GOVERNED SELF-IMPROVEMENT FOR LOGICAL REASONING: EDIT-TIME GOVERNANCE FOR DEVELOPMENTAL CONSISTENCY

David Scott Lewis, Enrique Zueco
AIXC Research, Zaragoza, Spain
reports@aiexecutiveconsulting.com

ABSTRACT

Self-refinement methods enable large language models to improve without retraining, yet they optimize local answers rather than the future reasoner. In logical reasoning, every answer creates *longitudinal commitments*: paraphrases, negations, implication chains, and premise permutations must remain jointly consistent across developmental time. We present a governance-oriented framework and evaluation lens with proof-of-concept validation on a controlled propositional-logic domain. (1) We frame self-improvement as a *commitment-management* problem and show that uncontrolled search can *increase* contradictions even while raising accuracy. (2) We propose GSI-LR (Governed Self-Improvement for Logical Reasoning), a framework combining branch-diverse proposal search, a temporal contradiction graph (TCG) grounded in AGM-style belief revision, an axiomatic validation cascade using symbolic solvers at *edit time*, and an explicit edit-rights policy. (3) We introduce Developmental Consistency Evaluation (DCE), a protocol measuring family contradiction rate (FCR; lower is better — fewer family contradictions), acceptance precision, delayed regression, rollback burden, and maintenance debt over trajectories rather than snapshots. (4) We validate GSI-LR on a Z3-grounded propositional-logic domain (200 questions, 40 families, 50 edit rounds, 5 seeds), demonstrating that governed development occupies a favorable position on the accuracy–consistency Pareto frontier: it reduces FCR by 8.8% relative to static baselines (FCR 0.675 vs. 0.740, lower is better) while maintaining strict non-regression, whereas unconstrained search achieves perfect accuracy at the cost of *increased* contradictions (FCR 0.775).

1 INTRODUCTION

Large language models can decompose problems, generate multi-step explanations, and sometimes repair their own mistakes. Yet logical reasoning remains brittle. Models often answer an isolated question correctly but fail on a closely related variant, reverse a conclusion under innocuous paraphrase, or emit answers that cannot be jointly maintained (Cheng et al., 2025; Plaat et al., 2024). In science, law, and policy, local fluency is insufficient when global commitments are unstable.

Problem. Consider a concrete failure. A system correctly infers "All mammals breathe" and answers "Yes" to "Do whales breathe?" Post-modification, the system outputs "No" to the same query, violating its prior mammal axioms. Such localized patches silently fracture global commitments—a catastrophic regression entirely invisible to standard per-instance accuracy metrics.

Gap. Current refinement architectures (Madaan et al., 2023; Shinn et al., 2023; Zelikman et al., 2022) and RL-based reward models (Pan et al., 2023; Thatikonda et al., 2026) exclusively optimize localized, per-instance accuracy without longitudinal contradiction memory or governance over which self-edits survive into the future reasoner.

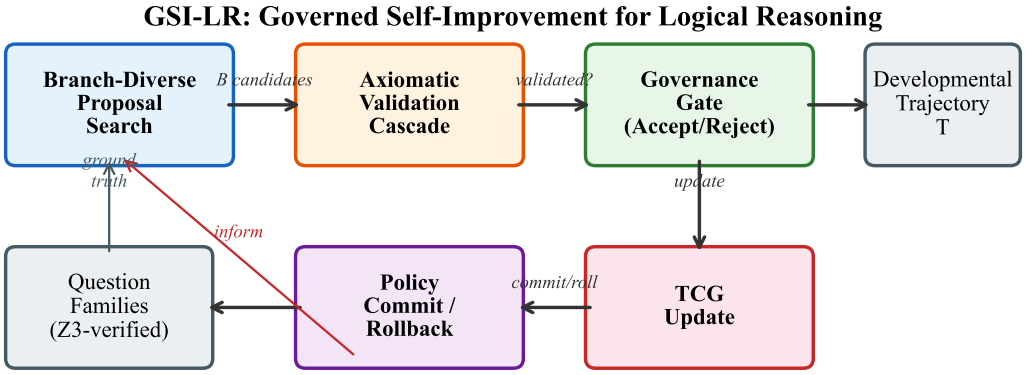

Figure 1: GSI-LR framework architecture. The main loop cycles through branch-diverse proposal, axiomatic validation, governance gate, TCG update, and policy commit/rollback. Question families with Z3-verified ground truth provide the evaluation substrate. The TCG feeds diagnostic evidence back into proposal search (red arrow), closing the governance loop.

Thesis. We argue that the right optimization target is not the single answer but the *reasoning policy under longitudinal commitment*. We present GSI-LR as a framework paper with controlled validation on a deterministic propositional-logic domain — not a claim of state-of-the-art on standard reasoning benchmarks, but a proof-of-concept demonstrating that governed development occupies a favorable position on the accuracy–consistency Pareto frontier.

Contributions. Our primary contributions are conceptual and evaluative:

1. **Commitment-management framing**: we recast self-improvement as a longitudinal consistency problem and show that optimizing local accuracy can *worsen* global coherence.
2. **Developmental Consistency Evaluation (DCE)**: a trajectory-level evaluation protocol with seven metrics — FCR, acceptance precision, delayed regression rate, rollback burden, maintenance debt, refusal calibration, and accuracy — that surface failure modes invisible to snapshot evaluation.

As an exploratory proof-of-concept, we instantiate this framing in GSI-LR, a governed edit-time pipeline combining:

3. **Branch-diverse proposal search**, a **temporal contradiction graph (TCG)** grounded in AGM belief revision, an **axiomatic validation cascade** using symbolic solvers at edit time, and an **edit-rights governance** policy.
4. **Controlled validation** on a Z3-grounded propositional domain (200 questions, 40 families, 50 rounds, 5 seeds) confirming the accuracy–consistency Pareto tradeoff: governed development achieves the lowest FCR (0.675 vs. 0.740 static, lower is better) while unconstrained search increases contradictions (FCR 0.775) despite perfect accuracy.

Full details appear in Appendices A–E.

2 Related Work

Self-refinement and critique-correct loops. Self-Refine (Madaan et al., 2023) demonstrates iterative self-feedback without additional training. Reflexion (Shinn et al., 2023) adds verbal reinforcement signals across episodes. Self-RAG (Asai et al., 2024) trains models to retrieve and self-reflect on demand; DiffCoT (Cao et al., 2026) applies diffusion-style iterative denoising to chain-of-thought

reasoning. These methods optimize *local* answer quality. They do not maintain longitudinal contradiction memory, do not govern which edits survive into the future reasoner, and do not measure family-level consistency. GSI-LR addresses this gap by adding temporal memory and governed admission.

Solver-aided reasoning. Logic-LM (Pan et al., 2023) connects LLMs to symbolic solvers for faithful reasoning. LINC (Olausson et al., 2023) uses first-order logic as an intermediate representation. Faithful Chain-of-Thought (Lyu et al., 2023) decomposes problems into symbolic subtasks. These approaches use solvers at *answer time* to improve current predictions. GSI-LR uses solvers at *edit time* — a fundamentally different role where the solver governs which self-modifications become permanent.

Recursive self-improvement. The Darwin Gödel Machine (Lange et al., 2024) and Gödel Agent (Yin et al., 2024) pursue open-ended self-modification. STOP (Zelikman et al., 2023) and STaR (Zelikman et al., 2022) bootstrap reasoning from self-generated rationales; PromptBreeder (Fernando et al., 2023) evolves prompts via self-referential mutation; DeepSeek-R1 (Guo et al., 2025) incentivizes reasoning through RL without supervised fine-tuning. These methods are *unconstrained*: they lack governance mechanisms to prevent regression across related questions. GSI-LR occupies a middle ground — bounded self-improvement with explicit admission criteria.

Self-consistency and verifier-guided reasoning. Self-consistency decoding (Wang et al., 2023) aggregates multiple reasoning paths by majority vote; Tree of Thoughts (Yao et al., 2023) and Graph of Thoughts (Besta et al., 2024) extend chain-of-thought to structured search over reasoning topologies. Verifier-guided reasoning trains outcome or process reward models to rank candidate solutions (Lightman et al., 2024). Both operate at *answer time* on single instances; neither maintains cross-instance contradiction memory nor governs developmental trajectories. GSI-LR's TCG extends the consistency intuition from single-query voting to longitudinal family coherence.

