WHEN DO LLM AGENTS REPRODUCE HUMAN SOCIAL STRUCTURE? AN IDENTIFIABILITY AUDIT OF GENERATIVE SOCIAL SIMULATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly used as agents to simulate human societies, and such simulators are often evaluated by whether the social structures they generate resemble real ones. However, realistic-looking social structure does not necessarily imply realistic agent behavior: the observed structure may arise from simulator architecture or even from superficial features of the prompt. We introduce the **Simulator Identifiability Audit**, a framework for testing whether emergent social structure is actually attributable to an LLM agent policy. The audit compares each simulator with an architecture-matched random twin that replaces LLM choices by chance while holding the rest of the simulator fixed, evaluates both at matched edge counts, and permutes labels to separate effects of simulated individuals from effects of their representations.

Across six social-simulation architectures, real and synthetic social networks, three commercial LLM providers, and an open-weight model, we find three consistent patterns. First, a minimal persona-agent architecture distilled from recurring design choices in published simulators is structurally silent: none of its evaluated network properties differs detectably from its random twin. Second, exposing peer traits makes the policy detectable, improving clustering through convergent selection of the same trait-attractive peers rather than homophily, while degrading degree structure. Third, apparent successes can be artifacts: memory-driven community formation disappears at matched edge counts, while names and list positions systematically bias partner selection, with female-associated names selected up to $8.7\times$ as often as male-associated names in a synthetic-name probe.

We release **ClassNetBench**, a benchmark for separating agent effects from simulator and presentation effects in generative social simulation.

1 INTRODUCTION

LLM-based generative agents produce believable social behavior (Park et al., 2023) and are moving from demonstration to application. In education, simulated classrooms are proposed for studying pedagogy and piloting interventions (Zhang et al., 2025; Lin et al., 2026), validated by *believability*: human judges cannot tell simulated episodes from real ones. When such a simulation then reproduces the structure of a real network—its communities, its popular and isolated members—the reproduction is read as evidence that the agents *behaved socially*, and the applications that matter (finding isolated students, composing groups, forecasting interventions) are built on that reading. This paper is about what that reading requires. Its question is not whether LLM agents *can* be social, but:

By what criterion may an observed social structure count as evidence that the agents produced it—rather than the simulator, or the words it happened to use?

Our answer is eliminative. We do not define social behavior; we identify what else could have produced the structure and ask whether it has been ruled out. Three rivals stand between structure and agent: the *machinery* (accumulating edge weights and keeping the strongest ties yields community-like structure even from *random* choices); the *presentation* (models prefer certain labels and list

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

positions regardless of who carries them); and *impossibility* (agents shown bare identifiers—no channel through which persons become knowable to one another—cannot select on attributes, however capable the model). Until each is excluded the observation says nothing about the agents; for the minimal template, we show, none is.

Making the criterion operational (Fig. 1). The agent policy's behavior is the quantity of interest; the distribution of graph statistics is the observable. The central control is a *volume-matched random twin*: the same simulator, score scale, and selection volumes, choosing at random. Against it we ask, per structural component: Is the policy *detectable* at all? Does a detected effect *improve or degrade* fidelity? Is an improvement *attributable to the simulated individuals* rather than to the labels they wear (Defs. 1–2)? Three devices carry the argument—the twin, comparison at matched edge counts, and label permutation; everything else (shuffle and scale-matched twins, equivalence margins, density sweeps) is a robustness check and lives in Appendix D. "Identifiability" is operational and null-relative throughout: distinguishability from the twin, not classical identifiability (Appendix D); the only global claims are the architecture-level impossibility results (Props. 1–3).

Testbed. Classroom friendship networks are an ideal proving ground: small (n=23–50), trait-instrumented, with robust regularities—heterogeneous degrees, communities, clustering, extraversion–degree coupling (Selfhout et al., 2010; Feiler & Kleinbaum, 2015). The program spans a private four-classroom study (160 students), two public SIENA datasets, and planted-mechanism synthetics; six architectures toggling information, choice-volume, memory, and backbone axes; and LLMs from three providers plus an open-weight model ($\sim$350 runs, $\sim$90k decisions, plus a 27.2k-decision probe).

Contributions. (1) *Framework* (§3): a three-stage separation of policy *detectability*, *fidelity effect*, and *social attribution*; the volume-matched random twin (and a stronger decision-level shuffle twin) as architecture-matched nulls; equivalence margins; and five architectural principles (P1–P3 proved, P4–P5 empirical) with pre-specified predictions. (2) *The attribution matrix* (§5): LLM policies do not reproduce edge volume—with memory or a backbone they form 35–56% fewer distinct ties than the observed network while their twins form nearly all—so every comparison is made at matched edge counts. There, T0, a minimal architecture distilled from recurring design choices in published simulators (Appendix H) and the audit's lower bound, is structurally silent, as P1–P3 predict; where peer traits make the policy detectable, they improve exactly one component—clustering, via *convergent popularity selection* rather than homophily, following persons rather than labels—while degrading degree, assortativity, and isolation; memory instead concentrates ties, and its apparent community gain is an edge-volume deficit in disguise. (3) *Presentation artifacts* (§6): to our knowledge the first systematic characterization of identifier preference and positional bias in *network-generating* LLM agents (both known in ranking and recommendation (Sakunkoo & Sakunkoo, 2025; Bito et al., 2025)), localized by a two-space permutation test—*the model prefers labels, not students*—with fairness-relevant loadings of human names. (4) *ClassNetBench* (§7): the audit as a benchmark, requiring no private data.

2 Related Work

LLM social simulation and its critics. Generative-agent systems simulate social networks and media dynamics (Park et al., 2023; Gao et al., 2023; Törnberg et al., 2023) and educational settings (Zhang et al., 2025; Lin et al., 2026), with surveys in Mou et al. (2026); Gao et al. (2024). A critical literature holds that role-play plausibility does not imply behavioral validity and that collective outcomes are often dominated by protocols and priors rather than the model (Li & Tao, 2026; Madden, 2025); audits find high sensitivity to implementation choices (Ye et al., 2026), operational validation against platform data exists (Tomašević et al., 2025), and validation is named the field's central open problem (Larooij & Törnberg, 2025). Emerging audits target behavioral paths and action provenance (Pandey & Jajoo, 2026; Liao, 2026). Closest in spirit is SimBench (Hu et al., 2026), which standardizes whether LLMs reproduce *human behavior* across 20 datasets. We ask a complementary, upstream question: when a multi-agent simulator appears to reproduce social *structure*, is that structure attributable to the agent policy at all? SimBench audits the map from human behavior to LLM behavior; we audit the map from LLM policy, through the simulator architecture, to emergent structure. Takami & Haruno (2026) introduced a calibrated rule-based and LLM-agent simulation framework for classroom social networks and evaluated its structural fidelity; the present audit asks instead whether apparent fidelity is attributable to the agent policy at all.

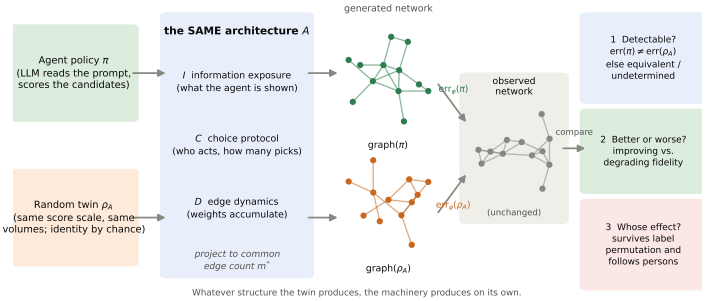


Figure 1: **The Simulator Identifiability Audit.** A simulator is $S = (A, \pi)$: architecture $A = (I, C, D)$ plus agent policy π. The policy and its random twin (same score scale and selection volumes, choosing at random) run through the *same* architecture; each generated network is projected to the cell's common edge count m^* and scored as component errors against the unchanged observed network. Stage 1: does the policy change the error (Def. 1)? Stage 2: in which direction? Stage 3: does an improvement survive label permutation and follow persons (Def. 2)?

Validation standards for generative network models. Network science accepts a generative model when it beats constrained randomness: Erdős–Rényi (Erdős & Rényi, 1959) controls size and density; degree-preserving rewiring controls the degree sequence (Maslov & Sneppen, 2002); statistical models (SAOMs, ERGMs) give attribute-informed references (Snijders et al., 2010); pattern-oriented modeling demands multi-pattern fit (Grimm et al., 2005). We adopt this standard (conditioned references as baselines or diagnostics, never competitors) and add the two controls LLM policies require.

3 The Simulator Identifiability Audit

Two parts of a simulator. A generative social simulator is a pair $S = (A, \pi)$. The *architecture* $A = (I, C, D)$ is everything that is not the agent's judgment: information exposure I (what a decision context reveals—own attributes, alter attributes, interaction history, and in what surface form), choice protocol C (who acts when and selects how many alters), and edge dynamics D (how choices become graph state: weight accumulation, new-edge gating, deduplication, projection). The *agent policy* π maps decision contexts to choices—an LLM, a rule, or random.

What is measured. Evaluation computes six structural components Φ as discrepancies from an observed network: degree-distribution KS, modularity, trait–position coupling $r(\text{focal}, \text{degree})$, clustering, degree assortativity, and isolate rate. Components are kept separate. All arms are compared *at a common edge count*: within each (architecture, site) cell every generated network—policy, twin, references—is projected to its m^* strongest edges, $m^* = \min\{m_{\text{obs}},$ smallest distinct-edge count among the cell's policy and twin runs$\}$, while the observed network is left unchanged. The question is thus "at equal edge volume, which generated network is closer to the observed statistics?"—necessary because concentrated LLM policies produce far fewer distinct ties than their twins (Appendix D). As m^* differs across architectures, absolute errors compare within a cell (robustness checks: Appendix D).

The twin. For any architecture A, the *volume-matched random twin* τ_A shares (I, C, D) exactly—the same candidate pools, selection protocol, memory bookkeeping, and edge dynamics under the same seeds—and replaces the policy's scores by chance (Algorithm 1, Appendix D). Where the architecture fixes the selection volume (top-k; T5's quota), the twin's volume is identical by construction and its scores are Uniform$(0, 1)$; drawing them from the model's own score sample instead changes no headline verdict (17/18 cell verdicts identical; the exception is T5 isolation, Appendix D). Where volume is score-dependent (threshold architectures T1, T2, T4), the twin draws scores i.i.d. from the audited model's empirical score distribution and per-decision selection counts from the audited runs of the same cell, so that neither score scale nor choice volume can separate the arms. It randomizes *who is chosen* and nothing else. Whatever structure the twin produces, the machinery produces on its own.

Definition 1 (policy effect on component error)**.** *In words:* does replacing the agent's choices by chance change how far the network is from the real one? *Formally, let τ_A be the architecture-matched random twin of π under A, and $\mathrm{err}_\varphi(\cdot)$ the distance of component φ from its observed value at matched edge count. The policy has a* detectable effect on the φ-error *under A on data $\mathcal{D}$ if $\mathrm{err}_\varphi(S(A,\pi)) \neq \mathrm{err}_\varphi(S(A,\tau_A))$ with statistical support (two-sided site-stratified permutation test, multiplicity-corrected; §4). A detectable effect is* fidelity-improving *if $\mathrm{err}_\varphi(S(A,\pi)) < \mathrm{err}_\varphi(S(A,\tau_A))$ and* fidelity-degrading *otherwise. When no effect is detected, the* mean error *is* twin-equivalent *if the 90% confidence interval of the pooled difference lies within a declared margin, and* undetermined *otherwise. The object tested is the fit error, not the statistic: "detectable" throughout means an effect on the evaluated error; raw statistics can shift without it (two-sided raw-value tests, Table 6).*

Two cautions. Detectability is not merit: a policy can separate from its twin by making the network *less* realistic (T5's degree structure below). And twin-equivalence bounds only the *mean evaluated error*, not the generated structure or mechanism; without equivalence, a non-detection is just that.

Definition 2 (label-permutation-robust improvement)**.** *In words:* does the improvement survive when we shuffle which label each simulated person wears? *A fidelity-improving effect on φ is* label-permutation-robust *if the* label-permuted *policy (labels permuted per run, traits fixed) itself still improves on the twin (magnitude and CI reported) and the effect is not carried by presentation position. This is necessary, not sufficient, for attributing the improvement to simulated individuals: a policy clustering on label* similarity *(shared prefixes) keeps aggregate modularity or clustering under relabeling while membership changes. Attribution to persons additionally requires that popularity follows persons across relabelings (two-space consistency, §6); we call an improvement* person-attributable *only when both hold.*

The product: a three-stage attribution matrix (architectures $\times$ components). Stage 1, detectability; stage 2, fidelity effect; stage 3, attribution to simulated individuals rather than their labels. Surviving all three stages is what we mean by evidence that a structure is the agents': each stage excludes one rival source.

