Identifiability screening · policy simulators

Raunak Mondal · Discord: raunakmondal07

1 · PROBLEM

what and why

Agent-based policy simulators expose dozens of tunable knobs, and the standard move is to calibrate all of them against data using simulation-based inference. The trouble is that many knobs are not recoverable even in principle: some are declared in the config schema but never read by any transition rule, and others move outcomes by less than the seed-to-seed noise. Calibration returns a clean posterior anyway, because a neural posterior estimator handed an uninformative parameter simply reproduces the prior — and, critically, that failure passes the field's standard validation check. Simulation-based calibration measures rank uniformity, and a posterior that returns the prior is perfectly rank-uniform, so the diagnostic everyone runs is exactly the one that cannot see this. We propose to build and validate an identifiability screen that runs before calibration and certifies which parameters are worth inferring.

2 · APPROACH

model / method, and why this one

We use a two-stage design: screen first, then calibrate only the certified subspace. The screen combines a common-random-numbers perturbation sweep (same seed, knob swept across its prior support, so any nonzero output delta is causal rather than noise) with a Morris/Sobol elementary-effects stage and a power analysis that reports the smallest effect size detectable at a given simulation budget. Each parameter comes back labeled inert, weak, or identifiable, with the label backed by a measured effect size and a detection threshold rather than a heuristic. Because simulation is the scarce resource (§5), the screen spends its budget sequentially rather than flat across the prior, concentrating runs where the weak/inert boundary is still ambiguous. Calibration then runs amortized neural posterior estimation — a neural spline flow over the certified subspace, with a learned set-transformer embedding of the raw event stream replacing hand-crafted summary statistics. We validate with SBC rank histograms, TARP expected-coverage tests, and posterior contraction (1 − Var_post/Var_prior). We chose this over running SBI alone because SBI alone cannot detect its own failure here, which is the technical crux: SBC is necessary but not sufficient, and the inert case is precisely where the insufficiency bites. We chose it over pure structural identifiability analysis (Lie-symmetry or profile-likelihood methods) because those need a closed-form likelihood, which a discrete-event simulator does not have. The screen is deliberately simulator-agnostic — it treats the simulator as a black box with a seed, so it transfers to any ABM. Preliminary results. We ran the screen on rfsim, our 6.3k-line research-field simulator (13 transition modules, 60-month horizon, ~860 events/run). A 400-draw prior-predictive sweep plus a common-random-numbers sweep over 11 policy parameters gives: Parameter Δ exposure (CRN, 0.05→0.95) max abs. Spearman ρ Verdict open_science_level +0.221 0.864 identifiable repository_control_strength –0.036 0.581 identifiable pre_travel_review_strength +0.014 0.787 identifiable electronic_monitoring_strength –0.014 0.126 weak audit_frequency –0.007 0.115 weak access_control_strength 0.000 0.105 inert need_to_know_strength 0.000 0.056 inert export_review_strength 0.000 0.055 inert investigation_capacity 0.000 0.052 inert cloud_control_strength 0.000 0.049 inert mitigation_strength 0.000 0.112 inert Six of eleven parameters are exactly inert — zero output change across the full prior range under a fixed seed. Source inspection confirms the mechanism: only 4 of 13 transition modules reference the policy object at all. Two event types (AccessGranted, NoncompliantTransfer) have zero variance across all 400 prior draws. Prior-predictive spread on the headline outcome is sd = 0.054, so the two "weak" parameters sit at roughly a quarter of the noise floor. Naive calibration of this simulator would have produced eleven confident-looking posteriors, six of them meaningless.

3 · DATASET

source, size, preprocessing

The primary data is simulator-generated: a corpus of 106 (θ, event-log) pairs from rfsim, $\theta \in$ R18 drawn from a uniform prior over policy and scenario parameters. Preprocessing converts each event log into a binned multivariate count series over 12 event types, plus asset-level exposure and detection timings — the set-transformer arm consumes the raw stream instead, held in a packed columnar format at ~14 KB per run (see §5 for why the native JSONL representation is not viable at this scale). Real-data anchoring comes from OpenAlex: 44 research subfields across AI/robotics, semiconductors, and synthetic biology, with roughly 13k authors, 3k institutions, and 500 topics already ingested and persisted in Neo4j by the existing pipeline. This supplies the observable side of the calibration target (publication cadence, cross-country collaboration rates). For generalization we run the screen on two external, mature ABMs with published calibration studies — Covasim and a Sugarscape-family economic model — to establish that weak-but-nonzero identifiability is the general case and rfsim's exact-zero result is the extreme end of a spectrum, not an artifact of one codebase.

4 · METRICS

how you'll know it worked

Screen precision/recall against ground-truth inertness. Ground truth is available two ways: static instrumentation of which knobs each module reads, and synthetic parameters injected with known effect sizes. Detection-power curve: smallest true effect size the screen flags, as a function of simulation budget N, reported for sequential vs. flat prior sampling. This is the number that makes the tool usable by others — a practitioner needs to know what their budget buys, not what a cluster buys. The headline demonstration: SBC rank histograms pass ($p > 0.05$, uniform) on inert parameters while posterior contraction sits at ≈ 0 and the screen flags them — the diagnostic gap, shown rather than argued. Posterior quality on the certified subspace: TARP expected-coverage error within ±2% of nominal; contraction ratio ≥ 0.5 on parameters labeled identifiable. Embedding ablation: learned event-stream embedding vs. hand-crafted summaries, measured by contraction at matched budget. Downstream consequence: whether policy rankings derived from screened vs. unscreened calibration actually differ.