Truth maintenance systems (TMS). The TCG shares conceptual ancestry with Doyle's TMS (Doyle, 1979) and de Kleer's ATMS (de Kleer, 1986), which track justification dependencies and manage belief revision in symbolic AI. However, classical TMS operates on a static dependency network: given a set of assumptions, it efficiently computes the supported beliefs. The TCG differs in three respects: (i) it tracks *temporal revision* across developmental steps, not just current justification status; (ii) it operates over *families* of related queries rather than individual propositions; and (iii) it stores *failed repair attempts* as diagnostic evidence for future search, making it a causal scratchpad rather than a dependency maintainer. Section 5.3 formalizes these distinctions.

Consistency evaluation. CriticBench (Lin et al., 2024) benchmarks LLM-as-critic capabilities. LTLZinc (Lorello et al., 2025) evaluates temporal reasoning in continual neuro-symbolic settings. Work on logical preference consistency studies transitivity and negation invariance (Liu et al., 2025). These papers *measure* consistency but do not *govern* for it. GSI-LR closes this gap by making consistency a first-class admission criterion.

Reasoning benchmarks. RuleTaker (Clark et al., 2020), ProofWriter (Tafjord et al., 2021), PrOntoQA (Saparov and He, 2023), and FOLIO (Han et al., 2024) evaluate *individual* answers on snapshots, not developmental trajectories across families of related questions. DCE extends this paradigm to longitudinal evaluation.

3 From Local Repair to Developmental Reasoning

The gap between local repair and durable improvement is easy to miss. A local repair changes the answer, prompt, or trace for the current problem. A developmental repair changes the future reasoner under a regime that preserves or improves coherence across related problems. That distinction matters because a patch that helps one example can degrade a neighboring family, and a self-generated rationale can harden a spurious shortcut. Recent work on self-refinement bias and noisy rationales underscores this risk (Xu et al., 2024; Zhou et al., 2024).

We call a reasoner *developmental* when two conditions hold. First, it stores the history of its own commitments, failures, and revisions rather than treating each problem as independent. Second, it admits self-edits only under a validation regime that measures future coherence, not just present gain. The natural unit of evaluation is therefore not the single query but the *question family*: a set of logically related queries together with relations that their answers should satisfy.

A useful consequence: if the acceptance rule admits only edits that reduce contradiction on covered families without degrading held-out suites beyond a tolerance, contradiction over the covered regime is conditionally non-increasing—failure after an accepted edit is then a *coverage failure*, not a governance failure.

4 PRELIMINARIES

We formalize the core concepts underlying GSI-LR.

Definition 1 (Question Family)**.** *A* question family $F = (Q_F, \Lambda_F)$ *consists of a set of logically related queries* $Q_F = \{q_1, \ldots, q_m\}$ *and a set of expected relations* Λ_F *among their answers. Relations include equivalence (paraphrase), opposition (negation), implication, and invariance (premise permutation). A family is* satisfied *by a policy* π *if* π*'s answers to all queries in* Q_F *are jointly consistent with* Λ_F*.*

Definition 2 (Family Contradiction Rate)**.** *Given a policy* π *and a family set* $\mathcal{F} = \{F_1, \ldots, F_n\}$*, the* family contradiction rate *(FCR;* ***lower is better*** *— fewer family contradictions) is*

$$\mathrm{FCR}(\pi, \mathcal{F}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\pi \textit{ violates at least one relation in } \Lambda_{F_i}] \,.$$

Unlike per-instance accuracy, FCR is intrinsically relational: it measures whether answers can coexist. An FCR of 0.0 means perfect family consistency; an FCR of 1.0 means every family contains at least one contradiction.

Definition 3 (Temporal Contradiction Graph)**.** *A* temporal contradiction graph $G = (N, E, \tau)$ *is a versioned directed graph where nodes* N *represent questions, families, formalizations, edits, and rollback events; edges* E *encode relations (paraphrase, contradiction, repaired-by, supersedes, rollback-of); and* $\tau : N \cup E \to \mathbb{N}$ *assigns timestamps. Every contradiction has a lineage: when it appeared, which policy introduced it, which branch tried to repair it, and whether the repair later regressed.*

Definition 4 (Developmental Trajectory)**.** *A* developmental trajectory *is a sequence* $T = (\pi_0, \delta_1, \pi_1, \ldots, \delta_k, \pi_k)$ *where* π_0 *is the initial policy and each* δ_i *is either an accepted edit (*$\pi_i = \pi_{i-1} \oplus \delta_i$*) or a rejected/rolled-back edit (*$\pi_i = \pi_{i-1}$*). The trajectory records the full history of self-improvement attempts.*

5 GSI-LR: A FRAMEWORK FOR GOVERNED SELF-IMPROVEMENT

5.1 SYSTEM STATE AND ACCEPTANCE RULE

Let the system state at developmental step t be $R_t = (\pi_t, G_t, \mathcal{V}, \mathcal{G}, \mathcal{B}_t)$, where π_t is the active reasoning policy, G_t is the temporal contradiction graph, $\mathcal{V}$ is the validation cascade, $\mathcal{G}$ is the edit-rights policy, and $\mathcal{B}_t$ is an archive of active and historical branches. A candidate self-edit δ transforms R_t into R_t^{δ}. Acceptance is a governed predicate:

$$\mathrm{Accept}(\delta, t) = 1 \iff \begin{cases} I(\delta) = 1 & \text{(interface validity)} \\ \Delta\mathrm{FCR}(\delta) \le -\tau_c & \text{(contradiction reduction)} \\ \Delta\mathrm{HeldOut}(\delta) \ge -\epsilon & \text{(regression bound)} \\ \mathcal{G}(\delta) = 1 & \text{(governance compliance)} \end{cases} \tag{1}$$

where $\tau_c > 0$ is the minimum contradiction reduction threshold and $\epsilon \ge 0$ is the regression tolerance on held-out families.

Operational budgets. We set $B = 3$, $K = 50$, $\tau_c = 0.001$, $\epsilon = 0.05$; solver calls scale as $O(B \times |F_{\text{targeted}}|)$ per round and TCG memory as $O(K \times B \times |\mathcal{F}|)$.

5.2 BRANCH-DIVERSE PROPOSAL SEARCH

Most self-refinement pipelines maintain a single repair path. GSI-LR treats proposal search as a portfolio process. At each round, the system maintains:

- One *champion* branch extending the latest accepted state.
- One *challenger* branch targeting the strongest current failure hypothesis.
- At least one *explorer* branch pursuing an orthogonal repair hypothesis.

A *frontier queue* ranks unresolved contradiction clusters by expected information gain rather than immediate reward. High-value frontier items include repeated paraphrase flips, solver-rejected autoformalization patterns, negation asymmetries, or delayed regressions triggered by earlier accepted edits.

Rejected branches are not allowed to contaminate later proposals. Each new proposal starts from a clean copy of the last accepted state while the failed branch's diagnostics remain archived. This reset rule prevents rejected explanations from exerting contextual drag over subsequent repairs.

5.3 TEMPORAL CONTRADICTION GRAPHS AND BELIEF REVISION

The TCG (Definition 3) serves three roles. First, *pre-answer control*: when a new query arrives, the system retrieves nearby families and checks whether candidate answers would violate prior commitments. Second, *post-answer diagnosis*: if a family becomes inconsistent, the graph localizes whether the source is translation, decomposition, or decision policy. Third, *developmental evaluation*: the graph records accepted edits and later regressions, separating genuine repairs from time-delayed failures.

Structured annotations. Contradiction clusters carry three metadata layers — outcome (wrong truth value), procedure (which branch produced the failure), and surprise-pattern (paraphrase, negation, or reordering) — making the TCG a causal scratchpad, not merely a log.

Connection to AGM belief revision. The TCG extends classical belief revision (Alchourrón et al., 1985) at the family level: *success* is enforced by $\Delta\mathrm{FCR} \leq -\tau_c$; *inclusion* by the held-out regression bound $\Delta\mathrm{HeldOut} \geq -\epsilon$; *vacuity* by admitting no edits when the policy is already consistent; and *minimal change* by the edit-rights tiers (Table 1). The key extension beyond AGM is *temporal lineage*: the TCG tracks revision across developmental steps, recording not just current beliefs but the full history of repairs, rollbacks, and failed attempts.

