OPEN WORLDS NEED RULE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-world AI systems are rapidly expanding through larger content spaces, richer task sets, more coherent agent behaviors, and more flexible language interfaces. This position paper argues that such progress masks a more fundamental bottleneck: systems may appear open because agents can see, say, or attempt more, while remaining closed at the mechanism layer that determines what the world can recognize, validate, execute, and reuse. We call this mismatch the *content trap*. We argue that the next bottleneck of open-world AI is not generating more content, tasks, or behaviors, but enabling the endogenous growth of executable world mechanisms. To this end, we propose *rule distillation*: treating LLM-generated intents, justifications, and consequence predictions not as direct world updates, but as candidate rule priors that must earn validity through mechanistic formalization and execution, pass finite admission checks including replay, and stay subject to audit tracing and failure rollback once they become reusable world mechanisms. Through a scarcity-driven trajectory in the Farm-Town simulator, we trace how one-off generated behavior can be distilled, under explicit checks, into executable and reusable mechanisms. Controlled studies in a ten-agent world support the acquisition step. In preregistered studies with two model families, once an arrangement made through an acquired relation is complete, acquired rules enforce the obligations it carries by refusing the independent actions that would break them; composing the initial actions leaves many of them unenforced, synthesized prohibitions close them only in part, and a generic obligation primitive provisioned in advance covers two of the four demands. Finite admission checks select candidates but do not certify them, and admitted rules can refuse unrelated lawful actions, so deployment also requires a sealed semantic challenge that tests both. With GPT-6 Astra and a supplied demand, multi-role interaction yields rules passing both sealed suites in 29 of 64 blocks, against 24 for full-information drafting and 2 for public facts alone, whereas with Claude Opus 5.5 both designer arms pass more often; participants given only their needs propose new relations, which Claude Opus 5.5 formalizes, with development feedback, into rules passing both suites. We therefore argue that the core progress of open worlds is not to generate more content, but to gradually acquire the ability to simulate real-world rules.

1 INTRODUCTION

Open-world systems face a persistent tension: agents should act in ways their designers did not anticipate, and those actions should reshape the world through mechanisms that accept them and produce traceable consequences (Stanley & Lehman, 2015; Lehman & Stanley, 2011; Clune, 2020; Team et al., 2021; Guss et al., 2019; Fan et al., 2022). Large language models (LLMs) enable richer dialogue, action descriptions, and narrative branches (Park et al., 2023; Yao et al., 2023; Shinn et al., 2023; Wang et al., 2023a; Li et al., 2023; Sumers et al., 2024; Xi et al., 2023), yet expanding content space does not by itself grow mechanism space: systems may expand what agents can see, say, or attempt while leaving fixed what the world can recognize, validate, and reuse. We call this the content trap. Acquiring executable mechanisms is a central research objective for open worlds; LLM-generated intents can supply hypotheses about missing rules. They already write content (Togelius et al., 2010; Maleki & Zhao, 2024; Summerville et al., 2018; Khalifa et al., 2020; Justesen et al., 2018; Cobbe et al., 2020), guide embodied exploration (Hafner, 2022; Küttler et al., 2020; Shridhar et al., 2021; Puig et al., 2018), and drive social dialogue in multi-agent frameworks (Wang et al., 2024; Qian et al., 2024; Guo et al., 2024). Rule distillation asks when a proposed interaction should change

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.

the world's executable rules, and requires that the resulting update preserve existing constraints and specify consequences for later use.

We argue that open-world systems should use LLMs as sources of rule priors (candidate hypotheses about missing rules) for executable world mechanisms. Rule hypotheses are candidate operational relations suggested by generated language: how actions are constrained by resources, tools, locations, and capabilities (Petroni et al., 2019; Roberts et al., 2020; Feldman et al., 2019; Bosselut et al., 2019; Sap et al., 2019a;b; Hendrycks et al., 2023; Weidinger et al., 2021; Bommasani et al., 2022; Bender et al., 2021); how interactions are shaped by rights, obligations, commitments, and feedback (Shoham & Tennenholtz, 1995; Singh, 2007; Morales et al., 2013; 2015); how institutions emerge from conflict, coordination, and repeated exchange. Rule distillation subjects these priors to executable definition and admission checks before they enter shared execution. Audits and rollback govern revisions when later interactions reveal conflicts. Generated actions, justifications, and consequence predictions express candidate regularities exposed by contextual constraints and interactional pressure. Distillation identifies their conditions, participants, resources, constraints, and persistent effects; formalization makes the relations executable; admission checks govern entry into the shared rule base. Farm-Town gives the framework a concrete world and evaluation setting. Agents act under resource, memory, relationship, and institutional constraints; candidate mechanisms require implementation, replay checks, and audit before admission. Alongside the 90-day trajectory, controlled studies compare existing-action composition with model-generated shared rules, and test acquisition from proposals, development feedback, and source construction, then sealed semantic checks, later use, and shared enforcement across four demands. This paper makes four contributions: the content-trap diagnosis; LLM-suggested rule hypotheses as an explicit research object; rule distillation as a paradigm for mechanism growth; and evaluation through consolidation, validation quality, enforcement, and reuse.

2 Content Expansion Is Not World Openness

Table 1: Scope and structure across research approaches (examples cited in Sections 1–2).

Research line and examples	Given structure	Expanded scope
Content generation PCGML (PCG via machine learning), PCGRL (via RL), multi-task PCGRL	Game rules, action spaces, playability criteria, evaluation functions	Maps, levels, tasks, items, events, content forms
Generative world models GameGAN, Genie	Interaction interfaces, action conditioning, dynamics priors, controllability objectives	Interactive environments, visual dynamics, controllable worlds
Embodied task environments MineDojo, Minecraft benchmarks	Physical and crafting rules, item systems, rewards, environment mechanisms	Tasks, language goals, knowledge sources, exploration
Lifelong LLM agents Voyager, DEPS, Ghost in the Minecraft	World dynamics, executable APIs, skill verification, environment feedback	Exploration strategies, skill libraries, tool use, task solving
Cross-environment agents SIMA, multiworld agents	Action interfaces, environment rules, success criteria, world feedback	Language instructions, transfer across environments, 3D execution
Generative social agents Generative Agents, Social Simulacra	Sandbox, observable events, action interface, memory and behavior updates	Memory, reflection, planning, social behavior, group dynamics
Executable model and rule synthesis WorldCoder, LLM-to-PDDL, mechanic generation, GAVEL	Representation, task objectives, evaluation constraints, playability fitness	Transition models, action models, game mechanics
Normative multi-agent systems Social laws, commitment protocols, IRON	Agent action space, world dynamics, norm language, designer objective	Deontic norms, commitments, and sanctions over a fixed action set
Rule distillation (this paper) Farm-Town, controlled studies V1–V11	Initial interface $\mathcal{R}_0$, bounded rule language, declared mechanism classes, supplied or need-driven demands, admission checks, sealed suites	Executable relations and obligations acquired from interaction, enforced once arranged through them

Content expansion adds maps, tasks, dialogue, and events; mechanism acquisition adds executable relations governing subsequent interactions. Table 1 compares what each research line expands and what structure it assumes as given; only executable model and rule synthesis and normative-system synthesis target operational relations, and rule distillation differs from both in what it installs and for whom. Rule distillation asks how interaction exposes candidate relations and how worlds validate and retain them for shared execution. Rule generation has substantial precedents. Mechanic generation synthesizes rules under design and playability constraints (Zook & Riedl, 2014); Code World Models translate rules and trajectories into executable simulators (Lehrach et al., 2026); WorldCoder learns a programmatic transition model through interaction (Tang et al., 2024); LLM-to-PDDL methods construct action models and revise them using validator and human feedback (Guan et al., 2023); action-model learning and inductive logic programming infer executable relations from examples (Yang et al., 2007; Muggleton, 1991); EnvGen adapts training configurations from agent performance feedback (Zala et al., 2024); GAVEL evolves executable game rules in Ludii under a playability fitness (Todd et al., 2024), and OMNI-EPIC generates environments and tasks as code under a model of interestingness (Faldor et al., 2024). These methods supply candidate-generation and environment-adaptation tools, and each selects candidates by a designer's objective. Normative multi-agent systems formalize the layer these candidates would enter: institutionalized power makes designated acts count as normative facts (Jones & Sergot, 1996), and electronic institutions specify enforceable interaction protocols (Esteva et al., 2001). That literature separates constitutive rules, which create institutional facts and act types, from regulative ones, which permit, oblige, or sanction (Boella & van der Torre, 2006), and it already synthesizes the regulative kind automatically, including on line from observed conflicts (Morales et al., 2013; 2015). What is new here is therefore not automation but the class of addition and where it comes from: those systems constrain an action set the world already executes, and the social constraints of a norm-governed society are otherwise specified as fixed during execution (Artikis et al., 2009), whereas an admitted candidate adds the transition and the persistent state an obligation would be about, is proposed in interaction rather than derived from a conflict monitor, and enters shared execution only through Eq. (4) and a sealed suite. Our position concerns admitting candidates to shared world execution: a revised predictor can improve agent knowledge without changing world transitions, whereas admission changes the executable library and can change those transitions for every later caller (Appendix C). Smart contracts encode escrow and collateral terms (Szabo, 1997), LLM game masters resolve actions in language (Vezhnevets et al., 2023), and self-organizing institutions amend their own rules (Pitt et al., 2012); rule distillation admits interaction-proposed relations only through executable checks.

Generating world forms is not growing world mechanisms. Procedural content generation scales maps, levels, tasks, and events, and PCGRL optimizes content against design objectives; GameGAN simulates dynamic game environments, and Genie generates action-controllable worlds from video or prompts (Kim et al., 2020; Bruce et al., 2024). Level variation preserves movement, collision, and reward rules; a learned world model adapts dynamics within its action interface. MineDojo, Voyager, and SIMA develop agent capability (Fan et al., 2022; Wang et al., 2023a;b; Zhu et al., 2023; Team et al., 2024) through tasks and external knowledge, automatic curricula and reusable skills, and language instructions followed across 3D environments; their boundary is the given rule system: skills recombine existing mechanisms but add no shared action types, resource relations, or institutional constraints.

Generative Agents models social behavior through memory, reflection, and planning (Park et al., 2022; 2023), and CAMEL scripts role-playing agent communication (Li et al., 2023), extending open-world research beyond physical exploration and task completion; larger societies go further: in Project Sid, LLM agents in Minecraft adopt constitutions, taxes, and roles (Altera.AL et al., 2024), and LLM populations converge on shared conventions (Ashery et al., 2025). In these systems the adopted norms bind through the agents' own behavior while the world executes the same initial actions for everyone, and the AI Economist optimizes a tax mechanism whose structure the designer fixes in advance (Zheng et al., 2022); behavioral coherence and adopted convention do not themselves change the operational structure.

Open-endedness is usually defined as novelty and learnability relative to an observer (Hughes et al., 2024), and the artificial-life literature separates variation inside a state space from change to the rules that generate it (Banzhaf et al., 2016). These trajectories expose the content trap: content, tasks, skills, and social behavior can expand while operational relations remain fixed at construction, and evaluating

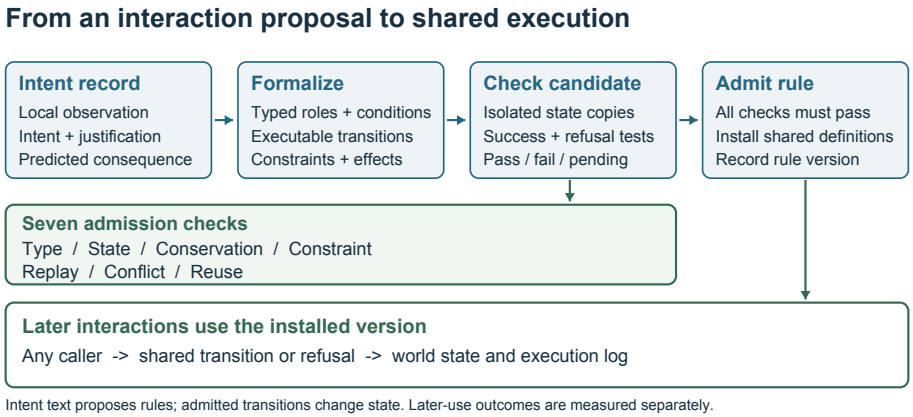

Figure 1: Conceptual architecture of rule distillation: unsupported intents reveal gaps in $\mathcal{R}_t$, all seven checks must pass before installation, and deployment requires the sealed challenge (Section 7).

openness by content diversity leaves mechanism acquisition unmeasured. Escaping it requires mining generated objects, tasks, dialogue, and events for candidate rule priors and consolidating them into verifiable, reusable world mechanisms.

3 Rule Distillation for Mechanism Growth

Rule distillation treats open worlds as mechanism growth systems (Figure 1). Actions, justifications, and consequence predictions express hypotheses about resources, tools, constraints, feedback, social relations, and institutions; we use "priors" operationally for these generated assumptions about feasibility and outcomes, and admission checks compare their preconditions, state changes, and refusal behavior with the declared contracts. A distilled candidate must specify applicability, participating entities and resources, state changes, constraints and permissions, failure or refusal, and behavior across analogous contexts. Admitted rules expand behavior when they add executable action types, trackable state variables, resource relations, binding protocols, or institutional conditions affecting later decisions. Definitions persist in a versioned rule base; later checks govern revision and withdrawal. Content generation completes a context, whereas rule distillation expands future actions; rule-growth evaluation tracks how these relations change available interactions, while object, task, and narrative counts measure content coverage.

4 A Simulator for Rule Distillation

Farm-Town provides a concrete world specification for mechanism growth. The world is a pair $(s_t, \mathcal{R}_t)$: s_t records environment, resources, agents, memory, relationships, and institutional conditions (illustrated in Figure 7, Appendix A), and $\mathcal{R}_t$ governs how actions are interpreted, executed, and constrained. Mechanism-level openness is the acquisition of validated operational relations in $\mathcal{R}_t$, as distinct from generated objects, tasks, or executable agent behaviors. Mechanism growth is relative to the initial state schema and execution interface: instances fill existing templates, skills compose available actions, and mechanism additions extend the operational relations governing later interactions; a fixed interpreter can support such additions (Appendix A.19 gives the classification criteria and a credit example). The platform separates scenario facts (locations, roles, resources, initial conditions, environmental constraints) from general mechanisms (action interpretation, state transition, conservation checking, social observation, governance, replay, auditing). Candidate transitions and guards enter the shared rule base and govern subsequent calls from any agent.

The intent interface generates agent i's intent $z_i \sim \pi_{\text{LLM}}(\cdot \mid o_i(s_t))$ from locally observable state $o_i(s_t)$, encoding goals, actions, justifications, anticipated consequences, constraints, and risk judgments. Intent records and admissible state deltas live in disjoint spaces, $z_i \in \mathcal{Z}$, $\Delta s_t \in \mathcal{D}_S$, $\mathcal{Z} \cap \mathcal{D}_S = \emptyset$: LLMs propose hypotheses about world operation, and the current rule base $\mathcal{R}_t$ determines state changes. Intents executable under $\mathcal{R}_t$ enter normal interaction; unsupported intents remain recorded proposals and evidence of mechanism gaps. $\text{Exec}(\mathcal{R}_t)$ denotes the supported intent forms; failing a known precondition, such as insufficient gold, does not alone identify a mechanism

gap. An unsupported form produces a proposal, $z_i \notin \mathrm{Exec}(\mathcal{R}_t) \Rightarrow u_i = \mathrm{Proposal}(z_i, s_t, \mathcal{R}_t)$. u_i records a proposed behavior, resource relation, interaction protocol, or institutional response that existing mechanisms cannot express, together with its triggering context; it changes no world state, and any admitted mechanism applies no earlier than the next eligible execution step. Unsupported intents are thus neither discarded, which would hide mechanism gaps, nor executed, which would contaminate world state.

Given proposal u_i, the platform forms a candidate $u_i \xrightarrow{\text{distill}} h_i$ by binding participants to typed roles, turning intended effects into state updates, and representing delayed effects as persistent obligations; its output must carry executable transitions and tests for success, refusal, and failure, and unresolved clauses remain pending. In the controlled studies, LLMs generate the executable rules and development checks select among revisions (Appendix D.11). Rule base updates are governed by a validation condition:

$$\mathcal{R}_{t+1} = \begin{cases} \mathcal{R}_t \cup \{h_i\}, & \mathrm{Valid}(h_i;\, s_t, \mathcal{R}_t) = 1, \\ \mathcal{R}_t, & \mathrm{Valid}(h_i;\, s_t, \mathcal{R}_t) = 0. \end{cases} \tag{1}$$

Revisions undergo the same validation; a rollback restores the prior rule base and is recorded. Admission checks compatibility; novelty evaluation identifies newly supported interactions relative to the initial interface, so a compatible but redundant candidate contributes no new coverage. Invalid, conflicting, or unstably executable hypotheses remain outside $\mathcal{R}_t$, with failed checks retained for inspection. Candidate mechanism h_i is written as an executable structure:

$$h_i = \langle \phi_i, \alpha_i, \tau_i, \kappa_i, \omega_i \rangle. \tag{2}$$

Here ϕ_i gives triggering world states, α_i participating entities and roles, τ_i the state transition over resources, locations, inventories, relationships, or institutional states, κ_i the capability, tool, permission, consent, ownership, or risk constraints, and ω_i downstream effects such as failure, refusal, delayed consequences, or changes to future actions. The schema requires all five components:

$$u_i \xrightarrow{\text{distill}} h_i \quad \text{only if} \quad h_i \in \mathcal{H}_{\text{mech}} = \{h : \mathrm{schema}(h) = \langle \phi, \alpha, \tau, \kappa, \omega \rangle\}. \tag{3}$$

The admission predicate $\mathrm{Valid}(h_i; s_t, \mathcal{R}_t)$ conjoins the checks for entry into the world rule base:

$$\mathrm{Valid}(h_i; s_t, \mathcal{R}_t) = C_{\text{type}} \wedge C_{\text{state}} \wedge C_{\text{conserve}} \wedge C_{\text{constraint}} \wedge C_{\text{replay}} \wedge C_{\text{conflict}} \wedge C_{\text{reuse}}. \tag{4}$$

The seven factors check typing; applicable state; conservation of resources, ownership, locations, and ledgers; capability, permission, consent, and institutional constraints; replay; conflicts with declared invariants; and reuse under alternative bindings. Tests run on isolated state copies with the candidate temporarily enabled; fail and pending both prevent installation, with their statuses retained. An admitted candidate installed in $\mathcal{R}_t$ is an *acquired* rule. These finite checks assess compatibility within the declared test domain; Appendix A.14 defines them and the unresolved-case policy.

5 Farm-Town Case Study and Experiments

In a 90-day, 10-agent Farm-Town trajectory, sustained scarcity prompts proposals across economic and institutional relations: 29 intent edges and 18 salient events distil into 13 mechanism families, 11 admitted and 2 pending (Appendix A.19 gives counting units and the relation-level test). The admitted families span market exchange and inquiry, collateralized food credit, labor-for-food settlement, dependency observation and public complaint, pooled purchase, evidence-backed law triggering, public work relief, and class-session skill transfer. In Figure 2, early edges cluster around Dale, Marcus, and Petra, where hunger and low gold expose barter, quote, and counter-offer mechanisms; middle-phase edges add tool lending, labor settlement, debt pressure, witness reporting, and evidence chains, and later edges connect Lewis and Yuna to public relief, governance, and teaching. Figure 3 records the staged registrations, from market protocols on days 1–7 to law, relief, and teaching on days 67–82. The most diagnostic family is COLLATERALIZED-FOOD-CREDIT: Dale, hungry and short of gold, offers Petra a sickle for bread, which the initial rule base executes as neither purchase, gift, nor borrowing, and a valid rule must link repayment status to redemption or forfeiture, the relation the controlled D1 demand tests. Figure 4 contrasts that arc with an institutional one in which debt pressure, a witness report, and a public complaint become an evidence-backed law trigger: broad law text is rejected, and only a rule carrying event tags, evidence, authority, and consequences is admitted. Figure 5 groups the agents' intents into these families by role. Measured claims rest on the controlled studies (Appendix D).

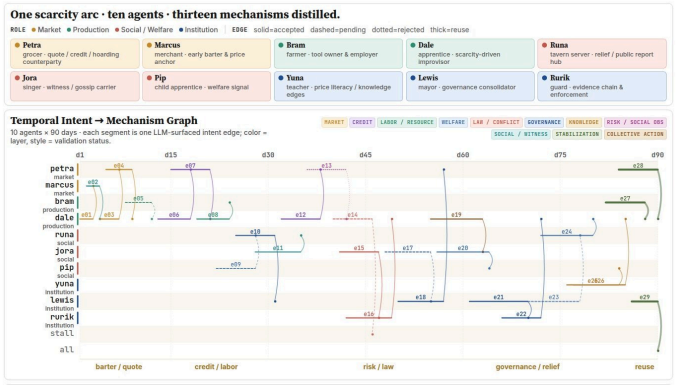

Figure 2: Temporal agent graph of the 90-day trajectory; edge style gives validation status.

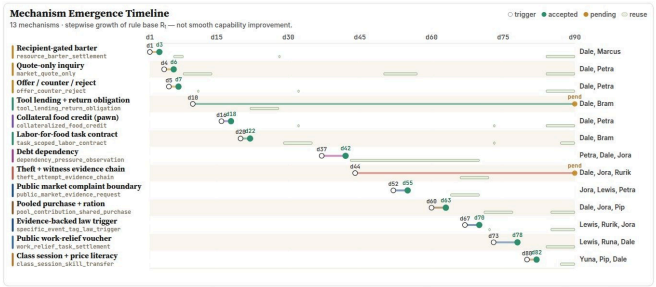

Figure 3: Mechanism emergence timeline of the 90-day trajectory: rule growth is stepwise.

Controlled implementation. The controlled studies (Table 2) use a dedicated implementation of a subset of the Farm-Town specification (Appendix A): ten agents and a library $\mathcal{R}_0$ of 17 initial actions, from movement and production to gifts, listings, loans, hiring, and law enforcement. Its trusted runtime dispatches every action under an authenticated actor and requires recorded approvals or contract authority for cross-party debits. A candidate bundles rules, which add transitions, and guards, which add refusal conditions to existing actions. Four demands supply mechanism gaps that $\mathcal{R}_0$ cannot express as a shared obligation: *D1*, food on credit secured by the debtor's tool; *D2*, a two-contributor purchase settling before a deadline with contributor-specific refunds after it; *D3*, an asset promised to a worker and awarded only on authenticated completion, cancellable before it; *D4*, custody of one asset rotating among three agents on a schedule. V1–V10 supply the demands; V11 gives each participant only its needs. Generation requests Claude Opus 5 or GPT-6 Astra in fresh contexts (GPT-6 Astra alone in V6, Claude models in V9–V11); a *block* is one demand, one requested model, and one repetition; a *binding* relabels participants and assets, each contract class running 20 times; synthesis has six request slots with development feedback, and admission is Eq. (4) on development cases. Two sealed suites run only after admission: the *standard* held-out suite runs the demand's contract classes at 20 bindings, and the *independent semantic* challenge runs eight classes per demand at 20 further bindings (160 cases per candidate), composed of unrelated legal actions, identity-mismatched evidence, replays after settlement, and later disposal. Both are authored once per demand from its public contract, frozen before the first live model call, and graded by host predicates over economic state, receipts, and refusal reasons, never by candidate rule text. A *complete chain* is a later task, planned by a fresh planner that sees only the installed library, in which an admitted candidate passes both suites, its rule is invoked, and goal and safety predicates hold.

Acquired rules enforce completed arrangements made through them; composed actions do not. With probes sealed before generation, V6 finds that the initial library with authored plans (the $\mathcal{R}_0$ *composition*) permits the independent-caller violation that each demand's *discriminating* probe targets (collateral returned before payment in D1, pledged gold spent in D2, the promised asset gifted to an outsider in D3) on every fixture of that demand at all 20 bindings (960/960), while none of the 86 admitted D1–D3 libraries permits it at any binding (0/1,720). V4E, with probes written afterwards, and V9, repeating V6 with Claude Opus 5.5, find the same differential. V8 (Appendix D.23) compiles a suite from the contract clauses after the V6 results, by fixed transformations that read no candidate

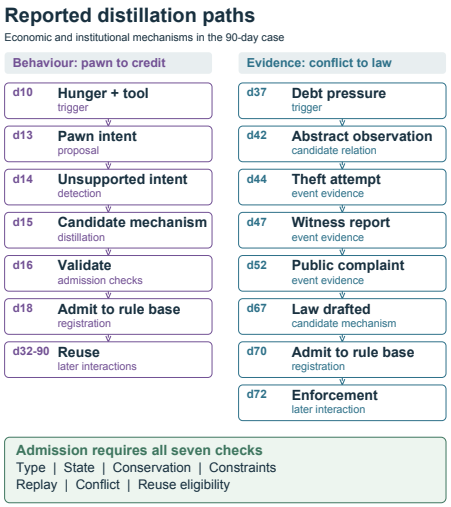

Figure 4: Economic and institutional paths; both require all seven checks of Eq. (4).

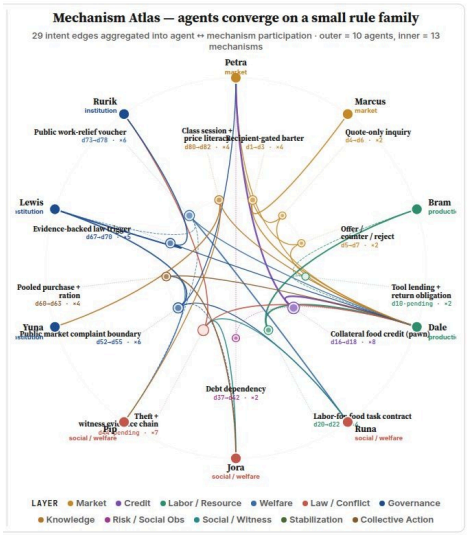

Figure 5: Mechanism atlas of ten agents and thirteen families (Appendix A.20).

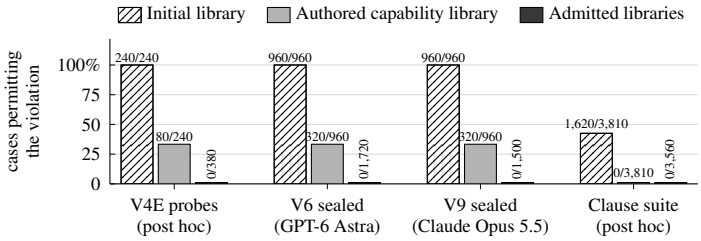

Figure 6: Cases permitting the violation, by library: the three discriminating V4E, V6 and V9 probes and every compiled case; the authored library installs the four reference candidates (Appendices D.17, D.21, D.23, D.24).

Table 2: Controlled studies, 17-action library. I/D: multi-role interaction/designer drafting; priv/pub/full: private, public-facts, or full requirements; Seq/Pool: sequential or pool search.

Study	Design	Headline result
V1	12 candidates; authored controller	all admitted; the controller fails both protection probes
V2	48 episodes; I vs. D	D3 cancellation gap in an admitted rule, found later
V3	16 sources; feedback vs. sampling	admitted 6/16 vs. 0/16
V4	16 blocks; I–Seq, I–Pool, D–Seq	both suites 6/16, 3/16, 4/16
V4E	V4 libraries vs. $\mathcal{R}_0$ composition	probes 0/380 vs. 240/240
V5	16 blocks (8 D–pub refusals)	both suites 7/16, 0/16, 7/16 (I–priv, D–pub, D–full)
V6	64 blocks, preregistered	both suites 29/64, 2/64, 24/64; probes 0/1,720 vs. 960/960
V7	V6 plans under sealed probes	goal met 33/96, 20/96, 31/96 vs. 0/96 for $\mathcal{R}_0$
V8	compiled clause cases, post hoc	violations 0/3,560 vs. 1,620/3,810 for $\mathcal{R}_0$
V9	V6 protocol, Claude models	Opus: both suites 16, 22, 32 of 64; probes 0/1,500
V10	synthesis; generic primitive	synthesis: semantic 0/64; primitive: D3–D4 0/32
V11	needs instead of demands	eligible 49/26/8 (Opus/Haiku/Sonnet); Opus both 8/64

or rule text, and finds the differential on D1, D2 and D4 over the 116 admitted V6 episodes. V8 also records the regression side that Section 7 asks to report alongside: admitted guards refuse an unrelated lawful loan or its return in 520 of 1,175 cases and a lawful disposal after settlement in 250 of 680, almost all guard lookup errors, which neither control refuses. The sealed challenge rejects 42 of the 44 libraries with such refusals, all of which passed the standard suite; the 55 passing both suites refuse 25 of their 750 lawful cases, all in 2 libraries, and permit none of 1,710 violations, against 640/1,710 under $\mathcal{R}_0$. Synthesized prohibitions only partly supply the obligation: IRON-style norms (Morales et al., 2013) synthesized from the V6 development conflicts (V10) close one of the three discriminating probes, the D1 early return (640/960 permitted), and permit all 400 compiled discharge and window violations, as $\mathcal{R}_0$ does. Enforcement, not cooperative reach, is what acquisition adds (Figure 6): it supplies the obligation relation that the initial interface lacks for the demand (Section 6).

Interaction and feedback turn proposals into admitted rules. With the participants' requirements private, multi-role drafting produced confirmed proposals in 62/64 blocks, 54/64 admitted selections and 29/64 selections passing both suites in V6, preregistered with GPT-6 Astra. A designer given every participant's requirements reached 64/64, 46/64 and 24/64, not separated from multi-role drafting by sign test on any tested endpoint (paired difference on both suites $+0.078$, 95% interval $[-0.053, +0.183]$, $p = 0.36$). A designer given public facts alone reached 37/64, 16/64 and 2/64 ($p < 0.0001$ on all three), returning in 26 blocks an ordinary plan that achieved the goal. Development feedback turns proposals into rules: on the same 62 sources and six-request budget, sequential feedback admitted 54 and independent sampling none ($p < 0.0001$), reproducing V3, and with Claude Opus 5.5 (V9) 23 of 52 against 12 ($p = 0.007$). All $p < 0.05$ here survive Holm correction across these tests (Appendix D.27). In V6, interaction supplied what the designer otherwise had to be given: the participants' requirements. With Claude Opus 5.5 (V9) the designer arms pass both suites more often (22/64 and 32/64 against 16/64; Appendix D.24), so this advantage depends on the model, while the enforcement differential, the feedback effect and the two-stage split recur.

Proposals form without a supplied demand. V11 withholds the demand on the 64 V6 fixtures: no common task is given, and each participant's private statement says only what it holds, needs and by when, can offer, and which loss it cannot accept, protecting the outcomes the V6 contracts protect with no arrangement vocabulary. Multi-role interaction then produced a confirmed shared-relation proposal in 49/64 blocks with Claude Opus 5.5, 26/64 with Haiku 4.5 and 8/64 with Sonnet 5; through the V6 synthesis and its development feedback, Opus's selections passed both sealed suites in 8/64 blocks, across D1, D3 and D4 (Appendix D.26). Proposals thus form with no demand stated.