5 · COMPUTE

training time, memory, dependencies

Sized against the actual allocation — 1× MI300X (192 GB VRAM), 240 GB system RAM, 20 cores, 2 TB disk, 100 MB/s — from measurements on the current codebase rather than estimates. The binding constraint is core count, not the accelerator. Simulation is a pure-Python discrete-event loop and cannot be moved to the GPU. In alternating A/B trials we measured 253 ms per 60-month run single-core, cut to 181 ms (1.4×) by memoizing the ontology traversals that dominate the profile — a change verified to produce bit-identical event logs (matching SHA-256 over every event field) with the test suite still green. At roughly 100 runs/s sustained across 20 workers: Corpus Wall-clock, 20 cores Packed storage 106 runs 2.8 hours 14 GB 7 28 hours 140 GB 10 runs A 107 sweep is affordable once, overnight; it is emphatically not affordable per ablation. This is the single fact that shapes the design. Rather than sampling the prior flat and hoping the budget covers the smallest effects, we allocate simulations sequentially — the surrogate proposes the next θ batch, concentrating runs near the decision boundary between "weak" and "inert." That is also what makes the detection-power curve budget-aware, which is the form practitioners actually need, since most people running an ABM do not have a large cluster either. Storage is where the naive plan breaks. rfsim writes event logs as JSONL at a measured 1,518 bytes per event, and a 60-month run emits ~860 events — 1.3 MB per run, so a 106-run corpus in native format is 1.31 TB, two-thirds of the allocation, before checkpoints or the OpenAlex snapshot. We therefore store the corpus packed columnar (int32 time, event type, asset, actor) at ~14 KB per run, keeping even a 107 corpus at 140 GB. Raw JSONL is retained only for a 104-run audit subsample so provenance checks remain possible. With Parquet-converted OpenAlex (~200 GB net), Neo4j, and flow checkpoints, total footprint lands near 440 GB. RAM and VRAM are both abundant, and

each earns its keep once. A simulation worker peaks at a measured 23 MB resident, so 20 workers are nowhere near the 240 GB ceiling — memory never throttles simulation. The payoff is downstream: the packed corpus stays fully resident during flow training, so there is no streaming dataloader and no disk I/O in the inner loop. The same logic justifies the 192 GB card. A neural spline flow over 18-dimensional θ needs under 5 GB, so the capacity is not for the model — it is that a 140 GB packed corpus fits in HBM, and with all 20 cores saturated by simulation we cannot afford a host-side loader competing for CPU. Total accelerator demand is modest, roughly 40–80 GPU-hours across the architecture and seed sweep, interleaved with simulation rather than serialized after it. Bandwidth matters once, at the start. The full OpenAlex snapshot is ~330 GB compressed and free from S3; at 100 MB/s that is a few hours, one time, after which the pipeline no longer depends on a rate-limited API key for the temporal analysis. Dependencies: ROCm PyTorch 2.x, sbi, zuko, DuckDB, numpy/scipy/scikit-learn. No custom kernels, no multi-node training, no distributed setup.

6 · DELIVERABLES

model, paper, demo

1. simscreen — a standalone, simulator-agnostic identifiability screening library (black-box interface: a callable plus a seed), released open-source with the detection-power methodology documented. 2. A certified calibration of rfsim, with posteriors reported only over the parameters that earn them, and the inert set reported as a finding. 3. A paper targeting ICLR 2027 (deadline late September 2026), with AISTATS 2027 as the fallback venue. The framing is the diagnostic gap: SBC cannot detect unidentifiability, and here is what to run instead. 4. The simulation corpus and full experiment configs, released so the numbers reproduce, with per-run git SHA and config hash. 5. A demo: point the screen at an unfamiliar simulator and get back a labeled parameter table in minutes.

7 · RISK

what breaks first, and plan B

what breaks first, and plan B What breaks first: the inert parameters in rfsim may read to reviewers as our own incompleteness rather than a general phenomenon. This is the main risk and we take it seriously. Plan B is the external-ABM arm, which is why it is in the plan from day one rather than as a stretch goal — if mature, published, widely-used simulators also show parameters below their own noise floor, the contribution stands independent of rfsim. And if they do not, "the screen catches real wiring bugs in policy simulators, here is one it caught" remains a genuine and useful result, just a smaller one. Second risk: the SBC-insensitivity point may be known folklore in the SBI community even if it is not crisply written down. We run a focused prior-art check in week one, before spending the GPU budget. If it is already published, we pivot the emphasis to the screening tool and the detection-power methodology, which are contributions regardless. Third: the simulation budget may prove too tight for the smallest effect sizes, since 20 cores caps us near 107 runs per overnight sweep. Mitigations are already measured: the ontology memoization above bought 1.4× for free, the remaining profile hotspot (process_rules._observed_classes_for_asset, 13% of runtime) is cacheable on the same argument, and the sequential allocator exists precisely to spend a small budget well. If the floor still sits above the interesting effect sizes, that floor is a reportable result — it tells ABM practitioners what their own hardware can and cannot certify. Fourth: the learned event-stream embedding may not beat hand-crafted summaries. That ablation is reported either way and the screen does not depend on the outcome. Fifth: ROCm friction. We stay on standard ops and smoke-test on a single GPU before launching the sweep. The device-selection code is already backend-agnostic. Interpretation note: rfsim’s outputs are simulated actors and hypothesized pathways, not claims about real individuals or institutions; this caveat is enforced in the codebase and carried into all released artifacts.