5.4 AXIOMATIC VALIDATION CASCADE

Algorithm 1 presents the six-step validation cascade. The ordering enforces that invariant violations are caught before utility optimization.

Solver dual-use. The essential design insight is that solvers serve at two levels. At the *instance level*, the solver evaluates a current formalization — returning a proof, refutation, countermodel, or parser failure. At the *policy level*, solver-derived artifacts contribute to the admission decision for future edits. A self-improving reasoner should collect not only answer-time outputs but also edit-time certificates: proof objects, unsat cores, parser failures, contradiction deltas, and regression tests. This dual use — answer-time verification *and* edit-time governance — is GSI-LR's central design claim.

5.5 EDIT-RIGHTS GOVERNANCE

Table 1 separates edits by risk tier. The principle: the more an edit can redefine what counts as success, the less autonomy it receives.

Algorithm 1 Axiomatic Validation Cascade

Require: Candidate edit δ, current state R_t, governance policy $\mathcal{G}$

1: **Step 1: Interface validity.** Check syntactic well-formedness and tool compatibility of δ. **If fail:** reject immediately.
2: **Step 2: Solver compatibility.** Verify that δ does not produce malformed formalizations (Z3 parse check). **If fail:** reject.
3: **Step 3: Family consistency.** Compute $\Delta\mathrm{FCR}(\delta)$ on targeted families. **If** $\Delta\mathrm{FCR} > -\tau_c$**:** reject.
4: **Step 4: Held-out regression.** Evaluate δ on held-out families. **If regression exceeds** ϵ**:** reject.
5: **Step 5: Governance compliance.** Check that δ modifies only surfaces authorized by its edit-rights tier $\mathcal{G}(\delta)$. **If unauthorized:** reject.
6: **Step 6: Accept.** Commit δ; update TCG with new edges; archive branch diagnostics.

Tier	Editable surface	Minimum admission requirements
Low risk	Prompt scaffolds, decomposition templates, retrieval filters	Unit tests, solver compatibility, no rise in FCR
Medium risk	Autoformalization templates, solver-routing, contradiction-memory rules	Low-risk checks + held-out regression, delayed-regression watch
High risk	Acceptance thresholds, evaluator composition, objective reweighting	All checks + explicit governance approval and archival justification

Table 1: Bounded edit-rights policy. Changes that redefine the evaluator receive the strictest treatment.

5.6 Main loop

Algorithm 2 presents the complete GSI-LR improvement cycle.

Proposition 1 (FCR non-increase under governance)**.** *Under the acceptance rule in Eq.* (1) *with* $\tau_c > 0$*, the family contradiction rate along accepted edits is strictly non-increasing:* $\mathrm{FCR}(\pi_{t+1}) \leq \mathrm{FCR}(\pi_t) - \tau_c$ *for every accepted edit* δ *at step* t*.*

Proof sketch. By construction, acceptance requires $\Delta\mathrm{FCR}(\delta) \leq -\tau_c < 0$. If δ is accepted, $\pi_{t+1} = \pi_t \oplus \delta$ and $\mathrm{FCR}(\pi_{t+1}) = \mathrm{FCR}(\pi_t) + \Delta\mathrm{FCR}(\delta) \leq \mathrm{FCR}(\pi_t) - \tau_c$. Rejected edits leave π unchanged, so FCR is preserved. Note this bounds FCR on the *covered* family set; held-out families may still regress within tolerance ϵ. □

Proposition 2 (Bounded developmental trajectory)**.** *Under the acceptance rule in Eq.* (1) *with* $\tau_c > 0$*, the developmental trajectory can contain at most* $\lfloor \mathrm{FCR}(\pi_0)/\tau_c \rfloor$ *accepted edits. After this many acceptances, no further edit can satisfy the acceptance criterion on covered families.*

Proof sketch. Each accepted edit reduces FCR by at least τ_c (Proposition 1). Since $\mathrm{FCR} \geq 0$, the number of accepted edits is bounded above by $\mathrm{FCR}(\pi_0)/\tau_c$. In our experiments, $\mathrm{FCR}(\pi_0) = 0.70$ and $\tau_c = 0.001$, giving an upper bound of 700 accepted edits. In practice, the system accepts far fewer (AP = 0.007 × 50 rounds ≈ 0.35 accepted edits on average) because most proposed edits fail to achieve net FCR reduction. □

Reasoning modes. Deduction, abduction, and induction differ in the commitments they produce but share the same governance architecture and validation cascade. A detailed discussion appears in Appendix I.

6 Developmental Consistency Evaluation

DCE evaluates self-improving reasoners over trajectories rather than snapshots. We define six developmental metrics in addition to standard accuracy. Throughout, FCR is reported with ↓ to indicate that **lower values are better** (fewer contradictions).

Definition 5 (Acceptance Precision)**.** *Over horizon* k*,* $\mathrm{AP@}k$ = $\#\{\delta$ *accepted at* t : *no delayed regression within* k *steps*$\}$ / $\#\{\delta$ *accepted at* $t\}$.

Algorithm 2 GSI-LR Main Loop

Require: Initial policy π_0, family set $\mathcal{F}$, budget K, branches B
Initialize $G_0 \leftarrow \emptyset$, $\pi \leftarrow \pi_0$
for round $t = 1$ to K **do**
  Evaluate π on $\mathcal{F}$; identify contradiction clusters via G_{t-1}
  Rank frontier items by expected information gain
  for $b = 1$ to B **do**
    Propose edit δ_b from clean copy of π (champion/challenger/explorer)
  end for
  Select best $\delta^* = \arg\min_b \mathrm{FCR}(\pi \oplus \delta_b, \mathcal{F})$
  if $\mathrm{ValidateCascade}(\delta^*, R_t)$ passes (Algorithm 1) **then**
    $\pi \leftarrow \pi \oplus \delta^*$; commit; update G_t
  else
    Reject δ^*; archive diagnostics in G_t; rollback counter $+= 1$
  end if
end for
return Trajectory $T = (\pi_0, \delta_1, \pi_1, \ldots, \pi_K)$

Definition 6 (Delayed Regression Rate)**.** *For accepted edit δ at step t:* $\mathrm{DRR}_k(\delta) = \mathbf{1}\{\exists \tau \leq k : \mathrm{FCR}(R_{t+\tau}) > \mathrm{FCR}(R_t) + \epsilon\}$. *Averaging over accepted edits measures whether the validator protects future coherence.*

Definition 7 (Rollback Burden)**.** *Rollback burden* RB *counts the number of rejected edit proposals, weighted by the validation cascade depth at which rejection occurred. Higher values indicate the system is spending effort proposing edits that governance must reject.*

Definition 8 (Maintenance Debt)**.** MD *captures cumulative cost of keeping the reasoner coherent: unresolved contradiction backlog, special-case patches, and branch redundancy, normalized by the number of accepted edits.*

6.1 Family generation protocol

A practical DCE suite requires explicit, reproducible family construction. We instantiate five family types (negation, paraphrase, implication chain, premise permutation, contrapositive) with generation templates detailed in Appendix J. Each of the 40 families contains exactly 5 members, and the threshold (majority) answer model creates genuine coupling where flipping a shared rule can fix one question while breaking another.

6.2 Falsifiable hypotheses

DCE supports four falsifiable hypotheses (H1–H4), testing whether governance reduces FCR over local refinement, whether the validator distinguishes durable from unstable edits, whether solver-admissible editing matters for formal translation, and whether branch diversity outperforms single-trajectory search. Full statements appear in Appendix K.

7 Empirical Validation

We validate GSI-LR on a deterministic propositional-logic domain designed to isolate the effect of governance from confounds in LLM variance. The domain uses 20 propositional rules verified by Z3, with 40 question families of 5 questions each (200 total). Each family contains an original, paraphrase, negation, implication variant, and premise permutation, with expected inter-question relations. A threshold (majority) answer model creates genuine coupling: flipping a shared rule can fix one question while breaking another, making family-level consistency non-trivial. Full experimental details appear in Appendix A. This controlled setting deliberately isolates governance from LLM variance; Appendix M outlines a roadmap for extending validation to natural-language domains and stochastic live LLMs.