Five architectural principles link a simulator's design to which components *could* be identifiable at all (P1–P3 proved as necessary conditions, Appendix C; P4–P5 empirical; each with a pre-specified prediction tested in §5–6):

- **P1 (information).** If I shows candidates as bare labels under attribute-exchangeable presentation (labels carrying no attribute information, e.g. a uniformly random relabeling), no policy can condition on *alter* attributes (selection is exchangeable over candidates given surface forms; Prop. 1): homophily, attribute assortativity, and alter-driven clustering are unavailable to it, while effects of the ego's *own* attributes are not excluded.
- **P2 (volume).** Under uniform mandatory top-k with deduplicating D (repeats become weight, not degree) and independent, time-homogeneous selection, concentrating an agent's choices lowers its expected number of distinct partners (Prop. 2); convergent choice across agents can additionally concentrate in-degree (§5).
- **P3 (memory).** A policy whose context excludes interaction history (and that carries no state of its own across rounds) cannot express reinforcement-driven community formation (Prop. 3)—a statement about which mechanism is expressible, not a guarantee that memoryless modularity gains are machinery (a policy can still cluster on static features, including labels), which is why the permutation control is also required.
- **P4 (concentration).** Under deduplicating D, local closure and degree spread compete through the policy's concentration; lifting one constraint redistributes error, and axes need not compose.
- **P5 (surface).** Policies carry preference mass over labels and positions; entity-level claims without decoupling controls are confounded.

4 Audit Protocol and Experimental Program

Protocol (full version in Appendix D). Project every generated arm in a cell to the cell's common edge count m^* and evaluate components separately. Compare the policy against conditional Erdős–Rényi

Table 1: Architectures: toggled axes and protocol.

	I traits	C volume	M memory	B backbone
T0 minimal	–	top-k	–	–
T1 +volume	–	threshold θ	–	–
T2 +traits	✓	threshold θ	–	–
T3 +memory	–	top-k	✓	–
T4 all	✓	threshold θ	✓	–
T5 backbone	✓	quota	–	fitted lognormal

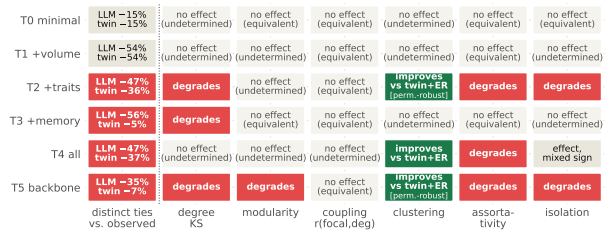


Figure 2: **Three-stage attribution matrix at matched edge counts** (in-vivo grid; GPT-4o-mini; all arms projected to the cell's m^*, Table 7). First column: mean shortfall of distinct ties relative to the observed edge count for the LLM and its twin (red: the LLM falls short by at least 5 points more than its twin). Stage 1: does the policy change the component *error* relative to its twin (two-sided stratified permutation, Holm)—the fit error, not the raw statistic (Table 6)? Non-detections are "equivalent" if the 90% CI lies within half the between-classroom SD of the observed statistic, else "undetermined" (degree KS is never called equivalent). Stage 2: improving (dark green also beats ER) or degrading. Stage 3: [perm.-robust] = the label-permuted policy itself beats the twin. Observed-density projection: Fig. 6.

(chance), a degree-preserving diagnostic, a trait-informed dyadic logistic reference, and its own twin. Test with site-stratified permutation (10,000 permutations), Holm-corrected within family, reporting per-site direction, bootstrap CIs, equivalence margins for non-detections, and edge shortfall. Apply the attribution controls: label permutation with statistics in real-entity and displayed-label spaces, and the surface-form probe (§6).

Data. *In-vivo*: four intact classrooms (n=40 each; 160 students) from a study conducted under institutional ethics approval [anonymized]; directed friendship nominations, symmetrized by union (a tie exists if either student nominated the other) to match the undirected weighted graphs the simulator generates, reciprocity reported descriptively (Appendix A); Big Five and classroom-climate z-scores; focal trait extraversion. Raw data are not distributable; aggregates and code are released. *Public*: Knecht classroom klas12b (Knecht, 2008; Snijders et al., 2010) (n=23; focal: delinquency) and Glasgow s50, the 50-girl excerpt of the Teenage Friends and Lifestyle Study (West & Sweeting, 1995; Michell & Amos, 1997; Pearson & Michell, 2000) (n=50; focal: sport)—modularity 0.67—both fetched at run time from the SIENA archive, never redistributed. *Synthetic*: a planted-mechanism generator for mechanism-recovery checks.

Architectures (Table 1). T0 is a minimal architecture distilled from recurring design choices in published simulators (Appendix H): persona-conditioned agent, bare-ID candidate list, mandatory top-k selection, weight-accumulating dynamics. **T1** adds volume freedom via a calibratable score threshold θ (prompt byte-identical to T0, Appendix J; instruction-based volume control was ignored in a pilot). **T2** adds alter traits to candidate entries; **T3** adds interaction memory (past-interaction counts, reciprocation flags); **T4** combines all three; **T5** is a backbone hybrid: per-agent activity quotas from a *fitted* lognormal, with trait-aware single selections.

Fairness by construction. Every architecture is calibrated by the same procedure on its own random twin (no API), so none is tuned on LLM behavior while another is not; twin calibration also replaces replay recalibration, which memory architectures break (Appendix E). Workhorse policy GPT-4o-mini (anchor study, Appendix A); in-vivo grid $6 \times \{\text{LLM}, \text{twin}\} \times 4 \times 5$ seeds $= 240$ runs plus a 60-run permutation arm, 0% fallback.

5 RESULTS I: THE IDENTIFIABILITY MATRIX

Reading Fig. 2: rows are architectures, columns components; blank = the LLM's choices made no detectable difference to that component's error; green = improved beyond the twin (dark: beyond chance too); red = made *worse*; bracket = survives label permutation.

First, edge volume: the policy does not reproduce how many distinct ties form (Fig. 2, first column). At the observed-density projection, every architecture that lets the policy act on peer information or history yields far fewer distinct ties than the observed network—and than its own twin: T3 misses 56% of m_{obs} against 5% for its twin, T5 35% vs. 7%, T2 47% vs. 36% (Table 5). Where LLM and twin fall short alike (T0 15%/15%, T1 54%/54%) the shortfall is the machinery's. Concentration (P4) is the one policy signature visible before any component is examined, and the reason every comparison below is made at a common edge count.

The minimal template is structurally silent, and its volume-only variant shows no detectable fidelity effect (T0, T1; P1–P3)—the predicted lower bound, observed. T0 is not a claim about any deployed system but a distillation of design choices that recur across the surveyed simulators (Appendix H); P1–P3 predict its silence, and the informative result is how little each added axis lifts it. At matched edge counts no component's error separates from the twin's for T0 or T1 (all two-sided Holm $p \geq .19$); T1 shifts raw coupling (Table 6) without changing its error—silent in error space, not structurally inert. For T0 the mean error is twin-equivalent on modularity (LLM$-$twin -0.004, 90% CI $[-0.010, 0.003]$; margin 0.028), coupling, assortativity, and isolation; degree and clustering are *undetermined* (clustering -0.013 $[-0.024, -0.002]$: a nominal advantage that reaches significance only at the sparsest projection of a density sweep and vanishes under the scale-matched twin, -0.003; Appendix D). At the observed-density projection T0's modularity beats ER (0.143 vs. 0.184)—but so does the twin's (0.149), and at m^* all three coincide (0.136/0.140/0.134). The apparent advantage is not attributable to the policy. The *shuffle twin* (Appendix D) gives the same verdict on all six components (paired Holm $p \geq .07$).

Peer traits make the policy visible—improving one component and degrading most others (T2, T5; P1, P4). At matched edge counts the policy is detectable on four (T2) or five (T5) of six components. It *improves* clustering: error 0.268 (T2) and 0.161 (T5) against twins at 0.575/0.522 and ER at 0.578/0.545 (Holm $p < .001$; 4/4 classrooms). The improvement is label-permutation-robust—the permuted policy itself beats the twin (T2: -0.267, 90% CI $[-0.289, -0.244]$; T5: -0.285 $[-0.313, -0.257]$)—and, unlike the minimal template, popularity under T2/T5 follows persons rather than labels (§6), so we call it person-attributable; it also replicates against the decision-level shuffle twin ($p < .001$). *The mechanism is not homophily.* Selected pairs are *not* more trait-similar than pool expectation ($1.05\times$ vs. $1.00\times$); instead, different egos converge on the same trait-attractive alters ($+0.91$ SD extraverted, $+0.84$ agreeable vs. offered)—attraction to status, not the affinity of like for like—building a triangle-rich core (Appendix A). The same convergence *degrades* degree (KS $+0.10$ (T2), $+0.12$ (T5) above their twins), assortativity ($+0.43$, $+0.38$), isolation ($+0.04$, $+0.02$) and, for T5, modularity ($+0.09$), all Holm $p \leq .02$—degradations a "beats the twin" criterion would have filed as absence of information. T5 isolates the trade-off (P2, P4): its backbone gives the twin the grid's best degree fit (KS 0.169 vs. ER 0.193, observed projection), which the LLM degrades to 0.506 by folding volume onto few partners. Coupling error shows no detectable effect although the raw statistic shifts strongly positive ($+0.42$, $+0.38$; Table 6).

Memory looked like the one community-forming policy—until edge counts were matched (T3; P3). At the observed-density projection, memory produced the program's best modularity (error 0.074 vs. twin 0.176; LLM$-$twin -0.102, 90% CI $[-0.115, -0.088]$; the permuted policy also beat the twin). But T3's LLM runs contain 56% fewer distinct edges than the observed network (max 69%) while its twin's fall short by 5%: lock-in re-selects the same partners. At matched edge counts the difference is $+0.008$ $[-0.004, 0.020]$ (Holm p=1), inside the equivalence margin for the pooled in-vivo estimate; on the public datasets no model separates from its twin either, though with 2–5 seeds per cell those are non-detections, not equivalence claims (Table 3). The community improvement is therefore *not supported after density matching and cannot be read as a policy contribution independent of edge volume*. What memory detectably does is concentrate (degree KS $+0.07$ above the twin, Holm $p < .001$) and lock popularity onto whichever partners were met first: familiarity-driven lock-in rather than reciprocal reinforcement (§6). We withdraw the pre-specified "genuine communities" reading (Appendix B); the audit caught its own false positive.

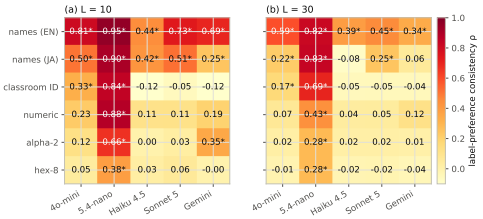


Figure 3: **Surface-form artifacts** (probe). Label-preference consistency ρ by format $\times$ model at (a) L=10 and (b) L=30 ($*$: Monte-Carlo $p < .01$ vs. protocol-matched random null). The gender-associated name effect is in Fig. 7 (Appendix I).

Exposing everything does not help (T4; P4). The "all axes" architecture reproduces T2's profile at matched density—clustering improves (0.361 vs. twin 0.569; also beats ER), assortativity degrades—and did not enlarge the set of improved components; T4 has no permutation arm, so its robustness is untested. Two pre-specified predictions failed informatively (T1, T4); both are absorbed by P2/P4 (Appendix B).

Not one model, not one dataset. The critical cells were re-run on the two public datasets with three providers (GPT-4o-mini, Claude Haiku 4.5, Gemini 3.6 Flash; per-dataset twin calibration; 240 runs), at the observed-density projection and at matched edge counts (Appendix A). (i) *T0 stays silent*: 0 of 36 tests certify; modularity never improves on the twin at matched density (8/8 cells; in one cell, Gemini on dense Knecht, it is *worse* than its twin, Holm p=.048, two-sided scale-matched check); one candidate (clustering on dense Knecht, 3/4 models, raw $p \leq .008$) fails the family and is recorded as undetermined. (ii) *T2's clustering improvement replicates and survives density matching*: direction 6/6, pooled certification 3/3 providers (Holm p=.0024), below twin in every closed-model cell at m^* (raw $p \leq .005$); T5 replicates in 6/6 (certified for GPT-4o-mini). (iii) *T3 is a density artifact on public data too*: at the observed projection it certifies on sparse s50 (2/3 providers) and reverses on dense Knecht (3/3), but public T3 runs fall 37% short of m_{obs} (twins 0%), and at matched edge counts no model separates from its twin on modularity on either dataset (8/8, raw $p \geq .17$). *Open-weight replication.* To rule out proprietary-API idiosyncrasy, the headline cells were re-run with an open-weight model (Qwen3-VL-32B-Instruct, text-only) served on-premises. Across both public datasets (Table 3; two seeds per cell, 0% parse fallback) Qwen3-VL-32B reproduces the closed-model pattern cell by cell. *T0 shows no detectable policy effect* (s50: modularity error 0.170 vs. twin 0.169, ER 0.177; Knecht: 0.092 vs. twin 0.056). *T2's clustering improvement replicates* on both datasets (s50: 0.306 vs. twin 0.410, ER 0.436; Knecht: 0.147 vs. 0.289, 0.281; $p_{\text{ER}} < .001$ in both) and persists at matched edge counts. *T3 behaves as for the closed models*: an apparent modularity gain on sparse s50 at the observed projection (0.121 vs. twin 0.169) that disappears at matched density (0.123 vs. 0.134, p=.32), and no gain on dense Knecht. With two seeds the one-sided twin test cannot reach α=.05 (exact floor $1/6$; $1/3$ two-sided): these are directional replications, not certified effects, with a proper test only against ER. The finding that apparent structure need not be agent-attributable is not an artifact of closed models.