A sealed semantic challenge completes admission, and later planners use the acquired rules. In V6, 111 of 116 admitted selections passed the standard held-out suite and 60 the sealed semantic challenge (55 both), failures concentrating in interleaving and replay after lawful disposal; in V4 every admitted selection passed the standard suite and the challenge split them 13 to 13, in four recurring classes. V2 had shown the same on one candidate whose D3 rule passed admission and 260/260 held-out cases yet permitted cancellation after completion, erasing the worker's settlement (Appendix D.10). The split recurs with Claude Opus 5.5 (16 of 89 admitted V9 and 12 of 21 V11 selections fail the challenge) and for norm synthesis (standard 31/64, challenge 0/64). Deployment therefore requires the sealed challenge, the second admission stage of Section 7. Fresh planners seeing only the installed library invoked the added rules in 71/128 multi-role and 71/128 full-information V6 tasks against 22/128 for public facts, with complete chains in 32/128, 45/128 and 2/128. Later tasks are cooperative by construction and the initial library completes them (27/32 in V4, 95/128 in V6); in 41 paired V6 slots the acquired library does not. When the goal instead requires the arranged obligation to be enforced against a third party (V7, Appendix D.22), only acquired libraries meet it (Table 2); in 21 of the 113 established acquired slots the planner arranged the relation through native actions that the acquired guard does not watch, and the violation succeeded.

6 Alternative Views and Objections

Skill libraries already grow what agents can do. Skill libraries address what agents can compose; mechanism acquisition addresses which obligations the shared world enforces. The $\mathcal{R}_0$ composition reaches every cooperative endpoint yet leaves the discriminating violations unrefused (Section 5). Acquisition extends test-driven synthesis (Shinn et al., 2023) to the shared rule base, admitting interaction-proposed relations, scored by the enforcement differential and the lawful actions they refuse. *Interaction adds nothing that a designer could not draft.* With requirements enumerated for both procedures, V4 finds 6/16 and 4/16 on both suites. The claim concerns elicitation: which demands arise and what requirements they carry need not be enumerated in advance; with GPT-6 Astra, V6 separates a designer given public facts alone from both drafting procedures on every endpoint, while with Claude Opus 5.5 both designer arms pass more often (Section 5). *Admission checks cannot certify LLM-written rules.* On their own they cannot: 13 of 26 admitted V4 and 56 of 116 V6 selections failed the sealed challenge; once a demand is known, its required consequences can be written as tests without writing the mechanism, so deployment requires the second stage of Section 7, with human-designed validators setting the standard (Appendix C).

The bottleneck moves to whoever writes the tests. A suite is written from the demand's contract once, before any candidate for it is generated, so it follows the demand rather than anticipating every

one; its cost scales with demands, not candidates, and it grades by host predicate rather than by case-by-case judgement. *A richer initial interface would make these demands instances.* It would, and Appendix A.19 counts them as instances relative to the declared interface; the question is not expressiveness but who may extend the shared library and under which checks, since a primitive provisioned in advance covers at most the demands it anticipates. On the same suites, a generic obligation primitive specified in advance to express the D1 and D2 contracts (V10) passes both suites only there, in none of the 32 D3 and D4 blocks under any planner, where the acquired V6 multi-role libraries pass in 8 and 14 of 16 (Appendix D.25). *Admitted rules are a privilege-escalation surface.* They are, which is why admission runs on isolated state and refuses any candidate that breaks a declared invariant or regression case; a proposer gaming the finite checks still faces the sealed challenge, which also rejects guards that refuse too much.

7 Measuring Open Worlds by Rule Growth

Open-world evaluation should measure rule-space growth alongside task success, behavioral diversity, content volume, and environment scale; provenance, admission records, and later-use traces identify validated, reusable mechanisms. Let U be unsupported intent proposals, H candidates distilled from them, and $\mathcal{R}_0, \mathcal{R}_T$ the initial and final rule bases. The primary object is rule growth:

$$\Delta\mathcal{R} = \mathcal{R}_T \setminus \mathcal{R}_0. \tag{5}$$

We propose measuring how unsupported intents become rules:

$$Y = |\{h \in H : \mathrm{Valid}(h) = 1\}| \,/\, |U|, \tag{6}$$

where Y is proposal-to-rule yield, which can exceed one when several admitted candidates share a proposal; over eligible confirmed proposals, the counting unit for U in the controlled studies, V6's yields are 54/62, 16/37, and 46/64 for multi-role, public-facts, and full-information drafting at 64 blocks. Declare its counting unit and report replay stability, rejection quality, auditability, and later reuse, assessing rejection explanations separately (Appendix B).

The controlled studies add two reporting requirements. Admission is reported in two stages: the finite checks of Eq. (4) govern entry into the versioned rule base, and a semantic challenge written from the demand's contract clauses, sealed before generation and never used for selection or repair, governs deployment into shared execution. The benefit of an acquired rule is reported as an enforcement differential: after each library's own lawful arrangement, the same independent-caller probes run under the initial and the acquired library, and the report gives the violations and the lawful actions each refuses, separating refusals by installed guards from those by initial preconditions, with cooperative completion reported alongside it. Cross-system comparisons should fix the initial interface and evaluate the same held-out contracts, each counted once regardless of rule implementation, reporting newly supported contracts, regressions, authoring cost, and reuse (Appendix B.8).

8 Conclusion

Open-world research should treat executable mechanism acquisition as a central objective. Rule distillation formalizes the relations that LLM agents propose, checks their admission to shared execution, and tracks their effects in later interactions. The controlled studies locate the contribution: acquisition adds enforcement, on two model families, of arrangements completed through the acquired relation, which synthesized prohibitions only partly supply; the demand for it can arise from participants' needs; and deployment requires a sealed challenge. Newly supported contracts, the enforcement differential reported with the lawful actions it refuses, and later reuse make mechanism growth measurable. That fixes a unit for benchmarks, which should hold the initial interface constant, pose demands it cannot meet, and score by what the world afterwards enforces and reuses rather than by how much content it produced; and a boundary for designers, since the callable interface is part of a world's specification and may grow, but only through explicit admission.

Reproducibility Statement

The controlled implementation (V1–V11 and V4E), its protocols, frozen configurations, fixtures, tests, and run reports are provided as supplementary material; workload seeds and participant bindings are fixed in the configurations, and every downstream evaluation is reproducible from the recorded model calls. V2–V5 used two requested models and the V4E probes were written after the V4 results were known; V6 is prospective, with one requested model, 64 blocks, four hypotheses recorded in advance and the enforcement probes sealed with the semantic challenge before any candidate was generated; V7 and V8 evaluate the recorded V6 artifacts on the host with no model call, V8 through a post-hoc suite compiled from the demand contracts. V9 repeats V6 prospectively with three Claude models, its hypotheses and sealed probes recorded before the first call; V10 evaluates a deterministic norm synthesis and a generic obligation primitive composed by three Claude planners on the V6 fixtures; V11 replaces each V6 demand by the participants' need statements, checked for arrangement vocabulary before its freeze.

AI Use Statement

AI tools assisted with the typesetting of this paper. The research agents evaluated in this paper are LLM-based; the requested models and their settings are described in Section 5 and Appendices D.11, D.15, D.20, D.24, D.25, and D.26.

References

Altera.AL, Andrew Ahn, Nic Becker, Stephanie Carroll, Nico Christie, Manuel Cortes, Arda Demirci, Melissa Du, Frankie Li, Shuying Luo, Peter Y. Wang, Mathew Willows, Feitong Yang, and Guangyu Robert Yang. Project Sid: Many-agent simulations toward AI civilization. *arXiv preprint arXiv:2411.00114*, 2024. URL https://arxiv.org/abs/2411.00114.

Alexander Artikis, Marek Sergot, and Jeremy Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic*, 2009.

Ariel Flint Ashery, Luca Maria Aiello, and Andrea Baronchelli. Emergent social conventions and collective bias in LLM populations. *Science Advances*, 11(20):eadu9368, 2025.

Wolfgang Banzhaf, Bert Baumgaertner, Guillaume Beslon, René Doursat, James A. Foster, Barry McMullin, Vinicius Veloso de Melo, Thomas Miconi, Lee Spector, Susan Stepney, and Roger White. Defining and simulating open-ended novelty: Requirements, guidelines, and challenges. *Theory in Biosciences*, 135(3):131–161, 2016.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Guido Boella and Leendert van der Torre. Constitutive norms in the design of normative multiagent systems. In *Computational Logic in Multi-Agent Systems (CLIMA VI)*, volume 3900 of *Lecture Notes in Artificial Intelligence*, pp. 303–319. Springer, 2006.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa,

Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction, 2019. URL https://arxiv.org/abs/1906.05317.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge (Jimmy) Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jeff Clune. Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence, 2020. URL https://arxiv.org/abs/1905.10985.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Luc Cary, Armando Solar-Lezama, and Joshua B. Tenenbaum. DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pp. 835–850, 2021. doi: 10.1145/3453483.3454080. URL https://doi.org/10.1145/3453483.3454080.

Marc Esteva, Juan A. Rodríguez-Aguilar, Carles Sierra, Pere Garcia, and Josep Lluis Arcos. On the formal specification of electronic institutions. In *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Artificial Intelligence*, pp. 126–147. Springer, 2001.

Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. OMNI-EPIC: Open-endedness via models of human notions of interestingness with environments programmed in code. *arXiv preprint arXiv:2405.15568*, 2024. URL https://arxiv.org/abs/2405.15568.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge, 2022. URL https://arxiv.org/abs/2206.08853.

Joshua Feldman, Joe Davison, and Alexander M. Rush. Commonsense knowledge mining from pretrained models, 2019. URL https://arxiv.org/abs/1909.00505.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pretrained large language models to construct and utilize world models for model-based task planning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.14909.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: a survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/890. URL https://doi.org/10.24963/ijcai.2024/890.

William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations, 2019. URL https://arxiv.org/abs/1907.13440.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1W0z96MFEoH.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values, 2023. URL https://arxiv.org/abs/2008.02275.

Edward Hughes, Michael D. Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktäschel. Position: Open-endedness is essential for artificial superhuman intelligence. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024. URL https://arxiv.org/abs/2406.04268.

Andrew J. I. Jones and Marek Sergot. A formal characterisation of institutionalised power. *Logic Journal of the IGPL*, 4(3):427–443, 1996.

Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Illuminating generalization in deep reinforcement learning through procedural level generation, 2018. URL https://arxiv.org/abs/1806.10729.

Ahmed Khalifa, Philip Bontrager, Sam Earle, and Julian Togelius. Pcgrl: procedural content generation via reinforcement learning. In *Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'20. AAAI Press, 2020. ISBN 978-1-57735-849-7.

Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with gamegan, 2020. URL https://arxiv.org/abs/2005.12126.

Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evol. Comput.*, 19(2):189–223, June 2011. ISSN 1063-6560. doi: 10.1162/EVCO_a_00025. URL https://doi.org/10.1162/EVCO_a_00025.

Wolfgang Lehrach, Daniel Hennes, Miguel Lazaro-Gredilla, Xinghua Lou, Carter Wendelken, Zun Li, Antoine Dedieu, Marc Lanctot, Atil Iscen, John Schultz, Marcus Chiam, Ian Gemp, Piotr Zielinski, Satinder Singh, and Kevin Murphy. Code world models for general game playing. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d8a12fde9e72444e1b356e8c37e53753-Abstract-Conference.html.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: communicative agents for "mind" exploration of large language model society. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Mahdi Farrokhi Maleki and Richard Zhao. Procedural content generation in games: a survey with insights on emerging llm integration. In *Proceedings of the Twentieth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE '24. AAAI Press, 2024. ISBN 1-57735-895-3. doi: 10.1609/aiide.v20i1.31877. URL https://doi.org/10.1609/aiide.v20i1.31877.

Javier Morales, Maite López-Sánchez, Juan A. Rodríguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. IRON: A machine for the automated synthesis of normative systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1389–1390. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

Javier Morales, Maite López-Sánchez, Juan A. Rodríguez-Aguilar, Wamberto Vasconcelos, and Michael Wooldridge. Online automated synthesis of compact normative systems. *ACM Transactions on Autonomous and Adaptive Systems*, 10(1), 2015. doi: 10.1145/2720024.

Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991. doi: 10.1007/BF03037089.

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393201. doi: 10.1145/3526113.3545616. URL https://doi.org/10.1145/3526113.3545616.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019. URL https://arxiv.org/abs/1909.01066.

Jeremy Pitt, Julia Schaumeier, and Alexander Artikis. Axiomatization of socio-economic principles for self-organizing institutions: Concepts, experiments and challenges. *ACM Transactions on Autonomous and Adaptive Systems*, 7(4):1–39, 2012.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs, 2018. URL https://arxiv.org/abs/1806.07011.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development, 2024. URL https://arxiv.org/abs/2307.07924.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model?, 2020. URL https://arxiv.org/abs/2002.08910.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. Atomic: an atlas of machine commonsense for if-then reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019a. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33013027. URL https://doi.org/10.1609/aaai.v33i01.33013027.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019b. URL https://arxiv.org/abs/1904.09728.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1–2):231–252, 1995. doi: 10.1016/0004-3702(94)00007-N.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

Munindar P. Singh. Formalizing communication protocols for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1519–1524, 2007. URL https://mlanthology.org/ijcai/2007/singh2007ijcai-formalizing/.

Kenneth O. Stanley and Joel Lehman. *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer Publishing Company, Incorporated, 2015. ISBN 3319155237.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents, 2024. URL https://arxiv.org/abs/2309.02427.

Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml), 2018. URL https://arxiv.org/abs/1702.00539.

Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.

Hao Tang, Darren Key, and Kevin Ellis. WorldCoder, a model-based LLM agent: Building world models by writing code and interacting with the environment. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2402.12275.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents, 2021. URL https://arxiv.org/abs/2107.12808.

SIMA Team, Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, Stephanie C. Y. Chan, Jeff Clune, Adrian Collister, Vikki Copeman, Alex Cullum, Ishita Dasgupta, Dario de Cesare, Julia Di Trapani, Yani Donchev, Emma Dunleavy, Martin Engelcke, Ryan Faulkner, Frankie Garcia, Charles Gbadamosi, Zhitao Gong, Lucy Gonzales, Kshitij Gupta, Karol Gregor, Arne Olav Hallingstad, Tim Harley, Sam Haves, Felix Hill, Ed Hirst, Drew A. Hudson, Jony Hudson, Steph Hughes-Fitt, Danilo J. Rezende, Mimi Jasarevic, Laura Kampis, Rosemary Ke, Thomas Keck, Junkyung Kim, Oscar Knagg, Kavya Kopparapu, Rory Lawton, Andrew Lampinen, Shane Legg, Alexander Lerchner, Marjorie Limont, Yulan Liu, Maria Loks-Thompson, Joseph Marino, Kathryn Martin Cussons, Loic Matthey, Siobhan Mcloughlin, Piermaria Mendolicchio, Hamza Merzic, Anna Mitenkova, Alexandre Moufarek, Valeria Oliveira, Yanko Oliveira, Hannah Openshaw, Renke Pan, Aneesh Pappu, Alex Platonov, Ollie Purkiss, David Reichert, John Reid, Pierre Harvey Richemond, Tyson Roberts, Giles Ruscoe, Jaume Sanchez Elias, Tasha Sandars, Daniel P. Sawyer, Tim Scholtes, Guy Simmons, Daniel Slater, Hubert Soyer, Heiko Strathmann, Peter Stys, Allison C. Tam, Denis Teplyashin, Tayfun Terzi, Davide Vercelli, Bojan Vujatovic, Marcus Wainwright, Jane X. Wang, Zhengdong Wang, Daan Wierstra, Duncan Williams, Nathaniel Wong, Sarah York, and Nick Young. Scaling instructable agents across many simulated worlds, 2024. URL https://arxiv.org/abs/2404.10179.

Graham Todd, Alexander Padula, Matthew Stephenson, Éric Piette, Dennis J. N. J. Soemers, and Julian Togelius. GAVEL: Generating games via evolution and language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2407.09388.

Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. Search-based procedural content generation. In *Proceedings of the 2010 International Conference on Applications of Evolutionary Computation - Volume Part I*, EvoApplicatons'10, pp. 141–150, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3642122388. doi: 10.1007/978-3-642-12239-2_15. URL https://doi.org/10.1007/978-3-642-12239-2_15.

Alexander Sasha Vezhnevets, John P. Agapiou, Avia Aharon, Ron Ziv, Jayd Matyas, Edgar A. Duéñez-Guzmán, William A. Cunningham, Simon Osindero, Danny Karmon, and Joel Z. Leibo. Generative agent-based modeling with actions grounded in physical, social, or digital space using Concordia. *arXiv preprint arXiv:2312.03664*, 2023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023a. URL https://arxiv.org/abs/2305.16291.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL http://dx.doi.org/10.1007/s11704-024-40231-1.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions, 2019. URL https://arxiv.org/abs/1901.01753.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and CraftJarvis Team. Describe, explain, plan and select: interactive planning with large language models enables open-world multi-task agents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023b. Curran Associates Inc.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Ethical and social risks of harm from language models, 2021. URL https://arxiv.org/abs/2112.04359.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey, 2023. URL https://arxiv.org/abs/2309.07864.

Qiang Yang, Kangheng Wu, and Yunfei Jiang. Learning action models from plan examples using weighted MAX-SAT. *Artificial Intelligence*, 171(2–3):107–143, 2007. doi: 10.1016/j.artint.2006.11.005.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Abhay Zala, Jaemin Cho, Han Lin, Jaehong Yoon, and Mohit Bansal. EnvGen: Generating and adapting environments via LLMs for training embodied agents. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=F9tqgOPXH5.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory, 2023. URL https://arxiv.org/abs/2305.17144.

Alexander Zook and Mark O. Riedl. Automatic game design via mechanic generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014. doi: 10.1609/aaai.v28i1.8788. URL https://ojs.aaai.org/index.php/AAAI/article/view/8788.

A FARM-TOWN WORLD SPECIFICATION

A.1 PURPOSE AND SCOPE

Farm-Town is a bounded open-world scenario for evaluating rule distillation under resource, social, economic, and institutional constraints. It supplies concrete world facts, including locations, roles, resources, items, initial inventories, initial institutional conditions, and environmental constraints; the rule distillation framework handles action interpretation, state transition validation, conservation maintenance, social observation logging, and reusable mechanism updates. The scenario provides a concrete, bounded, and auditable world in which agents generate action intents under real constraints, exposing mechanism gaps the current rule base cannot express.

Farm-Town separates scenario facts from general mechanisms. The scenario layer defines concrete world content, including farms, markets, forests, docks, resource nodes, agent attributes, and initial items; the mechanism layer maintains tick advancement, cognitive modules, the LLM intent interface, the impact engine, the rule layer, action templates, and the legal engine. Architecturally, the framework is a scenario-agnostic simulation kernel; Farm-Town supplies seed data, prompt context, constants, and session-level scenario resolution. The mechanism layer stores rule updates; the scenario layer supplies the initial state and contextual constraints.

Farm-Town is the experimental substrate for rule distillation. It provides resource, economic, social, and institutional pressure; general mechanisms carry candidate rule expression, validation, and reuse. Their separation makes rule growth an identifiable object of study.

The controlled studies (Table 2; Appendix D) run a dedicated implementation of a subset of this specification with ten agents, five locations, three resource nodes, two recipes, two initial laws, and an initial library $\mathcal{R}_0$ of 17 actions: moving, resting, eating, gathering, crafting, giving resources or items, listing and buying, borrowing and returning, hiring and completing jobs, teaching, filing complaints, enforcing laws, and recording refusals. Its clock advances one day at a time, raising hunger, restoring energy overnight, regenerating resource nodes under seeded weather, expiring stale listings, and recording overdue loans and jobs as audit events.

A.2 STATE REPRESENTATION

At each timestep t, the world state of Farm-Town is recorded as

$$s_t = \langle L_t, N_t, I_t, \mathrm{Rel}_t, E_t, G_t, M_t, P_t \rangle. \tag{7}$$

Here L_t denotes locations, spatial structure, and resource nodes; N_t denotes agents and their physiological, economic, cognitive, and action states; I_t denotes item definitions, item instances, tools, inventories, and recipes; Rel_t denotes social relationships, reputation, commitments, and social observations; E_t denotes events, action records, and state change trajectories; G_t denotes laws, governance, organizations, and institutional conditions; M_t denotes memory, beliefs, and knowledge edges; P_t denotes market listings, transactions, price history, property, and ledger state.

This state representation requires world mechanisms to ultimately act on trackable variables. Action outcomes must specify changes to, or required preservation of, location, resources, inventory, gold, relationships, legal status, market records, knowledge edges, or event records. The audit specification connects each candidate to its declared checks and to the state variables changed or preserved by execution.

World state and rule base jointly govern simulation evolution:

$$s_{t+1} = \mathcal{T}_{\mathcal{R}_t}(s_t, a_t), \tag{8}$$

where $\mathcal{R}_t$ is the current rule base, a_t is the action or interaction executed at timestep t, and $\mathcal{T}_{\mathcal{R}_t}$ denotes the state transition function constrained by the current rule base. Rule distillation asks whether $\mathcal{R}_t$ grows as the experiment progresses. The critical change in an open world occurs in the rule base, not in objects, events, or text added to s_t.

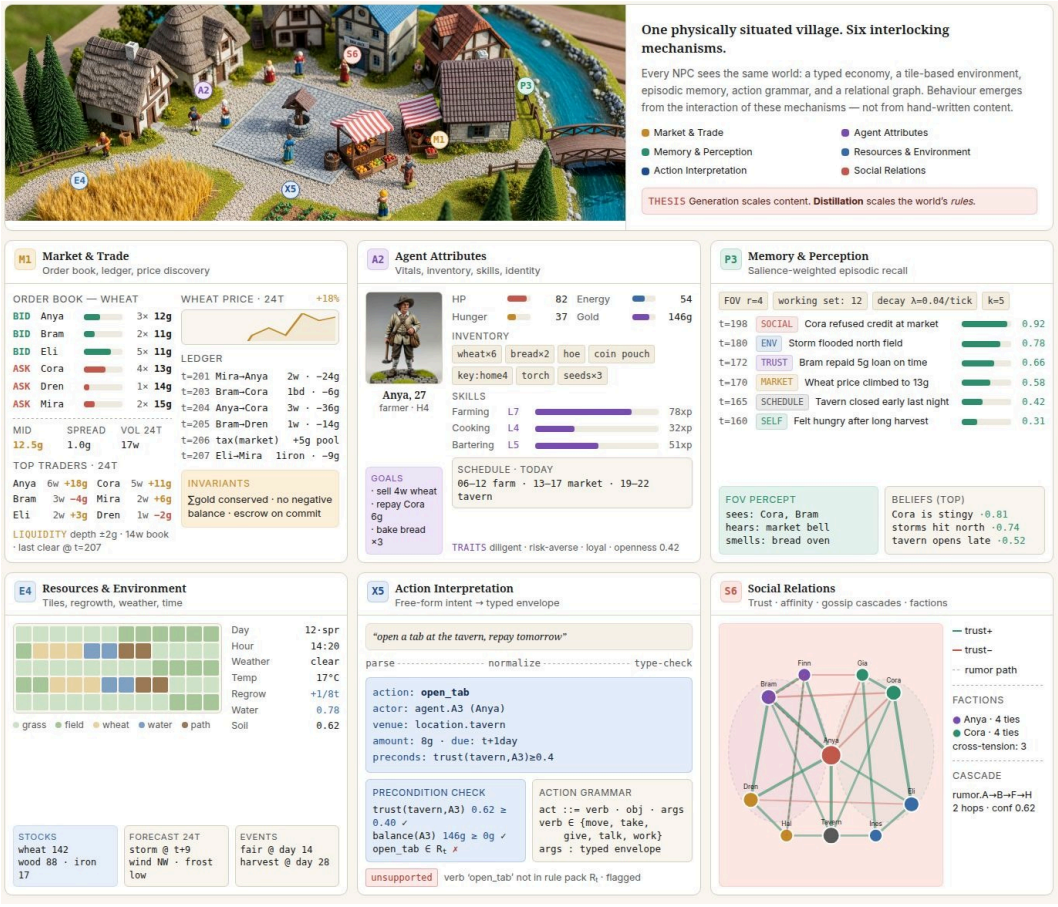

Figure 7: Extended conceptual illustration of the Farm-Town world state. The panels show, for one illustrative village, example observations, ledgers, intent records, and interfaces for the market, agent, memory, environment, action-interpretation, and social components of s_t; the action-interpretation panel shows an intent whose verb is not in $\mathcal{R}_t$ being recorded as unsupported rather than executed. Names and values are illustrative. Figure 1 gives the admission and execution procedure.

A.3 Time and Update Model

Farm-Town uses 96 discrete ticks per day to unify day-night cycles, action durations, resource regeneration, listing expiry, rent, long-term observation, and institutional aggregation.

For agent i executing action a at location l_t and arriving at $l_{t'}$ upon completion, time advancement is:

$$t' = t + \text{duration}(a). \tag{9}$$

At the specification level, a multi-tick action reserves the actor for the interval $[t, t')$, while background hunger, energy, resource, listing, and institutional updates continue on intervening ticks. If multiple actions complete at the same tick, an implementation must declare the deterministic ordering or conflict-resolution rule used for their state effects. This scheduling convention is part of the world specification; particular experiments must report whether they implement the full tick scheduler or a simplified fixture schedule.

Energy update is:

$$\text{energy}_{i,t'} = \text{energy}_{i,t} - c_{\text{action}}(a) - c_{\text{move}}(l_t, l_{t'}). \tag{10}$$

Hunger and energy evolve continuously as baseline physiological pressure:

$$\text{hunger}_{i,t+1} = \min(1,\ \text{hunger}_{i,t} + \delta_h), \tag{11}$$

$$\text{energy}_{i,t+1} = \max(0,\ \text{energy}_{i,t} - \delta_e). \tag{12}$$

Action costs, movement costs, rest recovery, feeding effects, and hazard states jointly constitute the survival constraints of the town. Hunger, energy, time, and distance directly shape each agent's actionable space, determining whether an agent seeks food, enters trade, requests assistance, shifts to foraging, engages in labor, or proposes action intents unsupported by the current rule base.

Daily updates process long-horizon cross-tick events, including resource node regeneration, market listing expiry, rent and property state updates, long-term social observation aggregation, governance state advancement, and persistent drive changes. Farm-Town is thus a continuously evolving dynamic world. Every candidate mechanism must be compatible with tick advancement, state updates, and long-term consequences to qualify as a stable rule.

A.4 Map, Locations, and Affordances

Farm-Town comprises a set of functionally differentiated locations. Location is a component of action feasibility: whether an action can execute depends on agent position, location type, resource nodes, available stations, market access, governance access, and surrounding visible social objects.

Core locations include the farm, market, tavern, forest, town hall, smithy, school, temple, pier, and beach. The farm supplies crops, seeds, and agricultural production conditions; the market supports listings, transactions, and price pressure; the tavern supports socializing, beverage crafting, and information diffusion; the forest supplies timber, herbs, wild food, and exploration opportunities; the town hall hosts governance, voting, laws, and property transactions; the smithy provides ore, tool crafting, and tool maintenance; the school supports learning, teaching, and knowledge diffusion; the temple hosts caregiving, healing, and moral reputation; the pier provides fishing and aquatic resources; the beach provides shells, marginal foraging, and low-value material sources.

Locations constrain actions through affordances. Foraging must occur at resource node locations; crafting must occur at locations matching the recipe's station kind; market transactions require valid listings and bilateral agent states; governance actions must connect to institutional locations or existing legal states; movement incurs distance-dependent energy costs. Intents must be interpreted through location, resource, tool, time, and institutional constraints before altering world state.

Resource nodes make locations auditable. Farm-Town includes resource nodes for forest wood, tree seeds, foxglove, mushrooms, wildlife, farm wheat, wheat seeds, pier fish, smithy iron, smithy copper, and beach shells. Each resource node is bound to a location, yield type, current yield, maximum yield, and regeneration rate. Fish originate from the pier fishery resource node, iron ore from the smithy vein, and wheat seeds represent inputs for the next agricultural cycle. Locations and resources together supply the constrained environment required for rule distillation.

A.5 Agents and Initial Heterogeneity

Agents in Farm-Town have heterogeneous initial states. The system includes roles such as farmer, merchant, blacksmith, physician, teacher, priest, town guard, apprentice, elder, child, and other townsfolk. Roles supply initial dispositions, visible opportunities, and social interpretation context; concrete behavior is jointly determined by state, constraints, memory, relationships, resources, and current world pressure. A farmer may pivot to trade or seek assistance under market pressure; a merchant may shift to procurement or negotiation under inventory shortage; an agent with low gold and high hunger may attempt to find food, request help, sell inventory, or propose unregistered exchange methods. Openness arises from the combination of states, constraints, and agent intents.

Agent i's state is represented as:

$$n_i = \langle l_i,\ \text{gold}_i,\ \text{hunger}_i,\ \text{energy}_i,\ \text{inv}_i,\ \text{skill}_i,\ \text{mem}_i,\ \text{rel}_i,\ \text{drive}_i,\ \text{belief}_i \rangle. \tag{13}$$

Here l_i is location, gold_i is gold holdings, hunger_i is hunger level, energy_i is energy, inv_i is inventory, skill_i is skill state, mem_i is memory, rel_i is social relationships, drive_i is current drive, and belief_i is belief or knowledge state. These variables and local observations jointly constrain action selection.

This heterogeneity triggers rule distillation in several ways. Resource scarcity surfaces rule priors related to foraging, exchange, lending, relief, and hoarding; role differences surface rule priors related to tools, skills, teaching, caregiving, and labor division; relationship differences surface rule priors related to trust, commitment, refusal, negotiation, and reputation; institutional differences surface rule priors related to laws, sanctions, governance, and collective decision-making. Agents propose action intents under concrete conditions of location, inventory, hunger, energy, relationships, and institutional state; rule distillation extracts candidate mechanisms from these intents.

The agent design of Farm-Town maintains the boundary between LLMs and world rules. LLMs generate action intents, justify choices, predict consequences, and produce dialogue; world state changes remain governed by the rule base, resource constraints, and validation mechanisms. Differences in inventory, hunger, roles, and relationships give agents distinct conditions for proposing rules.

A.6 Resources, Items, and Conservation

Farm-Town's resource system enforces finite resources and real inventory. Resources exist as resource nodes at concrete locations; each node has a resource type, location, current yield, maximum yield, regeneration rate, and harvestable output. Resource acquisition requires a resource node, harvesting conditions, action costs, and inventory changes; linguistic description alone is insufficient.

A resource node is represented as:

$$r = \langle loc(r),\, kind(r),\, yield_t(r),\, maxYield(r),\, regen(r),\, output(r) \rangle. \tag{14}$$

$loc(r)$ is the node's location, $kind(r)$ its resource type, $yield_t(r)$ its available yield, $maxYield(r)$ its capacity, $regen(r)$ its regeneration function, and $output(r)$ the harvested item type.

A harvest action is valid iff:

$$\begin{aligned} \mathrm{Harvest}(i, r, q) \text{ valid} \Leftrightarrow\ & q > 0 \wedge loc_i = loc(r) \wedge yield_t(r) \geq q \\ & \wedge \mathrm{Access}(i, r) = 1 \wedge energy_i \geq c_{\mathrm{harvest}}. \end{aligned} \tag{15}$$

Upon execution, resource node, inventory, and energy update simultaneously:

$$yield_{t'}(r) = yield_t(r) - q, \tag{16}$$

$$inv_{i,t'} = inv_{i,t} \cup \{\mathrm{item}(output(r), q)\}, \tag{17}$$

$$energy_{i,t'} = energy_{i,t} - c_{\mathrm{harvest}}. \tag{18}$$

This rule ensures resource changes have traceable origins. Items entering inventory must generate concrete item instances bound to owner, source node, and creation time. Node deduction, item generation, and energy consumption form the minimal causal chain of harvesting.