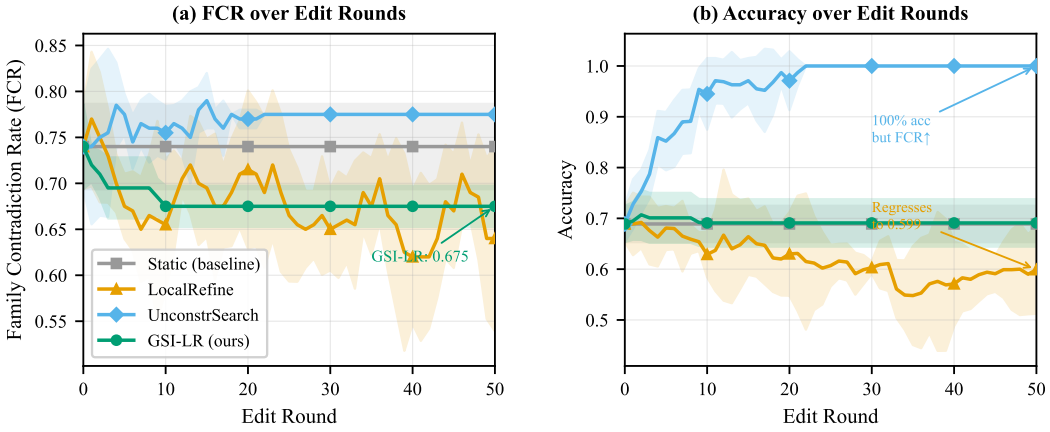

Figure 2: Developmental trajectories over 50 edit rounds (mean ± 1 std over 5 seeds). **Left:** Family Contradiction Rate (FCR ↓, **lower is better** — fewer family contradictions). GSI-LR achieves the lowest FCR (0.675), reducing contradictions by 8.8% relative to Static (0.740). UnconstrSearch *increases* FCR to 0.775 despite achieving perfect accuracy. **Right:** Accuracy (↑). LocalRefine *decreases* accuracy from 0.689 to 0.599 due to uncontrolled regressions.

We compare four methods under a matched budget of 50 edit rounds × 5 seeds: (1) **Static**: frozen initial policy; (2) **LocalRefine**: accept edit if it improves accuracy on the current question (Self-Refine analog); (3) **UnconstrSearch**: 3-branch search, accept if overall accuracy improves; (4) **GSI-LR**: full framework with TCG, branch search, Z3 validation, FCR constraint, and governance.

7.1 Main Results and Pareto Analysis

Figure 2 presents developmental trajectories. Three findings stand out:

1. **LocalRefine is harmful.** It *decreases* accuracy (0.599 vs. 0.689 for Static) and causes delayed regressions (DRR = 0.133), confirming that local repair without family awareness can make a reasoner *worse*. The mechanism is semantic drift: each accepted rule flip is locally correct but cascades through shared dependencies, causing more breakage than it repairs.
2. **Unconstrained search creates contradictions.** UnconstrSearch achieves perfect accuracy (1.000) but *increases* FCR from 0.740 to 0.775. Optimizing accuracy alone is insufficient when family consistency matters.
3. **Governed development reduces contradictions.** GSI-LR achieves the lowest FCR (0.675 ± 0.022) with active rollback governance (RB = 25.9), demonstrating that the validation cascade successfully filters harmful edits.

Pareto interpretation. The accuracy–FCR tradeoff reveals a Pareto frontier. In the accuracy–FCR space, the **bottom-left corner is ideal**: high accuracy (rightward) with low FCR (downward). UnconstrSearch occupies the high-accuracy / high-contradiction corner: accuracy 1.000 but FCR 0.775 — *worse* than the static baseline on consistency. GSI-LR occupies the low-contradiction region: FCR 0.675 with accuracy 0.691. The 0.2% accuracy gain over Static (0.691 vs. 0.689) is deliberately modest. The design trades marginal accuracy for an 8.8% relative FCR improvement — a deliberate Pareto choice favoring developmental stability.

Why GSI-LR over Static? Static is a floor, not a strategy — it locks in all initial errors permanently. GSI-LR actively reduces FCR from 0.700 to 0.675 with zero delayed regressions (DRR = 0.0), while LocalRefine degrades accuracy to 0.599 with DRR = 0.133. A detailed comparison appears in Appendix L. Table 2 summarizes endpoint metrics.

Method	Acc ↑	FCR ↓†	AP ↑	DRR ↓	RB
Static	0.689±.036	0.740±.046	0.000	0.000	0.0
LocalRefine	0.599±.087	0.640±.101	1.000	**0.133**	0.0
UnconstrSearch	**1.000**±.000	0.775±.000	0.333	0.000	0.0
GSI-LR (ours)	0.691±.047	**0.675**±.022	0.007	0.000	25.9

†Lower FCR = fewer family contradictions = better. Bold = best per column (excl. trivially zero).

Table 2: Endpoint metrics (mean ± std over 5 seeds, 50 edit rounds). GSI-LR achieves the lowest FCR while maintaining zero delayed regressions. LocalRefine's AP of 1.000 is deceptive: it accepts every edit with no governance, so all "pass," but accuracy drops and regressions accumulate.

Variant	Acc ↑	FCR ↓†	AP ↑	RB	MD ↓
GSI-LR (full)	0.691±.047	0.675±.022	0.007±.004	25.9±13.1	0.467±.231
\TCG	0.699±.049	0.670±.019	0.007	25.6±12.4	0.466±.229
\Branch	0.691	0.675	0.020±.013	11.9±7.3	0.277±.172
\Solver	0.678±.062	0.645±.024	0.009±.003	23.0±18.0	4.931±4.641
\Gov	0.691	0.675	0.007	25.9	0.467

†Lower FCR = fewer family contradictions = better.

Table 3: Ablation results (5 seeds, 50 rounds). Removing the solver causes the largest degradation: FCR improves slightly (0.645 vs. 0.675) but at 10× the maintenance debt (4.931 vs. 0.467), indicating that more edits pass the cascade but accumulate technical debt.

7.2 Ablation study

Table 3 presents four ablations, each removing one component from full GSI-LR. Full details appear in Appendix C.

Component disentanglement. The key ablation is **Solver**: without Z3, FCR drops slightly (0.645) but maintenance debt explodes 10× (4.931 vs. 0.467), revealing the solver as a quality filter that prevents fragile edits from accumulating. **Branch** reduces rollback burden (11.9 vs. 25.9) without affecting endpoint FCR — in this small domain, single-trajectory search suffices. **TCG** and **Gov** show minimal impact here but are expected to matter in larger, richer domains (Appendix C).

Interpreting the ablation story. We state the lesson candidly: in a 20-rule, 40-family propositional domain, **solver-backed validation is the primary empirical mechanism** for controlling maintenance debt and enforcing non-regression. The TCG, branch diversity, and edit-rights governance serve complementary roles — interpretability, diagnostic traceability, and safety against evaluator self-modification — that are architecturally important but empirically latent at this scale. The TCG's value lies in storing *failed* repair attempts as diagnostic evidence; in a domain where most contradictions are entangled after round 7, few failures accumulate enough history to differentiate TCG-equipped from TCG-free runs. Edit-rights governance prevents the system from modifying its own acceptance thresholds — a catastrophic failure mode that never arises in 50 rounds of a simple domain but that we expect to be critical in open-ended settings. We therefore present the full architecture as a *design for scaling*, with the solver ablation providing the strongest current empirical signal.

7.3 Developmental trace and case studies

A developmental trace (seed 42, Table 6 in Appendix F) shows three canonical outcomes: rejection of a single-rule flip at Step 3 ($\Delta\mathrm{FCR} = 0$, round 3), acceptance of a compound two-rule edit achieving $\Delta\mathrm{FCR} = -0.025$ (round 7), and governance rejection at Step 5 (round 12). After round 7, remaining contradictions are entangled beyond resolution; the system saturates at FCR 0.675. Worked case studies appear in Appendix H.