6 Results II: Presentation Artifacts

Setup. Label preference is a property of a *single scoring decision*, not of network dynamics, so a decision-level probe strips the dynamics away: the simulator's prompt (ID-only variant, Appendix J), a fresh randomized candidate list per decision, exposures controlled by design. 80 cells (25,600 decisions; plus three forced-full-scoring controls, Appendix I): 5 models (GPT-4o-mini, GPT-5.4-nano, Claude Haiku 4.5, Claude Sonnet 5, Gemini 3.6 Flash) $\times$ 6 label formats (classroom IDs, numerals, letter pairs, 8-char hex, English and romanized Japanese names; gender-balanced, frequency-ranked frozen rosters) $\times$ $L \in \{10, 30\}$, plus L=60 for four formats; 320 decisions per cell in 4 blocks. Statistic: cross-block Spearman consistency of exposure-normalized label popularity vs. a protocol-matched Monte-Carlo null ($\rho \approx 0.00$); zero parse failures.

Models prefer labels—worst for human names (Fig. 3). Label preference follows a semantic gradient: human names $\gg$ structured IDs $\gg$ random alphanumerics. English names reach $\rho = 0.95$ (GPT-5.4-nano, L=10) and 0.81 (GPT-4o-mini); *every* model shows significant name preference—

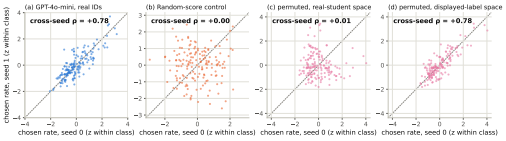


Figure 4: **Labels, not students.** Cross-seed popularity (seeds 0 vs. 1, standardized within site; one point per student in a–c, per displayed label in d; ρ = mean Spearman over ten seed pairs $\times$ four classrooms): default policy (ρ=.78); random twin (.00); label-permuted policy in *real-student* space (.01) vs. *displayed-label* space (.78).

including Claude Haiku 4.5 ($\rho = 0.44$), which is completely immune on all structured-ID formats ($\rho \approx 0$, replicating its in-vivo behavior). Artifact load is format-dependent: high-entropy hex labels eliminate the preference for four of five models (a practical mitigation) but not for GPT-5.4-nano ($\rho = 0.38$).

Name preference is a fairness-relevant failure mode. In all five models and both rosters, female-associated names are preferred or at parity: point estimates of the female/male chosen-rate ratio are at or above parity in all 20 model$\times$roster$\times L$ cells, with 17/20 95% bootstrap CIs over names excluding 1, and range from 1.0 after rounding (Claude Haiku 4.5, Japanese names, L=30; CI 0.92–1.10) to **8.7** (GPT-5.4-nano, English names, L=30; CI 5.9–13.1; Appendix I, Fig. 7), and commoner names are preferred (frequency-rank correlations -0.34 to -0.71). The gender-associated coefficient remains positive after adjusting for name frequency and length in all 20 model$\times$roster$\times L$ cells (bootstrap CI excluding zero in 16/20), so it is not a frequency effect in disguise; because rosters are synthetic and names are the only manipulated attribute, this is a *gender-associated name effect*, not a measured effect of gender. Per-name popularity is correlated *across providers* (raw ρ 0.37–0.82; 0.22–0.78 after regressing out gender and frequency): the ecosystem shares a name prior that ensembling will not remove.

Labels, not students (Fig. 4). Run the simulator with per-run bijective label permutation and compute cross-seed popularity consistency in two spaces. For the minimal template, real-student space collapses ($\rho = 0.01$) while displayed-label space persists at the unpermuted level ($\rho = 0.78$): the preferred *labels* persist, the preferred *students* disappear. For T2 and T5 the pattern inverts: popularity follows *persons* (real-student $\rho = 0.90$, 0.83; displayed-label $\rho = 0.04$) and tracks the trait the policy converges on (popularity–extraversion $\rho \approx 0.7$), so their clustering gain is attributable to individuals, not to label identity or similarity. Under T3 popularity is consistent in neither space ($\rho \leq 0.06$): lock-in on whichever partners were met first (Appendix I).

Positional bias. With candidate order randomized per decision, models still select 87–99.9% of candidates from the first 10 of 30 positions (random control: 33.6%); the two artifacts dissociate (Haiku: extreme position bias, no ID preference; GPT-5.4: both). The bias lives in *which* candidates a model writes about under a selection budget, not in a score gradient: forced to score every candidate under the base "sorted by descending score" instruction, scores fall with position in every model (Spearman -0.93 to -0.15), but with that clause removed (entries in input order, or in any order) the decline vanishes in four models (three return one constant score for all 30 candidates) and shrinks to $\rho = -0.17/-0.22$ for GPT-5.4-nano (Appendix I)—an instruction-induced pattern the first control would have misread as intrinsic. Randomized here, it only reduces effective choice diversity (P4); unrandomized, it is a direct structural confound.

7 Results III: ClassNetBench

The audit is packaged as **ClassNetBench**[1]: public-data loaders, the planted-mechanism generator, references, twins, the probe, and a one-command runner; no private data required.

[1] `https://anonymous.4open.science/r/ClassNetBench-24CD` (anonymized); public data are fetched from their original host, not redistributed (`DATA_TERMS.md`).

7.1 DOES THE AUDIT RECOVER KNOWN MECHANISMS?

An audit paper must audit itself. We instantiate six *ground-truth policies* with oracle access (random; homophily; popularity; triadic closure; reciprocity; memory reinforcement), run each through the identical shell against a held-out run of the same policy, and cross-evaluate all 6×6 policy–target pairs (Appendix G). *Specificity is exact*: the random policy certifies nothing, and nothing certifies on the random target. *Sensitivity*: 4/5 pre-specified signatures are recovered; the miss is diagnostic—oracle memory reinforcement expresses as *dyadic persistence* (reciprocity certified), matching the in-vivo finding that memory yields concentration, not community structure (§5). Off-diagonal certifications form one equivalence class, {homophily, popularity, triadic} on clustering—the classical homophily/transitivity confound (Shalizi & Thomas, 2011): the audit recovers mechanism *equivalence classes*, not point identification. Scaling to $n \leq 500$ keeps specificity exact and recovers 14/15 non-random signatures (19/20 expected outcomes including the random null; Appendix G).

8 DISCUSSION

Design guidance. Expose the information whose signature you want; constrain concentration if degree structure matters; compare arms at matched edge counts; and always run the twin and the permutation arm—"our agents formed communities" and "our pipeline manufactures communities from any choices" are one control apart.

Implications. We have measured not sociality itself, but whether observed social structure is attributable to the agent policy rather than to the simulator architecture or presentation artifacts. P1 formalizes one consequence of architecture: no policy, however capable, can condition its choices on information that its decision context does not disclose; the minimal template's silence is that consequence observed. Believability-validated simulators with named agents inherit every artifact documented here—popularity by name association and commonness, choices by list position, structure by accumulation dynamics—a concrete, fairness-relevant failure mode for applications involving minors (on synthetic rosters, not observed harm) and a reason surface-form confounding must be controlled before any entity-level claim.

Limitations. The principles are necessary conditions: detectability makes a component testable, not correct. Sites are few (four classrooms, two public datasets); inference is simulation-level, conditional on the sites. Claims are scoped to *persona-conditioned, network-generating LLM simulators*. The controls are agnostic to *how* the policy is conditioned (prompt personas, activation steering or persona adapters each define a π with an identically built twin); by P1, ego-side steering cannot make alter-attribute components identifiable unless alter attributes enter the context. Free-form dialogue agents and dynamic link-formation architectures (feed-mediated follow ties as in OASIS or AgentSociety, Yang et al., 2024; Piao et al., 2025) change C and D; they lie outside this audit and need architecture-specific twins. Matched edge counts compare generated arms against the *unchanged* observed network (which arm is closer to the fixed target at equal generated volume, not whether a sparsified observed network would look alike); a very small m^* (Table 7) constrains higher-order structure for every arm, hence the density-sweep, seed-wise-pairing and observed-density checks (Appendix D). Two analysis choices were made after seeing data and are disclosed as such: the primary comparison moved from the observed-density projection to matched edge counts once the edge-volume deficit appeared (both are reported: Fig. 2 vs. Fig. 6), and the equivalence margins were fixed after the grid had run (Appendix D). Each matrix cell uses one workhorse policy (replication covers four); evaluation is cross-sectional. Twins are calibrated to the same structural targets as the evaluation (Appendix F), which is conservative for improvement claims but not cross-fitted.

Conclusion. We have not said what it is for an LLM agent to behave socially. We have said what an observed structure must survive before it can count as evidence of it: it must differ from what the same machinery produces from chance; it must move the network toward the real one; and it must follow the simulated persons rather than the words that name them. Under the minimal template, nothing survives. Passing the audit earns a simulator the right to be evaluated, not the presumption of validity; we provide the definitions, controls, principles, and benchmark to make that test routine for persona-conditioned, network-generating LLM simulators.

AI USE STATEMENT

LLMs are the object of study; this statement concerns generative AI used in producing the paper. Following the ICLR 2027 AI Policy for Authors, among tasks with *required* disclosure we used generative AI tools to implement methods (code implementation and debugging, under the authors' direction), to assist in the writing of proofs (the write-up of Propositions 1–3, which the authors verified line by line), and to provide feedback on research methodology and experiments (critical feedback on framing and on robustness checks, which the authors then specified and ran). We did not use generative AI tools to generate synthetic data sets, to develop the theoretical model or conceptual framework, to formulate mathematical claims or provide critical ingredients for their proofs, to propose or refine hypotheses, to clean or reformat data, to support qualitative analysis, or to interpret results; assisting with translation is not applicable. Among tasks with *recommended* disclosure we used generative AI tools to draft and edit parts of the paper for readability, to create and edit software code, and to create scientific figures from scripted data. We have reviewed all AI-assisted work: every reported number is regenerated by the released scripts, and no AI system generated or altered data or results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work audits systems proposed for use on data about minors and exposes a fairness-relevant failure mode (name- and position-driven popularity) before deployment; in-vivo data are not distributable and contain no names, rosters are synthetic. Audit tools could be used to overfit simulators to audits; held-out sites and planted mechanisms mitigate this.

REPRODUCIBILITY STATEMENT

Code, frozen calibrations, aggregated results, and verbatim prompts: `https://anonymous.4open.science/r/ClassNetBench-24CD`; `reproduce.sh` rebuilds the CPU-only pipeline without private data or API keys.

REFERENCES

Ethan Bito, Yongli Ren, and Estrid He. Evaluating position bias in large language model recommendations. *arXiv preprint arXiv:2508.02020*, 2025.

Paul Erdős and Alfréd Rényi. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6: 290–297, 1959.

Daniel C Feiler and Adam M Kleinbaum. Popularity, similarity, and the network extraversion bias. *Psychological Science*, 26(5):593–603, 2015.

Chen Gao et al. S^3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*, 2023.

Chen Gao et al. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1259), 2024.

Volker Grimm et al. Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science*, 310:987–991, 2005.

Tiancheng Hu, Joachim Baumann, Lorenzo Lupo, Nigel Collier, Dirk Hovy, and Paul Röttger. SimBench: Benchmarking the ability of large language models to simulate human behaviors. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.17516.

Andrea Knecht. *Friendship selection and friends' influence: Dynamics of networks and actor attributes in early adolescence*. PhD thesis, Utrecht University, 2008.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Maik Larooij and Petter Törnberg. Validation is the central challenge for generative social simulation: A critical review of LLMs in agent-based modeling. *Artificial Intelligence Review*, 59(1):15, 2025. doi: 10.1007/s10462-025-11412-6.

Yiming Li and Dacheng Tao. AI agents alone are not (yet) sufficient for social simulation. *arXiv preprint arXiv:2603.00113*, 2026.

Junchi Liao. Auditing provenance sensitivity in LLM agent action selection. *arXiv preprint arXiv:2607.20827*, 2026.

Jingzhe Lin, Hengbin Yu, Yongdan Zeng, and Fangwei Zhong. EduMirror: Modeling educational social dynamics with value-driven multi-agent simulation. *arXiv preprint arXiv:2606.07948*, 2026.

Emma Rose Madden. Evaluating the use of large language models as synthetic social agents in social science research. *arXiv preprint arXiv:2509.26080*, 2025.

Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.

Lynn Michell and Amanda Amos. Girls, pecking order and smoking. *Social Science & Medicine*, 44 (12):1861–1869, 1997.

Xinyi Mou et al. From individual to society: A survey on social simulation driven by large language model-based agents. *ACM Computing Surveys*, 2026.

Atharva Pandey and Gautam Jajoo. Reason-mediated behavioral models for auditing LLM social simulators. *arXiv preprint arXiv:2607.24649*, 2026.