The item system distinguishes item definitions from item instances. A definition d describes category, usage, nutrition, value, tool properties, or material properties; an instance x describes a concrete item's owner, location, quantity, state, and provenance. Formally:

$$d = \langle kind,\, properties,\, baseValue,\, effects \rangle, \tag{19}$$

$$x = \langle def(x),\, owner(x),\, loc(x),\, qty(x),\, state(x),\, source(x) \rangle. \tag{20}$$

This distinction allows items of the same kind to have different owners, origins, and states. Wheat instances may originate from farm harvest, market purchase, gift transfer, or trade settlement; rule validation can trace their provenance and ownership chain.

Conservation is the foundational constraint of the resource and item system. Every generation, consumption, transaction, and crafting operation must leave corresponding inputs and outputs in world state. A candidate mechanism that produces items without specifying a resource node, recipe input, transaction source, or institutional transfer path fails conservation checking. The resource and item system thus supplies the baseline validation condition for rule distillation: action intents may propose to acquire, craft, sell, give, or consume, but world mechanisms must account for where resources originate, who owns items, and how states change.

A.7 Tools and Production Constraints

The tool system binds action capability to items, skills, and location conditions. Tools have type, durability, efficiency, and an applicable resource set. A tool can expand the resource range accessible to an agent or alter the cost of harvesting, crafting, or maintenance. Tools cannot take effect as narrative attributes; they must exist as concrete item instances in inventory or at a location.

A tool instance is represented as:

$$tool = \langle kind(tool),\ owner(tool),\ durability(tool),\ efficiency(tool),\ uses(tool)\rangle. \tag{21}$$

The resource access function is defined as:

$$\begin{aligned}\mathrm{Access}(i,r) = 1 \Leftrightarrow\ & requiredTool(r) = \emptyset \\ & \vee \big(\exists\, tool \in inv_i :\ kind(tool) \in requiredTool(r) \wedge durability(tool) > 0\big).\end{aligned} \tag{22}$$

For resources requiring no tool, $requiredTool(r)$ is empty; for ore, timber, fishing, or advanced crafting, tool conditions determine action feasibility and efficiency. Use reduces tool durability:

$$durability_{t'}(tool) = durability_t(tool) - c_{\mathrm{use}}(tool, r). \tag{23}$$

Efficiency degrades below a durability threshold; at zero, the tool fails. This constrains long-term acquisition and production: agents must maintain, buy, craft, or find alternative tools.

A recipe ρ is represented as:

$$\rho = \langle inputs(\rho),\ outputs(\rho),\ station(\rho),\ tools(\rho),\ skill(\rho),\ duration(\rho)\rangle. \tag{24}$$

A craft action is valid iff:

$$\begin{aligned}\mathrm{Craft}(i,\rho)\ \text{valid} \Leftrightarrow\ & inputs(\rho) \subseteq inv_i \wedge loc_i \in station(\rho) \\ & \wedge tools(\rho) \subseteq inv_i \wedge skill_i \geq skill(\rho) \\ & \wedge energy_i \geq c_{\mathrm{craft}}(\rho).\end{aligned} \tag{25}$$

Upon execution:

$$inv_{i,t'} = inv_{i,t} - inputs(\rho) + outputs(\rho), \tag{26}$$

$$energy_{i,t'} = energy_{i,t} - c_{\mathrm{craft}}(\rho). \tag{27}$$

Crafting intents require material consumption, tool requirements, location constraints, skill thresholds, and item generation. New recipes must share this structure: available inputs, defined outputs, a legitimate station, interpretable tool and skill conditions, and reuse in later world states.

Production-rule candidates connect materials, tools, skills, and locations to executable transformations. LLMs may propose intents such as "craft a fishing rod," "improve a tool," "prepare food," or "repair farm equipment." These intents expose candidate priors about material transformation, tool efficiency, production chains, and skill dependencies. The action label identifies the request; a mechanism must specify its operational relations. The platform requires these priors to enter recipe, tool, and state transition structures as verifiable mechanisms.

A.8 MARKET, LISTINGS, AND TRANSACTIONS

Farm-Town's market system expresses ownership, price, exchange, and resource flow. Every market transaction must consist of real items, real gold, and explicit ownership transfer. Selling an item first creates a listing; upon transaction, the buyer's gold decreases, the seller's gold increases, item ownership transfers, the listing closes, and a transaction record enters the ledger.

A listing is represented as:

$$\ell = \langle seller,\ item,\ price,\ loc,\ status,\ t_{\text{created}} \rangle. \tag{28}$$

A listing is valid iff:

$$\begin{aligned} \text{List}(seller, item, price) \text{ valid} \Leftrightarrow\ & owner(item) = seller \\ & \wedge\ status(item) = available \\ & \wedge\ price > 0. \end{aligned} \tag{29}$$

Upon listing creation:

$$status(item)' = listed, \tag{30}$$

$$listing(\ell) = active. \tag{31}$$

Transaction settlement is valid iff:

$$\begin{aligned} \text{Settle}(buyer, seller, item, p) \text{ valid} \Leftrightarrow\ & listing(item, seller, p) = active \\ & \wedge\ gold_{buyer} \geq p \\ & \wedge\ owner(item) = seller. \end{aligned} \tag{32}$$

Upon execution:

$$gold'_{buyer} = gold_{buyer} - p, \tag{33}$$

$$gold'_{seller} = gold_{seller} + p, \tag{34}$$

$$owner'(item) = buyer, \tag{35}$$

$$holder'(item) = buyer, \tag{36}$$

$$loc'(item) = l_{buyer}, \tag{37}$$

$$inv'_{seller} = inv_{seller} \setminus \{item\}, \tag{38}$$

$$inv'_{buyer} = inv_{buyer} \cup \{item\}, \tag{39}$$

$$listing'(item) = closed. \tag{40}$$

The transaction ledger records:

$$tx = \langle buyer,\ seller,\ item,\ p,\ t,\ loc \rangle. \tag{41}$$

This mechanism requires gold, items, ownership, holder, location, inventory, and ledger state to remain consistent. If a simulator represents holder through item ownership only, the inventory equations are projection constraints on the same item instance: the item cannot remain in the seller's inventory while being owned by the buyer. Transaction records must match the gold balances, item instances, listings, and ledger entries produced by each exchange. These conditions validate economic rules.

The market system supports surfacing multiple classes of rule priors. Hunger and low gold may trigger intents to buy, borrow, barter, purchase on credit, sell inventory, or seek wage labor; resource scarcity may trigger price increases, hoarding, substitute goods, negotiation, or relief; relationship differences may trigger discounts, gifts, refusals, or exploitation. The baseline market mechanism covers only listing and settlement; more complex credit, contracts, wages, leasing, auctions, and price formation can enter rule distillation as candidate mechanisms. Every candidate economic mechanism must maintain item, gold, ownership, and ledger consistency.

A.9 Social Interaction, Intent, and Recipient Agency

Farm-Town's social interaction system decomposes multi-agent behavior into intent, expression, interpretation, response, and state effect. Social behavior is not unilaterally determined by language; any action involving another agent's resources, commitments, consent, refusal, cooperation, coercion, or obligations requires recipient interpretation and response.

A social intent is represented as:

$$\eta = \langle speaker,\ recipient,\ actType,\ content,\ requested,\ offered,\ context,\ risk \rangle. \tag{42}$$

Here $speaker$ is the initiator, $recipient$ the receiver, $actType$ the interaction type (request, proposal, exchange, threat, gift, commitment, negotiation), $requested$ and $offered$ the resource or action terms, $context$ the world pressure, and $risk$ the potential risk or norm sensitivity.

Recipient response is:

$$d_j = \mathrm{Respond}(j, \eta, s_t), \quad d_j \in \{accept,\ reject,\ counter,\ ignore,\ report,\ retaliate\}. \tag{43}$$

Responses depend on the recipient's resources, relationships, beliefs, risk assessment, legal state, and goals. A request's resource transfers and obligations take effect only when the recipient accepts, the relevant mechanism conditions hold, and all resource and permission constraints are satisfied. Rejection records social observations, relationship changes, or subsequent conflict, without transferring resources or creating obligations.

The state effect of social interaction is:

$$s_{t'} = \mathrm{ApplyProtocol}(s_t, \eta, d_j), \tag{44}$$

where $\mathrm{ApplyProtocol}$ determines resource transfer, relationship change, memory write, commitment generation, reputation change, or event recording based on interaction type. This protocol layer ensures social behavior is bilateral and auditable. Requesting help, borrowing, exchanging, hiring, committing, and coercing all require explicit initiator, recipient, terms, response, and consequences.

Social-rule candidates specify how responses create obligations and change later interactions. LLMs frequently propose intents carrying social rule priors, such as requesting relief, negotiating price, committing to repayment, refusing trade, maintaining reputation, sanctioning breach, or seeking collective support. The platform retains these as structured interactions. World updates remain governed by the execution rules. Candidate social mechanisms must specify who holds response authority, what conditions trigger obligations, how refusals are handled, how breaches are recorded, and how future relationships are affected. Social rules thus enter world mechanisms through interaction structure.

A.10 Governance, Law, and Institutional Pressure

Farm-Town's governance system expresses laws, collective decisions, sanctions, institutional constraints, and collective responses. Governance mechanisms depend on events, evidence, trigger conditions, severity computation, and state effects. Laws cannot take world effect through natural language declaration; an institutional rule must specify applicable events, trigger predicates, enforcement authority, sanction or constraint type, and resulting state changes.

A law or institutional rule is represented as:

$$\lambda = \langle scope,\ trigger,\ authority,\ severity,\ sanction,\ duration \rangle. \tag{45}$$

Here $scope$ is the applicable domain, $trigger$ the activation condition, $authority$ the enforcement agent or recognition mechanism, $severity$ the severity function, $sanction$ the penalty or constraint, and $duration$ the persistence or expiry condition.

An event record is represented as:

$$e = \langle kind,\ actor,\ target,\ witnesses,\ loc,\ tags,\ factors,\ t \rangle. \tag{46}$$

When law λ is activated by event e such that $trigger_\lambda(e) = 1$, severity is computed as:

$$sev = severity_\lambda(e.factors), \tag{47}$$

and the state effect is:

$$s_{t'} = \text{ApplySanction}(s_t, \lambda, e, sev). \tag{48}$$

Law triggers must be checkable. Overly broad triggers cause institutional overgeneralization; overly narrow triggers prevent reuse across analogous contexts. Candidate laws and norms must specify event type, evidence source, severity rules, enforcement authority, and state effects.

Governance pressure arises from resource conflicts, transaction disputes, violence, high-risk social behavior, repeated harm, public safety, and collective resource management. Agents may propose intents to enact rules, report behavior, sanction breach, request protection, reform transaction institutions, or establish public relief mechanisms. The platform treats these intents as surfacing signals for institutional rule priors. Candidate institutional mechanisms must connect to events, evidence, trigger conditions, and consequences, not remain as textual norms of the form "this should be prohibited."

Governance also tests rule distillation's rejection capacity. A candidate law lacking trigger events, evidence conditions, enforcement authority, or state effects cannot enter the rule base. A sanction mechanism that cannot specify responsibility attribution, severity, and consequences cannot alter world state. Institutional growth requires auditable rules, not narrative claims of authority.

A.11 Learning, Knowledge, and Recipe Discovery

Farm-Town represents learning, teaching, and knowledge diffusion as trackable world state changes. Agents cannot acquire stable capabilities through dialogue or memory alone; new knowledge must enter skill state, knowledge edges, or reusable recipe structures. This design makes learning mechanisms affect the future actionable space.

Agent i's acquisition of knowledge or skill k is represented as:

$$\text{Learn}(i, k) \Rightarrow skill'_{i,k} = skill_{i,k} + \Delta_k, \tag{49}$$

$$edge(i, k, source, t) = 1. \tag{50}$$

Here $skill_{i,k}$ denotes experience or proficiency in skill k; $edge(i, k, source, t)$ is a knowledge edge formed at t, with $source$ recording its origin: teaching, observation, practice, reading, or social diffusion. Teaching gives the teacher teaching experience and the student target knowledge or skill experience; the world records the event and knowledge edge.

Knowledge diffusion also bears on rule distillation. Many candidate mechanisms require distinguishing between "a rule exists in the world" and "an agent knows how to invoke it." A recipe may be discovered in the world, yet only agents holding the corresponding knowledge edge can use it reliably; a tool may exist in the item system, yet agents must learn its usage or improve efficiency through practice. This design keeps rule growth and knowledge growth distinct: the rule base defines how the world can operate; knowledge state defines whether a given agent can invoke a mechanism.

Recipe discovery is a concrete instance of the learning mechanism. A candidate recipe ρ comprises inputs, outputs, location, tools, skill, and duration:

$$\rho = \langle inputs(\rho),\ outputs(\rho),\ station(\rho),\ tools(\rho),\ skill(\rho),\ duration(\rho) \rangle. \tag{51}$$

Agents may form candidate recipe leads through attempt, observation, or teaching. When leads reach a threshold and pass conservation, tool condition, location affordance, and output definition checks,

the candidate recipe enters the rule base. Infeasible combinations are recorded as failed leads or rejection cases without contaminating item definitions or the recipe library. Recipe discovery can supply a pathway for rule distillation: LLMs propose crafting intents, and the world checks their material transformations. Counting follows the declared initial interface. Filling slots in an already supported recipe template is an instance-level addition; registering an additional production relation changes the operational library. A new item name alone does not establish mechanism growth.

A.12 Initial Rule Library

Farm-Town's initial rule base $\mathcal{R}_0$ provides baseline mechanisms for survival, movement, resources, items, market, social interaction, governance, and learning. It defines the experimental starting boundary, enabling subsequent rule growth to be measured precisely. Formally:

$$\begin{aligned}\mathcal{R}_0 = \mathcal{R}_{\text{time}} \cup \mathcal{R}_{\text{metabolism}} \cup \mathcal{R}_{\text{movement}} \cup \mathcal{R}_{\text{resources}} \\ \cup \mathcal{R}_{\text{items}} \cup \mathcal{R}_{\text{market}} \cup \mathcal{R}_{\text{social}} \cup \mathcal{R}_{\text{governance}} \cup \mathcal{R}_{\text{learning}}.\end{aligned} \tag{52}$$

$\mathcal{R}_{\text{time}}$ covers ticks, action durations, daily updates, and long-term aggregation. $\mathcal{R}_{\text{metabolism}}$ covers hunger growth, energy consumption, rest recovery, feeding effects, and hazard states. $\mathcal{R}_{\text{movement}}$ covers movement, distance costs, and position updates. $\mathcal{R}_{\text{resources}}$ covers resource nodes, harvesting, regeneration, seeding, and node deduction. $\mathcal{R}_{\text{items}}$ covers item definitions, item instances, inventories, tool durability, and crafting. $\mathcal{R}_{\text{market}}$ covers listings, settlement, gold and item ownership transfers, and the transaction ledger. $\mathcal{R}_{\text{social}}$ covers social intents, recipient responses, relationship changes, memory writes, and social observations. $\mathcal{R}_{\text{governance}}$ covers law triggering, event recording, severity computation, sanction effects, and governance aggregation. $\mathcal{R}_{\text{learning}}$ covers skill experience, teaching, knowledge edges, recipe discovery, and knowledge diffusion.

The initial rule base provides a finite, auditable starting point for mechanism acquisition. Rule distillation in the experiment targets:

$$\Delta\mathcal{R} = \mathcal{R}_T \setminus \mathcal{R}_0, \tag{53}$$

the set of new mechanisms surfaced, validated, and consolidated through agent interaction during the run. Behavior interpretable under $\mathcal{R}_0$ is ordinary interaction within existing rules; an intent that cannot be executed under $\mathcal{R}_0$ and generates a candidate mechanism h enters rule distillation.

A.13 Evolvable Mechanism Classes

The evolvable mechanism set $\mathcal{H}$ in Farm-Town defines the types of mechanisms rule distillation can grow. These classes specify candidate structures; interaction and validation determine which candidates are proposed and admitted. A candidate mechanism $h \in \mathcal{H}$ may belong to the following classes.

New action types include novel harvesting, cooperative, caregiving, production, teaching, negotiation, or institutional actions. Action types must connect to applicability conditions, participating entities, state transitions, constraints, and downstream effects.

New resource relations cover location-specific resources, seasonal regeneration, resource depletion, resource substitution, common-pool governance, or overuse consequences. These must connect to resource nodes, yield changes, harvesting conditions, and long-term availability.

New items and recipes cover tools, food, medicine, building materials, trade goods, or composite items. Items require explicit definitions, usage, properties, provenance, and production pathways; recipes require inputs, outputs, tools, location, and skill conditions.

New interaction protocols include lending, hiring, negotiation, credit, surety, partnership, compensation, gifting, or assistance-seeking. Interaction protocols must retain participants, terms, acceptance or rejection, resource transfer, obligation state, and breach consequences.

New social observation and relationship mechanisms cover reputation change, trust accumulation, betrayal, bystander response, caregiving relationships, group pressure, or coalition formation. These must affect future relationships, memory, response spaces, or action dispositions.

The sixth class covers institutional mechanisms: law triggers, voting rules, common resource management, sanction mechanisms, relief institutions, property rules, or governance thresholds. Institutional mechanisms connect events, evidence, enforcement authority, severity, and state effects.

These classes supply the mechanism expression space for rule distillation. They constrain rule growth to checkable structures. Behavioral descriptions, social narratives, and event completions must specify the relations required for execution.

A.14 Validation Criteria

A candidate h enters the rule base only if it satisfies the validation function:

$$\mathrm{Valid}(h; s_t, \mathcal{R}_t) = C_{\mathrm{type}} \wedge C_{\mathrm{state}} \wedge C_{\mathrm{conserve}} \wedge C_{\mathrm{constraint}} \wedge C_{\mathrm{replay}} \wedge C_{\mathrm{conflict}} \wedge C_{\mathrm{reuse}}. \quad (54)$$

C_{type} checks type consistency across entities, resources, roles, locations, and state variables. A transaction requires buyer, seller, item, and price; a harvest candidate must include resource node, yield, and actor; a legal candidate must include event, trigger condition, and state effect.

C_{state} checks whether the candidate mechanism can be applied to the current world state. A mechanism cannot depend on nonexistent items, undefined locations, unowned resources, inactive laws, or absent participants. A candidate is executable only if these preconditions hold.

C_{conserve} checks resource, item, gold, ownership, location, and ledger consistency. Resources cannot arise from nothing; items cannot have multiple owners; gold transfers are bilateral; listings must match final inventories; harvesting and crafting require traceable sources and consumption.

$C_{\mathrm{constraint}}$ checks whether capability, tool, permission, consent, legal, and institutional conditions are satisfied. A candidate mechanism cannot bypass tool requirements, skill thresholds, recipient refusal, ownership restrictions, or legal constraints.

C_{replay} checks repeat execution from saved pre-admission states with fixed random inputs where needed. Declared state-level outcomes must agree. This check records repeatability on the admission tests; future stability is evaluated through subsequent interactions.

C_{conflict} checks whether the candidate mechanism conflicts with the existing rule base $\mathcal{R}_t$ on a declared finite test domain. The check covers invariant suites and regression cases for conservation, governance, ownership, protected refusal, and execution-budget behavior. Unresolved interactions retain pending status for further testing.

C_{reuse} checks whether the unchanged candidate definition passes declared tests under alternative participant or resource bindings before admission. These tests assess eligibility for reuse and prevent a candidate from passing only in the triggering instance. Actual post-admission reuse is counted only when later interactions invoke the registered mechanism.

These checks define a finite admission protocol. Candidate formation yields the five-tuple, an executable transition handler, and explicit tests. A test record contains an initial state, role binding, requested action, expected state predicates, and execution budget. Expected predicates follow from the declared contract and simulator invariants; repeating the generator's textual judgment is insufficient. The protocol first checks types and references, then executes the handler in an isolated state copy. Resource and permission checks inspect the resulting state difference. Refusal tests must preserve the specified protected state. Replay repeats saved tests with fixed random inputs; reuse eligibility changes role bindings while keeping the handler fixed. Conflict checks cover declared invariants and regression tests for the existing library. Timeout, unresolved references, and missing expected outcomes remain pending. A failed check blocks admission.

Admission commits a versioned rule after the declared tests pass. Future calls provide separate observations of reuse and failure. A rollback restores the executable rule version. Admission establishes compatibility over the declared finite test domain. The protocol defines the test records and outcomes that each implementation must report.

A.15 Evidence and Logs

The proposed evidence protocol requires each Farm-Town experiment to retain structured records. These records answer three questions: how mechanism gaps emerge, how candidate mechanisms form, and how validation results alter the rule base. Each end-to-end case contains at minimum:

$$\mathcal{E} = \langle s_{\text{pre}},\, z,\, u,\, h,\, V(h),\, \Delta s,\, r_{\text{replay}},\, a_{\text{audit}},\, reuse \rangle. \tag{55}$$

Here s_{pre} is the world state prior to candidate mechanism formation; z is the action intent surfaced by the LLM; u is the unsupported intent proposal; h is the candidate mechanism; $V(h)$ is the validation result; Δs is the state difference after mechanism execution; r_{replay} is the replay result; a_{audit} is the audit conclusion; and $reuse$ is the subsequent reuse record.

For positive cases, evidence must exhibit the complete chain:

$$s_{\text{pre}} \to z \to u \to h \to \mathcal{R}_{t+1} \to \Delta s \to reuse. \tag{56}$$

This chain requires that the agent naturally proposes an intent under concrete constraints, the current rule base cannot execute it, the system forms a candidate mechanism from it, the candidate passes validation and enters the rule base, and the mechanism subsequently produces trackable state changes.

For rejection cases, evidence must identify the failing validation check:

$$\begin{aligned} \text{Invalid}(h) \iff \exists\, C_k \in \mathcal{C} : \neg C_k, \\ \mathcal{C} = \{C_{\text{type}}, C_{\text{state}}, C_{\text{conserve}}, C_{\text{constraint}}, C_{\text{replay}}, C_{\text{conflict}}, C_{\text{reuse}}\}. \end{aligned} \tag{57}$$

For example, missing resource provenance corresponds to C_{conserve} failure; missing recipient consent corresponds to $C_{\text{constraint}}$ failure; holding only in a single scenario corresponds to C_{reuse} failure; conflicting with existing laws or ownership corresponds to C_{conflict} failure.

Evidence records further distinguish generated content, candidate mechanisms, and world facts. LLM outputs are stored as intent evidence; candidate mechanisms are stored as structured rule hypotheses; world state updates derive only from validated mechanism execution. This distinction ensures the auditability of rule distillation: researchers can trace each new rule to its originating context, the checks it passed, the state changes it produced, and the subsequent scenarios in which it was reused.

A.16 Trial Fixtures and Reachability

Farm-Town fixtures specify reproducible initial states; agent interaction and the rule base jointly determine outcomes. Fixtures set state without predetermining behavior. A fixture may specify agent hunger, gold, inventory, location, relationships, resource levels, market listings, or legal state; it cannot predetermine that an agent will propose a specific intent, that a recipient will accept a request, that a law will pass, or that a candidate mechanism will enter the rule base.

Formally, a trial fixture is represented as:

$$F = \langle s_0,\, \mathcal{R}_0,\, A_{\text{agents}},\, C_{\text{constraints}},\, Q_{\text{question}} \rangle, \tag{58}$$

where s_0 is the initial world state, $\mathcal{R}_0$ the initial rule base, A_{agents} the participating agent set, $C_{\text{constraints}}$ the resource, relationship, institutional, or location constraints, and Q_{question} the question the experiment aims to observe. A fixture defines reachability conditions: whether a class of mechanism gaps can be surfaced by agent interaction from that state.

For example, a resource pressure fixture may set an agent to high hunger and low gold, with exchangeable food and a potential trading partner co-located. The fixture does not dictate whether the agent proposes to buy, barter, beg, steal, or borrow; it supplies sufficient state conditions for these intents to be reachable under an open policy. A governance fixture may set repeated public harm events, bystanders, and an initial legal gap; it makes institutional response, reporting, sanctioning, or public rule proposals observable without prescribing a particular law.

Reachability is the core design criterion for fixtures. A rule distillation case is valid evidence only if:

$$\mathrm{Reachable}(z;\, s_0, \mathcal{R}_0) = 1, \tag{59}$$

meaning intent z can arise naturally from the agent given the state, memory, relationships, resources, and institutional constraints. Evidence of interaction-triggered acquisition requires agents to select intents under these conditions. The fixture supplies world pressure and leaves acceptance, transfer, sanction, and legislative outcomes to the participants and execution rules. The evidence chain records the path from world pressure to the surfaced mechanism gap.

A.17 Configuration for Mechanism Evaluation

Mechanism evaluation specifies the initial world, rule library, agent population, generation settings, and admission procedure. The following configuration schema organizes these inputs for re-execution, replay, and comparison across rule versions:

$$\Theta = \langle scenarioId,\, sessionId,\, seed,\, model,\, temperature,\, ticks,\, agents,\\ \mathcal{R}_0,\, s_0,\, mode,\, promptConfig,\, validationConfig\rangle. \tag{60}$$

Here $scenarioId$ identifies the scenario, $sessionId$ the session, $seed$ controls randomness, $model$ and $temperature$ record LLM configuration, $ticks$ records run length, $agents$ the participating agent set, $\mathcal{R}_0$ and s_0 the initial rule base and world state, $mode$ the run mode, $promptConfig$ the intent interface settings, and $validationConfig$ the rule validation settings.

Experiment output includes at minimum:

$$O = \langle world_{\mathrm{pre}},\, world_{\mathrm{post}},\, intentLog,\, proposalLog,\, candidateLog,\\ validationLog,\, stateDiff,\, replayLog,\, auditLog,\, reuseLog\rangle. \tag{61}$$

Here $world_{\mathrm{pre}}$ and $world_{\mathrm{post}}$ store pre- and post-run states; $intentLog$ records LLM-generated action intents; $proposalLog$ records unsupported intents; $candidateLog$ records candidate mechanisms; $validationLog$ records validation results; $stateDiff$ records state differences produced when candidate mechanisms enter the rule base; $replayLog$ records replay results; $auditLog$ records consistency checks; and $reuseLog$ records reuse of new rules in subsequent contexts.

Reproducibility requires state, mechanism, and evidence consistency. State consistency means that, for runs with the same s_0, $\mathcal{R}_0$, seed, and configuration, agreement and differences in intents, candidate definitions, and validation decisions are recorded. Mechanism consistency means that validated mechanisms reproduce the specified state transitions in replay. Textual similarity alone supplies no such check. Evidence consistency means that every added rule is traceable to its originating intent, proposal, candidate mechanism, validation result, and subsequent reuse record.

Experiment records retain natural language and structured state evidence for reproducibility. Natural language evidence includes intents, justifications, dialogue, and explanations; structured evidence includes resource changes, inventory changes, gold transfers, relationship changes, legal state, event records, knowledge edges, transaction ledgers, and rule base updates. Replay checks compare the specified state changes under the same rule version, starting state, action, and random input.

A.18 Relationship to the Main Text

Farm-Town evaluates mechanism acquisition under explicit world constraints. The central argument of the main text is that open-world progress should shift from content expansion to mechanism growth; LLMs surface candidate rule hypotheses; and rule distillation converts these priors into executable, verifiable, and reusable world mechanisms. Farm-Town connects this argument to a bounded, auditable experimental environment.

Farm-Town supplies concrete world facts, constraining agent intents by location, resources, inventory, energy, relationships, market conditions, and institutional state; trackable state variables, enabling

candidate mechanisms to be validated through state transitions, conservation, consistency, and reusability; and mechanism gaps of several kinds, enabling LLM-surfaced rule priors to be tested across resource, economic, social, learning, and governance scenarios.

The case study and controlled experiments examine the same operational questions. In the Dale–Petra credit case, repayment determines whether collateral can be released; the controlled credit tests vary repayment, default, ownership, and later attempts to dispose of the collateral. Pooled-purchase tests vary contributions, settlement, and refunds. These contract-level correspondences connect the trajectory's mechanism descriptions to the state-transition tests in Appendix D. The trajectory describes their development among interacting agents; the controlled implementation measures acquisition, enforcement, and reuse under specified comparisons.

Farm-Town's specification supports reproducibility and interpretation. It makes explicit what the initial world is, what the initial rule base contains, which mechanisms can grow, how candidate rules are validated, and how experiment logs are recorded. The main text uses this evidence to support a more general paradigm claim: open-world systems can, through rule distillation, consolidate rule hypotheses surfaced by LLMs into their own operational mechanisms.

A.19 Mechanism Boundaries and Case Evidence

Mechanism acquisition is defined relative to a declared representation. The initial state schema and execution interface provide the reference for classifying later changes. This reference remains fixed for comparison; the world may subsequently acquire additional state fields or transition relations. Classification follows the added operational relation and its state effects, with names and rule-file entries serving as identifiers.

The evaluation protocol specifies a procedure for deciding membership in $\mathrm{Exec}(\mathcal{R}_t)$. A schema-based procedure compares the normalized intent with the declared action templates, then checks the matched action's preconditions separately. An unresolved parse retains undecided status. Evaluations record the normalized intent, matched template or failure reason, and rule-library version. A model-assisted matcher requires a separate assessment of gap-classification accuracy. Evaluations must report proposal volume and detection accuracy separately.

An instance fills the arguments of a supported contract, recipe, or law. A skill selects or composes available actions while preserving their world-defined conditions and effects. A mechanism addition extends a relation used by the world to decide which transitions are admissible or which consequences persist. A universal interpreter can encode the changing library as data. This encoding preserves the distinction at the declared application interface; no increase in the interpreter's computational expressiveness is required. Endogenous acquisition identifies interaction as the source of a proposed addition. Implementation and admission autonomy are separate attributes of the acquisition procedure.

Collateral credit provides a concrete test. A sequence of transfers can move bread and a sickle between agents. Enforceable credit also records a future obligation and conditions redemption or forfeiture on repayment or default. If the initial contract mechanism already expresses those relations, the agreement is an instance. If a new persistent obligation or enforcement relation must be added, the change belongs at the mechanism layer. Section 5 supplies the reported terms of the Dale–Petra case. The initial market specification supplies listings, settlement, ownership transfers, and a ledger. The credit specification additionally connects repayment status to collateral release or forfeiture. This connection defines the relation to test against the initial library. Implementation records must identify the handler that enforces it.

The same test applies to governance and teaching. The initial library already includes law triggering, sanctions, and knowledge transfer. A newly supplied trigger value or class-session record can instantiate those mechanisms. A new relation among evidence, authority, obligations, and future admissible actions can extend them. Classification follows the changed relation. The 13 families organize the trajectory by mechanism class; relation-level comparison determines whether a family adds transition relations and identifies those additions. Market exchange, quote inquiry, and counter-offers must be compared with the initial settlement and recipient-response mechanisms. Labor settlement, pooled purchase, and work relief must be compared with the available transfer and participation rules. Dependency observation and public complaint must be compared with social observations and memory writes. Law triggering and class-session skill transfer must be compared

with the existing governance and learning rules. For each family, the relevant question is which applicability, obligation, or consequence relation changes. Admission and mechanism novelty are separate properties.

A state-distinction test. Consider a deterministic legacy handler that reads only a declared projection $p(s)$ of world state. If $p(s_1) = p(s_2)$, the handler receives the same input for the same action and returns the same output. Suppose a proposed contract instead requires different outcomes, $q(s_1, a) \neq q(s_2, a)$. The legacy handler cannot implement both requirements without changing the information it reads or the relation it computes. The argument follows directly from equality of the handler's inputs. It applies to that declared interface.