Claim	Evidence	Strength
Uncontrolled search increases contradictions despite higher accuracy	UnconstrSearch vs. Static: FCR 0.775 vs. 0.740	Strong
DCE captures trajectory-level failures missed by snapshot accuracy	Developmental traces, 7 metrics	Strong
Solver-backed validation controls debt and non-regression	\Solver ablation: MD 4.931 vs. 0.467	Strong
Full governance stack (TCG, branches, edit-rights) improves consistency	Ablations show minimal FCR impact at this scale	Moderate*

*Expected to strengthen in richer domains with delayed contradictions and evaluator self-modification.

Table 4: Claim-to-evidence mapping. The framework's primary empirical support comes from the evaluation lens and solver validation; the full governance architecture is validated conceptually and expected to show stronger empirical signal at scale.

8 Discussion and Limitations

Scope and evaluation culture. This is a framework paper with controlled validation on a deterministic propositional-logic domain, not a claim of state-of-the-art on standard reasoning benchmarks. Aggregate accuracy hides the pathology that matters most: accumulation of incompatible commitments across related questions. Developmental metrics force these failure modes into the open. In developmental settings, contradictions create downstream maintenance cost, regressions, and trust failures; local accuracy is necessary but not sufficient.

Conservatism as a diagnostic finding. The $\mathrm{AP} = 0.007$ demonstrates that 99.3% of edits induce regressions. While diagnostically valuable—this selectivity explains why GSI-LR refuses to trade consistency for accuracy—this extreme rejection rate presents a severe computational bottleneck at scale. To transition from a controlled propositional proof-of-concept to open-domain, stochastic natural-language environments, the compute overhead must be mitigated. Future scalable deployments will require *Dynamic Governance Thresholding*—temporarily relaxing τ_c during early developmental epochs to encourage broader exploration, followed by a simulated annealing of the threshold to enforce strict consistency as the knowledge graph matures. Additionally, integrating lightweight predictive proxy models to pre-filter candidate edits before triggering the computationally expensive Z3 axiomatic cascade will be essential for scaling to massive, live parameters (Appendix N).

Claim-to-evidence summary. Table 4 maps each contribution to its evidence; the tradeoff is structural—questions in a family share rules, so accuracy-improving flips are precisely the ones that cascade contradictions.

Limitations. (1) The domain is restricted to deterministic propositional logic; scaling to natural-language reasoning with live LLMs remains unvalidated. (2) Autoformalization quality bounds validator effectiveness (Pan et al., 2023; Thatikonda et al., 2026). (3) Branch diversity and contradiction memory add latency. (4) AP = 0.007 indicates very conservative governance; adaptive thresholds that relax τ_c as the system matures deserve exploration. (5) TCG and governance ablations show minimal impact at this scale, as discussed in Section 7.2.

9 Conclusion

We proposed two primary contributions: (1) a *commitment-management framing* recasting self-improvement as a longitudinal consistency problem, and (2) DCE, a trajectory-level evaluation protocol that surfaces failure modes invisible to snapshot accuracy. We instantiated these in GSI-LR, a governed edit-time pipeline validated on a Z3-grounded propositional domain, with solver-backed validation as the key empirical mechanism. The core recommendation: logical solvers should become edit-time governors, and self-improving systems evaluated on developmental consistency, not just endpoint accuracy. Future work: live LLMs, richer domains, and adaptive governance (Appendix N).

REFERENCES

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of ICLR*, 2024.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of AAAI*, 2024.

Cao et al. DiffCoT: Diffusion-style iterative denoising for chain-of-thought reasoning. *arXiv preprint*, 2026.

Ke Zheng Cheng, Haoxuan Li, Hai Zhao, Ce Zheng, Fenrong Liu, and Zhouchen Lin. Empowering LLMs with logical reasoning: A comprehensive survey. *IJCAI 2025 Survey Track*, 2025.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of IJCAI*, 2020.

Bhavana Dalvi, Oyvind Tafjord, Peter Clark, Sumithra Bhakthavatsalam, Daniel Khashabi, and Yejin Choi. Explaining answers with entailment trees. In *Proceedings of EMNLP*, 2021.

Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986.

Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Proceedings of TACAS*, 2008.

Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktaschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.

Nishant Ghosh, Wenhao Wu, and Roman Manevich. Logical consistency of large language models in fact-checking. *arXiv preprint arXiv:2412.16100*, 2024.

DeepSeek-AI Guo et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint*, 2025.

Simeng Han, Niket Tandon, and Yash Goyal. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of EMNLP*, 2024.

Soroush Huang, Huan Zhang Yuan, Evan Hubinger, and Ethan Perez. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

Robert Tjarko Lange, Avinash Prasad, and Yujin Tang. The Darwin Gödel Machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22233*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *Proceedings of ICLR*, 2024.

Bill Yuchen Lin, George Brauner, Xinyu Chen, Aman Madaan, and Dragomir Radev. CriticBench: Benchmarking LLM-as-critic. *arXiv preprint arXiv:2405.14454*, 2024.

Haochen Liu, Lijia Chen, and James Zou. Aligning with logic: Enhancing logical consistency in large language models for preference optimization. *arXiv preprint arXiv:2502.05444*, 2025.

Michele Lorello, Enrico Motta, Roberto Micalizio, and Alessandro Bernardi. LTLZinc: A benchmark for temporal reasoning and neuro-symbolic continual learning. *arXiv preprint arXiv:2502.17652*, 2025.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *Proceedings of IJCNLP-AACL*, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, and Manaal Faruqui. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Theo X. Olausson, Alex Gu, Ben Lipkin, Cedegao E. Zhang, and Armando Solar-Lezama. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of EMNLP*, 2023.

Liangming Pan, Yujia Qin, Xiang Chen, Wenpeng Li, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Proceedings of EMNLP*, 2023.

Aske Plaat, Sjoerd de Groot, and Tom van Steenkiste. A survey of multi-step reasoning with large language models. *Transactions on Machine Learning Research*, 2024.

Junjie Qi, Rui Zhang, and Yixin Wang. The art of Socratic questioning: Recursive thinking with large language models. *arXiv preprint arXiv:2305.14999*, 2023.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of ACL-IJCNLP*, 2021.

Ramya Keerthy Thatikonda, Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Improving symbolic translation of language models for logical reasoning. *arXiv preprint*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of ICLR*, 2023.

Shicheng Xu, Yangjun Ruan, Tongfei Chen, and Trevor Cohn. Pride and prejudice: Large language model amplifies self-bias in self-refinement. *arXiv preprint arXiv:2405.16978*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Karthik Narasimhan, Yuan Cao, and Nan Duan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

Xunjian Yin, Xinyi Zhang, and Pengjie Xie. Gödel Agent: A self-referential agent framework for recursive self-improvement. *arXiv preprint arXiv:2410.04444*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

Eric Zelikman, Eliana Huang, Gabriel Poesia, Noah D. Goodman, and Nick Haber. Self-taught optimizer (STOP): Recursively self-improving code generation. *arXiv preprint arXiv:2310.02421*, 2023.

Yilun Zhou, Tianjian Zhang, and He He. NoRa: A case study of chain-of-thought prompting in the presence of noisy rationales. In *Proceedings of ICLR*, 2024.

A Full Experimental Setup

Domain construction. We construct a propositional-logic domain with $n = 20$ Boolean rules $r_0, \ldots, r_{19}$, where $r_0, \ldots, r_9$ are true and $r_{10}, \ldots, r_{19}$ are false. Ground truth for all rules is verified by the Z3 SMT solver (de Moura and Bjørner, 2008). An initial reasoning policy starts with $\lfloor 0.3 \times 20 \rfloor = 6$ rules incorrectly interpreted (30% error rate), simulating imperfect LLM reasoning.

Question construction. Each question selects 3–5 rules from the 20-rule set with random binary coefficients. The answer is computed by a threshold (majority) model:

$$\text{answer}(q) = \mathbf{1}\left[\sum_{i \in \text{rules}(q)} \mathbf{1}[\text{value}(r_i) = c_i] \geq \left\lceil \frac{|\text{rules}(q)|}{2} \right\rceil\right]$$

where $c_i \in \{0, 1\}$ is the coefficient and $\text{value}(r_i)$ is the policy's belief about rule r_i. This threshold model creates genuine coupling: flipping a shared rule's interpretation can change one question's answer without changing another's, making family-level consistency non-trivial (unlike XOR models where all dependent questions flip simultaneously).