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

Michael Pearson and Lynn Michell. Smoke rings: Social network analysis of friendship groups, smoking and drug-taking. *Drugs: Education, Prevention and Policy*, 7(1):21–37, 2000.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. AgentSociety: Large-scale simulation of LLM-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*, 2025.

Annabella Sakunkoo and Jonathan Sakunkoo. Name of thrones: Evaluating how LLMs rank student names, race, and gender in status hierarchies. *arXiv preprint arXiv:2504.10797*, 2025.

Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. Simulated students in tutoring dialogues: Substance or illusion? In *Proceedings of ACL*, 2026.

Maarten Selfhout, William Burk, Susan Branje, Jaap Denissen, Marcel van Aken, and Wim Meeus. Emerging late adolescent friendship networks and Big Five personality traits: A social network approach. *Journal of Personality*, 78(2):509–538, 2010.

Cosma Rohilla Shalizi and Andrew C Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods & Research*, 40(2):211–239, 2011.

Tom AB Snijders, Gerhard G van de Bunt, and Christian EG Steglich. Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60, 2010.

Kyosuke Takami and Masahiko Haruno. From rule-based to LLM-based agents: A calibrated simulation framework for classroom social networks. In *Artificial Intelligence in Education (AIED 2026)*, Lecture Notes in Computer Science, pp. 380–388. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-29760-0_42.

Aleksandar Tomašević, Darja Cvetković, Sara Major, Slobodan Maletić, Miroslav Anđelković, Ana Vranić, Boris Stupovski, Dušan Vudragović, Aleksandar Bogojević, and Marija Mitrović Dankulov. Towards operational validation of LLM-agent social simulations: A replicated study of a Reddit-like technology forum. *arXiv preprint arXiv:2508.21740*, 2025.

Petter Törnberg, Diliara Valeeva, Justus Uitermark, and Christopher Bail. Simulating social media using large language models to evaluate alternative news feed algorithms. *arXiv preprint arXiv:2310.05984*, 2023.

Patrick West and Helen Sweeting. Background rationale and design of the West of Scotland 11-16 study. *MRC Medical Sociology Unit Working Paper*, (52), 1995.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, et al. OASIS: Open agent social interaction simulations with one million agents. *arXiv preprint arXiv:2411.11581*, 2024.

Jinyi Ye, Lei Cao, Ding Chen, and Emilio Ferrara. Stop drawing scientific claims from LLM social simulations without robustness audits. *arXiv preprint arXiv:2605.18890*, 2026.

Zhihao Yuan, Yunze Xiao, Ming Li, Weihao Xuan, Richard Tong, Mona Diab, and Tom Mitchell. Towards valid student simulation with large language models. *arXiv preprint arXiv:2601.05473*, 2026.

Zheyuan Zhang et al. Simulating classroom education with LLM-empowered agents. *Proceedings of NAACL*, pp. 10364–10379, 2025.

A THE IN-VIVO ANCHOR STUDY

Table 2: Mechanism of the clustering gain (T2 policy vs. controls): mean trait z of selected minus offered candidates, and ego–alter trait-distance ratio vs. uniform-selection expectation.

	trait-aware	default	random
extraversion Δ	$+0.91$	-0.03	0.00
agreeableness Δ	$+0.84$	-0.01	0.00
closeness Δ	$+0.77$	-0.01	-0.01
friction Δ	-0.39	-0.03	$+0.01$
trait-distance ratio	1.05	1.00	1.00

Data and protocol. Four intact classrooms (n=40 each; 160 students) with directed friendship nominations and psychometric traits (Big Five and classroom-climate z-scores); ethics approval and consent [anonymized]; raw data not distributable. Observed networks: density 0.19–0.28, modularity 0.36–0.49, arc reciprocity 0.54–0.68, r(extraversion, degree) $\in \{0.54, -0.10, 0.33, 0.22\}$. The simulator is the T0 template (T=20 rounds, empty initial graph, per-classroom calibrated edge parameters; candidate pools of 30). Audited policies: GPT-4o-mini, GPT-5.4-nano/-mini/-full, Claude Haiku 4.5, Claude Sonnet 5, Gemini 3.6 Flash, a trait-aware GPT-4o-mini variant, a rule-based scorer, and the random-score control; 5 seeds $\times$ 4 classrooms each; realized parse-fallback 0% in every reported run but one (0.13%); provider-incident-contaminated runs were discarded and re-executed (App. F).

Unconditional summary. *SFI-5* is a descriptive composite reported only here; calibration itself uses the corresponding SFI-6 objective, and neither enters inference. The calibration objective (`config/sfi_weights.json`; `calibrate_arch.py`, `calibrate_bench.py`) is SFI-6, the weighted sum $\mathrm{KS}_{\text{degree}} + |\Delta \text{ density}| + |\Delta \text{ modularity}| + |\Delta\, r(\text{extraversion}, \text{degree})| + 0.5\,|\Delta \text{ isolate rate}| + 0.5\,|\Delta \text{ arc reciprocity}|$ of simulated-minus-observed discrepancies, each on its native scale (KS and rates in $[0, 1]$, correlations in $[-1, 1]$; no further normalization); lower is better. SFI-5 drops the reciprocity term (weight 0), because the evaluated graphs are undirected union graphs, and is pooled over classrooms. All Stage-1–3 inference is component-wise (§3; Appendix D); no composite enters any test. Pooled SFI-5: degree-preserving diagnostic 0.205; trait-logit reference 0.462; ER 0.705; Gemini 3.6 Flash 0.706; *random-score control 0.710*; Claude Sonnet 5 0.735; GPT-5.4-nano 0.751; GPT-5.4 0.766; rule 0.774; Claude Haiku 4.5 0.776; GPT-4o-mini 0.803; GPT-5.4-mini 0.815; trait-aware 1.067. The best LLM variant and the random-score control are indistinguishable; no pairwise difference among the eight LLM variants and the rule baseline survives Holm correction.

Conditional component results (density-conditioned; pooled over classrooms): degree KS — ER 0.194, all default variants 0.24–0.28 (worse); coupling $|\Delta r|$ — trait-logit 0.092, generative variants 0.27–0.37 ($\approx$ ER 0.32); clustering — all default $\approx$ 0.47 ($\approx$ ER 0.45); modularity — ER 0.184, *all* simulator variants 0.13–0.16 including the rule baseline (0.150) and the random-score control (0.158): the advantage ships with the machinery.

Provenance analyses. (i) Projection: on unprojected binary union graphs the modularity advantage over density-matched references vanishes (-0.01 to $+0.015$); after top-m projection it is $+0.02$–0.05. (ii) Ego-side ablations (remove persona or climate blocks): no significant change. (iii) ID-only ablation (no trait information at all): identifier-preference consistency persists at 0.65; modularity at random-score level. (iv) Two-space permutation (Fig. 4, main text). (v) Replay recalibration (App. E): no default variant improves significantly; only trait-aware does ($1.18 \rightarrow 0.96$, Holm p=.0009), remaining last.

Trait-aware mechanism. Conditional $|\Delta r|$ improves to 0.233 and clustering to 0.191 (only generative variant to beat the trait-logit reference on clustering, Holm p=.0006), while degree KS ($0.24 \rightarrow 0.47$) and assortativity ($0.25 \rightarrow 0.66$) collapse. Selected pairs are *not* more trait-similar than pool expectation (distance ratio 1.05 vs. 1.00 for controls); selected candidates are extraverted ($+0.91$ SD), agreeable ($+0.84$), high-closeness ($+0.77$), low-friction (-0.39): convergent popularity selection.

Table 3: Open-weight replication (Qwen3-VL-32B-Instruct, text-only, served on-premises; two seeds per cell, 0% parse fallback). Left: the observed-density projection (m_{obs}) used in the first analysis. Right: the paper's primary analysis at the cell's common edge count m^* (m^*/m_{obs} shown), where the T3 modularity difference disappears. p_{twin} and p_{ER} are the one-sided run-level permutation tests used for certification on the public datasets (improvement direction; Appendix D); with two seeds per arm the exact one-sided floor is $1/6$ (a two-sided test would floor at $1/3$), so p_{twin} cannot reach α=.05 and only the ER column (100 replicates) is a proper test. Numbers regenerate from `openweight_eval.py` and `density_sensitivity_bench.py`.

			observed-density projection					matched edge count m^*			
Dataset	Arch.	Comp.	LLM	Twin	ER	p_{ER}	m^*/m_{obs}	LLM	Twin	ER	p_{twin}
Knecht (dense, n=23)	T0	modularity	0.092	0.056	0.061	0.992	92/92	0.092	0.056	0.058	0.83
Knecht (dense, n=23)	T2	clustering	0.147	0.289	0.281	0.000	79/92	0.158	0.370	0.332	0.17
Knecht (dense, n=23)	T3	modularity	0.134	0.044	0.061	1.000	35/92	0.134	0.175	0.157	0.17
Glasgow s50 (sparse, n=50)	T0	modularity	0.170	0.169	0.177	0.368	74/74	0.170	0.169	0.182	0.66
Glasgow s50 (sparse, n=50)	T2	clustering	0.306	0.410	0.436	0.000	74/74	0.306	0.410	0.442	0.17
Glasgow s50 (sparse, n=50)	T3	modularity	0.121	0.169	0.177	0.001	65/74	0.123	0.134	0.133	0.32

B Pre-Specification Log

Predictions were written into the architecture design document before any LLM run of the grid; the dated results log in the released design document (`docs/architecture_comparison_design.md`) preserves the ordering.

arch	predicted	observed
T0	nothing (anchor replication)	as predicted: no detected effect; twin-equivalent on modularity, coupling, assortativity, isolation; degree and clustering undetermined
T1	degree spread ↑	**miss**: threshold volume under-densifies LLM and twin alike (protocol cost; P2)
T2	coupling + clustering, reduced degree damage	clustering improving (robust); coupling directional only; degree still broken (P4)
T3	reciprocity/stability; possibly genuine communities	modularity improving at m_{obs} projection, survives label permutation, but **miss**: gain vanishes at matched density (56% edge shortfall); memory concentrates, degrades degree
T4	best pure-LLM case	**miss**: reproduces T2's profile (clustering improves, assortativity degrades); no new improved component (P4 interference)
T5	degree by construction + coupling via LLM	clustering improving; degree degraded by LLM concentration (twin KS 0.169 vs. LLM 0.506) — sharpened P2/P4

C Principles P1–P5: Full Statements and Propositions

We formalize P1–P3 as necessary-condition propositions. Throughout, a simulation run consists of decision contexts $c_{i,t}$ presented to agents i at rounds $t = 1..T$; a policy π maps contexts to selections $S_{i,t} \subseteq \mathcal{C}_{i,t}$ from the presented candidate set; edge dynamics D aggregate selections into a graph.

Proposition 1 (Information impossibility)**.** *Suppose the context $c_{i,t}$ contains no candidate attributes, and candidate presentation is* attribute-exchangeable*: the joint distribution of contexts is invariant under any permutation of the attribute vectors among candidates (e.g., candidates appear as labels assigned by a uniformly random bijection; fixed labels satisfy this only when the policy's choices do not depend on the label strings themselves—see the Remark below). Then for every policy π, selection is conditionally independent of candidate attributes given the context; hence for any attribute x and any exposure pattern, $\mathrm{Cov}(x_j, \mathbf{1}[j \in S_{i,t}]) = 0$ in expectation, so no policy can use candidate attributes in selection under such an architecture. Consequently, components whose only route to the policy runs through alter-attribute-dependent choice (attribute assortativity, homophilous clustering) cannot separate from the twin* through that route*. The proposition does not exclude effects of the*

ego's own attributes (which are in the context) on activity or repeat choices, so ego-attribute–degree coupling remains expressible and is tested empirically rather than ruled out.

Proof. Under attribute-exchangeability, permuting attributes among candidates leaves the context distribution—and therefore the selection distribution— unchanged, while permuting the attribute labels in the covariance. Averaging over the uniform permutation gives $\mathrm{Cov}(x_j, \mathbf{1}[j \in S]) = \mathrm{Cov}(x_{\sigma(j)}, \mathbf{1}[j \in S])$ for all σ, which forces the covariance to its permutation mean, 0. Hence selection carries no information about candidate attributes, and functionals that depend on candidate attributes only through alter-attribute-dependent choice coincide in expectation with those of any identity-randomizing policy, in particular τ_A; functionals routed through the ego's own attributes are not covered. □

Remark. With *fixed* (non-random) labels, exchangeability can fail through the policy's label preferences—this is precisely the artifact channel of §6, and why Definition 2 requires label permutation: it *restores* the exchangeability premise.