For collateral credit, two states can agree on current inventories while differing in repayment status. A redemption contract may release collateral in one state and retain it in the other. Inventory transfer alone cannot determine this distinction. An existing handler that already reads debt status and enforces the required relation can implement it as an instance. The test supplies a concrete criterion for classifying an extension. Per-admission classification compares the initial and installed handlers. The Farm-Town specification states the contract-level relations; Appendix D.14 reports the static differences for six V3 admissions in the repeated evaluation and the D3 state at which an added guard changes the decision.

Table 3: Farm-Town case evidence and its evaluation units. Each row connects a concrete mechanism to the corresponding analysis.

Case content	Presented in the paper	Evaluation unit
Scope and counts	One 90-day, 10-agent trajectory; 29 edges, 18 events, 13 families; 11 admitted and 2 pending.	Edges, events, and families are separate units. Yield uses the unsupported-proposal count; rejection rates use the declared candidate count.
Collateral credit	Sickle ownership, consent, lock, food transfer, debt terms, default and redemption or forfeiture.	Mechanism classification depends on which obligation and enforcement relations the initial library already expressed.
Institutional rule	Broad law text is rejected; the admitted trigger specifies evidence, authority, severity and consequences.	Validation records identify the failed condition; reference judgments evaluate rejection correctness.
Later reuse	Timeline and atlas describe availability and subsequent use across agents and contexts.	The case tracks subsequent use across participants and contexts; controlled comparisons use matched tasks and explicit denominators.
Comparative conditions	Appendix B.7 defines four conditions.	The controlled implementation measures acquisition and later use under these conditions (Appendix D).

The case identifies economic and institutional relations that require persistent world state and shared enforcement. In the controlled studies, LLMs generate candidate rules, executable development checks select installations, and fresh planners choose later actions. V3 compares feedback with independent sampling under equal installation authority and a three-call budget. V4 holds both feedback access and synthesis budget equal across the two search procedures (Appendices D.11 and D.15). These experiments measure acquisition and later use of the contract relations described in the case analysis.

A.20 Mechanism-atlas label index

Figure 5 in Section 5 groups ten agents around thirteen mechanism families. Table 4 supplies the labels at text size. The index transcribes the agent roles and mechanism names; the figure retains the dates and participation marks.

Table 4: Agent roles and mechanism names in the historical atlas.

Role group	Agents
Market	Petra, Marcus
Production	Bram, Dale
Social / welfare	Runa, Jora, Pip
Institution	Yuna, Lewis, Rurik
Mechanism labels	
Recipient-gated barter	
Quote-only inquiry	
Offer / counter / reject	
Tool lending + return obligation	
Collateral food credit (pawn)	
Labor-for-food task contract	
Debt dependency	
Theft + witness evidence chain	
Public market complaint boundary	
Pooled purchase + ration	
Evidence-backed law trigger	
Public work-relief voucher	
Class session + price literacy	

B Rule-Growth Evaluation Metrics

This appendix expands the evaluation protocol introduced in Section 7. We evaluate a rule-growth run using an initial rule base $\mathcal{R}_0$, a final rule base $\mathcal{R}_T$, unsupported intent proposals U, candidate mechanisms H, accepted mechanisms H^+, rejected mechanisms H^-, replay records, audit records, and reuse records. Measuring these quantities requires open-policy intent logs, unsupported proposal records, candidate mechanism reviews, implementation records, replay traces, audit logs, and rollback records. Unsupported proposals are recorded separately from world-state updates, keeping mechanism gaps auditable even after rejection or rollback.

B.1 Mechanism Growth

Mechanism growth measures the net expansion of the rule base:

$$G = |\mathcal{R}_T| - |\mathcal{R}_0|. \tag{62}$$

The set of newly consolidated mechanisms is $\Delta\mathcal{R} = \mathcal{R}_T \setminus \mathcal{R}_0$, as defined in Equation (5). When mechanisms are typed, we also report class-specific growth:

$$G_c = |\{h \in \mathcal{R}_T \setminus \mathcal{R}_0 : \mathrm{class}(h) = c\}|. \tag{63}$$

The class c may denote an action type, resource relation, market protocol, social protocol, governance mechanism, learning mechanism, item definition, or recipe. The metric measures the operational mechanisms entering the world. Event and description counts characterize content expansion.

When withdrawal is allowed, let $A = |\mathcal{R}_T \setminus \mathcal{R}_0|$ and $D = |\mathcal{R}_0 \setminus \mathcal{R}_T|$. Then $G = A - D$; $G = A$ requires $D = 0$. Additions and withdrawals should be reported separately. Splitting one transition into several rules can increase A without changing behavior. Comparisons therefore require a common rule language, a declared granularity, and a consistent treatment of equivalent mechanisms. A family count groups related mechanisms and must be reported separately from counts of candidate versions or rule definitions.

B.2 Proposal-to-Rule Yield

Proposal-to-rule yield measures how often unsupported intents produce validated mechanisms:

$$Y = \frac{|H^+|}{|U|}. \tag{64}$$

This expands Equation (6). Joint reporting of yield, replay stability, rejection quality, and auditability evaluates both the rate and the quality of admission. An admission procedure is assessed on the consistency of its accepted mechanisms as well as their number.

The ratio Y counts admitted candidate mechanisms per unsupported proposal. Several candidates may arise from one proposal, and one candidate may consolidate several proposals; consequently Y need not lie in $[0, 1]$. Reporting requires an explicit proposal–candidate relation and a convention for candidate revisions. An instance resolution rate counts proposals associated with at least one admitted candidate, divided by $|U|$. An abstraction ratio divides those resolved proposal instances by the number of distinct admitted relations they map to; it is undefined when no relation is admitted. The proposal–candidate map makes repeated requests and candidate revisions visible. Both ratios are undefined when $|U| = 0$. The 29 graph edges and 13 families in Section 5 are trajectory-level units; proposal and candidate logs supply the denominators for these ratios.

B.3 Replay Stability

Replay stability measures consistent execution after restoring the relevant pre-state:

$$S = \frac{|\{h \in H^+ : \mathrm{ReplayPass}(h) = 1\}|}{|H^+|}. \tag{65}$$

A replay pass requires state-level consistency. Textual similarity does not count. Replay must recover the intended causal chain: resource nodes decrease when harvesting succeeds, inventories and ownership update when exchange succeeds, recipient decisions gate bilateral actions, market listings correspond to real item instances, and ledger entries match final state.

An independent stability assessment declares test states and comparison criteria separately from admission. If $\mathrm{ReplayPass}$ reuses the admission check, S is reported as repeatability conditional on that check. Independent replay measures consistency across the separately specified states. External fidelity uses observations of the corresponding external process. For $|H^+| = 0$, both S and mean reuse are undefined.

B.4 Reuse

Reuse measures whether accepted rules affect later interactions:

$$U_{\mathrm{reuse}} = \frac{1}{|H^+|} \sum_{h \in H^+} \mathrm{ReuseCount}(h). \tag{66}$$

Reuse is measured through later invocations of the registered mechanism by agents, trades, social protocols, or governance events. Admission records local executability; subsequent execution records actual use. The count follows calls through the rule base. Comparisons fix the post-admission observation horizon and report exposure time and eligible interaction opportunities. Total calls measure usage frequency, while the distinct contracts invoking each rule measure interaction coverage. These complementary units distinguish repeated use from coverage growth.

B.5 Rejection Quality

Rejection quality measures whether the system can identify why a candidate mechanism fails. Each rejected mechanism $h \in H^-$ should be associated with at least one failed validation condition:

$$\begin{aligned} &\mathrm{Invalid}(h) \iff \exists\, C_k \in \mathcal{C} : \neg C_k, \\ &\qquad \mathcal{C} = \{C_{\mathrm{type}}, C_{\mathrm{state}}, C_{\mathrm{conserve}}, C_{\mathrm{constraint}}, C_{\mathrm{replay}}, C_{\mathrm{conflict}}, C_{\mathrm{reuse}}\}. \end{aligned} \tag{67}$$

Missing item provenance indicates conservation failure; missing recipient consent indicates constraint failure; a mechanism that only fits one scenario indicates reuse failure; conflict with an existing ownership, market, or legal rule indicates rule-base conflict. A justified rejection enforces the admission boundary between rule priors and world facts.

Rejection quality pairs the validator's failure label with a justified reference judgment or an explicit disagreement analysis. The label records the decision reason; the reference assessment evaluates whether the decision correctly excludes an invalid candidate. This assessment also identifies useful candidates rejected by overly restrictive checks. The conjunction defining Valid fails when at least one constituent check fails; several checks may fail together.

B.6 AUDITABILITY

Auditability requires each accepted rule to have a complete evidence chain:

$$\mathrm{Audit}(h) = 1 \iff \exists \langle s_{\mathrm{pre}}, z, u, h, V(h), \Delta s, r_{\mathrm{replay}}, a_{\mathrm{audit}}, reuse \rangle. \tag{68}$$

The chain records the pre-state, LLM intent, unsupported proposal, candidate mechanism, validation decision, state difference, replay result, audit result, and later reuse. Equation 68 measures record completeness. Checking whether those records substantiate admission requires consistency checks across rule versions, test inputs, state differences, and decisions. A reuse record may report zero observed invocations over its declared horizon. Record completeness and decision correctness are evaluated separately. This separates generated content, candidate mechanisms, and world facts: LLM outputs are intent evidence; candidate mechanisms are structured hypotheses; world-state updates derive only from validated mechanism execution.

B.7 COMPARATIVE CONDITIONS

Four comparison conditions separate content generation, intent recording, and permission to update the shared rule base. Section 5 retains the Farm-Town trajectory and adds controlled measurements, with settings and full results in Appendix D.

Fixed-rule simulation. The world keeps $\mathcal{R}_t = \mathcal{R}_0$. This tests stability within the initial rules.

Content generation. The system may generate descriptions, events, tasks, or dialogue, while $\mathcal{R}_t$ remains fixed. This condition measures surface expansion.

Open intent only. LLMs may propose unanticipated intents, and the system records unsupported proposals, but rule-base updates are disabled. This condition measures visibility of mechanism gaps.

Rule distillation. Unsupported proposals may become candidate mechanisms, pass validation, enter $\mathcal{R}_t$, replay, and later be reused. This condition measures mechanism growth.

The comparative output is:

$$\left(|\Delta\mathcal{R}|,\ Y,\ S,\ U_{\mathrm{reuse}},\ |H^-|,\ \mathrm{AuditRate}\right). \tag{69}$$

The comparison records which conditions install executable rules and support later use. Content counts record generated descriptions, events, tasks, and dialogue.

$\mathrm{AuditRate}$ is defined over admitted candidates as $|\{h \in H^+ : \mathrm{Audit}(h) = 1\}|/|H^+|$; if $H^+ = \emptyset$, it is reported as undefined.

The four conditions isolate access to the rule layer: the first three hold the rule base fixed, giving zero rule growth by design. Algorithm comparisons give competing acquisition procedures the same representation, installation authority, and admission criteria. Direct rule generation and existing-action composition address different alternatives. Their evaluation measures supported interactions, later execution outcomes, and computational or human cost. The supplementary study implements direct generation and skill composition controls. Direct and interaction-conditioned synthesis achieve the same contract coverage, while the skill control isolates the difference between cooperative execution and shared enforcement.

B.8 COMPARISON BY INTERACTION CONTRACTS

A comparison can standardize its unit at the level of required behavior. Fix a common initial library and a finite suite $\mathcal{Q}$ of interaction contracts before candidate generation. A contract contains a request, typed role bindings, admissible initial states, required effects, and refusal cases. Each system receives the same development cases. Held-out bindings and states evaluate the final library with a shared checker. A contract passes only if all its required success and refusal cases pass within the same execution budget. Identifiers and rule-file counts do not enter this decision.

Let $\mathcal{P}_0 \subseteq \mathcal{Q}$ and $\mathcal{P}_T \subseteq \mathcal{Q}$ be the contracts passed by the initial and final libraries. Report the number of newly passed contracts $|\mathcal{P}_T \setminus \mathcal{P}_0|$ and lost contracts $|\mathcal{P}_0 \setminus \mathcal{P}_T|$ separately. When $|\mathcal{Q} \setminus \mathcal{P}_0| > 0$, the

fraction of initially unsupported contracts newly passed supplies a bounded coverage measure. An empty eligible set is reported as not applicable. Splitting one handler into several files cannot change these scores when the tested behavior is unchanged. If interfaces cannot express a common contract suite, report results by interface; a pooled ranking would have no common denominator.

The contract suite fixes the behavioral scope of comparison. Its expected outcomes require independent justification, including explicit normative choices for institutional rules. Report suite construction and human judgments alongside computational cost. A skill-library baseline may compose any existing action under the same budget and receives credit whenever it satisfies the contract. The comparison measures coverage under the declared interface and search budget. Contract tests supply the proposed scores; the 29 intent edges and 13 families keep their trajectory-level units. Rule-growth evaluation connects mechanism gaps, candidate validation, executable admission, and reuse in future world operation.

C Extended Discussion of Alternative Views

Objection 1: Stronger content generation suffices. This view equates openness with surface richness: larger environments, more coherent tasks, more believable characters, and longer-horizon consistent events seem to naturally produce more open worlds. We argue this conflates content space with mechanism space. Content generation expands what a world can present, but cannot ensure the world learns how to operate. As long as new content is interpreted by the same action space, state transitions, resource constraints, and interaction protocols, it produces only more variants within existing mechanisms. Rule distillation targets additional operational rules acquired through interaction. The stronger the content generation, the more concealed the content trap: the world appears more open while its rule layer remains fixed at initial design.

Executable skills and world mechanisms define distinct targets of growth. Voyager accumulates executable skills within Minecraft's environment API (Wang et al., 2023a). DreamCoder learns reusable program abstractions that support later synthesis (Ellis et al., 2021). POET jointly generates environments and their solutions (Wang et al., 2019). These approaches study executable growth through skills, abstractions, and generated environments. Mechanic generation (Zook & Riedl, 2014) and Code World Models (Lehrach et al., 2026) directly address executable rules. Rule distillation centers mechanism acquisition on interaction-triggered proposals, continuing admission, and the consequences of accumulated world rules. The evaluation follows the operational relations that later shared interactions obey.

WorldCoder synthesizes a transition predictor from environmental experience and uses that predictor for planning (Tang et al., 2024). Guan et al. construct PDDL action models, with validators checking syntax and human feedback correcting semantic omissions (Guan et al., 2023). ARMS learns action preconditions and effects from plan examples (Yang et al., 2007); inductive logic programming learns relational hypotheses using examples and background knowledge (Muggleton, 1991). These approaches can implement parts of candidate formation. Learning and admission remain distinct decisions: a predictor is assessed against an environment, whereas an adopted simulator rule determines subsequent transitions. The same learned program can occupy either role. Rule distillation requires the second role to be explicit, versioned, and accountable to the world's existing contracts. The admitted world rule is the evaluation object: its demand origin, executable formalization, versioned installation, and effects on later shared interactions form a traceable acquisition chain.

Normative-system synthesis changes rules governing multiple agents. IRON monitors undesirable states and synthesizes compact norms to prevent recurrence (Morales et al., 2013; 2015). Rule distillation evaluates binding world relations through their demand origin, executable formalization, admission, and subsequent use. The relations include obligations and resource transformations alongside permissions and prohibitions. The competitive comparison specified in Appendices B.7 and B.8 gives both procedures the same installation authority, contract tests, and accounting of specification effort; V10 runs its norm-synthesis side on the V6 fixtures (Appendix D.25). The R0+ control measures execution capacity. V3 compares sequential feedback with our independent-sampling control under the same installation authority, source information, and three-request budget (Appendix D.11).

Objection 2: LLMs do not encode genuine real-world rules. Rule distillation turns LLM-suggested regularities into testable mechanism hypotheses. The objection concerns correlations, biases, and narrative patterns learned from text. Generated actions, justifications, and consequence predictions enter the pipeline as candidate hypotheses. Executable definitions and admission checks determine their effects on shared world state. Admission requires consistent state changes and reuse eligibility across analogous contexts. LLMs supply hypotheses; the validation pipeline tests which mechanisms satisfy the world's contracts.

Objection 3: Rules should be written directly by human designers. Rule distillation makes rule authoring continuous and auditable under human-defined objectives and constraints. Designers specify the initial world, admission criteria, and protected state. Resource relations, social interactions, institutional conflicts, and environmental interventions generate additional mechanism requirements as they combine. Rule distillation turns these interaction-driven requirements into candidate rules for review. Designers supply boundaries, interaction exposes gaps, and validation determines admission. This division connects explicit human design to sustained mechanism acquisition.

Objection 4: Simulation-based validation may be circular. Simulator validation evaluates executable transitions, state consistency, resource and rights constraints, and reuse. Correspondence to an external process is evaluated with observations of that process. Rule distillation makes implicit rule hypotheses explicit, executable, and auditable, with declared tests governing their admission. The evidence chain records accepted cases, rejected candidates, and cross-scenario tests. These records support independent assessment of both the admission decision and subsequent behavior.

Two production rules can each conserve resources and replay consistently while assigning different yields to the same inputs. Measurements of the target process distinguish their external fidelity. Economic and institutional rules additionally specify acceptable obligations and sanctions. Executable consistency, external fidelity, and normative choices therefore have explicit evaluation criteria and evidence sources.

Objection 5: Rule growth can amplify bias or instability. Institutional rules distribute access, obligations, and sanctions across agents. Their evaluation examines both execution consistency and the distribution of benefits and costs. Social laws, commitment protocols, and synthesized norms provide established objects of study in normative multi-agent systems (Shoham & Tennenholtz, 1995; Singh, 2007; Morales et al., 2013; 2015). Rule admission specifies acceptable consequences alongside mechanical checks. The audit protocol requires records of who receives benefits and who bears costs when a rule is reused. Evidence of harmful outcomes calls for review of the rule and its continued use. Fairness assessment uses these distributional records in addition to the validity predicate. Research on language-model reliability and social consequences motivates explicit evaluation of the priors entering this process (Bender et al., 2021; Hendrycks et al., 2023; Weidinger et al., 2021; Bommasani et al., 2022).

Mechanism acquisition is evaluated through interaction coverage, consistency, and maintenance cost. Continued evaluation records regressions and reuse alongside the number of installed rules. Rule interactions undergo conflict checks as the library grows; overlapping mechanisms can be consolidated. Mechanisms with conflicting effects enter revision, consolidation, or withdrawal. Comparisons with fixed-rule systems report coverage and cost under the same interaction contracts. These measurements connect rule growth to the world's ability to support and enforce subsequent interactions.

D Controlled Experiments on Executable Mechanism Acquisition

Scientific object. The experiments test acquisition of shared executable relations that govern subsequent actions. The controlled experiments run on a dedicated implementation of the Farm-Town specification. Ten agents share a library of 17 initial actions covering resource use, market settlement, social exchange, governance, and learning. Ordinary borrowing, hiring, and teaching are supported controls. The two target families specify additional relations: repayment-dependent collateral disposition and contributor-dependent purchase settlement or refund. Table 6 specifies their roles. The controlled world contains five locations, three resource nodes, two recipes, and two initial

Table 5: Controlled studies. All start from the same 17-action library; both suites means the standard held-out suite and the independent semantic challenge.

Study	Sources and comparison	Recorded outcome (unit)
V1	Public contract specification; direct vs. interaction-conditioned synthesis; authored skill controller	Each of 12 candidates covers every class of its family (15/15 credit, 13/13 pool) at 20 bindings; skill controller 14/15 and 12/13, failing only the protection probes
V2	D1–D4; private-role interaction vs. full-information planner; two models; 48 episodes	A later sequence exposed a D3 cancellation gap in an admitted candidate
V3	16 interaction sources (13 confirmed); sequential feedback vs. independent sampling, three calls each	Predeclared repeat evaluation: admitted 6/16 vs. 0/16 sources
V4	16 blocks; multi-role (I) vs. centralized (D) drafting with equal facts; sequential vs. pool search with equal feedback	Admitted 11/16, 6/16, 9/16 blocks for I–Seq, I–Pool, D–Seq; both suites 6/16, 3/16, 4/16; chains 8/32, 4/32, 8/32 tasks
V4E	26 admitted V4 libraries and two authored controls; identical independent-caller probes after the lawful arrangement	Probe cases: authored $\mathcal{R}_0$ composition, three probes succeed in 80/80 each; admitted libraries, 0/380 on those probes (0/1,180 on all nine)
V5	16 new blocks; private multi-role (I–priv) vs. public-facts centralized (D–pub) vs. full-information centralized (D–full) drafting; V4 synthesis and suites; eight D–pub blocks ended in provider refusals	Eligible 14/16, 5/16, 14/16; admitted 11/16, 2/16, 8/16; both suites 7/16, 0/16, 7/16 for I–priv, D–pub, D–full
V6	64 new blocks, one requested model; the three V5 source arms; sequential feedback in every arm and independent sampling on I–priv at the same budget; probes sealed before generation; V4 later tasks	Eligible 62/64, 37/64, 64/64; admitted 54/64, 16/64, 46/64; both suites 29/64, 2/64, 24/64 for I–priv, D–pub, D–full; feedback 54/62 vs. sampling 0/62 admitted; discriminating probes 0/1,720 under admitted libraries against 960/960 under $\mathcal{R}_0$; chains 32/128, 2/128, 45/128
V7	48 V6 blocks (D1–D3); each V6 planner's recorded later-task plan re-executed under the sealed probes with an independent-caller violation; no model call	Load-bearing goal 33/96, 20/96, 31/96 for I–priv, D–pub, D–full against 0/96 for $\mathcal{R}_0$
V8	Post hoc; the four demand contracts transcribed as clauses and compiled into violation cases by fixed transformations; the 116 admitted V6 libraries and two authored controls at five bindings; no model call	Violations permitted in 1,620/3,810 cases under $\mathcal{R}_0$, 0/3,810 under the authored capability library, 0/3,560 under admitted libraries
V9	The V6 protocol on its 64 blocks with Claude Opus 5.5, Sonnet 5 and Haiku 4.5 through Claude Code; hypotheses and sealed probes recorded before the first call	Opus: both suites 16/64, 22/64, 32/64 for I–priv, D–pub, D–full; feedback 23/52 vs. sampling 12/52 admitted; discriminating probes 0/1,500 under admitted libraries against 960/960 under $\mathcal{R}_0$ (0/4,560 on all nine). Sonnet and Haiku: both suites 0/64 in every arm
V10	IRON-style norm synthesis from the development conflicts of the 64 V6 blocks, with no model call; a generic obligation primitive composed by three Claude planners; V6 evaluation and the compiled clause suite	Synthesis: admitted 32/64, standard suite 31/64, semantic challenge 0/64; discriminating probes 640/960 against 960/960 under $\mathcal{R}_0$; discharge and window cases 400/400, as under $\mathcal{R}_0$. Primitive: both suites 30/64 authored, 27/64 and 20/64 for Opus with full and public information, 6, 5, 13 and 1 for Sonnet and Haiku; D3 and D4 0/32 in every condition
V11	The 64 V6 blocks with each demand replaced by the participants' need statements; multi-role drafting and V6 synthesis	Eligible 49/64, 26/64, 8/64 for Claude Opus 5.5, Haiku 4.5, Sonnet 5; Opus admitted 21/64, both suites 8/64, discriminating probes 0/180; Sonnet and Haiku admitted 1/64 each, both suites 0/64

laws. It advances in days with per-action energy costs and instantiates a subset of the full specification. Hunger and energy use a 0–100 scale. These choices are fixed in the saved configurations.

Table 6: Target interaction contracts. Coverage counts complete interactions.

Family	Participating roles	Required relation
Collateral credit	Debtor, creditor, collateral item	Food transfer creates an obligation; repayment permits redemption; unpaid default after the deadline permits forfeiture.
Pooled purchase	Buyer, partner, seller, purchase item	Contributions remain reserved until complete funding permits settlement; expiry permits contributor-specific refunds.

Executable distillation. Claude receives the public interaction specification and initial action definitions. Its JSON output pairs $h = \langle \phi, \alpha, \tau, \kappa, \omega \rangle$ with executable rules. Typed role declarations identify participants; preconditions encode applicability and constraints; ordered effects change state and record consequences. The interpreter bounds state reads, arithmetic, and effects. A failed action preserves its input state atomically. Candidate execution excludes assertion answers and arbitrary code. Consent is a Boolean response supplied with the simulated action; participant identifiers and consent are caller-supplied inputs to this interface; authenticated participants enter the runtime with V2 (Appendix D.7).

Generation and experimental groups. Group 1 traces proposals through admission and subsequent execution. Five pressure fixtures elicit separate proposer and recipient responses without naming either target family. For each family, the first qualifying non-refused proposal in fixture order may receive formalization and one revision. Selection records an exact proposal quotation and its semantic classification before formalization. Public contracts subsequently guide implementation, separating proposal content from specification-assisted synthesis.

Group 2 compares interaction-conditioned synthesis with direct synthesis in Table 7. The first call produces an interaction or a design plan; the next two calls produce a draft and one feedback-based revision. Both procedures receive identical public interfaces and development evidence. The final revision, as generated, determines the candidate. Incomplete outputs and unsuccessful trials remain in the allocated trial budget.

The four manuscript conditions comprise fixed rules, generated surface content, recorded open intents, and rule distillation. The first three are structural controls sharing a supported-action driver. Actual interaction outputs provide their surface and intent artifacts. These controls preserve the initial executable library. Direct synthesis adds another generation procedure, while an authored skill controller supplies a strong behavioral comparison. The six conditions combine structural controls, two sampled synthesis procedures, and an authored action-composition controller.

Group 3 evaluates the full validator, seven leave-one-gate-out settings, and an all-gates-disabled setting. The candidate bank includes authored valid/defective controls and every generated draft and revision. Authored defects and naturally generated errors retain separate provenance. Disabling a gate changes its admission role while retaining its recorded evidence.

Table 7: Prespecified V1 settings. Workload seeds identify simulator fixtures.

Setting	Value
Requested generation model	`claude-opus-5`; medium effort; fresh contexts; tools disabled
Targeted trials	Two families $\times$ two procedures $\times$ seeds $\{11, 23, 37\}$; three attempted calls per trial
Pressure calls	Five fixtures $\times$ two calls; up to two further calls per matching family
Skill-controller budget	20 actual base actions per request; up to 20 separately logged preflight actions
Evaluation workload	Seeds 1000–1019; unchanged admitted rules; separate admission-reuse and held-out bindings

Finite admission checks. The type gate checks bundle structure and DSL validity. The state gate evaluates positive development contracts and state invariants. Conservation checks every candidate invocation against the trusted transfer-only policy for these two families: resource totals must remain unchanged, and the candidate cannot authorize production by changing accounting counters. Test-family metadata fixes the policy independently of the candidate's declared family. Trusted initial actions and clock updates account separately for legitimate production or consumption. The constraint gate evaluates required refusals. Replay reruns fixed states and compares the resulting states. Conflict checks initial-rule collisions and regression equivalence. Reuse checks alternative bindings designated for admission. Each gate returns pass, fail, or pending; admission requires every enabled gate to pass. Held-out outcomes remain outside this decision. Failure matrices record check overlap and the admission decision under each ablation.

Common-world comparison and reuse. Comparative cases begin from ordinary shared initial states. Each method establishes its own agreement through actual actions. External assertions evaluate resources, ownership, custody, and required refusal behavior; canonical record layouts are excluded. The skill controller composes existing actions using full state and private memory. Cooperative endpoints receive full credit. Independent initial-action probes bypass that controller and test premature collateral return or disposal of a reserved item. Lifecycle, refusal, and enforcement results remain separate; family coverage requires every designated case to pass. Precreated obligation records serve only as representation diagnostics.

Outcome measures. Evaluation includes multi-day default and refund schedules. Metrics distinguish new and lost contracts, family coverage, regressions, and actual rule invocations. Each candidate is aggregated across held-out bindings before reporting the three trial outcomes, means, and ranges. Scheduled reuse records opportunities and successful or failed invocations under unchanged rules. Exact replay measures interpreter consistency. Model identifiers, tokens, cost, and elapsed time document generation effort.

Cumulative acquisition. For each family, selection takes the first admitted final candidate in the prespecified trial order. Credit and pooling are then admitted sequentially into one shared library. Two frozen cross-family schedules test whether a credit encumbrance blocks disposal into a pool until redemption and whether a pool reservation blocks collateralization until refund. Each schedule uses 20 held-out bindings. These 40 cases are separate from the per-family coverage counts. An authored combined library supplies a separately labelled implementation control. Both assessments use their libraries unchanged.

Experimental configuration. The protocol, prompts, public contexts, and test settings were fixed before primary generation. Each targeted trial requests Claude Opus 5 with medium reasoning effort in a fresh context. Table 7 specifies the shared synthesis budget and simulator inputs. Evaluation executes the generated candidates without further model calls. All 12 targeted trials and all five pressure trials are included.

D.1 Controlled Acquisition and Reuse Results

All targeted synthesis trials supported the specified interactions. All 12 final candidates were admitted and passed every target-family contract class in the held-out suite (Table 8). Direct synthesis and interaction-conditioned synthesis each achieved complete coverage in all three trials per family. Credit candidates passed 15/15 classes and 300/300 binding-specific executions per trial. Pooled-purchase candidates passed 13/13 classes and 260/260 executions per trial. Every class was exercised under 20 bindings drawn from evaluation seeds 1000–1019. These bindings vary participants and quantitative state within each declared interaction class. Coverage is aggregated by candidate across the bindings. Candidate-level coverage therefore had mean 100% and range 100–100% in each family–procedure group.

All 36 targeted model calls parsed successfully. Each trial retained its draft and final revision, and every draft also passed admission. The evaluated candidate was always the allocated final revision, as generated. Each candidate added four executable definitions to the 17-action initial library. The two procedures achieved equal coverage under the shared public specification and feedback budget.

These results support executable formalization, finite-check admission, and subsequent enforcement of the specified relations.

Table 8: Every targeted generation trial. "Classes" requires success across all 20 bindings of each semantic class; "Executions" counts those bindings. "+Rules" counts added executable definitions. Every row also passed all 13 supported control classes, or 260/260 control executions. Seeds identify workloads.

Family	Procedure	Seed	Admission	Classes	Executions	+Rules
Credit	Direct	11	admitted	15/15	300/300	4
Credit	Direct	23	admitted	15/15	300/300	4
Credit	Direct	37	admitted	15/15	300/300	4
Credit	Interaction	11	admitted	15/15	300/300	4
Credit	Interaction	23	admitted	15/15	300/300	4
Credit	Interaction	37	admitted	15/15	300/300	4
Pool	Direct	11	admitted	13/13	260/260	4
Pool	Direct	23	admitted	13/13	260/260	4
Pool	Direct	37	admitted	13/13	260/260	4
Pool	Interaction	11	admitted	13/13	260/260	4
Pool	Interaction	23	admitted	13/13	260/260	4
Pool	Interaction	37	admitted	13/13	260/260	4

Table 9: Held-out contract coverage by condition: a class counts when all 20 bindings pass. Generated rows show the three trials per family; the fixed-rule, generated-content, and recorded-intent conditions share the supported-action driver and the same coverage.

Condition	Credit (15 classes)	Pool (13 classes)
Fixed rules / generated content / recorded intent	6	3
Authored state-aware skill library	14	12
Direct rule generation	15, 15, 15	13, 13, 13
Interaction-conditioned distillation	15, 15, 15	13, 13, 13

Supported initial behavior was preserved. Every admitted library passed all 13 supported control classes under all 20 bindings. The controls cover market exchange and ordinary borrowing, together with hiring, teaching, and the existing institutional handlers. Each targeted candidate passed 260/260 executions of the same control classes. Each candidate also passed its independent world-enforcement class in all 20 bindings. The authored skill controller passed 14/15 credit classes and 12/13 pool classes, with its single failed class in each family confined to those protection probes. The generated libraries therefore supplied a shared restriction that remained effective when the probe bypassed the cooperative controller.