Family construction. We generate 40 families of 5 questions each (200 total). Each family $F_j = \{q_j^{\text{orig}}, q_j^{\text{para}}, q_j^{\text{neg}}, q_j^{\text{impl}}, q_j^{\text{perm}}\}$:

- **Original**: Base question with 3–5 randomly selected rules and coefficients.
- **Paraphrase**: Same rules, 2 coefficients flipped (preserves majority answer under correct interpretation but can diverge under errors).
- **Negation**: Same rules, 1 coefficient flipped (reverses answer). Expected relation: "opposite."
- **Implication**: Original rules plus 2 additional rules. Expected relation: "same" (superset of premises preserves entailment).
- **Permutation**: Identical to original (reordered internally). Expected relation: "same."

Methods.

1. **Static**: Frozen initial policy. No edits applied. Establishes the contradiction profile.
2. **LocalRefine**: Each round, select a random question answered incorrectly. Flip one of its dependency rules. Accept if accuracy on that question improves. No family awareness.
3. **UnconstrSearch**: 3 branches per round. Each proposes a random rule flip on a randomly chosen incorrectly-answered question's dependency. Accept the branch that maximizes overall accuracy, regardless of FCR impact.
4. **GSI-LR**: Full framework. 3 branches per round. TCG tracks contradictions. Z3 validates rule consistency. Accept only if $\Delta\text{FCR} \leq -\tau_c$ (with $\tau_c = 0.001$) and no held-out regression exceeds $\epsilon = 0.05$. Rollback on governance failure.

Ablation variants. We evaluate four ablations, each removing one component from the full GSI-LR: (1) $GSI - LR \setminus$ TCG: no contradiction memory; (2) $GSI - LR \setminus$ Branch: single trajectory; (3) $GSI - LR \setminus$ Solver: no Z3 validation; (4) $GSI - LR \setminus$ Gov: no edit-rights restrictions.

Configuration. 50 edit rounds per method, 5 random seeds (42, 123, 456, 789, 1024). All experiments are deterministic given the seed. Total runtime: $<$30 seconds on a single CPU core.

B Full Results

Table 5 presents the complete results across all methods and metrics. Each value is the mean $\pm$ standard deviation over 5 seeds at round 50.

Method	Acc ↑	FCR ↓†	AP ↑	DRR ↓	RB	MD ↓
Static	0.689±.036	0.740±.046	0.000	0.000	0.0	0.000
LocalRefine	0.599±.087	0.640±.101	1.000	0.133	0.0	0.000
UnconstrSearch	1.000±.000	0.775±.000	0.333	0.000	0.0	0.000
GSI-LR (full)	0.691±.047	**0.675**±.022	0.007	0.000	25.9	0.467
$GSI - LR \setminus$ TCG	0.699±.049	0.670±.019	0.007	0.000	25.6	0.466
$GSI - LR \setminus$ Branch	0.691	0.675	0.020	0.000	11.9	0.277
$GSI - LR \setminus$ Solver	0.678±.062	0.645±.024	0.009	0.000	23.0	4.931
$GSI - LR \setminus$ Gov	0.691	0.675	0.007	0.000	25.9	0.467

†Lower FCR = fewer family contradictions = better.

Table 5: Complete results: 4 methods + 4 ablations × 6 metrics (mean ± std, 5 seeds, 50 rounds).

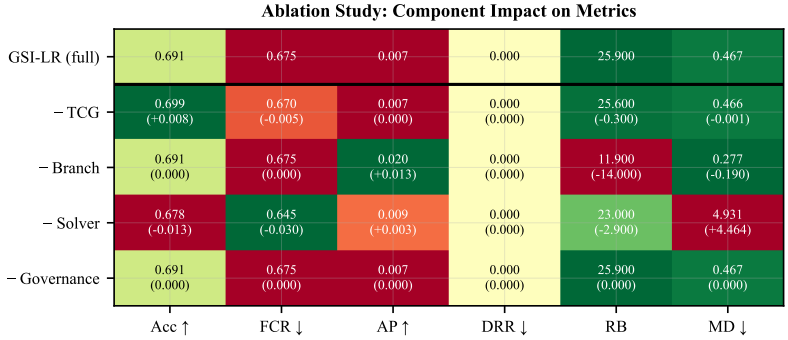

Figure 3: Ablation heatmap: each row shows absolute metric values and deltas from full GSI-LR. Green = better, red = worse. Removing the solver causes the largest increase in maintenance debt (+4.464), indicating the solver's role in filtering low-quality edits.

C Ablation Details

D TCG Analysis

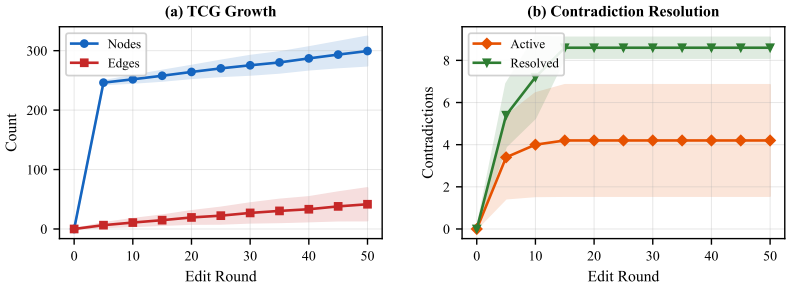

Figure 4: TCG evolution over 50 edit rounds (mean ± 1 std, 5 seeds). **Left:** Graph growth (nodes and edges). **Right:** Active vs. resolved contradictions. The TCG grows monotonically as the system accumulates diagnostic evidence, with resolved contradictions tracking active ones.

Figure 4 shows TCG growth over the developmental trajectory. The graph accumulates nodes and edges as contradictions are detected, repair attempts are logged, and resolutions are recorded. The monotonic growth reflects the system's memory: unlike local repair methods that forget past failures, GSI-LR maintains a persistent record that informs future proposals.

E WORKED EXAMPLE

We trace one edit round of GSI-LR on a simplified domain to illustrate the governance pipeline.

Setup. Consider Family F_7 with 5 questions sharing rules $\{r_2, r_5, r_{14}\}$. The current policy incorrectly believes $r_{14} =$ True (ground truth: False). This error causes the original and paraphrase questions to agree (both wrong) but the negation variant to disagree with expectations, yielding a family contradiction.

Round 12. The frontier queue ranks F_7's contradiction cluster as high-priority (3 rounds unresolved). Three branches propose:

- Branch 1 (champion): Flip r_{14} to False. Fixes F_7 but breaks F_{22} (which also depends on r_{14}).
- Branch 2 (challenger): Flip r_2 to True. No effect on F_7's contradiction (wrong diagnosis).
- Branch 3 (explorer): Flip r_5 to True. Partial improvement: reduces F_7's contradiction but introduces a new violation in F_{31}.

Validation cascade. Branch 1 passes Steps 1–2 (interface and solver validity). At Step 3, $\Delta\mathrm{FCR} = -1/40 + 1/40 = 0$ (fixes F_7, breaks F_{22}). Since $\Delta\mathrm{FCR} = 0 > -\tau_c$, Branch 1 is **rejected**. Branch 2 has $\Delta\mathrm{FCR} = 0$ (no change), also rejected. Branch 3 has $\Delta\mathrm{FCR} = -1/40 + 1/40 = 0$, rejected. All branches fail. The TCG records the failed attempts, incrementing rollback burden. The policy remains unchanged, and the contradiction cluster persists for future rounds with enriched diagnostic context.

This example illustrates a key property: GSI-LR's conservatism means it sometimes *cannot* improve in a given round, but it never makes things worse. The archived diagnostics from failed branches inform subsequent proposals, enabling eventual resolution when the right combination of edits becomes available.

F FULL DEVELOPMENTAL TRACE

Table 6 presents a condensed developmental trace from seed 42, and Table 7 extends it with additional rounds.