Proposition 2 (Degree–volume constraint)**.** *Suppose C mandates exactly k selections per agent per round and D deduplicates (repeated selection of the same alter increments weight, not degree). Let U_i be agent i's number of distinct selected alters after T rounds. Then (i) $U_i \le \min(kT, n-1)$ for every policy; (ii) if agent i's per-round selection probabilities over alters are time-homogeneous, $p^i = (p_{i1}, \ldots, p_{in})$, and rounds are conditionally independent given p^i, then $\mathbb{E}[U_i] = \sum_j \left\{1-(1-p_{ij})^T\right\}$ is Schur-concave in p^i: any concentration of the selection distribution (in the majorization order) weakly decreases $\mathbb{E}[U_i]$, strictly so away from degenerate coordinates. Consequently a policy can create out-degree heterogeneity only* downward *— by concentrating some agents' choices, which simultaneously removes unique edges — and out-degree spread beyond the twin's is purchasable only at the price of unique-edge mass (the concentration trade-off, P4).*

Proof. (i) is immediate. (ii): $f(p) = 1-(1-p)^T$ is concave on $[0,1]$; a sum of a common concave function applied coordinate-wise, $\sum_j f(p_{ij})$, is a symmetric concave function of p^i and hence Schur-concave. Strictness follows from strict concavity of f on $(0,1)$ for $T \ge 2$. □

Proposition 3 (Memory requirement)**.** *Suppose the context $c_{i,t}$ excludes interaction history and its distribution is measurable with respect to static features only (candidate presentation is history-independent), and that the policy carries no persistent state across rounds beyond the supplied context (as for stateless API calls, the case throughout this paper). Then for every policy the selection processes across rounds are conditionally independent given static features: $\Pr(j \in S_{i,t} \mid j \in S_{i,t'},\ \textit{static}) = \Pr(j \in S_{i,t} \mid \textit{static})$ for $t' < t$. Hence tie reinforcement beyond the level induced by static features is impossible, and any community-formation mechanism defined by conditioning on past ties cannot be expressed by any policy. The conclusion is deliberately limited to* history-dependent *mechanisms: communities driven by static features remain expressible, and "static features" includes the displayed labels themselves (a policy preferring same-prefix identifiers forms label communities without memory and independently of any person attribute). Attribute-exchangeability (Prop. 1) excludes person-attribute communities but not label communities. Label permutation (Def. 2) removes communities tied to particular label* identities *but not those built on label* similarity *(shared prefixes), which survive relabeling with changed membership; excluding them requires the two-space consistency check (§6). No claim that memoryless modularity gains "must be machinery" follows from this proposition; that is an empirical question answered cell by cell.*

Proof. By the no-persistent-state assumption the policy is a (possibly random) function of the current context alone; if the context's distribution at t is $\sigma(\text{static})$-measurable and independent of past selections given static features, so is the selection. Conditional independence gives the displayed equality; reinforcement functionals that compare the two sides are therefore at their static-feature baseline. □

P4 and P5 are stated as empirical regularities with mechanism arguments (P4 follows qualitatively from Prop. 2(ii): the same concentration parameter governs distinct-partner loss and repeated-pair closure; P5 is documented in §6).

D AUDIT PROTOCOL: FULL VERSION

Steps. (1) Project every generated arm in a cell to the common edge count m^* (observed network unchanged); evaluate components separately—no composite index. (2) References: conditional Erdős–Rényi (chance), degree-preserving rewiring (diagnostic only), and a trait-informed dyadic logistic reference (17 parameters: intercept plus $|z_i - z_j|$ and $z_i + z_j$ per trait; fitted in-sample as an attribute-signal upper reference). (3) Run the twin. (4) Test π vs. ER and π vs. twin per component: site-stratified permutation tests (10,000 permutations) on pooled means, Holm-corrected within family, with per-site direction consistency reported ($n/4$ sites); inference is explicitly simulation-level, conditional on the observed sites. (5) Attribution controls: per-run label permutation with statistics computed in *real-entity* and *displayed-label* spaces; the surface-form probe (§6); position-specific selection rates. (6) Optional mechanism recovery on planted synthetics. (7) Report matrix + flags; structural claims are licensed only for cells passing both definitions.

Calibration fairness. Every architecture is calibrated by the same procedure on its own random twin (random search, 100 trials $\times$ 3 reps per site; no API): no architecture is tuned on LLM behavior while another is not. Memory architectures make decisions depend on the evolving graph, which breaks replay-based recalibration; twin calibration is the uniform substitute (Appendix E). The workhorse policy is GPT-4o-mini; the in-vivo anchor study additionally spans eight LLM variants from three providers (Appendix A). In-vivo grid: $6 \times \{\text{LLM}, \text{twin}\} \times 4 \times 5$ seeds $= 240$ runs plus a 60-run permutation arm; realized fallback rate 0% in all grid runs.

The random twin, step by step. Algorithm 1 gives the twin as it is implemented (`src/simulate.py`, `RandomScoreLLM` in `src/llm_agent.py`). Three facts about it matter for reading the audit. (i) *Everything except the scores is shared with the audited run.* The twin is not a separate generator: it is the same simulator invoked with the API-backed scorer swapped for a random one. At a given seed, round, and ego, the per-decision random stream is the same as the audited run's, so the twin's 30-candidate pool is exactly the pool the LLM would have been shown; the shuffled agent order, the T5 quota schedule, the p_{new} gate, the weight increments, and the projection to m^* are the same code with the same calibrated parameters. Memory annotations (T3, T4) are computed for the twin exactly as for the LLM, but the random scorer never reads them—which is the point: the twin realizes the architecture with an agent that conditions on nothing. (ii) *Volume is matched, not calibrated.* Under a fixed top-k (T0, T3) or a quota slot (T5) the twin selects exactly as many candidates per decision as the LLM; under a score threshold (T1, T2, T4) the twin does not apply θ at all but draws its selection count for each decision uniformly from the multiset $\mathcal{K}_\pi$ of per-decision counts logged in that cell's LLM runs (all seeds pooled; zero-selection decisions included), so the twin reproduces the LLM's volume distribution exactly rather than a threshold-induced approximation of it. (iii) *Score scale.* For threshold architectures the twin's scores are drawn from a frozen empirical sample $\mathcal{S}_\pi$ of the audited model's own scores (20,480 scores per closed model, pooled from that model's decisions in the surface-form probe, §6; for the open-weight model, from its own architecture runs), because with Uniform$(0, 1)$ scores the top of a 30-candidate pool would always clear θ and the threshold would bind on one arm only. For fixed-volume architectures the reported twins drew i.i.d. Uniform$(0, 1)$ scores. Under the *rank* weight rule this is immaterial: the selected set is a uniformly random k-subset either way and the score never enters the dynamics. Under the *score* and *rank*$\times$*score* rules (calibrated in three T0, two T3, and three T5 classroom cells, and in both public T0 and T5 cells) the score enters as the per-selection weight increment, so the twin's increments (top order statistics of 30 uniforms) and the LLM's (its own discrete scale) differ in dispersion. The scale-matched check below re-runs every such cell with $\mathcal{S}_\pi$ scores and finds 17 of 18 verdicts unchanged, every headline verdict among them (the exception is T5's isolation degradation, which is no longer detected). The label-permuted arm is not a twin: it is the LLM itself, run with a fixed permutation of the identifiers shown in the candidate list (statistics then computed in real-entity and displayed-label space).

Scale-matched twin check. We re-ran every fixed-volume twin cell (T0, T3, T5: four classrooms $\times$ five seeds in-vivo, and all 22 public cells) with scores drawn i.i.d. from $\mathcal{S}_\pi$ instead of Uniform$(0, 1)$—the *scale-matched twin*—holding everything else fixed, and repeated the matched-edge-count tests (`scale_matched_twin.py`; `scale_matched_twin_tests.csv`, `scale_matched_twin_bench.csv`). *In-vivo* (Table 4): 17 of 18 architecture$\times$component verdicts are identical under the paper's Holm family. The one change is T5's isolation degradation ($+0.018$, Holm p=.020), which is no longer detected

Algorithm 1 Volume-matched random twin τ_A for architecture $A = (I, C, D)$, one run at seed s. Lines marked $\star$ are byte-identical to the audited run at the same seed; the unmarked lines (the score draw and the resulting selection) are the only difference.

Require: roster V; rounds T=20; pool size P=30; calibrated ϑ_A (top-k, or max volume $k_{\max}$ with threshold θ, or quota (μ, σ) with k=1 per slot; edge rule α, weight rule, rank weights $a_1 \geq a_2 \geq \ldots$, p_{new}); for threshold architectures the frozen score sample $\mathcal{S}_\pi$ and the cell's logged LLM selection counts $\mathcal{K}_\pi$

Ensure: weighted undirected graph G (evaluation projects it to the cell's m^* strongest edges)

$\star$ $G \leftarrow \emptyset$; directed history $h \leftarrow 0$
if A is T5 **then**
  $\star$ draw budgets $Q_i \sim \text{round LogNormal}(\mu, \sigma)$, clipped to $[1, n-1]$, and scatter each agent's Q_i slots over rounds with the seed-derived stream; $k_{it} \leftarrow$ slots of i in round t
end if
for $t = 1, \ldots, T$ **do**
  $\star$ shuffle the agent order
  for each ego i (skip if A is T5 and $k_{it} = 0$) **do**
    $\star$ $r \leftarrow \text{seed}(s, t, i)$; pool $\mathcal{C} \leftarrow P$ candidates sampled from $V \setminus \{i\}$ with stream r (all of $V \setminus \{i\}$ if $n-1 \leq P$)
    $\star$ if A has memory, annotate each $j \in \mathcal{C}$ with h_{ij}+h_{ji} and the reciprocation flag {the LLM reads this; the twin never does}
    if fixed volume (T0, T3, T5) **then**
      $\sigma_j \sim \text{Uniform}(0, 1)$ i.i.d. for $j \in \mathcal{C}$; $k \leftarrow$ top-k (T0, T3) or k_{it} (T5) {reported runs; the scale-matched variant draws $\text{Uniform}(\mathcal{S}_\pi)$ here}
    else
      $\sigma_j \sim \text{Uniform}(\mathcal{S}_\pi)$ i.i.d. for $j \in \mathcal{C}$; $k \sim \text{Uniform}(\mathcal{K}_\pi)$ {threshold architectures; θ is *not* applied to the twin}
    end if
    selected $(j_1, \ldots, j_k) \leftarrow$ the k highest-σ candidates in $\mathcal{C}$, in descending σ {a uniformly random k-subset of the pool}
    for $\ell = 1, \ldots, k$ **do**
      $\star$ $\Delta \leftarrow \alpha a_\ell$ **(rank)** $\mid$ $\alpha \sigma_{j_\ell}$ **(score)** $\mid$ $\alpha a_\ell \sigma_{j_\ell}$ **(rank×score)**, per the calibrated weight rule
      $\star$ **if** $\{i, j_\ell\} \notin G$ **and** a draw from stream (r, ℓ) exceeds p_{new}: **skip** j_ℓ {new-edge gate}
      $\star$ $w_{\{i,j_\ell\}} \leftarrow w_{\{i,j_\ell\}} + \Delta$ (create the edge if absent); $h_{ij_\ell} \leftarrow h_{ij_\ell} + 1$
    end for
  end for
end for
return G

(+0.010, p=.37). Every headline cell is unchanged: T0 is silent on all six components under both twins (largest $|\Delta|$ 0.035; Holm $p \geq .38$), T3's modularity is silent (+0.008 vs. +0.009; p=1.0), and T5's clustering improvement and the T3/T5 degree and T5 modularity/assortativity degradations are detected with the same sign under both. The two twins are themselves indistinguishable except for T5 (modularity, clustering, assortativity; Holm $p \leq .044$)—the architecture with top-1 selection and score-based rules in three of four classrooms—where the scale-matched twin is the stricter null: the LLM's assortativity degradation shrinks from +0.380 to +0.246 and remains detected (p<.001). T0's in-vivo clustering, coded undetermined, moves toward zero (−0.013 → −0.003; p=.69). *Public data* (22 cells × 2 projections × 6 components = 264 two-sided tests; Holm within cell): the two twins are distinguishable in 8 of 264 tests (chance expectation 13), and verdicts change in 19—at the same rate in rank-rule cells, where the two twins are identical in law (7/96), as in score-rule cells (12/168), i.e. seed noise at n=5 per arm (raw-p floor .008; Holm ≈ .048). The critical cells are unchanged: no T3 cell shows a modularity improvement at m^* under either twin (0/8), and T5's clustering improvement has the same direction in 6/6 cells and the same four detections under both. The T0 clustering candidate on Knecht, recorded undetermined in §5, is detected at the floor in one cell under the reported twin (Claude; Holm p=.047) and in three under the scale-matched twin (GPT-4o-mini and Gemini on Knecht, Gemini on s50; p=.041–.047), while the same component moves toward zero in-vivo; we keep the undetermined coding. The released runner exposes the

Table 4: **Scale-matched twin check** (in-vivo grid; matched edge counts). Pooled LLM−twin component error for the fixed-volume architectures under the reported twin (Uniform$(0, 1)$ scores; U) and the scale-matched twin (scores from the model's empirical sample; S), and the difference between the two twins' own errors, $\Delta_{\text{twin}} = \text{err}(\tau_{\text{U}}) - \text{err}(\tau_{\text{S}})$, which equals S−U in the notation of the first two columns. Bold: two-sided site-stratified permutation test, Holm within component family over all six architectures, $p < .05$. †: the two twins differ (Holm $p < .05$). Weight rules per classroom A–D: T0 rank/score/rank/rank×score; T3 rank/rank/score/score; T5 rank/score/rank×score/score.

