalised to sum to 1)	relative weights

J. Decision Layer

At period boundaries the engine may pose typed questions to a decision layer: a *Choose* question selects among declared options, a *Score* question places the situation on a rubric, and a *Noul* question returns the probability that a stated condition holds. The default engine is a deterministic rule table; a model-backed engine can be configured through a server-side proxy, but the deterministic table is used for the search tier and as a fallback. The decision layer is never the source of a displayed number; it adjusts trajectory parameters, and the run records the number of calls, the mean confidence, the escalation rate below the automation threshold, and whether a fallback occurred, all of which are surfaced in the interface rather than hidden.

K. Accounting Identities, Their Timing, and the Evidence Pack

No macro quantity is set directly. Every headline metric is a stated summation over the agent population, and the accounting identities that connect those summations must hold or the run fails loudly. The timing matters for interpreting a result: during the trajectory the identities are asserted at each period **for the full-population path only** (the reduced-agent search tier deliberately skips them, which is why that tier is flagged as an approximation), and the engine re-checks them on the treated population at the final period, throwing an *AccountingViolationError* rather than returning numbers. The displayed metrics are then adjusted for seasonality and growth *after* those level computations, so the identities are asserted on the un-seasonally-adjusted levels and a seasonal adjustment cannot silently break or mask an identity.



Each run also assembles an evidence pack that groups its checks into population, network, aggregation, optimisation, reproducibility, and decision categories, and reports exactly which checks passed. Guardrail checks and risk alerts are presented with the result. A failed identity indicates that the run is not internally consistent and that its numbers should not be used; a failed direction check indicates that the network moved against its documented sign.

L. Data and Dashboard Layer

The dashboard and data-intelligence pages read from the same generated population the Simulation Lab simulates, so a figure shown on those pages can be traced back to the generator and to the provenance ledger. Earlier versions rendered invented sample data with no provenance, which was itself a criticism raised during evaluation; the mock data has been removed. Two series that the population cannot honestly support—a multi-year employment trend and a cost-of-living index over time—were deliberately dropped rather than fabricated, because the town is a single Census-2011 snapshot and the engine has no time series for it. They were replaced with cross-sectional facts the population genuinely supports, such as worker status by age band and sector composition.

V. THE AI POLICY ADVISOR

The AI Policy Advisor is an implemented, optional service that accepts a natural-language policy goal and returns a small set of structured candidate policies, each with a name, description, expected benefits, expected risks, and the population segment affected. It is independent of the Simulation Lab: it proposes policies, and the Lab evaluates them.

A. Multi-Stage Reasoning Pipeline

Rather than a single monolithic prompt, policy generation is modelled as a graph of reasoning stages using LangGraph [14]. The pipeline has three nodes. The first, dashboard extraction, loads six multi-domain datasets (economy, demographics, education, healthcare, infrastructure, and governance) with Pandas and reduces them to statistical summaries, highlights, and anomalies. The second, insight generation, prompts a hosted LLM with those statistics and the user's goal to produce a small number of natural-language insights about the town's current state. The third, policy generation, prompts the LLM with those insights and the original goal to produce a structured JSON array of candidate policies. Splitting the work keeps each LLM call focused on a narrower task than an end-to-end prompt would require, and makes the intermediate insights available as explanatory context alongside the final recommendations. LLM calls are wrapped with retry with backoff and a small in-memory cache for robustness.

B. Service, Integration and Credential Handling

The pipeline is exposed through a FastAPI service [13], which defines a health endpoint and a policy endpoint that returns the generated policies together with the chart extracts and any non-fatal pipeline warnings. The frontend calls the policy endpoint and renders each candidate's benefits, risks, and affected groups; warnings are surfaced to the user without blocking the response. The LLM provider key is held server-side in the backend environment and is never shipped to the browser. This distinction matters because any client-side, build-time environment variable is compiled into the public JavaScript bundle and would be readable by every visitor. The Advisor service is optional, and the rest of the platform runs without it.

VI. DATA FOUNDATION: FROM AGGREGATE RECORDS TO CENSUS-ANCHORED MICRODATA

During project evaluation, reviewers raised two related concerns about the underlying data: that the available corpus of town-level records across six domains was too small to credibly train and test agent-level models, and that a considerably larger record count would be expected for the project to be considered adequately grounded. Investigation surfaced a more fundamental issue than record count. Internal consistency checks showed that demographic sub-totals summed to the reported total population in fewer than six percent of the original records, indicating that the individual fields had been generated independently rather than as a coherent synthetic town.