Pressure-only interactions retained ordinary arrangements. The five pressure settings yielded four complete proposer–recipient pairs, all proposing immediate barter or ordinary tool lending (Table 10). One pair was incomplete because the proposer response did not satisfy the required format. None contained an eligible target-family proposal. All five settings remain in the denominator. These interactions measure proposal formation under pressure, while the targeted trials measure synthesis from a supplied specification.

A generated credit rule enforced repayment before redemption. The trace in Table 11 uses the first admitted final credit candidate in the prespecified trial order, evaluated on the repayment-and-redemption case with evaluation seed 1000. The debtor owns the collateral sickle; the other agent is the creditor. Their initial gold balances are 8 and 11, and each holds 10 bread. The agreement transfers three bread and requires four gold by day 2. All six recorded actions occur on day 0.

Creation moves the bread and places the debtor-owned sickle in the creditor's custody. Premature redemption is refused because repayment is incomplete, preserving the preceding state. Repayment then changes gold balances to 4 and 15. Redemption returns custody to the debtor and releases the

Table 10: All pressure fixtures. A model annotation applied the prespecified family criteria. "Eligible" requires a complete parsed pair containing a qualifying target-family proposal.

Fixture	Parsed pair	Proposed interaction	Eligible
1	Incomplete	No complete proposal; recipient states existing exchange terms.	No
2	Complete	Present wood-for-bread barter; recipient counterproposes a smaller immediate exchange.	No
3	Complete	Present-goods barter; future claims on the sickle and later earnings are explicitly excluded.	No
4	Complete	Ordinary same-day tool loan; proposed teaching remains independent of the loan.	No
5	Complete	Ordinary tool loan following an upfront grain gift; no conditional pooled purchase or repayment-dependent collateral.	No

lock. Duplicate repayment and duplicate redemption also preserve state on refusal. The table reports actual stored states from the generated-rule execution.

Table 11: Saved execution of a model-generated credit candidate. Bread and gold columns give debtor/creditor balances. The debtor owns the sickle throughout; custody changes as shown. T/F denotes the repayment flag.

Step	Outcome	Bread	Gold	Holder	Obligation
Initial	—	10/10	8/11	Debtor	Absent
Create	Success	13/7	8/11	Creditor	F; active; locked
Redeem early	Refusal	13/7	8/11	Creditor	F; active; locked
Repay	Success	13/7	4/15	Creditor	T; active; locked
Repay again	Refusal	13/7	4/15	Creditor	T; active; locked
Redeem	Success	13/7	4/15	Debtor	T; redeemed; unlocked
Redeem again	Refusal	13/7	4/15	Debtor	T; redeemed; unlocked

Sequential admission preserved both families and their composition. The prespecified selection takes the first admitted final candidate for each family, credit followed by pooled purchase. Both passed admission in that order, producing a 25-definition shared library. The resulting model-generated library passed all 40 joint checks and reproduced their saved-state results on all 40 replays. The denominator comprises two fixed composition patterns with 20 bindings each. In the first pattern, active credit blocks a pooled purchase of the collateral; repayment and redemption then permit that purchase. In the second, an active pool blocks use of the reserved item as collateral; expiry and refund then permit credit creation.

The combined library retained 300/300 credit executions, 260/260 pooled-purchase executions, and 260/260 supported control executions. The separately labelled authored-reference library also passed the 40 joint checks. The joint workload measures sequential compatibility and scheduled reuse of fixed definitions. The two candidates are drawn from the twelve reported generation trials.

D.2 Validator Ablations

The full validator admitted both authored valid controls and all 24 generated versions, and rejected every authored defect. The bank contains 35 candidate versions: two valid bundles, nine deliberately defective bundles, and all model-generated drafts and final revisions. Drafts and revisions retain their shared generation lineage. Table 12 reports all nine gate settings. The authored defects test specified failure modes; model error rates are measured from the generated-candidate population.

Removing the constraint gate admits a candidate that omits recipient consent; it passes 280/300 credit executions and fails the remaining 20. Removing conservation admits a candidate that creates gold and declares matching production in its own accounting record. That candidate still passes all 300 behavioral contract executions: the accounting identity alone accepts its self-authorized source. An independent trace audit finds 180 successful held-out creation actions, each adding one gold; in a

Table 12: Admission under all validator interventions. Columns count admitted versions. "Behavioral failure" counts admitted authored defects failing at least one held-out interaction class; resource-policy violations are assessed separately in the text.

Gate setting	Valid (2)	Defective (9)	Generated (24)	Behavioral failure
All seven	2	0	24	0
Without type	2	0	24	0
Without state	2	0	24	0
Without conservation	2	1	24	0
Without constraint	2	1	24	1
Without replay	2	0	24	0
Without conflict	2	0	24	0
Without reuse	2	0	24	0
All disabled	2	9	24	8

representative action, physical gold rises from 279 to 280 while the production counter rises from 0 to 1. The transfer-only admission policy rejects these changes. Behavioral coverage and resource integrity therefore require separate checks. Disabling all gates admits all nine defects, eight of which fail the behavioral suite; the ninth is the forged-accounting case. All thirteen supported control classes remained intact for these admitted defects. Contract validity and regression preservation therefore require separate measurements.

Conservation and consent checks each changed admission decisions on this bank. The other five single-gate removals preserved the same decisions through overlapping checks. The complete check vectors identify the overlap responsible for unchanged decisions.

Generation effort. Generation used 46 calls: 36 for targeted synthesis and ten for pressure interactions. Forty-five outputs met the required format. The reported generation cost was USD 24.75. Both synthesis procedures received the same call and revision budgets.

Scheduled invocations and replay. Each generated credit candidate receives 640 new-rule invocations across held-out schedules: 260 successful transitions and 380 expected refusals. Each pool candidate receives 700: 300 successes and 400 expected refusals. The evaluator schedules these invocation opportunities under unchanged rules. All credit evaluations replay consistently in 560/560 cases and all pool evaluations in 520/520, including the 260 control cases in each evaluation. Replay confirms deterministic execution; contract assertions provide the separate behavioral test.

Invocation counts match each executed request to its resulting state and exclude clock-only observations. Across all twelve generated runs, every scheduled positive request succeeded. Refusals are counted separately against their specified contract conditions.

D.3 Executable Interface and Complete Contract Inventory

The evaluation fixes 28 executable case classes. Tables 13 and 14 enumerate all 15 credit and 13 pooled-purchase classes used in primary comparison. Every class begins in an ordinary world and executes its own setup. Refusal requires unchanged state for the refused interaction. The 20 held-out bindings vary participants and quantities within these fixed classes. Several classes share individual predicates; the class count measures declared test programs. Precreated-obligation diagnostics and the two joint composition patterns have separate evaluation records.

Acquired relations extend the initial world capabilities. Table 15 distinguishes the capabilities available before synthesis from the conditional relations supplied by the selected generated candidates. The initial world supports transfers, ordinary loans, listed-asset purchases, and wage escrow. Credit adds a relation between debt, collateral release, and default. Pooled purchase connects multiple contributions to settlement and refunds. Both mechanisms use the world's existing asset reservations to constrain subsequent actions.

The comparison tests whether these acquired relations govern later interactions when callers bypass the cooperative controller. Reservation checks apply to gifts, listings, and borrowing. Initial market

Table 13: Complete credit contract-class inventory. Setup means successful credit creation unless creation itself is the requested refusal.

ID	Semantic class	Required executed behavior
C01	Repayment and redemption	Refuse early redemption; accept repayment and redemption; refuse repeated repayment and redemption.
C02	Creditor refusal	Refuse creation when the supplied consent response is false.
C03	Wrong collateral owner	Refuse creation when the declared debtor does not own the item.
C04	Negative food amount	Refuse creation with a negative requested bread transfer.
C05	Self-dealing	Refuse creation when debtor and creditor are the same agent.
C06	Past-due terms	Refuse creation with a deadline before the current day.
C07	Insufficient food	Refuse creation when the creditor lacks the requested bread.
C08	Duplicate agreement	Create once; refuse another creation using the same agreement identifier.
C09	Shared collateral protection	After setup, refuse unmediated ordinary return by the actual holder and disposal by the debtor.
C10	Unpaid redemption	After setup, refuse redemption without repayment.
C11	Participant binding	Refuse repayment, redemption, and default requested under the wrong participant role.
C12	Premature default	Refuse forfeiture before the deadline.
C13	Unpaid default	Advance three days; refuse late repayment; transfer collateral ownership on default; refuse repeated default.
C14	Paid default refusal	Repay, advance three days, and refuse forfeiture of the paid collateral.
C15	Insufficient repayment	Create credit for a debtor with insufficient gold; refuse repayment.

Table 14: Complete pooled-purchase contract-class inventory. Setup reserves the buyer's contribution and the seller's asset; funded setup also executes the partner's contribution.

ID	Semantic class	Required executed behavior
P01	Joint settlement	Refuse incomplete settlement; accept the partner's contribution and settlement; refuse repeated joining and settlement.
P02	Seller refusal	Refuse opening when the supplied seller response is false.
P03	Partner refusal	After setup, refuse contribution when the partner response is false.
P04	Inconsistent price	Refuse opening when the two specified shares do not sum to the price.
P05	Duplicate agreement	Open once; refuse another opening with the same identifier.
P06	Insufficient buyer deposit	Refuse opening when the buyer lacks its specified contribution.
P07	Shared reservation protection	After setup, refuse seller disposal and ordinary return by the actual holder, bypassing the controller.
P08	Incomplete funding	After setup, refuse settlement before the partner contributes.
P09	Premature cancellation	Refuse cancellation before expiry.
P10	Participant binding	Refuse joining, settlement, and cancellation under the wrong participant role.
P11	Insufficient partner deposit	After setup, refuse joining when the partner lacks its contribution.
P12	Unfunded expiry refund	Advance three days after setup; refuse late joining and settlement; refund the buyer; refuse repeated cancellation.
P13	Funded expiry refund	Advance three days after funded setup; refuse late joining and settlement; refund both contributors; refuse repeated cancellation.

Table 15: Initial capabilities and conditional relations added by the selected generated candidates.

Mechanism	Initial capability	Added conditional relation
Collateral credit	Resource transfers and ordinary loans that change custody while preserving ownership.	Food credit is secured by debtor-owned collateral. Repayment permits redemption; unpaid default after the deadline transfers collateral ownership to the creditor.
Pooled purchase	Listed-asset purchases and wage escrow funded by one employer.	Two contributions fund one purchase. Settlement requires complete funding, the agreement's asset reservation, and a date at or before the deadline. Cancellation after expiry refunds each contributor's deposit.
Shared asset protection	Ownership, custody, and reservation checks govern gifts, listings, and borrowing.	Credit and purchase agreements reserve assets in shared state. Existing gift, listing, and borrowing actions then refuse those reserved assets; agreement completion releases them for later use.

purchase instead uses listing validity, seller ownership, buyer funds, and location, without an additional custody or matching-reservation condition. Ordinary return applies to ordinary loans and has no repayment or deadline condition. A refusal on a precreated credit agreement therefore reflects agreement-type incompatibility. Each method establishes its own agreement in the common-world comparison, so subsequent tests assess the restrictions produced by that method's setup.

Repayment-conditioned collateral release. The first admitted final credit candidate under the prespecified trial order links repayment to collateral release. A redemption request names an existing agent and a collateral-credit agreement that identifies its pledged asset. Redemption requires the agreement to be active and marked as repaid, the requesting agent to be both its recorded debtor and the asset's owner, and the asset to remain locked to that agreement. These conditions determine redemption eligibility for this candidate.

A successful transition returns custody to the debtor, releases the collateral lock, marks the agreement as redeemed, and sets its closing date to the current day. Asset ownership and all resource balances remain unchanged. The debtor's memory and the world event history record the redemption, identifying the agreement, collateral, creditor, and date. State changes commit atomically; a failed action leaves the world unchanged. The transition makes repayment govern collateral release through shared world state, so subsequent actions observe the completed agreement and the released asset.

D.4 E1: INTERACTION-DRIVEN ACQUISITION EPISODES

The v2 protocol evaluates task-demand episodes as sources of candidate operational relations. Each episode begins from a Farm-Town fixture, a public action interface, and private role preferences. The protocol crosses four demand settings, three initial-condition strata, Claude Opus 5 and GPT-6 Astra, and two source procedures. This design yields 48 acquisition episodes. The four demand settings cover cross-time resources, distributed inputs, verifiable performance disputes, and competing timed access. The strata mark supported tasks, potential mechanism gaps, and legally unsatisfiable tasks. The labels define the evaluation matrix and remain hidden from generation contexts.

The interaction source procedure records bounded role dialogue and action attempts. It permits at most six source calls per episode. A final proposal requires affected roles to approve the same proposal version. The full-information source procedure gives a planner the same public world facts and role preferences in a fresh context. It permits the same six-call budget and reserves terminal role confirmations within that budget. Both procedures retain ordinary completion, refusal, incomplete outputs, and absence of a proposal in the 48-episode denominator.

A candidate operational relation enters formalization only when the source record contains a concrete proposed relation. The same requested model family supplies the source process and the later formalizer for that episode. The formalization context contains the recorded demand, the source transcript or plan, the public DSL, the initial action interface, and the trusted kernel contract. It omits target mechanism names, reference implementations, held-out outcomes, fixed rule identifiers, and proposal-specific oracle answers. The formalizer may produce one draft and one feedback revision

in the main branch. A separate self-check branch receives the original requirements and a generic self-check instruction under the same revision budget. Twelve blinded annotation batches label source records. Their cards use the same public action-result projection as source prompts and omit candidate code, later execution outcomes, generating-model names, and source-condition names. The annotation records identify their model source and serve as auxiliary evidence.

The v2 implementation separates three authority scopes. Consent scope belongs to role authorization: generated booleans and copied role names cannot authorize another participant. Bundle scope belongs to installed candidate rules and guards: host installation assigns bundle domains and prevents unrelated bundles from inheriting contract authority. Kernel scope belongs to the trusted runtime: authenticated actors, one-use approvals, trusted receipts, immutable fields, resource accounting, and clock advancement remain outside candidate control. These scopes define the fixed experimental substrate; acquisition counts changes to the installed candidate relations.

D.5 E2: POST-ADMISSION USE AND FINITE CONTINUOUS ADMISSION

The reuse protocol evaluates whether a later planner selects and executes an admitted relation in a new task. The M stratum contains 16 acquisition processes, formed by four demand settings, two requested models, and two source procedures. Each process receives two later tasks. One task changes participant bindings. The other changes timing or interference structure. Each later task is evaluated under the initial library and under the library updated by that process. The crossed design creates 64 reuse slots.

Each reuse slot receives at most two model decision calls. The prompt exposes the current public action interface and the public rule list for the selected library arm. It does not expose the acquisition transcript, the generating model's plans, hidden oracle labels, or an instruction to use the new rule. The planner may use initial actions, select an admitted rule, or decline. The record separates opportunity, selection, actual invocation, and the combined execution-and-goal outcome. The follow-up separately evaluates final task-state predicates. A missing admitted rule keeps the slot in the denominator with zero rule-reuse opportunities.

Predeclared offers bound the participants, assets, resource amounts, and execution times authorized for each later task. Both library arms receive the same offers. The authenticated caller must be one of the action's participants. Each action requires a one-use authorization and is committed only if its resulting state change respects the offer. Third-party resource changes, unrelated asset changes, excess spending, and actions outside the authorized time window are rejected.

The changed-time D4 reuse variant preserves the semantic custody calendar. D4 asks whether timed access to a specific asset follows the advertised day schedule. Rewriting that calendar would change the task target. The slot therefore records the timing/interference variant while keeping the custody phase days fixed for D4. Other changed-time tasks may shift later phase days according to the frozen reuse fixture transform.

The R0+ comparison uses four authored template families, one for each demand setting. Their declared parameters and authored plans define a positive control for shared execution. The templates are fixed before primary generation. R0+ measures execution capacity supplied by these templates; comparisons between acquisition algorithms require the same installation authority and contract tests. The analysis compares the execution capabilities of R0, R0+, and acquired candidate libraries separately.

Continuous admission evaluates interaction-source M candidates in four ordered sequences: D1–D2–D3–D4 and D4–D3–D2–D1 for each model. V2 checks earlier tasks under the combined library and the new candidate's task with the initial library. Appendix D.10 compares these evaluation conditions. Rejection, rule collision, missing candidates, and regressions remain in the sequence. Rollback calibration separately tests restoration of the prior state and library.

D.6 E3: CANDIDATE VALIDITY, FEEDBACK, AND ADMISSION ABLATIONS

Candidate validity is evaluated for every draft, feedback revision, and self-check revision. Feedback supplies the verifier's development-test results; self-check supplies the original specification and a generic revision instruction under the same budget. Each branch is scored on its own candidate.

Admission uses seven finite checks. The type check validates candidate syntax, role types, and installation compatibility. The state check evaluates executable positive cases and invariants. The conservation check rejects unlicensed resource creation, candidate accounting writes, and self-certified production. The constraint check evaluates required refusals and authorization boundaries. The replay check verifies deterministic execution from saved states. The conflict check detects rule collisions and regression evidence. The reuse check requires designated alternative bindings before admission. Held-out cases are recorded and excluded from admission decisions.

Gate ablations are decision ablations over one fixed evidence vector. The validator records all seven check outcomes once, then derives the full setting, seven leave-one-gate-out settings, and the no-admission-gates setting from that vector. Kernel protections, DSL schema checks, authenticated dispatch, install collisions, immutable fields, and trusted accounting remain enabled in every setting. Checks may overlap; the report records both changed decisions and decisions preserved by the remaining checks.

The held-out suite uses 20 participant and asset bindings, indexed 2000–2019. These bindings change role identifiers and asset names. Within each binding, separate cases test action order, boundary timing, repeated calls, coexistence, and interference. Binding indices identify simulator inputs. The held-out suite evaluates every available candidate separately from admission and cannot feed back into candidate admission or revision.

D.7 V2 Experimental Settings

V2 extends the controlled Farm-Town action semantics with authenticated participants, scoped authorizations, trusted receipts, and independently specified acquisition and reuse tasks. These experimental settings define which relations a candidate can add and which permissions remain fixed.

All 48 acquisition episodes begin from fresh model contexts. Public action observations contain the attempted action, its arguments, outcome, refusal reason, descriptive events, and authorization outcome. They exclude private runtime state. The blinded annotation cards use the same observation projection. Planned actions, attempted actions, and successful executions are counted separately. The experimental matrix, scenarios, admission tests, and counting units are fixed before generation.

The primary call budget is capped at 572 model calls. The cap comprises 48 source procedures with up to six source calls plus draft and feedback formalization, 48 self-check revisions, 64 reuse slots with up to two decision calls, and 12 blinded annotation batches. CPU validation, held-out bindings, replay, gate ablations, continuous admission streams, baseline execution, and rollback checks reuse saved outputs and do not consume primary generation calls. The budget covers the 48-episode study and its associated evaluations.

The frozen public context contains the action interface, DSL grammar, trusted-runtime rules, and scenario-visible facts. It excludes scenario labels, reference candidates, held-out tests, candidate-specific assertion answers, and results from other branches. Executed reuse slots are analyzed using their original model responses and execution outcomes; unavailable slots remain in the planned denominator.

The analysis separates mechanism acquisition from execution support. New action names, locks, records, and rule counts are diagnostic objects. Admission and complete acquisition-with-reuse are reported separately. The latter requires a traceable source, successful installation and independent execution checks, and actual use in a later task under the admitted library. Strong skills and R0+ templates measure execution capacity; the source-to-admission and later-use records measure acquisition.

D.8 V2 interaction-driven acquisition results

Feedback revisions admitted candidate relations in 2/16 M-stratum episodes, and both candidates passed every finite held-out check. The V2 run completed all 48 acquisition episodes and all 64 later-task reuse slots. In the M stratum, 9/16 source records produced an eligible relation: 6/8 from D and 3/8 from I. Both admitted candidates came from D; I admitted 0/8. Draft and self-check branches admitted none.

All 16 S episodes produced no eligible mechanism. Their source outcomes were ordinary completion in 6 cases, ordinary attempts without verified goal completion in 8 cases, one refusal, and one unconfirmed proposal. The U stratum produced 5/16 eligible proposals; every available U candidate was rejected by the validator. The acquisition denominator includes admitted candidates, no-candidate outcomes, and lawful exits.

The admitted timed-access candidate passed 200/200 held-out cases, and the admitted labor-settlement candidate passed 260/260, each over 20 bindings. One distributed-input feedback candidate remains pending because participant rebinding prevented completion of its admission and held-out evaluations. It remains in the candidate denominator and contributes no admission or held-out success.

Primary acquisition-with-reuse success was 0/16 in the M-source denominator. Later-task evaluation retained all 64 slots. All 32 R0 slots executed; 1/32 passed the combined execution-and-goal check. Among 32 post-admission arms, 28 had no admitted library and were classified as unavailable; the remaining 4 evaluated slots all failed the combined check. The two D4 post-admission slots executed 4 successful generated-rule actions in total. The strict metric requires successful dispatch and satisfied task-state predicates together. The primary prompt listed rule signatures without stating the caller-participation requirement and suggested delegation beyond the permitted scope. Appendix D.9 evaluates an additional sample with explicit caller instructions.

Table 16: V2 source outcomes over all 48 acquisition episodes.

Model	Src	Str	N	Elig	Ord.	Unver.	Ref.	None/unconf.	Adm.+held
Claude	D	M	4	3	0	1	0	0	1
Claude	D	S	4	0	2	2	0	0	0
Claude	D	U	4	2	0	0	2	0	0
Claude	I	M	4	2	0	2	0	0	0
Claude	I	S	4	0	0	3	1	0	0
Claude	I	U	4	3	0	0	1	0	0
GPT	D	M	4	3	0	1	0	0	1
GPT	D	S	4	0	3	1	0	0	0
GPT	D	U	4	0	0	0	2	2	0
GPT	I	M	4	1	0	2	1	0	0
GPT	I	S	4	0	1	2	0	1	0
GPT	I	U	4	0	0	0	4	0	0

Rows are grouped by requested model, source procedure, and hidden stratum. Elig. means a confirmed candidate relation entered formalization. Ord. means ordinary completion. Unver. means ordinary actions were attempted but did not satisfy the frozen observable goal check. Adm.+held counts feedback-branch candidates admitted and passing all finite held-out checks.

Table 17: Candidate branch outcomes under the full validator.

Branch	Candidates	Admit	Reject	Pending	No cand.
draft	13	0	13	0	35
feedback	14	2	11	1	34
self-check	14	0	14	0	34

The feedback branch is the deployment branch. Draft and self-check revisions had zero admissions. No-candidate rows remain in the 48-episode denominator but do not enter candidate-level validator decisions.

Evaluation criteria. Execution checks determine the reported outcomes. Auxiliary model annotations contribute to generation cost. The continuous-admission and rollback experiments evaluate finite calibration cases.

D.9 Later Use with Explicit Caller Instructions

This additional eight-task study supplies explicit caller-participation and authorization requirements to the later-task planner. The primary experiment exposed rule signatures without these instructions. Each task is a fresh model draw using the original candidates; its outcomes are analyzed separately from the primary 64-task evaluation.

Table 18: Validator decision ablations over one fixed evidence vector.

Setting	Draft A/R/P/NC	Feedback A/R/P/NC	Self A/R/P/NC
All gates	0/13/0/35	2/11/1/34	0/14/0/34
No type	0/13/0/35	2/11/1/34	0/14/0/34
No state	0/13/0/35	2/11/1/34	0/14/0/34
No conserve	0/13/0/35	2/11/1/34	0/14/0/34
No constraint	0/13/0/35	2/11/1/34	0/14/0/34
No replay	0/13/0/35	2/11/1/34	0/14/0/34
No conflict	0/13/0/35	2/11/1/34	0/14/0/34
No reuse	0/13/0/35	2/11/1/34	0/14/0/34
No gates	13/0/0/35	14/0/0/34	14/0/0/34

A/R/P/NC denotes admitted, rejected, pending, and no candidate. All seven single-gate removals left the full-validator decisions unchanged on this bank. Disabling all admission gates admitted every available candidate, while kernel protections and DSL safety remained enabled.

Table 19: Later-task reuse slots over the full 64-slot denominator.

Lib	Src	Slots	Eval	Unavailable	Pass	Fail	No eval	New slots	New calls
R0	D	16	16	0	1	15	0	0	0
R0	I	16	16	0	0	16	0	0	0
post	D	16	4	12	0	4	12	2	4
post	I	16	0	16	0	0	16	0	0

Unavailable post-admission arms correspond to source processes with no admitted library and are retained separately from failures. The four evaluated post-admission slots all failed the combined execution-and-goal check; the two D4 post-admission slots nevertheless executed two successful generated-rule actions each. The strict check requires both successful dispatches and satisfied task-state predicates; any refused action counts against it, including an intended probe. The prompt/interface conditions described above are included in the failure analysis.

Table 20: Authored baseline controls by stratum.

Str	Arm	Fixtures	All pass	Coop. pass	Lawful refusal	Failed	Total
S	R0	4	0	0	0	88	440
S	skill	4	4	4	0	0	440
S	R0+	4	4	4	0	0	440
M	R0	4	0	0	0	528	1034
M	skill	4	0	4	0	242	1034
M	R0+	4	4	4	0	0	1034
U	R0	4	0	0	0	176	528
U	skill	4	0	0	4	88	528
U	R0+	4	0	0	4	88	528

Skill and R0+ solve all supported S fixtures. In M, skills complete cooperative endpoints but not all shared-enforcement checks; R0+ is a prespecified capability control. In U, lawful refusal is the expected behavior for skill and R0+; R0 lacks that control.

Table 21: Finite continuous-admission streams using the I-source branch.

Model	Order	Events	Admit	Reject	Missing
Claude	D1–D2–D3–D4	4	0	2	2
Claude	D4–D3–D2–D1	4	0	2	2
GPT	D1–D2–D3–D4	4	0	1	3
GPT	D4–D3–D2–D1	4	0	1	3

No stream admitted a new I-source candidate. The authored rollback calibration is separate: it injected an invalid minting rule, detected the fault, restored state and library, replayed the external event, and retained the failed version record.

Table 22: Model calls and reported generation costs.

Model	Stage	Calls	Cost records	Reported USD
Claude	acquisition	106	106	28.25
Claude	auxiliary annotation	6	6	3.16
Claude	reuse	18	18	1.86
GPT	acquisition	77	0	–
GPT	auxiliary annotation	6	0	–
GPT	reuse	18	0	–

Costs are provider-reported metadata only; a dash marks a GPT cost field with no provider record.

All eight slots cross both admitted candidates' two later tasks with both libraries, fixed before follow-up calls. Both arms received identical generic runtime context and caller/standing-offer clarification. Rules, libraries, tasks, permissions, model profiles, and the two-call budget were unchanged. No prewritten plans or forced generated-rule use were supplied. Each slot used one planning call. The protocol was frozen before the first follow-up model call.

Table 23: All eight interface-follow-up slots. Strict is the saved combined execution-and-goal outcome; State is the final task-state diagnostic. A/S are attempted/successful actual action dispatches; New counts generated action IDs separately from guards.

Source	Task	Library	Strict	State	A/S	New A/S
D3/Claude/D	New roles	R0	failed	pass	7/6	0/0
D3/Claude/D	New roles	Admitted	failed	pass	7/5	3/2
D3/Claude/D	Time/interf.	R0	failed	pass	8/6	0/0
D3/Claude/D	Time/interf.	Admitted	failed	pass	7/5	3/2
D4/GPT/D	New roles	R0	passed	pass	4/4	0/0
D4/GPT/D	New roles	Admitted	passed	pass	3/3	3/3
D4/GPT/D	Time/interf.	R0	passed	pass	5/5	0/0
D4/GPT/D	Time/interf.	Admitted	failed	pass	4/3	3/3

The D4 new-role task completed a traceable acquisition-to-later-use chain: the later planner selected the admitted activation, claim, and restoration actions, all three executed successfully, and the strict check passed. Across this separate follow-up, the strict outcome was 1/4 for admitted libraries and 2/4 for R0. The successful candidate originated from the full-information D procedure and had passed admission and all 200 finite held-out cases.

The final-state assessment passed 4/4 tasks under each library. It tests final task predicates independently of action refusals, authorization outcomes, clock advancement, and invariant checks. Tasks without a completed state evaluation remain unknown. The strict measure combines execution validity with task-state success. Dedicated independent-caller probes separately test whether the shared library enforces the contract.

The saved traces locate the strict failures in action refusals, with no goal-predicate failures. Both D3 admitted sequences completed registration and receipt-conditioned settlement, then attempted cancellation after settlement. Those cancellation actions were refused. The evaluated sequence included alternatives listed by the planner, so its strict score covers every listed action. D3 R0 plans attempted returns after loan closure; one also attempted rest with a false precondition. In both changed-time admitted slots, generated guards blocked an interference listing. D4 executed all three generated custody actions. The strict scores therefore combine plan sequencing, conditional-branch handling, and refusal scoring. The R0 results measure cooperative task completion; independent-caller enforcement remains a separate test object.

The original and additional samples use separate stochastic draws. Their comparison describes performance under the two prompt conditions. Both library arms retain every task and use the original generated candidates.

D.10 Temporal Rights and Joint-Library Validation

Completion establishes a settlement right. The D3 contract permits lender cancellation before work is completed and preserves the worker's settlement right after completion. We test this temporal condition with four action sequences: completion followed by settlement; cancellation before completion; completion–cancellation–return–settlement; and completion–cancellation–settlement. The first two sequences succeed. In the latter two, the V2 candidate permits cancellation after a trusted completion receipt, closes the arrangement, and blocks settlement. Cancellation leaves custody unchanged; return requires the borrower's own authenticated action.

The candidate passed the original 260/260 checks, which covered immediate settlement after completion and independent calls before completion. The added sequences test whether an intervening cancellation preserves the earned right. Rebinding participants tests identity variation; changing the action order tests this temporal obligation. The D4 later-use result is unchanged. Appendix D.12 evaluates four model revisions against 320 held-out checks each, including cancellation after completion and repeated terminal actions.

Validate both tasks in the combined library. Joint-library validation evaluates the new relation and previously supported tasks under the same installed rules. In V2, new-task checks used the candidate with the initial library, while earlier-task checks used the combined library. Separate D3–D4 evaluation passed 20/20 development cases under each library. The V3 combined-library protocol evaluates both tasks after installation (Appendix D.11).

D.11 V3: paired acquisition and temporal validation

V3.1 tests mechanism acquisition through confirmed interaction proposals, executable admission decisions, and later use of the installed rules. The experiment extends the four demands and trusted runtime in Appendix D.4. New interaction acquisition and revision of historical candidates are evaluated separately. Source collection adds participant confirmation of the same exact proposal and a public phase schedule. V2 and V3 therefore use different source-collection procedures; the paired algorithm comparison uses the shared V3 sources.

Interaction sources. Four demands, two models, and two separately generated interactions per combination produce 16 M sources. Requested models are `claude-opus-5` and `gpt-6-astra`, both with medium reasoning effort. Outputs are labelled by requested model; the returned Claude metadata also records Haiku usage. Eight control sources add one supported S task and one resource- or permission-constrained U task per demand, alternating the two models. Each source permits six calls: at most three alternating role turns, followed by participant confirmation of the latest unrejected proposal. Consent requires agreement on that exact proposal. Ordinary completion, refusal, and unconfirmed proposals are separate outcomes. Controls are reported separately from the M algorithm comparison.