	T0			T3			T5		
Component	U	S	Δ_{twin}	U	S	Δ_{twin}	U	S	Δ_{twin}
degree KS	-0.005	-0.007	-0.003	$\mathbf{+0.071}$	$\mathbf{+0.086}$	$+0.015$	$\mathbf{+0.115}$	$\mathbf{+0.111}$	-0.004
modularity	-0.004	-0.001	$+0.003$	$+0.008$	$+0.009$	$+0.001$	$\mathbf{+0.090}$	$\mathbf{+0.073}$	$-0.018^{\dagger}$
coupling	$+0.051$	$+0.035$	-0.013	-0.067	-0.040	$+0.027$	-0.050	-0.063	-0.013
clustering	-0.013	-0.003	$+0.009$	$+0.015$	$+0.029$	$+0.014$	$\mathbf{-0.362}$	$\mathbf{-0.322}$	$+0.040^{\dagger}$
assortativity	$+0.004$	-0.022	-0.019	-0.003	$+0.038$	$+0.041$	$\mathbf{+0.380}$	$\mathbf{+0.246}$	$-0.133^{\dagger}$
isolation	$+0.000$	-0.000	-0.001	-0.005	-0.013	-0.007	$\mathbf{+0.018}$	$+0.010$	-0.007

scale-matched twin (`twinemp`) for every architecture, and we recommend it whenever the calibrated weight rule is score-based.

Density conditioning and its failure mode. Each simulated weighted graph is projected onto its m_{obs} strongest edges, ties broken deterministically by (weight desc, source, target). The projection conditions on observed density *only if* a run has at least m_{obs} distinct edges; runs with fewer keep all their edges and the shortfall is recorded. This matters: LLM policies concentrate their choices on few partners, so their runs fall short far more often than their twins (Table 5; T1–T5 LLM runs miss 35–56% of m_{obs} in every run, while T3/T5 twins miss 5–7%). An LLM-vs-twin contrast at the m_{obs} projection is therefore partly a contrast at unequal density. We therefore evaluate every cell at a common edge count m^*, defined per (architecture, site) as $m^* = \min\{m_{\text{obs}},\ \min_{\text{LLM and twin runs}} \#\text{distinct edges}\}$; the label-permuted arm is projected to the same m^* but does not enter the minimum, conditional ER is re-sampled at m^*, and the observed network is unchanged (errors remain distances to the observed statistics). The realized m^* per architecture × site is listed in Table 7. This density-matched evaluation is the paper's primary analysis (Fig. 2); the observed-density projection is retained as Fig. 6 to show what an audit that does not match density would have concluded. The matched-density evaluation (Table 5, right; Fig. 2) preserves the T2/T5 clustering improvements, the T2/T5 degree and assortativity degradations, and the absence of detected effects for T0/T1, but *removes* the T3 memory→modularity improvement (-0.102 $[-0.115, -0.088]$ at m_{obs} vs. $+0.008$ $[-0.004, 0.020]$ at m^*): at equal density the memory policy's graphs are not more community-like than its twin's. The projection's own structural contribution is measured by the union-graph comparison (App. A).

Density sensitivity of the headline verdicts. Because m^* is set by the sparsest run in a cell (Table 7; e.g. 19 of 175 edges for T1/B), we re-evaluated LLM vs. twin under two alternatives (Fig. 5; `density_sweep.csv`): a *fraction sweep*, projecting every run to $f\, m_{\text{obs}}$ for $f \in \{0.25, 0.5, 0.75, 1\}$ (a run with fewer distinct edges keeps all of them; the share of such runs is recorded), and *seed-wise pairing*, projecting each LLM run and its same-seed twin to their common minimum, so that no run is cut by another seed's sparsity. The verdicts are stable. T2 and T5 clustering improve at every level and under pairing (differences -0.15 to -0.37; Holm $p < .001$; 4/4 sites). T3's modularity "gain" appears only at $f \geq 0.75$—exactly where 100% of LLM runs fall short of $f\, m_{\text{obs}}$ while twins do not (-0.055, -0.102); wherever densities are actually matched ($f \leq 0.5$; pairing $+0.011$ $[-0.004, 0.026]$; m^*) there is no improvement (Holm $p \geq .09$; at $f = 0.5$ the sign reverses). T0 modularity is silent at every level (Holm $p \geq .06$, and $\geq .74$ from $f = 0.5$ upward; pairing -0.006 $[-0.014, 0.002]$). One caveat: T0's clustering, coded undetermined at m^*, reaches Holm $p = .013$ at the sparsest level ($f = 0.25$, ≈ 44 edges on 40 nodes; -0.033, 4/4)—a small advantage visible only where the projected graph is highly fragmented; we keep the undetermined coding and report it here.

References. Conditional ER: $G(n, m_{\text{obs}})$. Degree-preserving: double edge swaps ($10m$ swaps). Trait-logit: dyadic logistic with intercept + $\{|z_i - z_j|,\ z_i + z_j\}$ per trait (ridge λ=10^{-3}, IRLS),

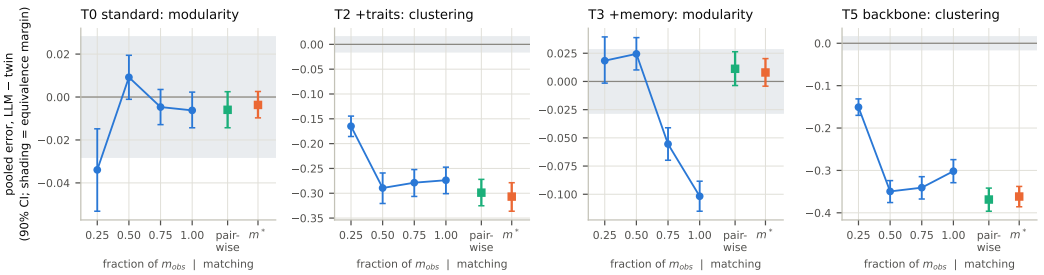


Figure 5: **Density sensitivity of the headline cells** (in-vivo grid). Pooled LLM−twin error with 90% site-stratified bootstrap CI when every run is projected to a fraction f of m_{obs} ($f = 1$: observed-density projection), under seed-wise pairwise matching, and at the cell minimum m^* (primary). Shading: the equivalence margin Δ_φ. T2/T5 clustering improvements hold everywhere; T3's modularity gain exists only at $f \geq 0.75$, where all LLM runs are short of $f\, m_{\text{obs}}$; T0 modularity never separates.

Table 5: Edge shortfall and density-matched re-test (in-vivo grid). Shortfall = fraction of m_{obs} missing when a run has fewer distinct edges than the observed network (mean (max) over 20 runs). Right: stage-1/2 coding at the common count m^* (Table 7; Fig. 2 adds the equivalence and permutation stages). Codes: 0 no detected effect; +/++ improving vs. twin (/and ER); − degrading; ± detected, inconsistent sign.

	shortfall, mean (max)			density-matched coding at m^*					
Arch.	LLM	twin	perm. LLM	degree	modularity	coupling	clustering	assort.	isolation
T0	0.15 (0.36)	0.15 (0.36)	–	0	0	0	0	0	0
T1	0.54 (0.85)	0.54 (0.89)	–	0	0	0	0	0	0
T2	0.47 (0.79)	0.36 (0.70)	0.46 (0.72)	-	0	0	++	-	-
T3	0.56 (0.69)	0.05 (0.19)	0.55 (0.68)	-	0	0	0	0	0
T4	0.47 (0.80)	0.37 (0.80)	–	0	0	0	++	-	±
T5	0.35 (0.49)	0.07 (0.26)	0.30 (0.43)	-	-	0	++	-	-

sampled at exactly m_{obs} dyads via Efraimidis–Spirakis weighted sampling without replacement. 100 replicates per site unless noted.

Twins. Protocol, dynamics, candidate pools, and calibrated parameters identical to the audited run (Algorithm 1). Threshold architectures (T1, T2, T4): scores i.i.d. from the audited model's frozen empirical score sample, per-decision selection counts drawn from the audited runs' logged counts. Fixed-volume architectures (T0, T3, T5): Uniform$(0, 1)$ scores, volume identical by construction; the scale-matched variant above draws from the empirical sample instead.

Terminology. Definition 1 is deliberately operational and null-relative: it tests *distinguishability from the architecture-matched random policy*, not identifiability in the classical sense ($P_{\theta_1}{=}P_{\theta_2} \Rightarrow \theta_1{=}\theta_2$ over a parameter space); two classically non-identifiable policies may both separate from τ_A. Failure to separate shows only the absence of a detectable policy signal relative to this null; separation is neither sufficient for nor equivalent to classical identifiability. Propositions 1–3 (Appendix C) carry the architecture-level impossibility claims; §7.1 validates the test on planted ground truth. Equivalence margins are substantive (half the between-classroom SD of the observed statistic) and disclosed in Appendix D, with Lakens (2017) as the reference standard.

Equivalence margins. Following Lakens (2017), non-detection becomes a bounded claim only through a declared smallest effect of interest. Ours is substantive: half the between-classroom standard deviation of the observed statistic, $\Delta_\varphi = \frac{1}{2}\,\text{SD}_{\text{sites}}(\varphi_{\text{obs}})$—a policy effect smaller than half the amount by which real classrooms differ from one another is negligible for the audit's purpose (modularity 0.028, clustering 0.015, assortativity 0.16, coupling 0.13, isolation 0.007). Degree KS has no observed statistic and is never called equivalent. We disclose that this margin was fixed after the observed-density analysis, when a machinery-relative margin ($\frac{1}{2}|\text{err}(\text{ER}) - \text{err}(\tau_A)|$) proved

degenerate at matched edge counts (ER and twin nearly coincide there); that margin and a $0.5\,\mathrm{SD}$ margin are reported for every cell (Appendix D), and every equivalence claim reads "equivalent mean error within Δ_φ," never "equivalent networks."

Table 6: Raw-statistic check at matched edge counts: pooled LLM−twin difference of the signed component *value* (not the error), two-sided site-stratified permutation test; * $p < .05$ uncorrected, ** Holm-corrected $p < .05$ within the component family. Degree KS is a distance and has no signed counterpart. Non-detection on the *error* does not imply the *structure* is unchanged: trait–position coupling shows no detectable effect on error for any architecture (Fig. 2), yet its raw value shifts under T1, T2, T4, T5 (Holm-corrected), and raw modularity shifts under T2 and T4 whose modularity *error* is undetermined—the policy moves the statistic away from the observed value in some sites and toward it in others, so the absolute error does not separate. Stage 1 therefore reads "effect on the evaluated error," as in Def. 1. From `density_raw_tests.csv`.

Component (raw value)	T0	T1	T2	T3	T4	T5
modularity	+0.004	-0.029	-0.115**	+0.010	-0.136**	-0.091**
coupling r(focal, deg)	-0.027	+0.149**	+0.424**	+0.047	+0.483**	+0.384**
clustering	+0.013	+0.009	+0.307**	-0.015	+0.208**	+0.362**
degree assortativity	-0.037	-0.041	-0.486**	-0.012	-0.509**	-0.401**
isolation	+0.000	+0.011	+0.043**	-0.013*	+0.065**	+0.022**

Decision-level shuffle twin. A stronger null built from the LLM's own logs: for every logged decision (round t, ego i) we keep its candidate pool (replayed exactly from the seeded sampler), its selection count, and its score vector in rank order, and re-assign the scored slots to a uniformly random subset of the pool; edge dynamics are then replayed with the cell's calibrated parameters (replay of the unshuffled logs reproduces the original evaluation to 10^{-16}). This preserves per-agent, per-round, per-decision score distributions and volumes exactly and destroys only the score-to-identity mapping. It is valid only for architectures whose decisions do not depend on the evolving graph (T0, T1, T2, T5); for memory architectures (T3, T4) the logged decisions are conditional on the graph the LLM itself produced, so a shuffled replay would not be a sample from any well-defined policy, and the i.i.d. twin remains the null. Three shuffles are drawn per LLM run; because they share that run's logs they are not independent replicates, and because each shuffle-average is built from one LLM run, the two arms are *paired*. We therefore average the three shuffles to one value per original run and test the per-run paired difference $d = \mathrm{err}(\mathrm{LLM}) - \overline{\mathrm{err}}(\mathrm{shuffle})$ with a site-stratified sign-flip permutation test (10,000 flips; statistic: mean over sites of the per-site mean difference; Holm within component family); the unpaired stratified test is reported alongside. Results (`shuffle_twin_tests.csv`): T0/T1 show no detectable effect on any component (paired Holm $p \geq .07$; smallest for T1 coupling); T2/T5 clustering improvements and degree/assortativity degradations replicate (paired $p < .001$).