Rather than patching the aggregate corpus or inflating it with independently sampled agents, the project replaced it for simulation purposes with a single town modelled at individual-citizen granularity. One agent per citizen of Pandharpur yields 98,923 agent-level records, which exceeds the scale bar named during evaluation, and every verified total is reproduced exactly by construction, which addresses the consistency concern directly. This is a deliberate change of granularity: a smaller number of internally consistent, individually varied citizens is more defensible for agent-level modelling than a larger number of aggregate records whose fields are mutually inconsistent. The canonical representation for simulation is the deterministic in-memory generator, not a stored table; a retained CSV artifact is kept only as a source/validation artifact, is never loaded by the Lab, and is linked to the generator and to the Census totals by a provenance



test. Fields that the Census does not record-income, sector, education, and housing-are modelled and tagged accordingly, so the system never presents a modelled quantity as measured data.

VII. PERSISTENCE, REPRODUCIBILITY, AND VERIFICATION

A. Persistence

A saved simulation run stores the policy name and type, the run identifier and seed, the headline and baseline metrics, the alerts, the engine versions, and an effectiveness score. Persistence works in two layers. The first and verified layer stores runs in the browser, so saving works and the saved reports view reads real runs without any backend. The second, optional layer mirrors the same run into a Supabase Postgres table when the project is configured with a Supabase URL and publishable key and a signed-in user identifier is available; a failure in that layer is reported to the user and never discards the local copy. A row-level-security migration defines three tables for profiles, simulations, and saved reports, each scoped to the authenticated user. Because the current sign-in is a local stub that does not provide a user identifier, the remote path is wired and type-checked but has not been exercised end to end; connecting real authentication is the remaining piece, and this is stated rather than implied.

The effectiveness score stored with a run is a deterministic ranking aid computed from the run's own metrics against its baseline. It aggregates weighted, normalised movements in employment, growth, happiness, inflation, inequality, and protest risk, with a midpoint denoting no measurable change against the no-policy baseline. It is a deterministic function of the result, not a forecast and not an LLM judgement.

B. Reproducibility

All randomness is seeded. The population generator hashes its inputs into a manifest, and the engine uses a seeded generator throughout, so the same population, policy vector, engine version, and seed produce identical results. The acceptance suite exercises this by running the same request twice and asserting equality of the point metrics, the baseline, and the trajectories. The seed is reproducibility metadata, not a user-facing policy parameter.

C. Verification Approach

Verification combines in-engine assertions with an automated acceptance suite. The engine asserts the Census totals and the accounting identities, validates the network's documented response directions, and emits the results as the evidence pack described in Section IV-K. The acceptance suite covers the population generator, the network directions, the accounting identities, determinism, the search, the zone partition, the policy-brief and governance gating, Wari seasonality, historical grounding, magnitude plausibility, and the wiring of the pages and persistence, so that the interface is shown to render computed output rather than mock samples.

VIII. RESULTS: VERIFICATION, UNCERTAINTY, AND SENSITIVITY

The results reported here are correctness, consistency and sensitivity results, not predictive-accuracy claims. They establish that the engine computes what it says it computes, that it is reproducible, and that its documented behaviour is stable under the sampling choices we examined; they do not establish that its magnitudes match reality, which is discussed in Section IX.

A. Population Fidelity

The generated population reproduces the verified Census anchors exactly: 98,923 agents; 50,645 male and 48,278 female; 20,054 households with a mean size of 4.93; 11,151 children aged 0–6; effective literacy of 81.11% for males and 72.45% for females; 30,855 workers (25,162 male, 5,693 female); and all 33 wards populated. Age-band and sex sub-totals are internally consistent for every agent, and enforced dependencies hold, including literacy that does not decrease across income classes and a strictly higher probability of working among literate than non-literate working-age agents. The generator supports substantially more behavioural cells than the ten representative groups of the earlier prototype, with no citizen silently dropped.

B. Network Behaviour

The network's documented response directions are validated on every run. Interventions are causal (*do*-operator by graph mutilation), and the checks confirm expected directions, such as a subsidy raising the probability of formal employment, a high budget share raising the probability of high inflation, and housing improving positive sentiment. A regression test also pins the causal consequence of the historical-grounding seam at the network level: the network's own response to the grounded band is strictly smaller than its response to the raw intensity/budget band, so grounding cannot be bypassed.