Paired synthesis. Both algorithms receive the same confirmed proposal, interaction, and public interface. Sequential feedback produces an initial candidate and two revisions using development-test feedback. Independent sampling produces three candidates from fresh contexts containing the same source material, without validator feedback. Both algorithms have a three-call synthesis budget, the same requested model and reasoning effort, and identical rule language, installation permissions, and validator. Model-generated rules and guards are evaluated as generated. A shared development ranking selects one candidate: admitted candidates precede type-passing, type-pending, type-failing, and absent candidates. Ties use passed gates, unique passed development cases, and earlier candidate order. Pending candidates remain undeployed. Held-out outcomes do not enter candidate selection.

Temporal tests and historical revision. Development uses bindings 0 and 1; held-out evaluation uses bindings 2000–2019 and separate action-order cases. D3 specifies cancellation before work completion and preservation of the worker's settlement right after completion. Custody return requires the holder's authenticated action. The completion–cancellation–settlement counterexample enters development (Appendix D.10). Held-out cases add repetitions, mixed identities, completion after cancellation, and replay after settlement. D4 tests custody-time boundaries, unauthorized calls, and replay at expiry. The two historical D3/D4 candidates receive these checks. Four D3 revision runs

cross two models with two repeats, each allowing three feedback calls. Historical revisions have a separate denominator of four.

Combined-library checks. Admission evaluates each candidate with the current rule library. For each algorithm and demand, the first admitted source in a fixed ordering supplies a candidate for ordered cross-demand pairings. Each pairing checks both tasks under the combined library at binding 2000. Selection uses admission alone. Candidates whose rule names conflict cannot be jointly installed. The comparison reports the number of available pairings, including zero.

Later-task planning. Each M source receives the two task variants from Appendix D.5. The design contains 32 shared R0 tasks and 32 post-admission tasks per algorithm: 96 distinct tasks in total. Tasks without an available library remain in the denominator. Each executable task permits two calls under the same permissions. A fresh planner observes the available rules and chooses a setup–progress–finish or setup–progress–cancel sequence. If it does not select a branch, the full proposed action sequence is evaluated and marked ambiguous.

Semantic scoring. Goal satisfaction uses the prespecified task predicates, intermediate custody checkpoints, and final state. D3 completion evidence must match a job arranged during setup. Rule use counts only successful execution of installed added rules. Refusal and task completion are scored separately. Added-guard protection requires a valid action that succeeds without the added guard under the same actor and permissions, but whose effect violates an independent contract or scope predicate. This counterfactual replay changes no actual state. Refusal of an otherwise permitted rest action is unexpected; refusals without a conclusive semantic classification remain unknown. Invalid action arguments and state changes during refusal are scored separately. Safety checks cover invariants, resource conservation, and contract predicates. The combined execution-and-goal score is retained as a secondary measure.

Denominators. Each algorithm retains all 16 M sources; each model–algorithm cell contains eight. Measures count confirmed proposals, admissions, and selected candidates passing every held-out check. A complete acquisition-to-use task requires admission, all held-out checks passing, actual added-rule use, goal satisfaction, and safety. Qualifying tasks are reported out of 32; sources with at least one qualifying task (any-task) and with both tasks qualifying (both-task) are each reported out of 16. Unavailable and unknown outcomes remain in the denominators. Both source summaries were specified after protocol freeze and before deployment or reuse evaluation. The eight controls and four historical revisions are separate analyses. Deterministic bindings define stress instances; workload seeds initialize the simulator.

Initial and repeat evaluations. The initial evaluation completed part of the planned synthesis calls. Before inspecting comparative scores, we specified a complete repeat of all 32 strategy conditions: both algorithms on all 16 M sources, with the same three-call budget and unchanged evaluation protocol. The two evaluations share the 24 interaction sources, four historical revision runs, and 32 R0 tasks. The repeat generates new candidates, deployments, combined-library checks, and post-admission task outcomes. Results retain every planned strategy condition and are reported separately for the two evaluations.

D.12 Revision of the historical settlement rule

All four selected D3 revisions passed admission and all 320 held-out checks per candidate. Each run received the same historical rule and development feedback under a three-call budget. The held-out suite contains 16 check classes at each of 20 bindings. Its three temporal classes exercise completed work followed by mixed-identity cancellation retries, cancellation followed by completion, and mixed replays after settlement. The resulting trajectories show that valid cancellation before completion succeeds; completed work retains an executable settlement path; repeated terminal calls preserve the settled state.

Historical candidates under the new checks. The saved V2 D3 candidate passed 32/34 development/admission cases and was rejected. The completion–cancellation–settlement case failed at both bindings: cancellation closed the contract and blocked the worker's earned settlement. These failures

triggered the constraint and reuse gates. The saved D4 candidate passed 28/28 development/admission cases and remained admitted. These are two historical-candidate checks, separate from the four model revisions and their held-out results.

Preserving earned settlement rights. The two selected GPT revisions add a requirement inside the cancellation rule. Cancellation is permitted only while the conjunction of fresh job-receipt count, matching worker, and receipt date at or after registration is false. The existing five guards remain unchanged. Claude run 1 adds the job-count and receipt-date condition to both a cancellation guard and the rule itself. Its settlement rule also checks the worker's completion count against its registration baseline. Claude run 2 preserves a receipt-backed settlement entitlement when cancellation follows completed work. Settlement is permitted while the agreement is open or the entitlement remains outstanding; settlement verifies the receipt and discharges that entitlement.

Table 24: Selected outputs from four revisions of the same historical D3 candidate. Each run has a three-call budget.

Run	Selected call	Executable protection	Held-out
Claude 1	3	Cancel guard and in-rule receipt check	320/320
Claude 2	1	Receipt-derived entitlement survives cancellation	320/320
GPT 1	1	In-rule cancellation requirement	320/320
GPT 2	1	In-rule cancellation requirement	320/320

Candidate selection. Selected model-generated programs are evaluated and deployed as generated. Claude run 1 produces an admitted revision on its third call; its first two calls yield an invalid program and a rejected candidate, respectively. The other three runs select their first candidate under the same development ranking. The four historical revisions remain separate from acquisition through new interaction sources.

D.13 V3 RESULTS: ACQUISITION AND LATER USE

Sequential feedback admitted candidates in 6/16 source episodes in the repeat evaluation, and all six selected candidates passed every held-out check. Independent sampling admitted candidates in 0/16 episodes under the same three-call budget and installation permissions. Feedback completed the acquisition-to-use chain for 4/16 sources on at least one later task, 1/16 on both tasks, and 5/32 task slots (Table 25).

Sources and paired synthesis. The shared source stage produced confirmed proposals for 13/16 M sources. The other three comprised two ordinary completions and one ordinary attempt whose goal was failed or unverified; all three remained ineligible for synthesis. In the four S controls, existing actions completed three tasks and one source produced a confirmed proposal. The four U controls produced two refusals and two confirmed proposals. Controls retain their separate source outcomes. Both algorithms therefore synthesized on the same 13 eligible M sources while retaining the denominator of 16.

In the repeat evaluation, feedback admitted 6/8 sources in the GPT-6 Astra arm and 0/8 in the Claude Opus 5 arm. The initial evaluation admitted 5/8 and 1/8, respectively. Repeat-evaluation admissions were 2/4 each for D1, D3, and D4, and 0/4 for D2; initial counts were 3/4, 1/4, 2/4, and 0/4 in the same demand order. Each evaluation rejected all 13 eligible independent-sampling episodes at development contract checks. The initial feedback evaluation also admitted six candidates that passed all held-out checks; its later-use counts were 4/16 any-task, 3/16 both-task, and 7/32 tasks. Across all 16 paired sources, six admission outcomes favor feedback, zero favor sampling, and ten admit neither, in each evaluation. Five source identities are admitted in both evaluations; D1/Claude/run 2 is admitted only in the initial evaluation, and D3/GPT/run 1 only in the repeat evaluation. The two evaluations share sources and remain separate comparisons.

Semantic discrimination by the admission tests. Development tests distinguish syntactically valid candidates by their compliance with the intended agreements. All 39 independent-sampling

candidates passed type checks, but all were rejected by development contract tests. All 13 selected independent-sampling candidates failed constraint and reuse; 12 of 13 also failed state and conflict. These gates share evidence and cannot be added as independent failures. One D1 candidate ties repayment to a fixed debtor identity and cannot transfer to a different debtor. A D2 candidate completes the cooperative purchase but permits pledged gold to leave escrow through an ordinary resource transfer. A D3 candidate does not establish the promised asset reservation, preventing subsequent settlement. Feedback admitted selected candidates in six source episodes; its two GPT D2 selections were rejected for the same gold-transfer bypass. Two unselected Claude feedback candidates received no completed admission decision; they remain pending and are not classified as contract rejections.

Table 25: Paired synthesis and complete acquisition-to-use outcomes. A: admitted; H: admitted and every held-out check passed. Any and Both count sources; Tasks counts qualifying later tasks. Each evaluation retains all planned denominators.

Evaluation	Algorithm	A	H	Any	Both	Tasks
Repeat	Feedback	6/16	6/16	4/16	1/16	5/32
Repeat	Sampling	0/16	0/16	0/16	0/16	0/32
Initial	Feedback	6/16	6/16	4/16	3/16	7/32
Initial	Sampling	0/16	0/16	0/16	0/16	0/32

Task completion, activation, and unavailable slots. Table 26 separates goal satisfaction from safety, actual rule activation, and combined execution-and-goal scores. A qualifying task requires admission and every held-out check to pass, followed by goal satisfaction, passing safety checks, and a successful added-rule call. Both feedback evaluations executed 12 later-task slots. The remaining 20 had no admitted library and retain NA status, with goal, safety, and strict scores unknown. Independent sampling retained all 32 slots with the same NA status. These slots earn no complete-chain credit; their unknown task outcomes remain distinct from measured failures. In the repeat evaluation, feedback recorded four measured goal failures, compared with three in the initial evaluation.

Table 26: Later-task measures, with 32 planned slots per row. Use: at least one successful added-rule invocation; Chain: qualifying complete acquisition-to-use task; Strict: combined execution-and-goal score. NA counts unavailable-library slots. A dash denotes an unavailable or inapplicable measure. Sampling and R0 rows are identical across evaluations; R0 records are shared.

Evaluation/arm	Goal	Safety	Use	Chain	Strict	NA
Repeat/feedback	8/32	12/32	6/32	5/32	7/32	20
Initial/feedback	9/32	12/32	8/32	7/32	7/32	20
Both/sampling	—	—	0/32	0/32	—	32
Shared R0	26/32	30/32	0/32	—	17/32	0

The shared R0 arm completed 26/32 goals and passed safety checks in 30/32 slots. Four goal values and two safety values remain unknown; two goals were measured failures. Its strict score was 17/32. R0 measures task completion with existing actions, and the acquisition-chain measure is inapplicable. In the repeat evaluation, feedback recorded 14 unknown refusal events; the initial evaluation recorded eight unknown events and one independently verified correct refusal. Both recorded zero verified added-guard refusals, unexpected refusals, unsafe refusals, and invocation-schema failures.

In the second GPT interaction for D4, the selected custody-rotation rule completes activation, progression, and closure under a changed-time task with interference. Custody moved $a_0 \to a_1 \to a_2 \to a_0$ across days 0–2; the contract closed, the item lock cleared, and both goal and safety passed.

Combined libraries and historical revision. Each evaluation selected three admitted feedback candidates, one each for D1, D3, and D4, yielding six ordered pairings. Both tasks passed all checks in every pairing at binding 2000: 152/152 cases per evaluation. Independent sampling supplied no admitted pairings. The four historical revisions also passed admission and all 320 held-out checks per candidate (Appendix D.12).

D.14 INSTALLED MECHANISM DIFFERENCES IN V3

Comparing the initial and acquired rule libraries identifies the shared executable relations added by the six repeat-evaluation admissions (Table 27). Each retains the 17 initial rule definitions and adds rules governing the formation, fulfillment, and termination of agreements. Initial borrowing and return move custody; the D1 rules additionally couple credit payment or bread return to custody release. Initial job completion pays escrow; the D3 rules additionally condition ownership settlement and cancellation on the specified job's trusted receipts. The D4 rules connect custody changes to stored phases and dates. Four bundles also register guards on initial actions; D1/r1 and D4/r1 register none. The runtime evaluates registered guards before dispatching their target actions for every caller of the shared library.

Table 27: Acquired relations for the six selected feedback candidates in the repeat GPT evaluation. Rows identify demand and source repeat. R/G counts added rules and guards implementing the three demand relations. The 17 initial rule definitions remain unchanged in every row.

Source	R/G	Acquired relation	Initial actions with added guards
D1/r1	3/0	Credit status and asset reservation condition custody release on payment or bread return.	None
D1/r2	3/7	Secured-credit status and asset reservation govern payment, return, and closure.	Gathering, giving, listing, borrowing, buying, returning, hiring
D3/r1	4/6	A promise links the asset and borrowing agreement to completed work and ownership settlement.	Giving, listing, buying, borrowing, hiring, completing work
D3/r2	3/7	The asset promise makes authenticated work completion govern ownership settlement and cancellation.	Borrowing, hiring, completing work, listing, giving, buying, returning
D4/r1	4/0	Agreement phase and current custodian govern dated handovers while ownership remains fixed.	None
D4/r2	4/8	Agreement phase, custodian, and asset reservation govern dated handovers and closure.	Giving, borrowing, listing, buying, returning, hiring, completing work, gathering

Both D3 candidates change whether an otherwise valid gift is permitted. After an ordinary return, the recorded item has owner and holder a_0, no lock, and colocated participants; the authenticated owner consents to giving the item. This state satisfies the initial gift rule's ownership, possession, reservation, location, and consent conditions. The installed promise guard reads the added promise relation and refuses the gift, leaving the world state unchanged. This connects the classification in Appendix A.19 to a concrete change in executable permissions. Guard counts include installed conditions that overlap with existing checks.

D.15 V4: EQUAL-FEEDBACK SEARCH AND SOURCE CONSTRUCTION

V4 tests executable mechanism acquisition across 16 paired world–model blocks: four demands, two requested models, and two repetitions. Each block compares multi-role drafting (I) with centralized drafting (D), and compares two synthesis procedures on the same I source. Table 28 specifies the three conditions.

Source construction. Both source arms receive the same participant facts, preferences, goal, and initial interface, together with their own previous drafts. I rotates speaking roles across three calls; D uses three centralized drafting and self-revision calls. The third response fixes the proposal or ordinary plan. Each participant then accepts or refuses those exact terms on the basis of its own observation and preference. D1 has two participants; D2–D4 have three. Confirmation cannot change terms or execute actions. Eligibility requires a shared-relation proposal and unanimous confirmation. Ordinary plans require the same confirmation and execute through the existing authenticated action interface. Source outcomes distinguish eligible proposals, ordinary plans, refusals, and unavailable or invalid proposals. Both I synthesis conditions receive the same confirmed proposal.

Table 28: V4 budgets per block. A source with a final proposal or ordinary plan adds one confirmation call per participant. I is generated once and shared by its two conditions. Synthesis requires source eligibility; ineligible blocks remain in planned denominators.

Condition	Source drafting	Synthesis requests	Later calls/task
I–sequential	Three role turns	6	≤ 2
I–pool	Same I proposal	6	≤ 2
D–sequential	Three central drafts	6	≤ 2

Search and selection. The procedures share the bounded rule language, 17 initial actions, installation permissions, development checker, and six request and validation slots. Sequential search revises the previous candidate in calls 2–6. Pool search drafts two candidates independently, revises each, then revises the best two among calls 1–4; call 6 does not inspect call 5. When a scheduled parent is unavailable, search uses the best syntactically valid candidate in the fixed parent pool. A single pool parent can supply both revisions; an empty pool triggers a fresh draft from the original source. Every revision receives the same set of development-test outcomes for its parent candidate. All six slots remain scheduled after an early admission. Selection uses development results only, ordered by admission, type tier, passed gates, unique passed cases, and earliest call. Invalid candidate outputs consume their allocated attempts; pending candidates remain undeployed. The selected candidate, confirmed proposal, and initial and acquired rule libraries are fixed before held-out evaluation.

Held-out execution. Admitted selections undergo the standard suite at 20 role bindings and an independent semantic challenge at another 20. The challenge defines eight classes per demand, giving 160 cases per candidate. Case classes, sequence templates, and expected-state predicates are fixed before generation. Candidates are evaluated as selected. Executed actions, resulting world states, and authenticated evidence of completed obligations determine pass, fail, or unknown. Selections without an admitted library receive NA for deployed tests. Bindings and cases index repeated checks of the selected candidate.

Later use and scoring. Each condition retains two later-task slots per block (32 total): a new-role binding and a schedule/interference variant. The latter shifts later phase times for D1–D3, preserves D4's specified custody calendar, and asks the planner to include an interfering ordinary action when available. The common R0 library supplies another 32 slots, assessed by goal and safety. Planners receive the actual library and task, with identical task permissions and standing offers across conditions. Execution allows one planning call and at most one required-party confirmation, preserving the V3 executor and refusal classifications. A complete chain requires confirmed source eligibility, admission, every standard and semantic check passing, goal and safety both true, and a successful added-rule invocation. Reported denominators are tasks/32, sources with at least one complete task/16, and sources with both/16. Missing libraries remain NA within planned denominators. Goal and safety also use matched available pairs with explicit false and unknown outcomes.

Experimental controls and analysis. Primary comparisons use admitted-and-all-held-out-pass sources/16, discordant paired counts, paired differences, and exact two-sided sign tests; Wilson intervals describe marginal proportions. Results are also grouped by requested model and demand. Generation requests Claude Opus 5 or GPT-6 Astra with medium reasoning effort, using a fresh context without tools for each call. Workload seeds initialize the world scenarios; model attribution and sampling controls follow Appendix D.11. The same input and time limits apply to all attempts. Unavailable or invalid responses consume their allocated budgets without additional attempts. The protocol, experimental conditions, validation criteria, and evaluation inputs are fixed before generation. Results are reported in Appendix D.16.

D.16 V4: SEARCH, SOURCE, AND SEMANTIC OUTCOMES

I–Seq admitted selected bundles for 11/16 source blocks, with 6/16 passing both held-out suites; I–Pool admitted selected bundles for 6/16 source blocks, with 3/16 passing both held-out suites; D–Seq admitted selected bundles for 9/16 source blocks, with 4/16 passing both held-out suites. Complete acquisition-to-use chains occurred in I–Seq 8/32, I–Pool 4/32, D–Seq 8/32.

I–Seq and I–Pool share each interaction source; D–Seq uses centralized source construction. Each condition retains 16 world/model blocks (eight per model, four per demand) and 32 later-task slots. A chain requires a confirmed eligible source, admission, both held-out suites passing, goal and safety, and successful invocation of an added rule. Unavailable libraries retain NA; a false chain predicate is not an observed task failure.

Table 29: V4 outcomes. Admission and suite denominators are sources; chain slots are tasks; any/both-task columns are sources.

Condition	Admit	Both suites	Chain slots	Any task	Both tasks
I–Seq	11/16	6/16	8/32	4/16	4/16
I–Pool	6/16	3/16	4/32	2/16	2/16
D–Seq	9/16	4/16	8/32	4/16	4/16

Marginal 95% Wilson intervals for both-suite source proportions: I–Seq [0.185, 0.614]; I–Pool [0.066, 0.430]; D–Seq [0.102, 0.495].

Selection statuses (admitted/rejected/pending/no eligible source): I–Seq 11/3/0/2; I–Pool 6/8/0/2; D–Seq 9/4/1/2.

Table 30: Results by requested model. Every planned block is retained.

Group	Condition	Admit	Both suites	Chain slots	Any task	Both tasks
Claude Opus 5	I–Seq	4/8	3/8	4/16	2/8	2/8
Claude Opus 5	I–Pool	4/8	3/8	4/16	2/8	2/8
Claude Opus 5	D–Seq	5/8	2/8	4/16	2/8	2/8
GPT-6 Astra	I–Seq	7/8	3/8	4/16	2/8	2/8
GPT-6 Astra	I–Pool	2/8	0/8	0/16	0/8	0/8
GPT-6 Astra	D–Seq	4/8	2/8	4/16	2/8	2/8

Table 31: Results by demand. Every planned block is retained.

Group	Condition	Admit	Both suites	Chain slots	Any task	Both tasks
D1	I–Seq	3/4	1/4	2/8	1/4	1/4
D1	I–Pool	3/4	0/4	0/8	0/4	0/4
D1	D–Seq	2/4	1/4	2/8	1/4	1/4
D2	I–Seq	2/4	0/4	0/8	0/4	0/4
D2	I–Pool	0/4	0/4	0/8	0/4	0/4
D2	D–Seq	1/4	0/4	0/8	0/4	0/4
D3	I–Seq	3/4	2/4	0/8	0/4	0/4
D3	I–Pool	2/4	2/4	2/8	1/4	1/4
D3	D–Seq	3/4	0/4	0/8	0/4	0/4
D4	I–Seq	3/4	3/4	6/8	3/4	3/4
D4	I–Pool	1/4	1/4	2/8	1/4	1/4
D4	D–Seq	3/4	3/4	6/8	3/4	3/4

Observed paired differences (A minus B): I–Seq/I–Pool 0.1875 (16/16); I–Seq/D–Seq 0.125 (16/16).

Added rules were successfully invoked in 18/32 I–Seq tasks (50 invocations), 10/32 I–Pool tasks (24 invocations), and 16/32 D–Seq tasks (50 invocations). Invocation counts were observed for every task.

Table 32: Source construction outcomes over 16 world–model blocks. Ordinary plans achieve the goal using the initial action library.

Source	Eligible proposals	Ordinary plans achieving the goal
I	14/16	2/16
D	14/16	2/16

Table 33: Component outcomes as true/false/unknown. Each row accounts for all 16 sources; unevaluated NA components remain within unknown.

Condition	Admit	Standard	Semantic	Both
I–Seq	11/5/0	11/0/5	6/5/5	6/10/0
I–Pool	6/10/0	6/0/10	3/3/10	3/13/0
D–Seq	9/7/0	9/0/7	4/5/7	4/12/0

Table 34: Exact paired comparisons of admission plus both suites, over 16 planned source pairs. Only discordant observed pairs enter the two-sided sign test; unknown pairs remain listed.

A/B	Both	A only	B only	Neither	Unknown	p
I–Seq/I–Pool	2	4	1	9	0	0.375
I–Seq/D–Seq	3	3	1	9	0	0.625

Table 35: Later-task components as true/false/unknown, each over 32 planned slots. Execution+goal additionally requires all attempted actions to succeed. NA means no admitted library; no-library NA and other unknowns are not measured task failures.

Condition	Goal	Safety	Goal+safety	Execution+goal	NA
I–Seq	16/6/10	22/0/10	16/6/10	11/11/10	10
I–Pool	8/4/20	10/2/20	8/4/20	5/7/20	20
D–Seq	15/3/14	18/0/14	15/3/14	7/11/14	14
R0	27/1/4	30/0/2	27/1/4	18/14/0	0

Table 36: Available-library matched slots against the shared R0, with identical source block and task index. Scores are true/false/unknown; full 32-slot totals remain above. Task equality is true/false/unknown.

Condition	Pairs	Acquired	R0	Same task
I–Seq	22	16/6/0	19/1/2	22/0/0
I–Pool	12	8/4/0	11/1/0	12/0/0
D–Seq	18	15/3/0	15/1/2	18/0/0

Table 37: Action-level validation and refusal judgments during later tasks. Counts are events, not independent sources; an unclassified refusal has no verified correctness judgment.

Action outcome	I–Seq	I–Pool	D–Seq	R0
Invalid action arguments	0	2	0	0
Verified unwarranted refusals	0	0	0	0
Unclassified refusals	25	8	14	7
Refusals violating safety	0	0	0	0
Verified added-rule enforcement	0	0	0	0
Verified correct refusals	0	0	0	4

Independent semantic outcomes. Each admitted selection undergoes eight semantic test classes at 20 bindings, giving 160 checks of one candidate. All scheduled semantic checks have observed outcomes; the following tables report passed/failed counts. Unadmitted selections receive NA because they supply no library for semantic evaluation. Failure categories may overlap within a case.

Table 38: Independent semantic test outcomes across all admitted candidates.

Condition	Passed	Failed	Total
I–Seq	1636	124	1760
I–Pool	840	120	960
D–Seq	1320	120	1440

Table 39: D1 semantic classes, passed/failed. Expected cases per class in I–Seq/I–Pool/D–Seq: 60/60/40.

Class	I–Seq	I–Pool	D–Seq
Third-party payments and debt discharge	60/0	60/0	40/0
Unauthorized closure followed by lawful repayment	60/0	60/0	40/0
Debt discharge after partial payment	60/0	60/0	40/0
Replay after disposal of redeemed collateral	20/40	0/60	20/20
Repeated payment after settlement	60/0	60/0	40/0
Attempted collateral disposal before repayment	60/0	60/0	40/0
Repayment with an unrelated loan in progress	20/40	0/60	20/20
Preservation of unrelated trades	60/0	60/0	40/0

Table 40: D2 semantic classes, passed/failed. Expected cases per class in I–Seq/I–Pool/D–Seq: 40/0/20.

Class	I–Seq	I–Pool	D–Seq
Settlement on day two followed by late cancellation	40/0	NA	20/0
Settlement and refund replay after funded expiry	40/0	NA	20/0
Repeated funding followed by completion	40/0	NA	20/0
Repeated refunds after partially funded expiry	40/0	NA	20/0
Attempted diversion of pledged funds before completion	20/20	NA	0/20
Preservation of unrelated income during refunds	36/4	NA	20/0
Settlement replay after disposal of a refunded asset	40/0	NA	20/0
Interleaved trading of unrelated assets	40/0	NA	20/0

Of 520 scheduled generation attempts, 518 returned interpretable responses: 240/242 for Claude Opus 5 and 278/278 for GPT-6 Astra. The two attempts without a usable response retain their allocated budgets; all 16 world–model blocks and all 32 later tasks per condition remain in the comparisons.

Table 41: D3 semantic classes, passed/failed. Expected cases per class in I–Seq/I–Pool/D–Seq: 60/40/60.

Class	I–Seq	I–Pool	D–Seq
Cancellation followed by unrelated and contracted work	60/0	40/0	60/0
Cancellation, return, and a new loan before completion	60/0	40/0	40/20
Repeated settlement after work completion	60/0	40/0	60/0
Replay after disposal of an earned asset	40/20	40/0	20/40
Earned rights after unrelated work and cancellation	60/0	40/0	60/0
Unrelated prior work followed by contracted work	60/0	40/0	60/0
Work for another employer followed by contracted work	60/0	40/0	60/0
Work by another worker followed by contracted work	60/0	40/0	60/0

Table 42: D4 semantic classes, passed/failed. Expected cases per class in I–Seq/I–Pool/D–Seq: 60/20/60.

Class	I–Seq	I–Pool	D–Seq
New borrowing after schedule expiry	60/0	20/0	60/0
Attempted disposal during the first custody interval	60/0	20/0	60/0
Replay after disposal of a returned asset	60/0	20/0	60/0
Gifting and replay during the second custody interval	60/0	20/0	60/0
Attempted reborrowing in the second custody interval	60/0	20/0	60/0
An unrelated loan across a custody interval boundary	60/0	20/0	60/0
Unrelated trades within each custody interval	60/0	20/0	60/0
An invalid recipient followed by the scheduled recipient	60/0	20/0	60/0

Table 43: Cases violating each semantic requirement; categories are not mutually exclusive.

Category	I–Seq	I–Pool	D–Seq
Forbidden transition accepted	20	0	20
Required legal action refused	100	120	80
Earned or acquired rights violated	0	0	20
Safety condition violated	20	0	40
Required state transition not achieved	4	0	0

Preserving an unrelated legal action. In the first D1 block, both selected I candidates passed all 200 standard checks, while the independent unrelated-loan challenge tested whether the acquired collateral mechanism preserved an existing legal action. The pool-selected candidate refused an authorized loan of an unencumbered tool because its collateral guard used a null lock identifier as a contract key, although the four-gold payment and collateral return still succeeded. The sequentially selected candidate from the same frozen source allowed the unrelated tool to move from its owner to the borrower and back, with authenticated borrowing and return receipts, and completed the original transaction.

The comparison uses the first D1 block in the Claude Opus 5 arm at binding 3000. In the recorded order, it is the first failed D1 case and the first passing case of the same challenge class. Authenticated receipts establish that the sequential candidate executes both borrowing and return; both traces transfer four gold and release the original collateral.

D.17 V4E: SHARED-ENFORCEMENT PROBES ON THE ADMITTED V4 LIBRARIES

Table 44: V4E: probe cases in which the violation succeeded, for the three discriminating probes; 20 bindings per library and block, admitted libraries pooled (count in parentheses). Guard decisive counts, out of all admitted cases, those whose refusal disappears when the installed guards are removed; the D2 probe runs after both contributions.

Probe (independent caller)	$\mathcal{R}_0$ comp.	Authored	Admitted (libraries)	Guard decisive
D1: creditor returns collateral unpaid	80/80	0/80	0/160 (8)	20/160
D2: contributor spends pledged gold	80/80	0/80	0/60 (3)	0/60
D3: promised asset gifted after return	80/80	80/80	0/160 (8)	160/160

V4E evaluates saved artifacts with no new model calls. It asks whether the relations acquired in V4 are enforced against callers who do not cooperate, measured on one instrument applied to every library: after a library's own lawful arrangement has been established, an independent caller attempts a base-action violation of the demand's obligation, and the outcome is classified by the resulting world state.

Arms. An arm is a library together with the plans that establish the lawful arrangement. The $\mathcal{R}_0$ composition uses the initial 17-action library with the authored native-action plans of the skill controller; the authored capability library installs the four authored reference candidates and uses their plans; each of the 26 admitted V4 selections installs its own candidate into the initial library and uses that candidate's plans. The two authored arms run on all 16 block fixtures; each acquired arm runs on its own block fixture. Every arm runs at bindings 3000–3019, bound exactly as in the semantic challenge, with two non-participants and a free asset added before any dispatch.

Probes and classification. Table 45 lists the nine probes. Each dispatches only base actions, so the identical probe is callable in every arm; approvals are presented for the acting principals only. A probe runs only after the arrangement predicate holds (D1: the creditor holds the collateral and the debtor has received three bread; D2: after both contributions, the pledge is held in escrow or under a lock or has left the contributors' balances; D3: the specified job has an authenticated completion; D4: the first holder has custody and the owner retains ownership). The violation predicate is evaluated before and after the probe. A case is *unprotected* when the predicate holds afterwards; *protected by guard* when it does not and at least one probe action was refused by an installed guard; *protected natively* when it does not and only initial preconditions refused, or nothing was refused. The remaining lawful phases are then executed, and a continuation counts as blocked only when an installed guard refuses a lawful action. The probes were authored after V4 generation and after the V4 results were observed, from the public D1–D4 contract clauses and the pre-existing V1 protection classes; they are a post-hoc instrument applied uniformly to fixed artifacts.