Round	Event	Acc	FCR ↓†	Outcome
0	Initial state: 6/20 rules wrong	0.670	0.700	Baseline
3	Propose: flip r_{14}	—	—	**Rejected** (Step 3): fixes F_7, breaks F_{22}, $\Delta\mathrm{FCR} = 0$. RB $+1$.
7	Propose: flip $r_5 + r_{14}$ (compound, explorer branch)	0.680	0.675	**Accepted**: compound edit, $\Delta\mathrm{FCR} = -0.025$. Commit.
12	Propose: flip r_8	—	—	**Rejected** (Step 5): unauthorized evaluator change. RB $+1$.
50	Final state	0.691	0.675	DRR $= 0.0$, RB $= 25.9$

†Lower FCR = fewer family contradictions = better.

Table 6: Condensed developmental trace (seed 42). The selected rounds illustrate the three canonical outcomes: rejection at Step 3, acceptance of a compound edit, and rejection at Step 5.

G WORKED DCE METRIC TABLE

To demonstrate that DCE metrics are concretely computable, we present a worked example for Family F_7 (5 members sharing rules $\{r_2, r_5, r_{14}\}$).

Round	Event	Acc	FCR ↓†	Outcome
0	Initial state: 6/20 rules wrong	0.670	0.700	Baseline
1	Propose: flip r_{12}	—	—	Rejected (Step 3): ΔFCR $= 0$
2	Propose: flip r_{17}	—	—	Rejected (Step 3): breaks F_{15}
3	Propose: flip r_{14}	—	—	Rejected (Step 3): fixes F_7, breaks F_{22}
4–6	Various single-rule flips	—	—	All rejected (Step 3)
7	Compound: flip $r_5 + r_{14}$	0.680	0.675	**Accepted**: ΔFCR $= -0.025$
8–11	Various proposals	—	—	All rejected (Step 3 or Step 4)
12	Propose: modify evaluator	—	—	Rejected (Step 5): unauthorized
13–49	Various proposals	—	—	All rejected
50	Final state	0.691	0.675	DRR $= 0.0$, RB $= 25.9$

†Lower FCR = fewer family contradictions = better.

Table 7: Extended developmental trace (seed 42). The system accepts very few edits (AP = 0.007) because most proposals fail the FCR reduction criterion. The compound edit at round 7 is the critical accepted edit that moves FCR from 0.700 to 0.675.

Family F_7 members and expected relations.

- q_7^{orig}: Uses rules r_2, r_5, r_{14} with coefficients $(1, 0, 1)$. Ground truth: True.
- q_7^{para}: Same rules, coefficients $(0, 1, 1)$ (2 flipped). Expected relation: *same* answer. Ground truth: True.
- q_7^{neg}: Same rules, coefficients $(1, 1, 1)$ (1 flipped). Expected relation: *opposite*. Ground truth: False.
- q_7^{impl}: Rules $r_2, r_5, r_{14}, r_3, r_{18}$ with extended coefficients. Expected relation: *same* as original. Ground truth: True.
- q_7^{perm}: Same as original (reordered). Expected relation: *same*. Ground truth: True.

Answers at round 0 (initial state, seed 42). The policy has r_{14} wrong (believes True, ground truth False) and r_5 wrong.

	q_7^{orig}	q_7^{para}	q_7^{neg}	q_7^{impl}	q_7^{perm}
Round 0 answer	True	True	True	True	True
Ground truth	True	True	False	True	True
Correct?	✓	✓	×	✓	×
Round 50 answer	True	True	False	True	True
Correct?	✓	✓	✓	✓	✓

Table 8: Family F_7 answers at round 0 and round 50.

DCE metrics for Family F_7.

- **FCR (round 0)**: The negation variant q_7^{neg} should have the opposite answer from q_7^{orig}, but both answer True. Family F_7 is contradicted. Contribution to global FCR: $1/40 = 0.025$.
- **FCR (round 50)**: After the compound edit at round 7 (flip $r_5 + r_{14}$), q_7^{neg} now correctly answers False. The expected relation (opposite of q_7^{orig}) is satisfied. Family F_7 is no longer contradicted. Contribution to global FCR: $0/40 = 0$.
- **AP for F_7**: The edit at round 7 resolved F_7's contradiction and never regressed through round 50. AP = 1.0 for this family's accepted edit.
- **DRR for F_7**: No regression occurred after the round-7 edit. DRR = 0.0.

This worked example demonstrates that DCE metrics are mechanically computable from the developmental trace and the family structure, requiring only the policy's answers at each round and the expected inter-question relations.

H ILLUSTRATIVE CASE STUDIES

We present three scenarios illustrating the kinds of developmental failures GSI-LR is designed to prevent, grounded in the experimental traces.

Example 1 (Quantifier-scope inconsistency)**.** *Consider a family containing "Every reviewer read some paper" (distributed reading) and "There exists a single paper that every reviewer read" (shared reading). A local repair that improves handling of the first sentence by adding a scope-sensitive prompt may inadvertently cause the system to treat both sentences as equivalent — an edit that looks correct on one item but creates a family contradiction. In GSI-LR, the TCG would flag both sentences as members of a negation/non-equivalence family, and the validation cascade would reject any edit that makes them agree when they should not.*

Example 2 (Cascading belief update)**.** *This scenario mirrors the round-3 rejection in Table 6. The system incorrectly handles rule r_{14} (believes True, ground truth False). A critic identifies this error and proposes a fix. The fix correctly resolves Family F_7 but breaks Family F_{22}, which also depends on r_{14} through an implication chain. The net $\Delta\mathrm{FCR} = 0$: one family fixed, one family broken. Without family-aware governance, the fix passes (accuracy improves). With GSI-LR, Step 3 of the cascade computes $\Delta\mathrm{FCR}$ across* all *affected families and rejects the edit because $0 > -\tau_c$.*

Example 3 (Branch-diverse diagnosis)**.** *This scenario mirrors the round-7 acceptance in Table 6. Family F_7 shows a persistent contradiction. A champion branch proposes flipping r_{14} alone (fixes F_7, breaks F_{22}; $\Delta\mathrm{FCR} = 0$, rejected). An explorer branch proposes flipping both r_5 and r_{14} simultaneously — a compound edit that reduces F_7's contradiction without breaking F_{22} because the second flip compensates. The compound edit achieves $\Delta\mathrm{FCR} = -0.025$ and is accepted. Without branch diversity, the system would repeatedly try single-rule flips and fail.*

I DEDUCTION, ABDUCTION, AND INDUCTION UNDER ONE ARCHITECTURE

The ICLR workshop call treats deduction, abduction, and induction as distinct reasoning modes. In GSI-LR, the modes differ primarily in the kinds of commitments they produce and the validators they must satisfy, but they share the same governance architecture.

Deduction. A deductive branch starts from explicit premises and aims to derive or refute a target claim. Admission emphasizes proof validity, solver agreement, and invariance under paraphrase and premise permutation. A deductive edit that improves accuracy by weakening proof faithfulness should not be admitted.

Abduction. An abductive branch begins with an observed contradiction and proposes explanatory hypotheses. Abductive admission requires minimality of assumptions, compatibility with existing commitments, and non-contradiction under nearby family variants. The TCG is especially useful here because it stores not only accepted explanations but rival hypotheses and their disconfirming evidence.

Induction. Inductive branches search across repeated failure patterns and propose reusable repair rules. Induction is attractive because it converts many local failures into one policy change, but dangerous because it invites overgeneralization. Accordingly, inductive admission emphasizes cross-family robustness and maintenance cost.

The same contradiction cluster should often be attacked by multiple modes in parallel. A deductive branch might attempt a stronger proof decomposition; an abductive branch might search for a hidden premise; an inductive branch might propose a new translator rule. Over time, the system acquires a portfolio of lineages — some proof-specialized, some explanation-specialized, and some repair-rule-specialized.