C. Accounting Identities and Reproducibility

The headline metrics satisfy their stated summation identities on every period of every full-population run, and a run that does not balance raises an accounting-violation error rather than returning numbers. Repeating a request with the same seed reproduces the point metrics, baseline, and trajectories identically.

D. Seed-Ensemble Uncertainty

Because a single deterministic run gives one trajectory, the Lab reports an uncertainty ensemble. The ensemble is a fixed set of deterministic seed runs: the chosen-seed trajectory plus a fixed internal number of additional seed variants (CONFIDENCE_ROUNDS = 12, so a default run reports **thirteen** members); the absolute count reported to the user is the ensemble size, not the constant. The displayed headline is the ensemble **median** rather than one arbitrary seed's value, which is why the point estimate is always bracketed by the reported interval. The reported quantities are an *improved share*—the share of simulated seed runs that moved in the improving direction—and an *empirical 5th-to-95th percentile* of the seed-run changes. These are deliberately **not** a probability, a confidence level, or a calibrated likelihood: a fixed ensemble of deterministic runs does not yield a statistical confidence interval, and the interface and its tests forbid that wording. Table VI records the ensemble for a representative twenty-four-month labour-market mission (₹12 crore, intensity 0.65) run at the default ensemble size; the baseline is the no-policy path from the same engine and population.

TABLE VI.

SEED-ENSEMBLE UNCERTAINTY FOR A 24-MONTH LABOUR-MARKET MISSION (₹12 CRORE; SEED 20260101). GDP IS A MODELLED BAND-DERIVED PROXY, NOT A FORECAST; ITS BASELINE IS ZERO BY CONSTRUCTION.

Metric	Baseline	Ensemble median (point)	Median change	Empirical 5th–95th pct	Improved share
Employment rate (%)	40.80	43.28	+2.48 pp	+2.31 to +2.71 pp	13/13
GDP-growth band proxy (%)	0.00	6.17	+6.17	+5.71 to +6.63	13/13
Mean income (₹)	3,195	3,407	+211.7	+203.4 to +231.1	13/13
Inflation band (%)	4.1	4.1	0.00	0.00 to 0.00	0/13 improved
Happiness index (0–10)	3.13	3.42	+0.29	+0.22 to +0.39	13/13
Protest risk (0–100)	36.65	36.61	−0.04	−0.11 to +0.02	11/13
Migration outflow (%)	0.107	0.100	−0.007	−0.021 to +0.006	9/13
Gini coefficient	0.521	0.510	−0.010	−0.013 to −0.007	13/13

Two honest readings follow. First, so-called “GDP growth” is a modelled band-derived proxy whose no-policy baseline is zero by construction, so the reported figure is a relative comparison against the engine's own baseline and not an economic forecast. Second, the intervals for protest risk and migration **span zero** (protest risk −0.11 to +0.02; migration −0.021 to +0.006), so the direction of those two outcomes is not resolved by the ensemble even though the headline median is slightly favourable. That caveat leads directly to the sensitivity analysis below.

E. Sensitivity Across Draw Counts

The network draws a finite number of Monte-Carlo samples per inference. We tested the stability of the inferred direction by re-estimating two migration quantities at four draw counts (100, 400, 1,600 and 6,400 samples), each with three fixed seeds, on the same population and network. Table VII reports the mean over the three seeds of the probability that an agent is in the “consider leaving” or “leaving” band, and of the rarer “leaving” band alone, under no policy and under a high-intensity labour policy.

TABLE VII.

MIGRATION-INTENT SENSITIVITY TO MONTE-CARLO DRAW COUNT (MEAN OF THREE SEEDS).

Draws	P(consider/leave) - none	P(consider/leave) - labour	P(leave) - none	P(leave) - labour
100	0.233	0.193	0.0100	0.0100
400	0.261	0.220	0.0025	0.0050
1,600	0.258	0.226	0.0029	0.0029
6,400	0.262	0.230	0.0038	0.0042



The coarser “consider or leave” band is directionally stable: the policy lowers it at every draw count. The rare “leaving” band is not. Its estimated policy effect changes sign across draw counts (the difference between policy and no-policy is zero at 100 draws, positive at 400, zero at 1,600 and positive at 6,400), and at the individual-seed level it is negative at 1,600 draws and positive at 6,400 for the same seed. This **migration-direction reversal** is a property of the estimator, not of the policy: when a probability of the order of a few tenths of a percent is estimated from a few thousand draws, sampling noise dominates the signal. The practical consequence, which we state rather than smooth over, is that the model can support a directional claim about *intent to leave or consider leaving*, but not about the magnitude or even the sign of the rarer *decision to leave* at these draw counts.