Results. Of 2,620 cases, none failed to establish its arrangement, none raised a host error, and none produced a runtime safety failure. Three probes discriminate between the arms. Under the $\mathcal{R}_0$ composition, the creditor's unpaid return, the contributor's transfer of pledged gold, and the owner's gift of the promised asset after the worker's return each succeed in 80/80 cases, and after the D2 and D3 violations the lawful endpoint is no longer reachable. Under the 26 admitted libraries no probe case succeeds. The decomposition shows what protects. For the D3 gift, an installed guard on gifting refuses the transfer in all 160 cases and is decisive in all 160: removing the guards lets the gift succeed. For the D1 return, 20 cases involve a base loan on the collateral, and the installed guard on return is decisive in those 20; in the other 140 cases the collateral is held under the candidate's own agreement, which no base action can release; the fallback transfer by the holder is then refused by an installed guard in 120 cases and by the initial ownership precondition alone in 20, and the ownership precondition would refuse it in all 140. For the D2 transfer, 20 cases are refused by a guard and 40 by the initial insufficient-funds precondition, because the acquired rule moved the pledged gold out of the contributor's balance. The authored capability library conditions disposal on the existence of a completion receipt and therefore permits the post-completion gift in 80/80 cases while every admitted library refuses it. No lawful continuation was refused by an installed guard in any arm.

Table 45: All nine V4E probes. Unprotected counts are violation successes over probe cases; the last three columns decompose the acquired arms: cases refused by an installed guard, cases refused by initial preconditions, and, out of all admitted cases, those in which removing the guards lets the violation succeed (decisive).

Probe	$\mathcal{R}_0$ comp.	Authored	Acquired	By guard	Native	Decisive
D1: creditor returns collateral unpaid	80/80	0/80	0/160	140/160	20/160	20/160
D1: owner lists and sells pledged tool	0/80	0/80	0/160	140/160	20/160	0/160
D2: contributor spends pledged gold	80/80	0/80	0/60	20/60	40/60	0/60
D2: seller gives away reserved asset	0/80	0/80	0/60	60/60	0/60	0/60
D3: owner disposes of promised asset while worker holds it	0/80	0/80	0/160	160/160	0/160	0/160
D3: after the worker's return, owner gifts promised asset	80/80	80/80	0/160	160/160	0/160	160/160
D4: owner sells during first interval	0/80	0/80	0/140	140/140	0/140	0/140
D4: first holder re-borrows in second interval	0/80	0/80	0/140	140/140	0/140	0/140
D4: second holder gifts asset	0/80	0/80	0/140	140/140	0/140	0/140

The six remaining probes are refused in every arm, including the $\mathcal{R}_0$ composition, because the initial library requires the actor to own an asset to give it and to hold an asset to list or lend it, and in the intended arrangements the owner and the holder differ. These probes measure initial capabilities: the guards that refuse them first are never decisive (0/160 for the D1 sale, 0/60 for the D2 seller disposal, 0/160 for the D3 disposal while the worker holds the asset, and 0/140 for each of the three D4 probes). The sealed semantic classes, which first attempt an early return to free the asset, test the D4 relations.

The D3 probe's first step is a lawful return that moves custody to the owner, and several acquired libraries settle title without redelivering custody, so their continuation does not reach the goal's holder; this is recorded as a failed continuation, not as a refusal. Grouped by V4 condition, unprotected counts are 0/500 for I–Seq, 0/260 for I–Pool, and 0/420 for D–Seq, and 0/660 for the 13 selections that passed both sealed suites. Binding repetitions are relabelings of one structural situation; the unit of the design is arm, block, and probe.

D.18 V5: INFORMATION LOCALITY OF PROPOSAL SOURCES

V4 held the facts available to the two source procedures equal: the multi-role arm and the centralized arm both received the union of all participants' observations and private preferences. V5 asks the question that motivates situated interaction: when participants' continuing requirements are private to the participants rather than enumerated for a designer, does multi-role interaction supply confirmed proposals and admitted rules that centralized drafting without those requirements cannot? Three source arms separate information locality from source structure on the same blocks.

Design. Four demands, two requested models (Claude Opus 5 and GPT-6 Astra, medium effort), and two repetitions give 16 blocks with new workload seeds. Every arm uses the V4 drafting structure: three drafting calls, then one terminal confirmation call per participant on the exact final proposal, using that participant's own observation and private preference. In the *private multi-role* arm, the speaking role rotates over the three drafting turns and each turn sees the public goal and interface, the speaking role's own observation and preference, and the prior public trace of earlier roles' statements and proposals. In the *public-facts centralized* arm, one designer sees the public goal and interface and all participants' observations, and no preference. In the *full-information centralized* arm, the designer additionally sees all preferences; this arm is identical to the V4 centralized arm. Eligibility requires a shared-relation proposal confirmed by every participant; ordinary plans require the same confirmation and execute through the existing authenticated interface. Every eligible source is formalized by the V4 sequential-feedback search with six request slots and development-only selection, and admitted selections receive the unchanged standard suite and the unchanged sealed semantic challenge. Per arm, the outcomes are eligible confirmed proposals, admitted selections, and selections passing both suites, each out of 16 blocks, with discordant paired counts and exact two-sided sign tests for the private multi-role versus public-facts centralized comparison, the full-information versus public-facts centralized comparison, and the private multi-role versus full-information comparison. Confirmation

refusals are tabulated by refusing role and named clause. The protocol, configuration, fixtures, and public context are hashed before the first call; no result-dependent change is permitted.

Protocol revision 1.1. All eight public-facts blocks whose requested model was Claude Opus 5 had a drafting call rejected by a provider content filter, so those blocks recorded a service outcome rather than the behaviour of a designer lacking the participants' requirements. A revision appended to the protocol before any repeat call, and without knowledge of any repeat outcome, defines a predeclared repeat of exactly those eight blocks on the same seeds, fixtures, participants, bindings and budget. The revision fixes in advance the only admissible change to the prompt, a single paragraph prepended to the instructions that restates mechanisms the original protocol already specifies and adds no fact, observation, preference, or hint; a block for which no response is obtained keeps its original outcome. The original results are reported unchanged alongside the repeat.

Results. V5 results are reported in Appendix D.19.

D.19 V5 RESULTS

Private multi-role interaction reached the full-information designer's rates on every outcome. Over 16 blocks per arm, the private multi-role arm produced confirmed shared-relation proposals in 14/16 blocks, admitted selections in 11/16, and selections passing both sealed suites in 7/16; the full-information centralized arm produced 14/16, 8/16, and 7/16; the public-facts centralized arm produced 5/16, 2/16, and 0/16 (Table 46). Ordinary plans confirmed by every participant and achieving the goal with the initial library occurred in 2/16, 3/16, and 2/16 blocks. No confirmation call in any arm returned a refusal; blocks without an eligible proposal ended either in a confirmed ordinary plan or, in the public-facts arm, without a parsed final draft. Over eligible confirmed proposals, the counting unit of Section 7, the proposal-to-rule yields are 11/14, 2/5, and 8/14 for multi-role, public-facts, and full-information drafting.

Table 46: V5 outcomes per source arm over 16 planned blocks. Eligible: shared-relation proposal confirmed by every participant; Admitted: selected candidate admitted; Standard: admitted and every standard held-out check passed; Both suites: admitted and both the standard suite and the sealed semantic challenge passed. Brackets give 95% Wilson intervals.

Arm	Eligible	Admitted	Standard	Both suites
Private multi-role	14/16 [0.64, 0.97]	11/16 [0.44, 0.86]	11/16	7/16 [0.23, 0.67]
Public-facts centralized	5/16 [0.14, 0.56]	2/16 [0.04, 0.36]	2/16	0/16 [0.00, 0.19]
Full-information centralized	14/16 [0.64, 0.97]	8/16 [0.28, 0.72]	7/16	7/16 [0.23, 0.67]

Service condition in the public-facts arm. Of 452 recorded calls, 440 parsed. All eight public-facts centralized sources in the Claude Opus 5 arm ended with the third drafting call refused by the provider's safety filter (10 of that arm's 24 drafting calls were refused; the same model completed all 92 drafting and confirmation calls of the private multi-role and full-information arms, and GPT-6 Astra completed all 264 of its calls). Those eight blocks therefore record no final proposal and are retained in the denominators as service outcomes rather than as decisions of a designer lacking the participants' requirements. Two Claude Opus 5 synthesis calls in the private multi-role arm returned malformed output and occupy their slots. Comparisons that involve the public-facts arm are reported over all 16 planned blocks, as specified.

Reading. With requirements private to the participants, the multi-role arm recovered the full-information rate on every outcome (Table 47: $p = 1.000$, 0.508, and 1.000), which replicates the V4 finding on new blocks and locates the contribution of interaction in elicitation. A designer given public facts alone reached fewer confirmed proposals and admitted rules. Over all 16 planned blocks, eight of them service outcomes in the public-facts arm, the both-suite sign tests of the private multi-role and full-information arms against the public-facts arm give $p = 0.016$. Three of the eight GPT-6 Astra public-facts sources ended in a confirmed ordinary plan that achieved the goal with the initial library, that is, the designer without the requirements chose an arrangement that the participants accepted but that added no shared relation. The demand breakdown mirrors V4: D4 admissions

Table 47: Exact paired comparisons over 16 planned blocks. I–priv: private multi-role; D–pub: public-facts centralized; D–full: full-information centralized. Only discordant observed pairs enter the two-sided sign test; no pair is unknown.

Blocks	A/B	Outcome	Both	A only	B only	Neither	p
All 16	I–priv / D–pub	Eligible	5	9	0	2	0.004
All 16	I–priv / D–pub	Admitted	0	11	2	3	0.023
All 16	I–priv / D–pub	Both suites	0	7	0	9	0.016
All 16	D–full / D–pub	Eligible	5	9	0	2	0.004
All 16	D–full / D–pub	Admitted	1	7	1	7	0.070
All 16	D–full / D–pub	Both suites	0	7	0	9	0.016
All 16	I–priv / D–full	Eligible	12	2	2	0	1.000
All 16	I–priv / D–full	Admitted	5	6	3	2	0.508
All 16	I–priv / D–full	Both suites	3	4	4	5	1.000

Table 48: V5 outcomes by requested model (8 blocks per cell).

	Eligible		Admitted		Both suites	
Arm	Claude	GPT	Claude	GPT	Claude	GPT
I–priv	6/8	8/8	5/8	6/8	2/8	5/8
D–pub	0/8	5/8	0/8	2/8	0/8	0/8
D–full	6/8	8/8	3/8	5/8	3/8	4/8

Table 49: V5 outcomes by demand (4 blocks per cell).

	Eligible				Admitted				Both suites			
Arm	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4
I–priv	4/4	4/4	4/4	2/4	4/4	1/4	4/4	2/4	3/4	0/4	2/4	2/4
D–pub	1/4	2/4	2/4	0/4	0/4	2/4	0/4	0/4	0/4	0/4	0/4	0/4
D–full	2/4	4/4	4/4	4/4	2/4	1/4	1/4	4/4	2/4	0/4	1/4	4/4

concentrate in the centralized arm with full information (4/4) and D2 remains the hardest demand for every arm.

Sealed semantic outcomes. The failing semantic classes of the admitted selections repeat those of V4: replay after disposal of redeemed collateral and repayment with an unrelated loan open (D1, 20/80 cases each in the private multi-role arm), disposal of an earned asset followed by replay (D3, 40/80), and the D2 refund and expiry classes, on which the single admitted private multi-role D2 candidate failed five classes, the two admitted public-facts D2 candidates failed the refunded-asset class in 40/40 cases, the unrelated-trade class in 20/40, and the unrelated-income class in 4/40, and the single full-information D2 candidate failed 4/20 cases in each of four classes. Unadmitted selections receive no semantic evaluation and are not counted as failures.

Predeclared repeat of the refused blocks. Under the predeclared revision all eight blocks produced a final designer response, none retaining the original service outcome. Table 50 gives the outcomes against the original outcomes of the other two arms on the same eight blocks. Combining these eight public-facts cells with the eight that completed in the original run gives a synthetic 16-block public-facts column of 10/16 eligible, 5/16 admitted, and 1/16 passing both suites; against it the multi-role arm's 14/16, 11/16, and 7/16 give paired differences of $+0.250$, $+0.375$, and $+0.375$ with exact two-sided sign tests of $p = 0.125$, $p = 0.146$, and $p = 0.070$. That column is synthetic, its cells coming from two runs under two protocol versions. V6 measures the same contrast at 64 blocks on one model with no refused call and finds it on every outcome, 62/64, 54/64 and 29/64 against 37/64, 16/64 and 2/64 ($p < 0.0001$; Appendix D.21).

Table 50: The eight refused public-facts blocks repeated under protocol revision 1.1, against the original outcomes of the other two arms on the same blocks.

Outcome (of 8)	D–pub, repeat	I–priv	D–full
Eligible confirmed proposals	5	6	6
Ordinary plans achieving the goal	2	2	2
Admitted selections	3	5	3
Passing both sealed suites	1	2	3

D.20 V6: A PROSPECTIVE SCALED STUDY OF ELICITATION, FEEDBACK, ENFORCEMENT, AND LATER USE

V6 repeats the V5 source design and the V4 synthesis, admission, sealed evaluation, and later-use stages on 64 new blocks with one requested model, adds the paired independent-sampling comparison at the V4 budget, and seals the enforcement probes together with the semantic challenge before any candidate is generated. Four hypotheses are recorded in the protocol before the first call. *Elicitation:* private multi-role drafting reaches the rate of full-information centralized drafting (paired difference within ± 0.10 on the both-suite endpoint) and exceeds public-facts centralized drafting on confirmed proposals, admitted selections, and selections passing both sealed suites, with the both-suite count per block as the primary endpoint. *Feedback:* on the same sources and the same six-request budget, sequential feedback admits more selections than independent sampling. *Enforcement:* on the probes that the initial library leaves unprotected, every admitted library refuses the violation in every case, the initial composition permits it in every case, and the authored capability library permits the post-completion gift. *Later use:* planners that see only the installed library invoke the added rules, and complete chains occur, in every source arm.

Design. Four demands with 16 repetitions each give 64 blocks with new workload seeds, all requesting GPT-6 Astra at medium effort. The three source arms are those of V5 and are unchanged: private multi-role drafting with a rotating speaking role that sees only its own observation and preference alongside the public goal, the public interface, and the prior public trace; public-facts centralized drafting with all observations and no preference; and full-information centralized drafting with all observations and all preferences. Each arm uses three drafting calls followed by one terminal confirmation call per participant on the exact final proposal, judged from that participant's own observation and private preference. Every eligible source in every arm is formalized by sequential-feedback search with six request slots and development-only selection. Every eligible multi-role

source is additionally formalized by independent sampling: six identical prompts from the same frozen source card, no feedback between them, and the same development-only selection key, so that the two strategies on one source form a pair at an equal budget. Selection is sealed before any held-out evaluation. Admitted selections receive the unchanged standard held-out suite and the unchanged sealed semantic challenge of eight classes per demand. The nine enforcement probes are applied at the sealed bindings to every admitted library after that library's own lawful arrangement, and to the initial composition and the authored capability library on the same block fixtures; probes that the initial composition leaves unprotected are the discriminating ones. Later use repeats the V4 procedure: two later tasks for each admitted selection of the three sequential-feedback conditions, planned by a fresh planner that sees only the installed library, with the same tasks under the initial library as the control for every block.

Analysis. Per arm, out of 64 planned blocks: eligible confirmed proposals, ordinary plans achieving the goal, admitted selections, selections passing all standard checks, and selections passing both suites, each with a Wilson 95% interval. The three arm comparisons are paired by block and reported as discordant counts with exact two-sided sign tests on eligible, admitted, and both-suite outcomes; the multi-role versus full-information comparison additionally reports a paired difference with a 95% interval, a Wilson interval on the multi-role-only share of discordant pairs rescaled by the discordant fraction. Feedback and sampling are compared by sign test over the multi-role sources that produced an eligible proposal. Enforcement is reported per probe and library kind as the cases in which the violation succeeded, with subtotals over the discriminating probes and over all nine, and with counts of cases in which an installed guard is decisive. Later use is reported per arm as invoked added rules and complete chains, both out of 128 planned tasks, with matched pairs to the control by block and task. Unknown and unavailable outcomes stay in the planned denominators and are not counted as failures; each p is a per-comparison value, and every prespecified endpoint is reported whatever its direction. The protocol, configuration, fixtures, public context, semantic challenge, and probe definitions are hashed before the first call, and no result-dependent change is permitted. Transport failures are re-dispatched at most twice, so recorded calls exceed request slots; a response that parses but is invalid is the slot's outcome, and a provider refusal is not retried.

Results. V6 results are reported in Appendix D.21.

D.21 V6: RESULTS

Elicitation. Private multi-role drafting produced a confirmed shared-relation proposal in 62 of 64 blocks, full-information centralized drafting in all 64, and public-facts centralized drafting in 37 (Table 51). In 26 of its remaining blocks, all 16 D4 blocks and 10 of the 16 D1 blocks, the public-facts designer achieved the goal with an ordinary action plan over the existing interface and proposed no relation; the full-information designer, given the same observations and the participants' requirements, proposed a relation in every block. In one block a participant rejected the final proposal at the confirmation call. Paired by block, private multi-role drafting exceeds public-facts drafting on all three compared outcomes (both suites, 27 discordant blocks to 0, $p < 0.0001$; admitted, 40 to 2, $p < 0.0001$; eligible, 25 to 0, $p < 0.0001$), and is not separated from full-information drafting on the primary both-suite endpoint (12 discordant blocks to 7, paired difference $+0.078$ with 95% interval $[-0.053, +0.183]$, $p = 0.36$) or on eligibility (0 to 2, paired difference -0.031 with interval $[-0.031, +0.010]$, $p = 0.50$); on admitted selections it is higher by interval but not by sign test (12 to 4, paired difference $+0.125$ with interval $[+0.003, +0.199]$, $p = 0.077$). The preregistered prediction for the multi-role versus full-information comparison was a paired difference within ± 0.10 on the both-suite endpoint; the point estimate, $+0.078$, lies within that margin and the interval extends to $+0.183$. Full-information drafting likewise exceeds public-facts drafting on all three outcomes (both suites, 23 discordant blocks to 1; admitted, 35 to 5; eligible, 27 to 0; $p < 0.0001$ each). Table 52 gives every pair count.

Feedback against independent sampling. Sequential feedback admitted 54 selections and independent sampling none on the 62 multi-role sources that produced an eligible proposal, both strategies receiving the same frozen source card and the same six request slots; the 54 discordant blocks all favour feedback ($p < 0.0001$). On the both-suite endpoint the counts are 29 and 0 ($p < 0.0001$). The sampled candidates were not malformed: every one of the 372 sampled slots returned a well-formed

Table 51: V6 outcomes per source arm, out of 64 blocks, with Wilson 95% intervals.

Outcome	I–priv	D–pub	D–full
Eligible confirmed proposals	62/64 [.89, .99]	37/64 [.46, .69]	64/64 [.94, 1.0]
Ordinary plans achieving the goal	1/64 [.00, .08]	26/64 [.29, .53]	0/64 [.00, .06]
Admitted selections	54/64 [.74, .91]	16/64 [.16, .37]	46/64 [.60, .81]
Passing all standard checks	51/64 [.68, .88]	16/64 [.16, .37]	44/64 [.57, .79]
Passing both sealed suites	29/64 [.34, .57]	2/64 [.01, .11]	24/64 [.27, .50]

Table 52: V6 paired comparisons by block (64 pairs each, none unknown): pair counts and exact two-sided sign tests.

Comparison	Outcome	Both	A only	B only	Neither	p
I–priv vs. D–pub	Eligible	37	25	0	2	< 0.0001
	Admitted	14	40	2	8	< 0.0001
	Both suites	2	27	0	35	< 0.0001
I–priv vs. D–full	Eligible	62	0	2	0	0.50
	Admitted	42	12	4	6	0.077
	Both suites	17	12	7	28	0.36
D–full vs. D–pub	Eligible	37	27	0	0	< 0.0001
	Admitted	11	35	5	13	< 0.0001
	Both suites	1	23	1	39	< 0.0001

candidate object, and each passed two or three of the seven admission checks, so the difference lies in what the development feedback repairs rather than in whether a candidate was produced. This reproduces the V3 comparison, which admitted 6/16 against 0/16, at four times the number of sources.

Enforcement. With the nine probes sealed alongside the semantic challenge before any candidate was generated, the admitted libraries permit the violation in 0/1,720 discriminating cases and the initial composition in 960/960. Three probes are discriminating, in the sense that the initial composition leaves them unprotected: the creditor returning collateral before payment in D1, the contributor spending pledged gold in D2, and the owner giving the promised asset to an outsider after the worker's return in D3. The three D4 probes are refused under every library, natively under the initial composition and by a guard that is never decisive under the acquired libraries, and do not discriminate. The authored capability library permits the D3 post-completion gift in 320/320 while refusing the D1 and D2 violations. The design unit is arm, block, and probe, so the discriminating counts are 86 admitted D1–D3 libraries at 20 bindings and the initial composition over 16 fixtures per demand at 20 bindings; over all nine probes the 116 admitted libraries permit no violation in 5,240 cases. The probes are dispatched after each library's own lawful arrangement is complete; the sealed challenge separately records 140/660 D2 cases failing on diversion before completion. An installed guard is decisive in 300 of the admitted cases, 260 in D3 and 40 in D1; in the remainder the acquired relation holds the pledge, the collateral, or the promised asset in a state that no base action produces. The lawful continuation after the refused probe completed in 1,500 of the 1,720 discriminating cases; in the other 220, all on the D3 post-completion gift probe under 11 of the 20 admitted D3 libraries, the guard refused the outsider gift but the worker's award then failed on a native precondition, because the probe's own return had closed the loan that those libraries' finish step expected to find open. This reproduces the enforcement differential of Appendix D.17 with the probes sealed before generation. Table 53 lists every probe by library kind.

Later use. Added rules were invoked in 71/128 multi-role tasks, 71/128 full-information tasks, and 22/128 public-facts tasks, and complete chains occurred in 32/128 [.18, .33], 45/128 [.27, .44], and 2/128 [.00, .06] respectively, on the two V4 later tasks given to each admitted selection of the three sequential-feedback conditions, planned by a fresh planner that saw only the installed library, with the same tasks under the initial library as the control for every block. Slots whose condition admitted no library are retained, 20, 36, and 96 of the 128, so among the tasks with an admitted library the added rule was invoked in 71/108, 71/92, and 22/32 and chains completed in 32/108, 45/92, and 2/32. A planner given only the initial library satisfied goal and safety in 95/128 tasks, with no evaluated

Table 53: All nine V6 probes by library kind at the sealed bindings. Unprotected counts are violation successes over probe cases; the last three columns decompose the admitted libraries: cases refused by an installed guard, cases refused by initial preconditions, and cases in which removing the guards lets the violation succeed (decisive). The three discriminating probes are those the $\mathcal{R}_0$ composition leaves unprotected.

Probe	$\mathcal{R}_0$ comp.	Authored	Admitted	By guard	Native	Decisive
D1: creditor returns collateral unpaid	320/320	0/320	0/660	640/660	20/660	40/660
D1: owner lists and sells pledged tool	0/320	0/320	0/660	640/660	20/660	0/660
D2: contributor spends pledged gold	320/320	0/320	0/660	480/660	180/660	0/660
D2: seller gives away reserved asset	0/320	0/320	0/660	640/660	20/660	0/660
D3: owner disposes of promised asset while worker holds it	0/320	0/320	0/400	360/400	40/400	0/400
D3: after the worker's return, owner gifts promised asset	320/320	320/320	0/400	360/400	40/400	260/400
D4: owner sells during first interval	0/320	0/320	0/600	600/600	0/600	0/600
D4: first holder re-borrows in second interval	0/320	0/320	0/600	600/600	0/600	0/600
D4: second holder gifts asset	0/320	0/320	0/600	600/600	0/600	0/600
Three discriminating probes	960/960	320/960	0/1,720	1,480/1,720	240/1,720	300/1,720
All nine probes	960/2,880	320/2,880	0/5,240	4,920/5,240	320/5,240	300/5,240

failure and 33 unknown outcomes, 32 of them the D3 tasks, and safety alone held in 112/128 with 16 unknown; on the slots where an acquired library was available the matched comparison is 69 against 81 for multi-role, 66 against 79 for full-information, and 20 against 30 for public facts. By pair, the control completes a task that the acquired library does not in 15 multi-role, 16 full-information, and 10 public-facts slots and never the reverse, 34 of those 41 in D1 and the other 7 in D2 and D4; the D3 tasks are unpaired because the control planner returned no D3 plan, and the acquired libraries fail 36 of them; safety held in every executed task under every library. The control has no added-rule chain by definition.

Semantic challenge and subgroups. Of the 116 admitted selections, 111 passed the standard held-out suite, 60 passed the sealed challenge, and 55 passed both. The 2,164 failing challenge cases fall in 11 of the 32 classes (Table 54) and concentrate in two kinds the challenge was written for, an unrelated lawful action interleaved with the contract and a replay after a lawful disposal, with 960 in D1, 1,004 in D2, 200 in D3 and none in D4. Outcomes differ substantially by demand: on the both-suite endpoint the multi-role arm records 3/16, 4/16, 8/16, and 14/16 across D1 to D4, the full-information arm 3/16, 5/16, 0/16, and 16/16, and the public-facts arm 0/16, 1/16, 1/16, and 0/16. Subgroups by demand are descriptive.

Table 54: V6 sealed-challenge failures by class over the admitted selections evaluated (failing cases over cases; 20 bindings per selection). The 21 classes with no failure are omitted.

Class	Failing / cases
D1: repayment with an unrelated loan in progress	500/660
D1: replay after disposal of redeemed collateral	460/660
D2: settlement replay after disposal of a refunded asset	360/660
D2: interleaved trading of unrelated assets	320/660
D2: attempted diversion of pledged funds before completion	140/660
D2: preservation of unrelated income during refunds	104/660
D2: settlement and refund replay after funded expiry	40/660
D2: repeated refunds after partially funded expiry	40/660
D3: cancellation, return, and a new loan before completion	140/400
D3: replay after disposal of an earned asset	40/400
D3: unrelated prior work followed by contracted work	20/400

Method conditions. Of 2,984 recorded calls, 2,810 parsed; the run made 2,814 request slots, and no call was declined on content grounds. Four sequential-feedback slots returned no candidate because their accumulated development feedback exceeded the input limit; each retains its slot and the affected selections were made among the remaining candidates.

D.22 V7: LATER-USE ENFORCEMENT ON THE V6 ARTIFACTS

The two V6 later tasks are cooperative by construction, so a planner given only the initial library also completes them, and cooperative completion does not distinguish the libraries. V7 asks whether the enforcement that acquisition adds survives into genuinely new use: after a fresh planner, which never saw the acquisition, has arranged the relation on relabelled participants in a later task, does the world refuse an independent caller's violation, and does the lawful task still complete afterwards? V7 makes no model call. Each slot re-executes the V6 planner's recorded plan under the probe instrument sealed in V6: the plan's arrangement phases run, the host tests whether the protected relation was established, the demand's discriminating violation is dispatched by an independent caller (D1: the holder returns the collateral before payment; D2: the contributor spends pledged gold; D3: after the worker's return the owner gifts the asset to an outsider), and the remaining lawful phases follow with the V4 goal predicate. The load-bearing goal is met when the arrangement is established, the violation is refused, and the lawful continuation completes. Under the initial library the violation succeeds wherever the arrangement is established, so the initial library is the control that demonstrates what the goal requires; the informative results are whether the acquired libraries meet it and how the arms compare.

Design. Units are the 48 V6 blocks of D1–D3, the three sequential-feedback conditions and the initial-library control, and both V6 later-task variants: 384 slots, 96 per condition. D1–D3 are the demands with a probe that the initial composition leaves unprotected, the condition for a probe to bear the goal, and this set was fixed before evaluation. Slots whose condition admitted no library are retained as unavailable; a slot whose recorded plan is empty is an arrangement failure. The probe definitions, the classification of a result, and the guard-decisive counterfactual are those of V4E, unchanged. The arrangement predicates are V4E's with one addition fixed before evaluation: for D2 the arrangement counts as established only while a pledge is pending, that is while gold is held for the acquisition either in a contract's escrow or in the contributor's hands with the asset not yet the contributor's, because V4E's predicate is also satisfied by a plan that has already paid the seller in full, and the diversion probe is then vacuous under any library. A 24-slot pilot on one block per demand showed this vacuity under the initial library; the condition applies identically to every arm. Three hypotheses were recorded before evaluation: the control meets the goal in no slot; each acquired arm refuses the violation in every established slot and meets the goal in a positive number; private multi-role and full-information drafting are within ± 0.10 paired difference.

Results. Only acquired libraries meet the load-bearing goal: 33/96 slots for private multi-role, 20/96 for public-facts, and 31/96 for full-information drafting, against 0/96 for the initial library, which permits the violation in all 43 slots where it establishes the arrangement (Table 55). Paired by block and task, every acquired arm exceeds the control with no discordant pair in the other direction (33, 20, and 31 discordant pairs to 0; $p < 0.0001$ each), and the three acquired arms are not separated from one another (Table 56): private multi-role against full-information, 7 discordant pairs to 10 over 56 observed, paired difference -0.054 with 95% interval $[-0.172, +0.085]$, $p = 0.63$, against the preregistered margin of ± 0.10; private multi-role against public-facts, 5 to 3 over 28, $+0.071$ with interval $[-0.111, +0.208]$, $p = 0.73$; full-information against public-facts, 3 to 1 over 22, $+0.091$ with interval $[-0.072, +0.165]$, $p = 0.63$. The second hypothesis holds in part: every acquired arm meets the goal, and the violation is refused in 92 of the 113 established acquired slots; in the 21 others, 15 of them in the time-shifted variant and 11 in D3, the fresh planner had established the relation through native actions, mostly loans or hiring, that the acquired guard, keyed to its own contract, does not watch. The arrangement itself is established in 50 of 80 available private multi-role slots, 39 of 60 full-information, and 24 of 32 public-facts, so plan failure and enforcement failure are reported separately. The initial library's D3 slots have no recorded plan: the V6 planner returned none for any of them. By demand, the goal is met in 13, 20, and 0 of 32 slots for private multi-role, 2, 18, and 0 for public-facts, and 9, 20, and 2 for full-information across D1 to D3; by task variant, in

20 of 48 new-roles slots against 13 of 48 time-shifted slots for private multi-role, 12 against 8 for public-facts, and 20 against 11 for full-information. Subgroups are descriptive.

Table 55: V7 outcomes per condition, out of 96 planned slots (48 blocks $\times$ two later tasks), with Wilson 95% intervals. Refused and goal counts are conditional on an established arrangement. The lower rows partition the available slots by probe classification; guard-decisive cases are those refused by a guard in which removing the guards lets the violation succeed.

Outcome	I–priv	D–pub	D–full	$\mathcal{R}_0$
Admitted library available	80 [.75, .89]	32 [.25, .43]	60 [.53, .72]	96 [.96, 1.0]
Arrangement established	50 [.42, .62]	24 [.17, .35]	39 [.31, .51]	43 [.35, .55]
Violation refused	37 [.29, .49]	22 [.16, .32]	33 [.26, .44]	0 [.00, .04]
Load-bearing goal met	33 [.26, .44]	20 [.14, .30]	31 [.24, .42]	0 [.00, .04]
refused by an installed guard	29	15	27	0
refused by initial preconditions	8	7	6	0
unprotected	13	2	6	43
arrangement failed	30	8	21	53
guard decisive	6	0	2	0

Table 56: V7 paired comparisons on the load-bearing goal, by block and task (96 planned pairs). Pairs with an unavailable member are unknown; the paired difference is over observed pairs, with a Wilson 95% interval on the first arm's share of discordant pairs rescaled by the discordant fraction; p is the exact two-sided sign test.