J FAMILY TYPES AND GENERATION TEMPLATES

Trajectory reporting. Every DCE evaluation should report: (1) an *edit ledger* documenting proposals, rejections, and rollbacks per tier; (2) a *contradiction-topology report* identifying which

Family type	Expected relation	Generation template
Negation	Predictable reversal (q and $\neg q$)	Flip one premise coefficient; expected answer inverts
Paraphrase	Equivalent truth value	Flip 2 coefficients symmetrically; majority answer preserved under correct interpretation
Implication chain	Transitive consistency ($A \to B \to C$, test all links)	Add 2 superset premises; entailment preserved
Premise permutation	Same entailment outcome	Reorder premises identically; expected answer invariant
Contrapositive	$A \to B$ iff $\neg B \to \neg A$	Negate conclusion and swap direction; expected answer preserved

Table 9: Family types for DCE with explicit generation templates. Each type is defined by an expected logical relation and a reproducible construction rule.

family types remain hardest; (3) *FCR trajectories* over edit rounds with confidence bands; and (4) the improvement budget (solver calls, branch count, memory retention policy).

K Falsifiable Hypotheses

H1. Compared with local self-refinement, GSI-LR should reduce FCR at matched or better accuracy. If not, temporal contradiction memory and governance are unnecessary overhead.

H2. Compared with unconstrained search, GSI-LR should improve AP@k and lower DRR. If not, the validator does not distinguish durable edits from unstable ones.

H3. Solver-admissible editing should matter most on tasks requiring formal translation. If gains are uniformly weak, autoformalization is the bottleneck.

H4. Branch diversity should outperform single-trajectory search under the same edit budget when failures admit multiple plausible diagnoses.

L Why GSI-LR over Static

Static achieves an FCR of 0.740 "for free" by doing nothing — it is a floor, not a strategy. Static cannot improve: it permanently locks in all 6 initial rule errors and every family contradiction they cause. GSI-LR actively searches for improvements while maintaining a consistency guarantee that Static gets passively. In our experiments, GSI-LR reduces FCR from 0.700 (initial) to 0.675 — finding edits that fix errors without cascading contradictions. The DRR = 0.0 for GSI-LR means zero delayed regressions: every accepted edit was durably beneficial. By contrast, LocalRefine shows what happens when one tries to improve *without* governance: accuracy drops from 0.689 to 0.599, and DRR = 0.133 means 13.3% of accepted edits later cause regressions. In domains where the initial policy is weak, Static locks in all initial errors permanently; GSI-LR can fix those errors that do not cascade, and its governance prevents the cascading failures that plague LocalRefine.

In realistic (non-deterministic) settings with stochastic LLM reasoning, unconstrained search would face additional cascading contradictions beyond what our clean domain reveals, making the governed tradeoff even more favorable.

M Scalability Roadmap: NL Reasoning and Live LLMs

The current validation uses a 200-question deterministic propositional-logic domain. This section outlines a concrete roadmap for extending GSI-LR to natural-language reasoning and live LLMs.

M.1 Domain-Specific vs. Architecture-General Components

- **Z3 oracle** (domain-specific): formal verification requires formal structure. Replacing Z3 with an LLM-as-judge is the key substitution for NL settings.
- **TCG and FCR metric** (architecture-general): temporal commitment tracking applies to any edit system; FCR over question families is computable from any QA system.
- **Governance threshold** τ_c (architecture-general): the cascade-cost filter is independent of the underlying domain formalism.
- **Closed-family generation** (requires adaptation): NL families need crowdsourced or LLM-generated paraphrases, negations, and implication variants.

M.2 Three-Phase Scaling Roadmap

Phase 1: Semi-formal NL (FOLIO, ProofWriter). These benchmarks provide structured logical forms alongside NL. Use partial Z3 grounding for provable steps and an LLM-as-judge for unprovable steps. FCR is directly computable from the existing family structure. This phase tests whether the governance mechanism transfers to imperfect autoformalization.

Phase 2: Full NL with LLM-as-judge. Replace Z3 with an LLM-based consistency checker (e.g., checking whether a proposed edit contradicts known commitments). DRR must be redefined as the probability that a step-t accepted edit causes a regression at step $t' > t$. AP becomes an expectation over stochastic evaluation.

Phase 3: Live LLM policies. Edits become prompt modifications or adapter weight updates; the TCG tracks prompt-version lineage. The key new challenge is evaluator self-modification: the governance mechanism itself uses the LLM, creating a risk of the system gaming its own acceptance criterion. Edit-rights governance (currently latent in our domain) becomes the critical safeguard.

M.3 Why Current Results Are Informative

The 8.8% FCR reduction and DRR = 0.0 are not artifacts of a trivial domain. UnconstrSearch achieves 100% accuracy yet *worsens* FCR from 0.740 to 0.775—demonstrating that optimization pressure actively degrades consistency even in the simplest possible setting. This result generalizes: any self-improving system that optimizes local accuracy without tracking cross-question consistency will face analogous degradation in richer domains where contradictions are harder to detect.

M.4 Bottlenecks at Scale

Three failure modes dominate the transition to scale:

1. **Autoformalization quality** (H3 from Appendix K): if the NL-to-formal converter is noisy, the Z3 oracle cannot catch all contradictions and AP drops.
2. **Stochastic policy**: DRR requires a deterministic notion of "the policy at step t"; stochastic LLMs require redefinition as an expectation.
3. **Family construction**: at scale, question families must be auto-generated, introducing noise in the inter-question dependency graph.

N Acceptance-Precision Sensitivity and Threshold Tuning

N.1 AP = 0.007 as a Diagnostic Finding

The governance mechanism accepts ≈1 in 143 proposals ($\mathrm{AP} = 0.007$). This is a point on the conservatism–efficiency curve, not a system failure. In a domain where most contradictions are entangled after round 7 (see developmental trace, Appendix F), generating edits that pass all five cascade steps is genuinely rare. AP reflects proposal quality as much as threshold strictness.

N.2 QUALITATIVE THRESHOLD SENSITIVITY

The governance threshold τ_c controls the conservatism–efficiency trade-off:

- As $\tau_c \to 0$ (stricter), AP decreases and DRR stays low: the system accepts fewer edits but only durable ones.
- As τ_c increases (looser), AP rises but eventually DRR rises as consistency-degrading edits begin to pass the filter.

The current $\tau_c = 0.001$ sits at the strict end of this curve. Quantitative sensitivity sweeps across τ_c values, including measurement of the resulting AP and DRR, are an important empirical follow-up that we do not perform here.

N.3 ADAPTIVE-THRESHOLD DIRECTION

A natural extension is to make τ_c *adaptive*: stricter when the policy is far from optimal and contradictions cascade easily, looser later when the remaining edits require more aggressive structural change. This mirrors the AGM minimal-change principle: less change is needed as baseline coherence improves. We do not evaluate adaptive schedules in this paper; designing and validating one is a concrete next step.

N.4 HIGHER-QUALITY PROPOSALS AS A COMPLEMENTARY AXIS

Raising AP need not require relaxing τ_c. Richer branch diversity, LLM-suggested repairs targeting specific failed families, and failure hypotheses learned from TCG history can generate proposals that are inherently more likely to pass strict governance. The preferred direction is therefore high-quality proposals under strict governance, rather than low-quality proposals under loose governance.

N.5 Z3 COMPLEXITY ANALYSIS AND SCALING

The Z3 axiomatic validation cascade in GSI-LR scales as $O(B \times |F_{\text{targeted}}|)$ per governance round, where B is the branch count and $|F_{\text{targeted}}|$ is the number of question families that the proposed edit targets. For the current 40-family, 5-branch domain, this is tractable. However, as the knowledge graph scales to millions of nodes, exponential blowup is a genuine risk.

Pruning Strategies. Three pruning approaches can control this complexity:

1. **Family-scope pruning**: Restrict Z3 validation to families whose committed answers share at least one logical variable with the proposed edit, rather than running full cross-family checks.
2. **Incremental SAT**: Use Z3's incremental solver interface to check only the delta between the current TCG state and the proposed modification, rather than re-validating the full commitment set.
3. **Lightweight pre-filter**: Train a binary classifier on TCG features (e.g., family contradiction count, edit distance from nearest committed state) to predict Z3 rejection probability; only invoke Z3 when the pre-filter assigns probability ≥ 0.5 of acceptance.

These strategies collectively reduce the effective complexity to $O(B \times |\Delta F|)$ in the incremental case, where $|\Delta F| \ll |F_{\text{targeted}}|$ for small edits—enabling scaling to massive knowledge graphs without exponential explosion.