F. Historical Calibration Findings

A calibration ledger records real, Pandharpur-relevant interventions, what actually happened, and how well the simulator’s magnitude is calibrated against that outcome. Several entries have a known direction and no evaluated magnitude; those are tagged uncalibrated, and none of the recorded entries is fully calibrated. For the representative labour mission above, the historical-grounding stage matched two labour-market predecessors (DAY-NULM, MGNREGA) and derived a grounded effect scale of approximately **0.106**. That scale is below the roughly 0.4 threshold at which the network’s discrete policy dimensions saturate: below that point every weakly grounded local policy maps onto the same lowest policy band, and neither its own intensity nor its budget changes the policy state the network sees. The grounding therefore bounds the modelled effect conservatively—a small local programme cannot produce a national-scheme-scale headline—but it does **not** resolve differences within the weak-grounding regime. This is a documented resolution limit, not a calibrated finding: magnitudes remain model assumptions, and the report says so wherever a magnitude appears.

G. Seasonal Wari Effects

Pandharpur hosts the annual Ashadhi Wari pilgrimage. The seasonal model treats pilgrims as temporary pressure-pilgrims are never added to the resident population and represents the town’s condition as the resident population plus a seasonal civic-pressure layer parameterised by documented footfall, anchored to the reported ~32 lakh visits over three days (2026) and ~27–28 lakh on Ekadashi (2025). The seasonal policy carries its own effect and no generic income effect: a policy on the pilgrimage-facilities channel is tested to move mean income by less than five percent while still recording its seasonal effect, and it grounds its characteristics against the Wari toll-exemption predecessor. The sentiment and protest-risk sensitivities to civic pressure are documented model assumptions with asserted direction and uncalibrated magnitude.

H. Governance, Budget and Channel Constraints

The three feasibility checks behave as specified. A policy whose declared domain is not one a named authority is competent to decide is rejected; a budget whose line items do not sum exactly to its stated total is rejected; and a total above the ₹200,000,000 model assumption is reported and clamped, never presented as a statutory limit. The gate is enforced in the engine, so the rejection cannot be bypassed from the interface. On the channel side, the implemented-versus-declared distinction is enforced rather than merely documented: a policy built only from declared channels grounds its characteristics but applies no causal effect and says so, and a multi-channel policy composes only its highest-precedence implemented family, with the engine warning that the other declared families are not independently composed.

I. Policy Search and Integration

The Differential Evolution search converges toward a known optimum on a controlled test, is seed-deterministic, produces a correct non-dominated front on a multi-objective test, and is reported together with an equal-budget random-search control so that the value of the search can be judged rather than assumed. On the integration side, the Dashboard, Data Intelligence, Alerts, Saved Reports, and Simulation Lab pages render computed population and engine data, the mock samples used in earlier versions have been removed, and saved runs persist through the browser layer and can be optionally mirrored to Supabase when it is configured.

IX. LIMITATIONS AND THREATS TO VALIDITY

The system is deliberately explicit about its boundaries, and a permanent model-limitations panel is shown in the Simulation Lab rather than buried in documentation. The principal limitations are as follows.

- **Uncalibrated magnitudes; insufficient historical calibration.** Response directions are validated, but magnitudes are not calibrated against an evaluated real programme, and no calibration-ledger entry is fully calibrated: the recorded interventions carry a direction and, in most cases, no evaluated magnitude. Anchoring income and sector to official household-consumption or district-industry tables is the work that would make sizes claimable. Instrument effects are correspondingly an uncalibrated prior shift applied every period.
- **No external validation.** The model has not been validated against an independent outcome dataset, nor reviewed by external domain experts, and it is not backed by a population survey of assumptions. Its agreement with the



historical cases it replays is directional at best, and a direction-only agreement is never presented as an accurate prediction.