Statistics. Multi-site: site-stratified permutation (10,000) on the pooled mean of per-site means; single-site: run-level permutation (5,000–10,000). Holm within each declared family. Sidedness: the in-vivo Stage-1 tests (Def. 1), the raw-value tests (Table 6) and the scale-matched checks are two-sided; the public-dataset *certification* tests (closed-model replication and the open-weight run; `bench_arch_eval.py`, `openweight_eval.py`) are one-sided in the improvement direction, because certification asks whether the LLM error lies below both the twin's and ER's. A one-sided non-detection is therefore silent about degradation; the one public cell reported as worse than its twin comes from the two-sided scale-matched check. Effect sizes: within-site Cliff's δ; per-site direction consistency n/N_{sites} reported beside every p. Probe statistics are tested against protocol-matched Monte-Carlo random nulls (1,000 simulations of the identical decision/exposure structure). Bootstrap CIs: 2,000–10,000 resamples. All inference is simulation-level, conditional on the observed sites. PYTHONHASHSEED pinned; all seeds in the repository.

E Replay Recalibration and Its Validity Boundary

Because the T0/T2 payload contains only the agent's own attributes and the candidate list — never the evolving graph — a completed run's logged decision sequence is invariant to the post-decision edge parameters $\{\alpha,\ \text{weight rule},\ \text{rank weights},\ p_{\text{new}},\ \text{add-if-absent}\}$. These can therefore be re-searched per model by replaying logged choices (300 configurations; selection on held-in seeds, evaluation on

Table 7: Common edge count m^* per architecture $\times$ classroom (observed m_{obs}: A 216, B 175, C 146, D 163). $m^* = \min\{m_{\text{obs}},\ \min_{\text{LLM, twin runs}} \#\text{distinct edges}\}$; the permuted arm is projected to the same m^* but does not enter the minimum.

Arch.	A	B	C	D
T0	211	161	99	104
T1	101	19	95	56
T2	81	112	31	100
T3	67	58	75	85
T4	177	45	98	32
T5	138	108	84	83

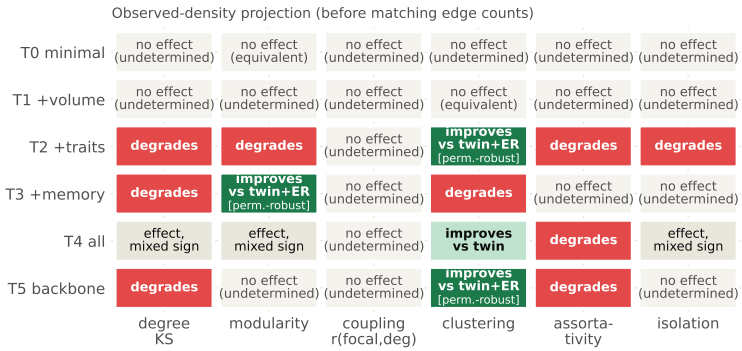


Figure 6: **Before density matching**: the same three-stage matrix at the observed-density projection (m_{obs}), i.e. what an audit that does not match edge counts would conclude. The T3 modularity cell reads "improves, perm.-robust"; it is a density artifact (Table 5).

held-out seeds), at zero additional API cost. Replay is bit-exact *conditional on* the logged candidate sets, model outputs, top-k, and random-number streams (validated by reproducing original runs exactly). Two boundaries: (i) top-k is prompt-embedded and NOT replayable — a model's scores under k=2 are not the truncation of its scores under k=4 (held fixed at the logged value); (ii) memory architectures break the invariance entirely (decisions depend on graph state), which is why the grid uses uniform twin-based calibration instead, with replay retained as a secondary fairness check where valid. Anchor-study outcome: no default variant improves significantly under per-model recalibration; only the trait-aware variant does ($1.18 \to 0.96$, Holm p=.0009, Cliff's $\delta = -1.0$), remaining last.

F Reproducibility and Incident Disclosure

Code and data. All code (simulator, architectures T0–T5, twins, evaluation, statistics), frozen inputs (rosters, per-model score distributions, calibrated parameters), aggregated result tables, and the verbatim prompt instances are at `https://anonymous.4open.science/r/ClassNetBench-24CD`; `classnetbench/reproduce.sh` reproduces the CPU-only pipeline end to end without private data or API keys. In-vivo classroom data are not distributable (Appendix A).

Calibration. Every architecture $\times$ site: random search, 100 trials $\times$ 3 repetitions, on the architecture's own random twin (no API); search spaces in the repository (`arch_defs.py`). Public-dataset calibration uses the identical procedure with dataset trait keys and focal traits.

Prompts. One base prompt for all providers (verbatim in the repository); per-architecture payload deltas are strictly additive: alter-trait entries (T2/T5), history annotations `past_interactions/reciprocated` (T3/T4); the C axis is purely mechanical (threshold θ) — byte-identical prompt to T0. Temperature 0 where supported; Claude Sonnet 5 accepts no temperature; Gemini 3.6 Flash at thinking level minimal; GPT-5.4-family reasoning effort left at API default. SDK versions pinned (openai 3.0.0; anthropic 0.121.0; google-genai 2.17.0).

Determinism. `PYTHONHASHSEED=0` throughout; per-decision RNG streams derived from (seed, round, agent); frozen rosters, score distributions, and calibrated parameters committed.

Incidents (all disclosed, contaminated runs discarded and re-executed to 0% fallback). (i) Gemini's 10,000 requests/model/day quota caused fallback contamination on two occasions (and is the likely cause of an earlier anchor-study incident); reruns were scheduled after quota reset. (ii) Gemini ignores `top_k`=1 and scores *all* presented candidates; with 30 trait-annotated candidates the response truncates at 512 output tokens ($\sim$10% parse failures) — re-run at 2048 tokens (0%). Related to the fill-to-limit completion bias observed in the T1 pilot (models fill to the stated selection limit regardless of instruction, 80/80 decisions).

Compute and cost. LLM usage $\approx$ 320k agent decisions across all studies (rough order US$100–150 total); everything else (twins, calibration, references, mechanism recovery, scaling) is CPU-only. All analyses are scripted and reproducible. AI assistance is disclosed in the main-text "AI Use Statement" in accordance with the ICLR 2027 AI Policy for Authors.

G ClassNetBench Details

Dataset cards. *knecht_klas12b*: Knecht (2008) Dutch classroom, n=23 after structural-missing removal, friendship wave C, traits {sex, age, ethnicity, religion, delinquency, alcohol} (z-scored, missing mean-imputed and counted), focal = delinquency. *glasgow_s50*: Teenage Friends and Lifestyle excerpt, n=50, wave 1, traits {alcohol, drugs, smoke, sport, family event}, focal = sport; a high-structure regime (modularity 0.67, clustering 0.49, arc reciprocity 0.69). Both fetched from the SIENA archive by `fetch.sh` under its research-use terms; never redistributed. *synthetic_**: planted-mechanism generator with homophily/popularity/block/reciprocity knobs and stored ground truth; presets {null, homophily, popularity, blocks, full}.

Interface and runner. `load(name)` $\rightarrow$ {nodes (z-scored traits), directed edges, focal trait, meta}; `run_bench.py -dataset X` produces targets, conditional references, and component evaluation of supplied run directories; `reproduce.sh` executes the full CPU-only pipeline (fetch $\rightarrow$ references $\rightarrow$ twin smoke $\rightarrow$ mechanism-recovery self-audit) in one command, verified end-to-end.

Validity notes. The architecture suite runs on public data unchanged (T0/T5 twins on s50 verified end-to-end: T0 twin KS 0.08–0.10 vs. conditional ER 0.076). Two validity notes: (i) the generator's planted popularity mechanism is recovered by the trait-logit reference (coupling error 0.077 vs. 0.499 for ER) and absent under the null preset; (ii) s50 constitutes a substantially more structured regime than the in-vivo classrooms (modularity 0.67 vs. 0.36–0.49), extending the benchmark beyond a single regime.

Mechanism-recovery suite (full). An audit paper must audit itself. We instantiate six *ground-truth policies* with oracle access (random; trait homophily; trait popularity; triadic closure; reciprocity; memory reinforcement) and run each through the identical simulator shell and audit machinery against *self-consistent targets*: the observed network for mechanism m is a held-out run of the same $(\text{policy}_m, \text{shell})$ simulator, so each target genuinely originates from its mechanism with matched strength (10 policy seeds vs. 10 twin seeds per pair; Holm within family). Because policy runs are target-independent, the full 6×6 policy–target cross-evaluation costs no additional simulation.

Results. *Specificity is exact*: the random policy certifies nothing on any target, and no policy certifies anything on the random target. *Sensitivity*: 4/5 pre-specified signatures are recovered on the diagonal (homophily and triadic $\rightarrow$ clustering; popularity $\rightarrow$ coupling, with its full fingerprint of six components; reciprocity $\rightarrow$ reciprocity, error 0.031 vs. twin 0.286). The one miss is diagnostic: the oracle memory-reinforcement policy expresses as *dyadic persistence* (reciprocity certified; modularity directionally better, 0.013 vs. 0.018, but n.s. at n=10)—consistent with the in-vivo finding that memory yields persistence and concentration, not density-matched community structure (§5). Off-diagonal certifications are few and *principled*: the clustering signature is shared within {homophily, popularity, triadic}, mirroring the classical non-identifiability of homophily and transitivity from cross-sectional structure (Shalizi & Thomas, 2011)—the audit correctly refuses to distinguish what the component set cannot distinguish, i.e., it recovers mechanism *equivalence classes* rather than overclaiming point identification.

Network-size scaling. Repeating the recovery suite at $n \in \{25, 50, 100, 250, 500\}$ (mechanisms {random, homophily, popularity, memory}; CPU-only): across the 20 policy$\times$size cells including the random null, 19/20 expected outcomes are recovered: the random policy certifies nothing at any size (5/5), and among the non-random mechanisms 14/15 signatures are recovered—homophily and popularity at every size, with certification breadth *growing* with n as statistical power increases. The single miss is itself interpretable: at n=500 with the candidate pool fixed at 30, the memory policy's re-encounter probability with previously chosen partners dilutes ($\approx 6\%$ per prior tie per decision), so the reinforcement mechanism barely expresses—another instance of the framework's central point, that the choice protocol (here, pool size relative to n) gates what any policy can express. The audit's validity is not an artifact of classroom scale. Full matrices: `confusion_matrix.csv`, `scaling_matrix.csv`.

H Design Choices in Published LLM Social Simulators

Table 8 records, for representative LLM social simulators, the design axes the audit toggles. Entries follow the papers' method descriptions; "n.r." = not reported. Two patterns motivate T0 and the controls: (i) no surveyed system reports an architecture-matched random-policy baseline or an identity/position control, and (ii) agents almost universally carry human names. The classroom-simulation systems closest to our setting (SimClass, EduMirror) expose peer descriptions and maintain interaction history, i.e. they sit between T2 and T4 in our grid; the network-generating systems (S^3, OASIS, AgentSociety) accumulate ties through repeated interaction choices from presented candidates or feeds, i.e. the mechanics of T0's edge dynamics with variable volume. T0 is therefore the *minimal common core* (persona-conditioned scoring of presented candidates with accumulating ties), not a description of any one system; T1–T5 add the axes on which the surveyed systems differ.

Table 8: Design axes of representative LLM social simulators (from published method descriptions; n.r. = not reported). Columns: alter attributes shown to the agent; interaction memory; volume per step fixed (top-k) or variable; ties formed by accumulation of interaction choices; identity/position controls; architecture-matched random-policy baseline; human names for agents.

System	alter attr.	memory	volume	tie accum.	id/pos. ctrl	random twin	names
Generative Agents (Park et al., 2023)	via dialogue	yes	variable	n.r.	no	no	yes
S^3 (Gao et al., 2023)	n.r.	n.r.	n.r.	yes (network sim.)	no	no	n.r.
Social-media sim. (Törnberg et al., 2023)	personas	n.r.	n.r.	feed-mediated	no	no	yes
SimClass (Zhang et al., 2025)	archetypes	dialogue history	variable	n.r.	no	no	yes
OASIS (Yang et al., 2024)	partial (profiles)	yes	variable	follow edges	no	no	yes
AgentSociety (Piao et al., 2025)	partial (profiles)	yes	variable	mutable given network	no	no	yes
EduMirror (Lin et al., 2026)	n.r.	yes ($\mathcal{H}_{<t}$)	n.r.	n.r.	no	no	yes
T0 (this paper)	no	no	fixed top-k	yes	yes (audit)	yes (audit)	no (IDs)

I Additional Artifact Analyses

Related work: simulated students. LLM "simulated students" support teacher training and tutoring research (Zhang et al., 2025; Scarlatos et al., 2026; Yuan et al., 2026), with validity work at the dialogue and learning level; their social-structural validity—a precondition for peer-relations applications—has not, to our knowledge, been audited.

Label-preference consistency by format (L=30; cross-block Spearman; $*$ = MC $p<.01$): names(EN) — 4o-mini 0.59^*, 5.4-nano 0.82^*, Haiku 0.39^*, Sonnet 0.45^*, Gemini 0.34^*; names(JA) — 0.22^*/0.83^*/-0.08/0.25^*/0.06; classroom IDs — 0.17^*/0.69^*/-0.05/-0.05/-0.04; numerals — 0.07/0.43^*/0.04/0.05/0.12; letter pairs — 0.02/0.28^*/0.02/0.02/0.01; hex-8 — -0.01/0.28^*/-0.02/-0.02/-0.04.