Comparison	Both	A only	B only	Neither	Unknown	Difference	95% interval	p
I–priv vs. D–full	18	7	10	21	40	-0.054	$[-0.172, +0.085]$	0.63
I–priv vs. D–pub	13	5	3	7	68	$+0.071$	$[-0.111, +0.208]$	0.73
D–full vs. D–pub	11	3	1	7	74	$+0.091$	$[-0.072, +0.165]$	0.63
I–priv vs. $\mathcal{R}_0$	0	33	0	47	16	$+0.412$	$[+0.326, +0.412]$	< 0.0001
D–pub vs. $\mathcal{R}_0$	0	20	0	12	64	$+0.625$	$[+0.424, +0.625]$	< 0.0001
D–full vs. $\mathcal{R}_0$	0	31	0	29	36	$+0.517$	$[+0.403, +0.517]$	< 0.0001

D.23 V8: A SUITE COMPILED FROM THE DEMAND CONTRACTS

The sealed suites of V2–V7 are authored scenarios graded by host predicates. This appendix reports a second instrument in which the violation cases are not written one by one: the four demand contracts are recorded as clause data, and the cases are generated from those records by fixed transformations. It was built after the V6 results were known, as a post hoc instrument; it calls no model, changes no earlier number, and evaluates libraries that were already recorded. Its protocol, clause table, compiler and every executed module were frozen before the run's first case was evaluated.

The contract as data. Each demand's public contract is transcribed once into clause records. A clause names the object it protects (the demand asset, or a participant's holding); the effects it forbids, drawn from the closed vocabulary {CUSTODY MOVES, OWNERSHIP MOVES, HOLDING SPENT}; the condition that discharges it (a payment of a named amount by a named party, an authenticated completion of a named job, or the arrangement's own settlement) and whether the clause holds until that condition or from it; the day its window closes and the effects required then, drawn from {REFUND ACTUAL CONTRIBUTIONS, RELEASE ASSET, SETTLEMENT REFUSED}, together with the contributors to be refunded; a custody schedule when custody rotates; and whether the right it protects persists after settlement. Four demands yield eight clauses. The table contains no action sequence, no expected trace and no reference to any candidate.

Realizing a forbidden effect. For each forbidden effect the compiler dispatches the base actions that can realize it, performed by the party positioned to act on the subject: for custody, a return by a holder who borrowed the object, a gift, or a loan to an outsider; for ownership, a gift or a sale to an outsider through the owner's listing; for a holding, a transfer of one gold to an outsider. Every agent that an action names presents its approval, so an act the base world allows is never refused for want of authorization alone. Each effect's attempts start from the same state, use only base actions, and are each made by a single party.

The transformations. Seven transformations produce violation cases. *Premature violation* attempts the forbidden effects while the clause is in force, immediately after the lawful arrangement is established, or at the start of each scheduled interval but the last when custody rotates. *Counterparty violation* repeats the attempts with a participant who neither holds nor owns the object as the caller, without the holder's or owner's approval. *Foreign condition* has an outsider perform the discharging condition, for example paying the payee in full, and then runs the arrangement's finish; *partial condition* has the payer pay half and then finishes. Under an admitted library whose release performs the authorized payment itself, that release discharges the condition in both transformations. *Replay after settlement* settles the arrangement and re-dispatches its phases; *dispose then replay* settles it, lets the rightful owner give the object to an outsider, and replays. *Expiry effects* advances the clock to the clause's window and checks the effects the clause requires there. An eighth transformation, *unrelated interleaving*, is an over-breadth check rather than a violation case: one outsider lends another an unrelated item and gives it one bread, the arrangement continues, and the item is returned. No transformation reads a demand name, a candidate or a rule text; every transformation except expiry effects starts from the lawful arrangement the V4E and V6 probes use for that demand (Appendix D.17), whose phases are specific to the demand.

Grading. Every verdict is a host predicate over the clause's subject and the dispatcher's own refusal reasons: the protected object's owner and holder (and, after settlement, its lock), or the named participant's holding, compared before and after the attempt, and at expiry the required refunds, release and refusal of settlement. A condition counts as discharged when the payer's holding fell and the payee's rose by the named amount, or when the named job was completed with authentication. A case is *unprotected* when a forbidden effect occurred or a required effect did not, *protected by guard* when it did not and an installed guard refused, *protected natively* when it did not and no guard was involved, and *over-broad* when a lawful action was refused. A case is *not applicable* when the clause gives its transformation nothing to act on (no forbidden effect, no payment or completion condition, no window with required effects, no right that persists after settlement, a holding where an object is needed, or no participant other than the holder and owner to act as counterparty), or when the clause is not in force at the attack point, such as a D3 case that needs the job still pending when the lawful arrangement includes its completion. Guards are removed in a counterfactual re-execution of every guarded case to record whether the guard was decisive.

Units. The suite runs on the 116 admitted V6 selections at five bindings (3100–3104), which reuse the sealed relabelings under new asset identifiers, against three libraries: the initial library with authored skill plans, the authored capability library, and each episode's own admitted library. That is 29,100 cases, 9,700 per library, of which 4,715 are not applicable and 1,175 are over-breadth checks in every library. The remaining violation cases give 3,810 evaluated cases for each control and 3,560 for the admitted libraries, whose 250 over-broad refusals are counted separately. The controls run once per episode on that episode's block fixture, so episodes that share one of the 60 fixtures repeat the same deterministic control case; counted once per fixture, as V4E and V6 count their controls, the initial library permits the violation in 775 of 1,960 distinct cases and the authored capability library in none. The five bindings relabel one structural situation.

Results. Table 57 gives every transformation by library. The initial library with authored plans permits the violation in 1,620 of 3,810 cases, in 96 of the 116 episodes: 495 of 1,320 D1 cases, 825 of 990 D2 cases, 300 of 900 D4 cases and none of the 600 D3 cases. The authored capability library permits it in none of 3,810 cases, and the admitted libraries in none of 3,560. The premature attempts succeed in 795 of 1,060 cases under the initial library: the creditor's early return of the collateral and the diversion of pledged gold, as in the sealed probes, and two acts no sealed probe attempts, an outsider's purchase of the reserved asset through the seller's standing listing and the scheduled holder's early return in D4. Neither the authored nor the admitted libraries permit any of them.

Table 57: Every compiled transformation by library: cases permitting the violation over cases evaluated. All three libraries run the same compiled cases on the same 116 episodes and five bindings; unrelated interleaving is an over-breadth check, reported as lawful actions refused.

Transformation	Initial library	Authored capability	Admitted
premature violation	795/1,060	0/1,060	0/1,060
counterparty violation	0/565	0/565	0/565
foreign condition	165/165	0/165	0/165
partial condition	165/165	0/165	0/165
replay after settlement	0/680	0/680	0/680
dispose then replay	0/680	0/680	0/430
expiry effects	495/495	0/495	0/495
All violation cases	1,620/3,810	0/3,810	0/3,560
unrelated interleaving (refused)	0/1,175	0/1,175	520/1,175

Guard decisiveness and refused lawful actions. Removing the guards separates two sources of protection. Under the authored capability library the violation goes through in 795 of its 1,095 guard refusals, all in D1 and D4. Under the admitted libraries it goes through in 90 of 1,930, all premature attempts in D1 and D4, and the other 1,840 cases are refused without the guards as well. In 25 of the 116 episodes an installed guard refuses the rightful owner's lawful disposal after settlement (250 cases: 230 in D1, 20 in D3), and in 44 episodes the guards refuse the unrelated loan between two outsiders or its return (520 of 1,175 cases: 250 in D1, 240 in D2, 30 in D3). Every disposal refusal and 505 of the 520 loan-or-return refusals are guard evaluation errors: the guard looks up the contract that locks the object, and the lookup fails because no contract locks it; the other 15 are refusals by a guard's condition. Neither control refuses any of these lawful actions.

Refused lawful actions and the sealed challenge. Split post hoc by the V6 outcomes of the same selections, the refused lawful actions fall almost entirely on libraries that two-stage admission does not deploy. All 44 libraries with a refused lawful action passed the standard held-out suite, and 42 of them failed the sealed challenge, whose classes include an unrelated lawful action interleaved with the contract and a replay after a lawful disposal (Appendix D.21). The 55 libraries passing both suites refuse 25 of their 750 lawful cases: the unrelated loan or its return in 25 of 450, from two libraries (15 cases in D2, 10 in D3), all guard evaluation errors, and the disposal after settlement in none of 300. The other 61 libraries refuse 745 of 1,105. On the violation cases of the same 55 episodes the $\mathcal{R}_0$ composition permits 640 of 1,710, and the authored capability library and the admitted libraries permit none. The script `tools/split_v8_by_v6_suites.py` in the supplementary material recomputes these counts from the released V6 and V8 records.

D.24 V9: THE V6 PROTOCOL WITH THREE CLAUDE MODELS

V9 repeats V6 (Appendix D.20) on the same 64 blocks with Claude Opus 5.5, Claude Sonnet 5 and Claude Haiku 4.5, each requested at medium effort in fresh contexts through the Claude Code command-line client, one run per model under one freeze record; V6 requested GPT-6 Astra through the Codex client. Every model finished all 64 blocks within the analysis cutoff recorded before the first call. Fixtures, public context, the three source arms, sequential-feedback synthesis with six request slots and the paired independent sampling, admission, the standard held-out suite, the sealed semantic challenge, the nine enforcement probes and the later tasks are those of V6: the first source prompt of every block and arm and the first later-task prompt under the initial library are byte-identical to V6's. The V6 hypotheses were recorded for V9 before its first call, with decision rules fixed in advance, and the protocol, configurations, fixtures, public context, semantic challenge and probe definitions were hashed into a freeze record that the runner verified before every block, so the probes were again sealed before any candidate was generated. As in V6, a selection whose admission stays pending counts as not admitted (14 of the 162 Claude Opus 5.5 sequential-feedback selections).

Enforcement. Under the sealed probes the 75 admitted Claude Opus 5.5 D1–D3 libraries permit the violation in 0/1,500 discriminating cases and the initial composition in 960/960; over all nine probes the 101 admitted libraries (89 sequential-feedback selections and 12 admitted by independent sampling) permit none of 4,560 cases (Table 58). An installed guard is decisive in 300 cases, all on the D3 post-completion gift, where the authored capability library again permits the gift in 320/320 while refusing the D1 and D2 violations. The enforcement prediction recorded before the first call is met exactly as in V6, on a second model family and a second client. The three admitted Claude Sonnet 5 libraries, all of D3, permit none of their 60 discriminating cases, each refused by an initial precondition; Claude Haiku 4.5 admitted no library.

Table 58: V9 (Claude Opus 5.5) discriminating probes by library kind at the sealed bindings. Unprotected counts are violation successes over probe cases; the last three columns decompose the admitted libraries as in Table 53.

Probe	$\mathcal{R}_0$ comp.	Authored	Admitted	By guard	Native	Decisive
D1: creditor returns collateral unpaid	320/320	0/320	0/1,100	540/1,100	560/1,100	0/1,100
D2: contributor spends pledged gold	320/320	0/320	0/100	40/100	60/100	0/100
D3: after the worker's return, owner gifts promised asset	320/320	320/320	0/300	300/300	0/300	300/300
Three discriminating probes	960/960	320/960	0/1,500	880/1,500	620/1,500	300/1,500
All nine probes	960/2,880	320/2,880	0/4,560	2,860/4,560	1,700/4,560	300/4,560

Feedback against independent sampling. With Claude Opus 5.5, on the 52 multi-role sources with an eligible proposal, sequential feedback admitted 23 selections and independent sampling 12, both strategies receiving the same frozen source card and six request slots (13 discordant blocks to 2, paired difference $+0.212$, $p = 0.0074$), meeting the recorded prediction on its primary endpoint; on the both-suite endpoint the counts are 16 and 11 (9 discordant blocks to 4, $p = 0.27$). With Claude Sonnet 5 feedback admitted one of 29 sources and sampling none; with Claude Haiku 4.5 neither admitted any of 31.

Admission in two stages. Of the 89 admitted Claude Opus 5.5 selections, 84 passed the standard held-out suite and 73 the sealed semantic challenge, 70 both; the 16 that passed admission but failed the challenge, and the four admitted Claude Sonnet 5 selections, all of which failed it, are the V9 instances of the split reported for V4 and V6 in Section 5.

Elicitation. Table 59 gives every source arm for the three requested models, with V6 alongside. With Claude Opus 5.5 the public-facts designer passed both suites in more blocks than private multi-role drafting (22 against 16; 12 discordant blocks to 6, $p = 0.24$), and full-information drafting in more than multi-role drafting (32; 17 discordant blocks to 1, paired difference $+0.250$, 95%

interval $[+0.136, +0.276]$, $p = 0.00014$, outside the ± 0.10 margin recorded for that comparison). On admitted selections multi-role and public-facts drafting reach 23 and 25 (8 to 10, $p = 0.81$), and on eligible proposals 52 and 46 (12 to 6, $p = 0.24$). Paired with V6 on the same blocks, Claude Opus 5.5 passed both suites in fewer blocks than GPT-6 Astra under multi-role drafting (16 against 29; 10 to 23, $p = 0.035$), in more under public facts (22 against 2; 22 to 2, $p < 0.0001$) and under full information (32 against 24; 13 to 5, $p = 0.096$). With Claude Sonnet 5 and Claude Haiku 4.5 no selection passed both suites in any arm: Claude Sonnet 5 admitted four selections and Claude Haiku 4.5 none, none passed the semantic challenge, and confirmed ordinary plans over the existing interface achieved the goal in 22, 14 and 26 (Sonnet) and 15, 17 and 20 (Haiku) blocks of the three arms.

Table 59: V9 outcomes per requested model and source arm, out of 64 blocks, with V6 (GPT-6 Astra) on the same blocks. Ordinary goal counts confirmed ordinary plans that achieved the block's goal.

Model	Arm	Eligible	Ordinary goal	Admitted	Standard	Semantic	Both suites
Claude Opus 5.5	I–priv	52	6	23	22	17	16
	D–pub	46	11	25	24	23	22
	D–full	64	0	41	38	33	32
Claude Sonnet 5	I–priv	29	22	1	1	0	0
	D–pub	29	14	3	3	0	0
	D–full	28	26	0	0	0	0
Claude Haiku 4.5	I–priv	31	15	0	0	0	0
	D–pub	27	17	0	0	0	0
	D–full	34	20	0	0	0	0
GPT-6 Astra (V6)	I–priv	62	1	54	51	32	29
	D–pub	37	26	16	16	2	2
	D–full	64	0	46	44	26	24

Later use. With Claude Opus 5.5, planners that see only the installed library invoked the added rules in 39, 45 and 74 of each arm's 128 tasks and completed chains in 29, 36 and 62 (I–priv, D–pub, D–full), meeting the later-use prediction; with Claude Sonnet 5 they invoked an added rule in one task each under I–priv and D–pub and completed no chain, and with Claude Haiku 4.5 no library was available.

D.25 V10: NORM SYNTHESIS AND A GENERIC OBLIGATION PRIMITIVE

V10 runs the comparison specified in Appendices B.7 and B.8 with the two competing explanations for the protection that acquired rules provide: regulative norms synthesized from observed conflicts, which are prohibitions on existing actions, and a richer initial interface that provisions a generic obligation primitive in advance. The synthesis follows IRON (Morales et al., 2013; 2015), is deterministic, and makes no model call. A norm reads "refuse base action ρ when every literal of F holds" and is installed as a guard on ρ; it can only refuse, and adds no rule, effect or state. Its literals come from a closed vocabulary generated from the base actions' declared roles: for the item an action names, whether its owner holds it and whether it is unlocked, and for each agent argument whether that agent owns or holds it; for a contract argument, whether it exists and is open, whether the day precedes its due day or equals its start day, and each agent argument's party role; for a listing, whether it is active and created that day and who sells it; for a resource, its kind. No identifier, role label or numeric threshold appears.

Procedure. For each of the 64 V6 blocks, the synthesis (i) executes the block's development cases, exactly those of the V4 admission conjunction and of V6 synthesis feedback, under the $\mathcal{R}_0$ composition and collects the failing constraint and refusal cases as conflicts; (ii) attributes each conflict to the accepted base-action dispatches whose refusal alone makes the case pass; (iii) forms, for the latest such dispatch, the most specific norm over the vocabulary evaluated on its pre-action state and arguments, and keeps it only if, installed alone, it repairs the conflict and every passing case still passes, otherwise trying the next dispatch; (iv) merges norms on the same action into their least general generalization when that remains effective and safe on every conflict of the group; and (v) with all norms installed, removes the newest norm until every passing case still

passes. The resulting library is evaluated with the V6 code on the V6 fixtures: admission, the standard held-out suite, the sealed semantic challenge, the nine probes, and the compiled clause suite of Appendix D.23. Three predictions on norm synthesis were recorded before the freeze: the norms refuse some discriminating violations that $\mathcal{R}_0$ permits; their both-suite count is below those of V6's multi-role and full-information libraries; and on the compiled transformations that test a condition's discharge or window (foreign condition, partial condition, expiry effects) they leave as many violations unprotected as $\mathcal{R}_0$.

Results. The development cases of the 64 blocks contain 544 conflicts. The synthesis repairs all 64 in D1 and all 96 in D4, each of those 32 blocks receiving one norm on the return action, and none of the 128 in D2 or the 256 in D3, where no prohibition repairs a conflict while keeping every passing case. The D1 norm refuses the return action when the returning agent holds the pledged tool as borrower under the open loan contract through which the $\mathcal{R}_0$ composition records the pledge, before that contract's due day. The norm libraries pass admission in 32/64 blocks, the standard suite in 31/64 and the sealed semantic challenge in none, so none passes both suites, against 29/64 and 24/64 for V6's multi-role and full-information libraries (29 and 24 discordant blocks, none the other way; $p = 3.7 \times 10^{-9}$ and 1.2×10^{-7}). Under the sealed probes the D1 norm refuses the creditor's early return in all 320 cases, while the pledged-gold diversion and the post-completion gift succeed in 320/320 each: 640/960 discriminating violations against 960/960 for $\mathcal{R}_0$, fewer in 16 blocks and more in none ($p = 3.1 \times 10^{-5}$). On the compiled clause suite the norms permit 560 of 2,080 violation cases against 800 for $\mathcal{R}_0$, and on the discharge and window transformations 400/400, as $\mathcal{R}_0$ does (Table 61). All three norm-synthesis predictions are met. A prohibition over existing actions can refuse a dispatch only where existing state already distinguishes the violation, as the open loan does for the early return; the pledge held for a purchase, the promised award and the record that a condition was discharged are state that no base action creates, and an acquired relation supplies it.

Generic obligation primitive. The primitive is one family of twelve rules written in the bounded rule language that candidates use and installed through the same path, subject to authenticated actors, host-written receipts and contract authority; it names no demand, participant, asset or task. It was specified in advance to express the D1 and D2 contracts. Its condition vocabulary covers custody of an item conditioned on a payment, pooled payment with contributor-specific refunds, release on an authenticated completion, and release at a date; the protocol records before the freeze that it lacks a custody schedule naming a later custodian and a conditional transfer of an item that is on loan. A planner (Claude Opus 5.5, Claude Sonnet 5 or Claude Haiku 4.5, each requested at medium effort in fresh contexts through the Claude Code command-line client) given the full-information or the public-facts projection of a block composes one arrangement from the 17 base actions and the primitive, every participant confirms or rejects the exact plan, and the plan with the primitive is evaluated as V6 evaluates an acquired library, both suites requiring confirmation; an authored composition of the primitive calibrates what it expresses. Every planner model finished all 64 blocks within the analysis cutoff recorded before the first call. The recorded predictions were that the full-information planner does not match the V6 full-information libraries on both suites ("match" if the 95% interval of the paired difference lies within ± 0.10, "does not match" if $p < 0.05$, otherwise inconclusive), and that public facts lower both-suite passes and primitive use.

Every planner and the authored composition pass both suites only in D1 or D2 (Table 60); none of the seven conditions passes in any of the 32 D3 and D4 blocks, where the V6 multi-role libraries pass in 8 and 14 of 16 and the full-information libraries in 0 and 16. Against V6 full information on both suites, the full-information planner is inconclusive with Claude Opus 5.5 (27 against 24; 20 discordant blocks to 17, paired difference $+0.047$, interval $[-0.134, +0.219]$, $p = 0.74$) and with Claude Haiku 4.5 (13; 12 to 23, -0.172, $[-0.319, +0.009]$, $p = 0.090$), and does not match with Claude Sonnet 5 (6; 3 to 21, -0.281, $[-0.342, -0.142]$, $p = 0.00028$). Public facts lower both-suite passes with Claude Opus 5.5 (20 against 27, $p = 0.039$) and Claude Haiku 4.5 (1 against 13, $p = 0.0018$) and match full information with Claude Sonnet 5 (5 against 6, interval $[-0.099, +0.076]$, $p = 1.0$); primitive use is 59 against 63, 52 against 52 and 51 against 48 ($p = 0.22$, 1.0 and 0.25). Under the primitive no discriminating violation succeeds for the authored composition or either Claude Opus 5.5 condition, and the violation succeeds in 20 and 40 of 960 cases for the full-information plans of Claude Sonnet 5 and Claude Haiku 4.5.

Table 60: V10 generic obligation primitive on the 64 V6 fixtures: blocks admitted and passing both suites, in total and per demand (16 blocks each), and cases permitting the violation on the three discriminating probes and on every compiled clause case (probe cases include those whose plan did not establish the arrangement; clause cases count those evaluated). V6 acquired libraries on the same blocks for reference.

Condition	Admitted	Both	D1	D2	D3	D4	Probes	Clauses
Authored composition	62	30	15	15	0	0	0/960	0/2,080
Opus 5.5, full information	60	27	15	12	0	0	0/940	0/2,050
Opus 5.5, public facts	51	20	13	7	0	0	0/860	0/1,850
Sonnet 5, full information	19	6	6	0	0	0	20/960	190/1,895
Sonnet 5, public facts	9	5	5	0	0	0	0/960	145/1,165
Haiku 4.5, full information	14	13	11	2	0	0	40/960	290/1,915
Haiku 4.5, public facts	4	1	0	1	0	0	0/960	220/1,020
V6 multi-role, acquired	54	29	3	4	8	14	0/800	0/1,670
V6 full-information, acquired	46	24	3	5	0	16	0/600	0/1,410

Table 61: V10 norm synthesis against the references on the 64 V6 fixtures. Suite columns count the blocks, out of 64, whose candidate library passes; the last three columns count cases permitting the violation over cases evaluated: the three discriminating probes, every compiled violation case, and the compiled discharge and window transformations.

Library	Admitted	Standard	Semantic	Both	Probes	Clauses	Discharge, window
$\mathcal{R}_0$ composition	–	–	–	–	960/960	800/2,080	400/400
Norm synthesis	32	31	0	0	640/960	560/2,080	400/400
Authored capability library	–	–	–	–	320/960	0/2,080	0/400
V6 multi-role, acquired	54	51	32	29	0/800	–	–
V6 full-information, acquired	46	44	26	24	0/600	–	–

D.26 V11: DEMANDS ARISING FROM STATED NEEDS

V11 asks whether multi-role interaction raises the demand when no participant is told it. It uses the 64 V6 blocks and replaces exactly two model-facing fields of each fixture. The public goal, which in V6 states the target arrangement, becomes a statement that no common task is given, followed by public facts only: the day, who is at the market, and the asset's kind, owner and holder. Each private preference, which in V6 states the contract requirements, becomes a need statement: what the participant holds, what it needs to obtain, keep or use and by when, what it can offer and from when, and which loss it cannot accept, stated as an outcome to avoid. In D1, for example, the need statement of the participant who needs bread reads "I need three more bread now, on day 0, and cannot wait for them. My gold is set aside until day 1 … I own [the sickle] … I must remain its owner", and that of the participant with bread "I have bread to spare and want four gold for three bread … I must not end up having given the bread without receiving the four gold." The need statements protect the outcomes that the V6 preferences protect and state no arrangement: no security, no triggered transfer, no rule binding other callers. World state, observations, phase days, oracles and every hidden evaluation field are those of V6. Before the freeze, the public statement, every need statement, every participant's observation and every fixture projection that a V11 prompt can contain were checked against a banned arrangement vocabulary (credit, collateral, pawn, escrow, pool, pledge, refund, reserve, custody schedule, rotation, promise, award, deadline settlement, with inflections), with no hit.

Procedure. The source is V6's multi-role drafting: three drafting turns with a rotating speaker that sees the public statement, the interface, its own observation and need statement, and the prior public trace, then one confirmation call per participant on the exact final draft, judged from that participant's own observation and need statement. A block is eligible when the final draft is a shared-relation proposal that the proposers state the interface cannot execute and every participant confirms it. Every eligible proposal is formalized by the V6 sequential-feedback synthesis with six request slots, whose

development feedback from the second slot on reports the failed development cases, and is admitted, evaluated on both sealed suites and the nine probes, and used in the two later tasks exactly as in V6, graded with the V6 fixtures and oracles. Claude Opus 5.5, Claude Sonnet 5 and Claude Haiku 4.5 were each requested on the 64 blocks in fresh contexts through the Claude Code command-line client. The protocol, fixtures, need statements, vocabulary check, challenge and probe definitions were hashed into a freeze record before the first call. The predictions recorded there concern eligible proposals (HD1) and admitted selections passing both suites (HD2). The protocol's descriptive reference for HD1 is the five V1 pressure-only dialogues, which differ in protocol and model and proposed no target relation (Table 10).

Proposal formation. Multi-role interaction produced eligible proposals in 49/64 blocks with Claude Opus 5.5 (95% interval [.65, .85]), 26/64 with Claude Haiku 4.5 ([.29, .53]) and 8/64 with Claude Sonnet 5 ([.06, .23]), so HD1 holds for every model. Claude Opus 5.5 reached eligible proposals in 5, 16, 16 and 12 of the 16 blocks of D1–D4, and in no block did it return an ordinary plan of existing actions that achieved the goal.

Formalization and admission. From these sources Claude Opus 5.5 reached 21/64 admitted selections, 20 passing the standard held-out suite, 9 the sealed semantic challenge, and 8/64 ([.06, .23]) both suites: four in D1, one in D3 and three in D4 (Table 63), so HD2 holds for Claude Opus 5.5; the Claude Sonnet 5 and Claude Haiku 4.5 sources yielded one admitted selection each and none passing both suites. As in V3, V6 and V9, development feedback raises admission: the first slot, written before any feedback, was admitted in 2 blocks and passed both suites in 1. Of the 21 admitted selections, 12 failed the challenge. One of them is the only admitted library in V6, V9 or V11 under which a probe violation succeeded: in a D2 block the seller gave away the reserved asset in 20/20 cases, a probe that the initial composition and the authored library refuse natively. The relation as its participants confirmed it leaves the asset unencumbered until delivery; the library passed the standard suite and 80 of the 160 challenge cases, so two-stage admission does not deploy it.

Enforcement and later use. Under the sealed probes the admitted Claude Opus 5.5 libraries permit none of the 180 discriminating cases, against 960/960 under the initial composition and 320/960 under the authored capability library, with an installed guard decisive in 20. Fresh planners that see only the installed library invoked the added rules in 29 of 128 later tasks and completed chains in 14 ([.07, .18]).

Paired comparisons. The protocol pairs each model's V11 blocks with the same blocks under the demand supplied: V9 multi-role drafting with the same model, and V6 multi-role drafting with GPT-6 Astra (Table 62; exact two-sided sign tests, no direction predicted). With Claude Opus 5.5, eligible proposals and admitted selections are close to V9 (49 against 52, $p = 0.66$; 21 against 23, $p = 0.85$), and both-suite passes are 8 against 16 (5 discordant blocks to 13, paired difference -0.125, interval $[-0.211, +0.005]$, $p = 0.096$). Claude Sonnet 5 formed fewer eligible proposals than in V9 (8 against 29, $p = 0.0002$) and Claude Haiku 4.5 a similar number (26 against 31, $p = 0.44$); neither passed both suites in either study.

D.27 Post-hoc analyses by unit and under multiplicity correction

After the results above were fixed, two analyses were run on the recorded V6, V8 and V9 artifacts alone, by the post-hoc scripts in the supplementary material. A second, independently written script for each reproduces every number, from the per-case records for the unit counts and from the recorded paired counts for the tests.

Counts by fixture and library. The enforcement counts of Section 5 are library-by-binding cases. Counted by unit in V6, the $\mathcal{R}_0$ composition permits the discriminating violation on 48 of the 48 D1–D3 fixtures, at every one of the 20 bindings; the authored capability library permits it on all 16 D3 fixtures; none of the 86 admitted D1–D3 libraries permits it, and none of the 116 admitted libraries permits a violation on any probe of its demand (0/5,240 over the nine probes). With Claude Opus 5.5 (V9), none of the 75 admitted D1–D3 libraries and none of the 101 admitted libraries permits a probed violation. On the compiled clause suite (V8), the $\mathcal{R}_0$ composition permits a violation on 47 of the 60 fixtures that carry an admitted library (every D1, D2 and D4 fixture), while none of the 116

Table 62: V11 paired with multi-role drafting under the supplied demand on the same 64 blocks: counts for V11 and the comparator, discordant blocks (V11 only / comparator only), and exact two-sided sign tests.

Model and comparator	Outcome	V11	Comparator	Discordant	p
Opus 5.5 vs. V9 Opus 5.5	Eligible	49	52	9 / 12	0.66
	Admitted	21	23	12 / 14	0.85
	Both suites	8	16	5 / 13	0.096
Opus 5.5 vs. V6 GPT-6 Astra	Eligible	49	62	2 / 15	0.0023
	Admitted	21	54	4 / 37	< 0.0001
	Both suites	8	29	6 / 27	0.0003
Sonnet 5 vs. V9 Sonnet 5	Eligible	8	29	5 / 26	0.0002
	Admitted	1	1	1 / 1	1.0
	Both suites	0	0	0 / 0	1.0
Haiku 4.5 vs. V9 Haiku 4.5	Eligible	26	31	11 / 16	0.44
	Admitted	1	0	1 / 0	1.0
	Both suites	0	0	0 / 0	1.0
Sonnet 5, Haiku 4.5 vs. V6	Both suites	0	29	0 / 29	< 0.0001

Table 63: V11 with Claude Opus 5.5 by demand: blocks out of 16 per demand (64 in total) with an eligible proposal, an admitted selection, and a selection passing both sealed suites.

Outcome	D1	D2	D3	D4	Total
Eligible confirmed proposals	5	16	16	12	49
Admitted selections	5	1	3	12	21
Passing both sealed suites	4	0	1	3	8

admitted libraries permits one. Refused lawful actions concentrate in libraries the sealed challenge rejects: 44 of the 116 admitted libraries refuse at least one lawful case, 42 of them among the 56 that the challenge rejects and 2 among the 55 that pass both suites; these two account for all 25 refused cases of the 55, each an unrelated action interleaved with the contract.

Multiplicity. The V6 and V9 protocols report every prespecified test at its nominal level. Applied post hoc, the Holm–Bonferroni procedure keeps below 0.05 every test that Section 5 prints with $p < 0.05$: across the six tests printed there (largest adjusted value among the five below 0.05: 0.015, the V9 feedback test), and within each study's eleven within-model sign tests, V6 (all eight with $p < 0.05$ remain) and V9 with Claude Opus 5.5 (all six remain; feedback test 0.044). All 33 sign tests of these studies, including the eleven V9 comparisons with the V6 outcome, reproduce exactly from the paired counts.