- **Sensitivity limits the finest claims.** As Section VIII-E shows, the direction of a rare outcome (migration decision) is unstable across draw counts, so the model can support directional claims about broader bands but not about rare-event magnitudes or signs at the draw counts tested.
- **Resolution limit in historical grounding.** Below a grounded effect scale of roughly 0.4, the network's discrete policy dimensions saturate, so weakly grounded local policies are treated alike; correcting this would require a higher-resolution or continuous policy dimension.
- **Modelled fields.** Income, sector, education, and housing are modelled, not measured; only population, the sex split, households, children aged 0–6, literacy, the SC/ST shares, and worker counts are verified Census figures.
- **Single town, closed economy.** The model covers one town. It does not represent migration between towns or spillover from neighbouring economies. External economic shocks can be stress-tested as hypothetical events, but the model has no recovery dynamics beyond an event's window and no real-world hazard rates.
- **Assumed geography.** The ward-to-zone mapping uses population-balanced contiguous ranges, which is an assumption rather than sourced ward geography.
- **Ordinal inflation.** Inflation is reported from a three-value band computed from aggregate demand, so it can take only those three values and should be treated as an ordinal band rather than a point estimate.
- **Channel composition and policy-duration edge cases.** Only one implemented channel family drives the network per run; multiple implemented channels are not additively composed. PolicyDuration is declared but has no outgoing effect on the network.
- **Persistence caveat.** The browser storage path is the verified one; the Supabase mirror is wired and type-checked but has not been exercised end to end because the current sign-in does not supply a user identifier.
- **Not a forecasting service, not peer-reviewed.** The platform is a counterfactual, directional comparison tool and carries no claim of predictive accuracy. Unlike the interactive scenario explorers that inspired its presentation, it has not been reviewed by external domain experts.

X. CONCLUSION AND FUTURE WORK

This paper has presented Meta Society, a smart-governance platform that combines an LLM-based policy advisor with an inspectable simulation engine. The AI Policy Advisor is implemented and exposes a three-stage reasoning pipeline, orchestrated with LangGraph and served through a FastAPI backend, that turns a natural-language goal and multi-domain town statistics into structured candidate policies. The Simulation Lab is likewise implemented: it generates one agent per citizen for a real town so that published Census 2011 totals are reproduced exactly; it models reactions with a 35-node Bayesian network (thirty core nodes plus five exogenous shock roots) whose tables are counted from the population wherever possible and whose interventions are causal; it grounds each policy's effect against real historical predecessors; it stress-tests policies under hypothetical external shocks applied identically to the policy and its baseline; it rolls the model forward with distinct channel lags and a model-generated counterfactual baseline; and it searches the policy-parameter space with Differential Evolution and NSGA-II selection, reported alongside a random-search control. Every headline metric is a summation over the agent population, and every run re-checks the accounting identities, the documented network directions, and reproducibility as an evidence pack. We have also documented why a regression-based agent model was replaced, how the project moved from an aggregate corpus of questionable internal consistency to census-anchored microdata, and how the engine's uncertainty and sensitivity behave.

Future work follows directly from the limitations. The highest-priority item is calibration: anchoring income and sector to official household-consumption and district-industry sources so that the model's magnitudes, and not only its directions, become claimable, together with external validation against an independent outcome dataset. Complementary work includes replacing the assumed ward-to-zone mapping with sourced ward geography; raising the resolution of the grounded policy dimension so that weakly grounded policies are no longer treated alike; composing multiple implemented channels additively rather than selecting one family; auditing the literacy definition against the primary Census publication; completing real authentication so that the Supabase persistence path can be verified end to end; and extending the population and network to additional towns so that the platform can compare policy outcomes across local contexts.

REFERENCES

- [1] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, suppl. 3, pp. 7280–7287, 2002.
- [2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.



- [3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann, 1988.
- [4] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.
- [5] R. Storn and K. Price, "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [6] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [7] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [9] J. Wu, B. Hu, J. Zhang, and D. Fang, "Multi-agent simulation of group behavior in E-Government policy decision," *Simulation Modelling Practice and Theory*, vol. 16, no. 10, pp. 1571–1587, 2008.
- [10] J. Liu, C. Chu, Y. Zhao, G. Aoki, and Z. Xiao, "Agentic AI for sustainable development: Leveraging large language model-enhanced agent-based modeling for complex policy strategies," *Emerging Media*, vol. 3, no. 3, pp. 401–413, 2025.
- [11] L. Hughes *et al.*, "AI agents and agentic systems: A multi-expert analysis," *Journal of Computer Information Systems*, vol. 65, no. 4, pp. 489–517, 2025.
- [12] Government of India, "District Census Handbook, Solapur-Census of India 2011," Office of the Registrar General & Census Commissioner, India, 2011.
- [13] Supabase Inc., "Supabase documentation." [Online]. Available: <https://supabase.com/docs>
- [14] LangChain Inc., "LangGraph documentation." [Online]. Available: <https://langchain-ai.github.io/langgraph/>