Positional bias (B_{top10}, share of selections presented at positions 1–10 of 30; expectation 0.333): Haiku 0.999; 5.4-mini 0.999; 4o-mini 0.991; Sonnet 0.990; Gemini 0.881; 5.4-nano 0.873; 5.4 0.870; random control 0.336.

Positional bias under forced full scoring (Table 9). In the main probe the model returns scores for up to k=4 of L=30 candidates, so B_{top10} measures which candidates it *chooses to score*; a model that reads the whole list but stops writing after four entries would look position-biased through

Table 9: **Positional bias under forced full scoring** (classroom IDs, L=30, 320 decisions per model, all 30 candidates scored). Three prompts differing only in the volume line: the base "sorted by descending score" clause (`-score_all`); "listed in the same order as the input candidates" (`-score_all_inorder`); no ordering instruction, "entries may be listed in any order" (`-score_all_free`). Under both re-runs every model returned all 30 scores in input order in 100% of decisions. Full: share of decisions returning all 30 scores. ρ_{pos}: Spearman correlation between presented position and score, pooled over decisions (n/a: constant scores). $\bar{s}_1/\bar{s}_{30}$: mean score at positions 1 and 30. Mono./Ramp: share of decisions with scores non-increasing in presentation order / arithmetic ramps; Const.: share returning one constant score. B_{top10}: share of the four selected candidates presented in positions 1–10 under the main truncated (k=4) protocol (expectation 0.333).

	sorted-by-score clause				input-order clause			no ordering clause			
Model	Full	ρ_{pos}	$\bar{s}_1/\bar{s}_{30}$	Mono./Ramp	ρ_{pos}	$\bar{s}_1/\bar{s}_{30}$	Const.	ρ_{pos}	$\bar{s}_1/\bar{s}_{30}$	Const.	B_{top10} (trunc.)
GPT-4o-mini	0.96	−0.93	0.84/0.01	0.96/0.00	+0.00	0.33/0.37	0.00	−0.04	0.41/0.40	0.00	0.963
GPT-5.4-nano	0.81	−0.53	0.72/0.51	0.33/0.14	−0.17	0.62/0.49	0.00	−0.22	0.62/0.47	0.00	0.731
Claude Haiku 4.5	1.00	−0.74	0.54/0.09	1.00/0.87	n/a	0.50/0.50	1.00	n/a	0.50/0.50	1.00	0.995
Claude Sonnet 5	1.00	−0.15	0.51/0.47	0.69/0.05	n/a	0.50/0.50	1.00	−0.00	0.50/0.50	0.99	1.000
Gemini 3.6 Flash	1.00	−0.67	0.69/0.04	0.57/0.47	−0.01	0.13/0.13	0.97	−0.01	0.16/0.16	0.94	0.996

truncation alone. We therefore re-ran the classroom-ID cell at L=30 (5 models × 320 decisions; 0 parse failures) with k=L and a prompt requiring exactly one score per candidate (`-score_all`; `position_forced_scoring.csv`), changing only the volume line and therefore keeping the base prompt's "sorted by descending score" clause (Appendix J). Every model complied in $\geq 81\%$ of decisions. Pooled over decisions the score falls with presentation position in every model (Spearman -0.93 GPT-4o-mini, -0.74 Haiku, -0.67 Gemini, -0.53 GPT-5.4-nano, -0.15 Sonnet 5; mean score at position 1 vs. 30: 0.84 vs. 0.01, 0.54 vs. 0.09, 0.69 vs. 0.04 for the first three), GPT-4o-mini and Haiku return non-increasing sequences in presentation order in 96% and 100% of decisions, and arithmetic ramps ($0.50, 0.48, \ldots, 0.01$: non-increasing, ≥ 25 distinct values, at most two step sizes) make up 87% of Haiku's decisions, 47% of Gemini's and 14% of GPT-5.4-nano's. A model that writes the list top-down can satisfy the sorting clause only by emitting a declining sequence, so this pattern is ambiguous between instruction-following and an intrinsic positional score bias, and the logs (score-sorted output, raw order not recorded) cannot separate the two.

Re-runs without the sorting clause. We therefore re-ran the identical cell (same seeds, pools and decisions) twice, replacing the sorting clause by (a) "listed in the same order as the input candidates" (`-score_all_inorder`; `position_forced_scoring_inorder.csv`) and (b) no ordering instruction at all, "entries may be listed in any order" (`-score_all_free`; `position_forced_scoring_free.csv`). Compliance was complete in both: 0 parse failures, all 30 scores returned in every decision, and—also under (b), where the order was free—every model wrote the entries in input order in 100% of decisions, so models write the list top-down by default and the logged order is their own. The decline disappears in both. Claude Haiku 4.5 returns the constant score 0.50 for every candidate in 100% of decisions under (a) and (b), Claude Sonnet 5 in 100% and 99%, and Gemini 3.6 Flash a constant (0.125; 0.157) in 97% and 94%: given bare identifiers and no ordering instruction they express no preference among candidates in this condition. This is consistent with P1, which forbids conditioning on alter attributes, but is not a test of it: a label-preferring policy would satisfy P1 equally, and the same models do discriminate by position under the truncated protocol. GPT-4o-mini returns a repeating non-monotone template ($0.30, 0.40, 0.50, 0.20, 0.60, \ldots$) uncorrelated with position (a: $\rho = +0.00$, p=.67; b: -0.04, mean 0.41 at position 1 vs. 0.40 at position 30). Only GPT-5.4-nano keeps a positional gradient, reproducibly but far weaker than under the sorted prompt: $\rho = -0.17$ (a) and -0.22 (b), carried mainly by an elevated first position (0.62 vs. 0.47–0.53 thereafter), with its unique highest score in positions 1–10 in 69% and 85% of decisions (expectation 33%). The score-level decline under the sorted prompt was therefore largely an artifact of the sorting instruction; we report all three conditions and base the claim on the two without it. What survives is (i) the selection-level bias of the truncated protocol—which four candidates a model writes about when it may write only four (87–99.9% from the first ten positions), a choice no sorting instruction dictates and the one the simulator's top-k interface actually exercises—and (ii) a mild first-position preference in GPT-5.4-nano's scores. Both are properties of presentation; randomizing candidate order per decision remains necessary, and prompt clauses that couple output order to score should be treated as part of the architecture I when a probe is designed.

Name attributes (all five models, both rosters): female/male chosen-rate point estimates at or above parity in all 20 cells, CI excluding 1 in 17/20 (the exceptions: Claude Haiku JA at L=30, Gemini EN at both L); ratios at L=30 from 1.0 after rounding (Claude Haiku, JA) to 8.7 (GPT-5.4-nano, EN; 95% bootstrap CI 5.9–13.1), 9.8 (6.5–17.7) at L=10; frequency-rank correlations -0.34 to -0.71. Frequency-and-length-adjusted gender coefficient positive in 20/20 cells, CI excluding zero in 16/20 (`gender_effect_ci.csv`). Fig. 7 shows the ratios with CIs.

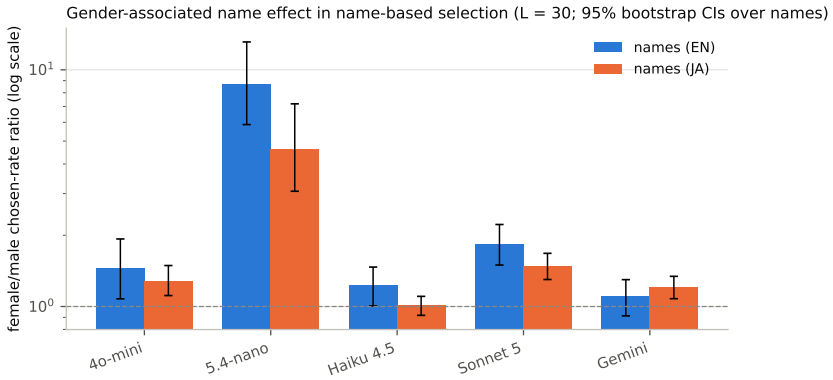


Figure 7: **Gender-associated name effect** (surface-form probe, L=30): female/male chosen-rate ratio per model and roster with 95% bootstrap CIs over names. Rosters are synthetic and gender-balanced; the effect is of the name string, not of any person.

Cross-model consensus (per-name popularity between providers): raw Spearman 0.37–0.82; 0.22–0.78 after regressing out gender and frequency — an ecosystem-level name prior.

Two-space consistency for T2/T5/T3 (permuted arms, 5 seeds × 4 classrooms; `two_space_t2t5.csv`): real-student ρ = 0.90 (T2), 0.83 (T5), 0.02 (T3); displayed-label ρ = 0.04, 0.04, 0.06; popularity–extraversion ρ = 0.71, 0.64, 0.01. Unpermuted arms give the same real-space values (0.94, 0.86, 0.05).

In-vivo two-space permutation: real-student space $\rho = 0.01$; displayed-label space $\rho = 0.78$ (unpermuted level 0.78); per-classroom values 0.64–0.86.

J VERBATIM PROMPTS

Every LLM decision in the program—all six architectures, all providers, the public-data replication, and the surface-form probe—uses the *same* prompt template; architectures differ only in what the `candidates` entries of the input JSON contain. The twin never sees a prompt (it draws scores from the audited policy's empirical distribution). Temperature 0 (where supported); the reply must be JSON only; three retries, then the decision is marked as fallback (realized fallback rate: 0% in every reported run).

Why a scoring interface. The JSON scoring interface is a deliberate stylization of the choice protocol C: it makes the selection volume and the score scale explicit, so that the twin can match them exactly. Free-form action selection (dialogue, posting, following) is a different C; the audit's controls attach to the architecture, not to the prompt phrasing, and transfer once volume and candidate exposure are logged.

Template (byte-identical across architectures).

```
You are a decision-making agent in a classroom social network simulation.
Given the input JSON, score each candidate in [0,1] where higher means more
likely to interact next.
Return ONLY valid JSON. Do not include any extra text.

Required output JSON schema:
```

```
{
  "scores": [
    {"id": "CANDIDATE_ID", "score": 0.0}
  ]
}

Constraints:
- Use only candidate ids provided in input JSON.
- Provide up to top_k items, sorted by descending score.
- score must be a number between 0 and 1.

Input JSON:
<one JSON object, see below>
```

Input JSON, minimal template T0 (in-vivo; illustrative values). The ego sees its own Big Five and classroom-climate z-scores and a list of 30 candidate *identifiers*—nothing about the candidates themselves. Key order is fixed.

```
{"self_id": "A007",
 "BFI_E_z": 0.62, "BFI_A_z": -0.15, "BFI_C_z": 0.30, "BFI_N_z": -0.88,
 "BFI_O_z": 0.11,
 "CCI_friction_z": -0.40, "CCI_satisfaction_z": 0.55, "CCI_closeness_z": 0.20,
 "use_persona": true, "use_climate": true,
 "candidates": ["A031", "A012", "A025", ..., "A003"],      # 30 ids, random order
 "top_k": 4}
```

What changes per architecture. Only the `candidates` entries and `top_k`; the instruction text above is unchanged.

Arch.	one `candidates` entry
T0, T1	`"A031"` (bare identifier; T1 applies the threshold θ to the returned scores mechanically)
T2	`{"id":"A031","BFI_E_z":1.2,"BFI_A_z":0.4,...,"CCI_closeness_z":-0.3}`
T3	`{"id":"A031","past_interactions":3,"reciprocated":true}`
T4	`{"id":"A031", <T2 trait fields>, "past_interactions":3,"reciprocated":true}`
T5	as T2, with `top_k` set per decision by the agent's fitted activity quota

In the pilot of instruction-based volume control the only change was the volume line ("Provide between 0 and top_k items . . . return an empty list if none"); models filled to the limit in 80/80 decisions, which is why T1 uses a score threshold instead and keeps the T0 prompt byte-identical. The three forced-full-scoring controls (Appendix I) likewise change only the volume line: to "Score EVERY candidate: return exactly one entry per candidate id (top_k equals the number of candidates), sorted by descending score." (sorting clause retained); to the same sentence ending ". . . , listed in the same order as the input candidates."; and to ". . . (top_k equals the number of candidates); entries may be listed in any order."

Label permutation arm. The prompt is unchanged; every identifier shown to the model (`self_id` and each candidate) is replaced by a per-run random bijection $\sigma_r(\cdot)$ of the real IDs, and returned ids are mapped back through σ_r^{-1} before the edge update. The model therefore sees a consistent but arbitrary labeling; the simulated persons (traits, history) are untouched.

Ablations and the surface-form probe (ID-only). Removing the persona block drops the five `BFI_*` keys; removing the climate block drops the three `CCI_*` keys (`use_persona`/`use_climate` flags are set accordingly). The probe (§6) uses the ID-only variant—`self_id`, the two flags, `candidates`, `top_k` (=4)—with the identifiers drawn from one of the six label rosters (classroom IDs, numerals, letter pairs, 8-char hex, English names, romanized Japanese names) and $L \in \{10, 30, 60\}$ candidates in fresh random order per decision.

Public datasets. The same template; the persona block carries the dataset's own attribute keys (focal trait and covariates, Appendix G) in place of the BFI/CCI keys, and T2 candidate entries carry the same keys for the alters.