Risk Subgraph Extraction in Cyclic Document Graphs with Structure-Aware Monte Carlo Search

Anonymous ACL submission

Abstract

Long documents such as contracts, statutes, software package metadata, and corporate filings often contain risks that emerge through mutually dependent records. LLM-guided Monte Carlo tree search (MCTS) can improve structured document reasoning, but standard tree rollouts become unstable on strongly connected components (SCCs), where the same logical record may be revisited indefinitely. We propose SA-MCGS, a structure-aware Monte Carlo graph search method that localizes SCCs, performs local LLM rollouts with relation-first evidence memory, and collapses each cyclic region into an inspectable risk subgraph. We evaluate a locked critical structural benchmark covering four authority-backed domains: Debian dependencies, SEC EX-21 subsidiary disclosures, German Civil Code cross-references, and CUAD contracts. Across 320 model-case pairs and four LLMs, SA-MCGS improves Root@3 from 58% to 81%, Risk-any from 79% to 98%, and Risk-all from 62% to 78% over an oracle-risk Top-3 full-SCC Naive baseline selected from ten attempts. The gains are larger under long-context pressure, and the returned subgraphs retain core evidence while removing more than half of the original SCC nodes.

1 Introduction

Many high-stakes document analysis tasks are graph problems hidden inside text (Hendrycks et al., 2021; Koreeda and Manning, 2021; Boulet et al., 2021). In contracts, one clause may define a deadline, another may suspend it, and a later amendment may restore only part of the obligation. Similar dependencies appear when statutory exceptions are narrowed by distant cross-references (Boulet et al., 2021; Holzenberger et al., 2020), or when software packages preserve old interfaces while dependent packages require newer ones to be present first (Debian Policy Manual, 2025a,b). Although each record can look plausible in isolation, the risk appears only when the directed dependencies among records are followed.

Large language models (LLMs) can read long contexts. Nominal context length alone does not guarantee reliable retrieval, multi-hop tracing, or aggregation under long inputs (Liu et al., 2024; Hsieh et al., 2024; Bai et al., 2024). Legal NLP benchmarks expose persistent difficulty with distributed evidence (Hendrycks et al., 2021; Koreeda and Manning, 2021; Chalkidis et al., 2022; Guha et al., 2023). We study a sharper failure mode in *cyclic* document regions, where evidence is revisitable and diffuse. A full-SCC one-shot prompt may find a suspicious node, but it is asked to simultaneously rank all records, produce a subgraph, and preserve multiple evidence endpoints in one generation.

A natural alternative is AlphaGo-style search (Silver et al., 2016; Kocsis and Szepesvári, 2006; Browne et al., 2012). Standard MCTS unrolls trajectories as trees. In a loopy graph, this makes the search state grow with walk prefixes instead of physical records. In an SCC, even a small set of records can therefore induce many repeated path copies. Related MCTS analyses describe this as a transposition, graph-history, and loop-induced uncertainty problem (Childs et al., 2008; Kishimoto and Müller, 2004; Saffidine et al., 2012; Moerland et al., 2020). Prior Monte Carlo graph search reduces duplicate transpositions for AlphaZero-style search by sharing states in an acyclic search graph (Czech et al., 2021). Other graph-to-directed-acyclic-graph (DAG) strategies, such as removing or orienting feedback edges to obtain an acyclic approximation (Eades et al., 1993; Berger and Shor, 1990), avoid cycles by discarding part of the cyclic structure. Neither line provides a mechanism for turning a strongly connected document region into a stable evaluation unit. Our setting requires localizing the SCC and collapsing it into an evidence-

preserving risk subgraph. The subgraph should retain explicit risk sources and supporting evidence.

We propose **SA-MCGS**, a structure-aware Monte Carlo graph search method for extracting risk subgraphs from cyclic document regions. SA-MCGS first finds SCC bottlenecks, then performs local window rollouts inside each SCC. Instead of treating risk as intrinsic to a single node, it stores relation-first evidence, where a node becomes risky when earlier evidence makes its constraint, exception, or handoff incompatible with the graph context. The final output is a dynamic core risk subgraph that can be inspected or passed upward as a collapsed evaluation point.

Our contributions are:

- We formulate *structural risk subgraph extraction*: given an SCC in a document graph, output a compact subgraph that retains the repair-relevant evidence endpoints, in place of single anomalous node.
- We introduce SA-MCGS, a structure-aware Monte Carlo graph search method that localizes SCCs, accumulates relation-level evidence through local rollouts, and distills persistent signals into a compact dynamic risk core. Its rollout policy uses upper confidence bound (UCB) selection augmented with progressive relation priors, critical-pair revisiting, and evidence-gated core pruning, so additional search budget is routed toward repeatedly repair-relevant relations, not arbitrary path copies.
- We build a frozen critical-stress benchmark from four authority-backed sources. Debian provides package-dependency graphs, SEC EX-21 provides subsidiary-disclosure graphs, the German Civil Code (BGB) provides statutory cross-reference graphs, and CUAD provides contract-clause graphs. The benchmark contains 320 model-case pairs evaluated with GPT-4o, DeepSeek-V3, Qwen2.5-72B, and Gemini 2.5 Pro. Across these settings, SA-MCGS improves Root@3, Risk-any, and Risk-all over an oracle-risk full-SCC direct-subgraph baseline. It also makes the retention cost visible through an explicit compression metric.

2 Task Formulation

Record graph. Let $G = (V, E)$ be a directed graph whose nodes are document records and whose edges denote typed dependencies such as reference, constraint, modification, trigger, or control. A record may be a clause, statute section, package metadata entry, subsidiary disclosure, or other domain unit. We use Tarjan's algorithm (Tarjan, 1972) to decompose G into strongly connected components. A component $S \subseteq V$ is a reasoning bottleneck when every node in S can reach every other node through directed dependencies.

Structural risk. We study relational risks whose evidence extends beyond suspicious wording. In the benchmark, each injected critical case modifies only existing records and introduces no new nodes. The core evidence contains a root record, a witness record, and sometimes affected records. The root and witness can each be locally plausible, but jointly create an incompatible obligation, temporal condition, ownership statement, or dependency handoff.

Output. Given a strongly connected component S, the system returns a risk subgraph $H \subseteq S$ and, when applicable, a ranking over risky records. We evaluate:

- **Root@3**: whether the injected root appears in the top three ranked records.
- **Risk-any**: whether H contains at least one repair-relevant risk endpoint.
- **Risk-all**: whether H retains all core risk endpoints. This criterion is strict since remediation often requires seeing both sides of the contradiction.
- **Compression**: $1 - |H|/|S|$, where higher values indicate smaller subgraphs. Compression is not optimized alone; a tiny subgraph that drops the evidence endpoints is not useful.

We report effective compression and effective OC as diagnostics in Appendix A; they are not primary success metrics.

3 Method

3.1 Why ordinary MCTS is not enough

Standard tree search treats a trajectory as a growing path, following the usual MCTS loop of selection, expansion, simulation, and backpropagation (Browne et al., 2012). In a DAG, this assumption is valid. Rollouts eventually terminate, and backpropagation updates a finite set of states. In

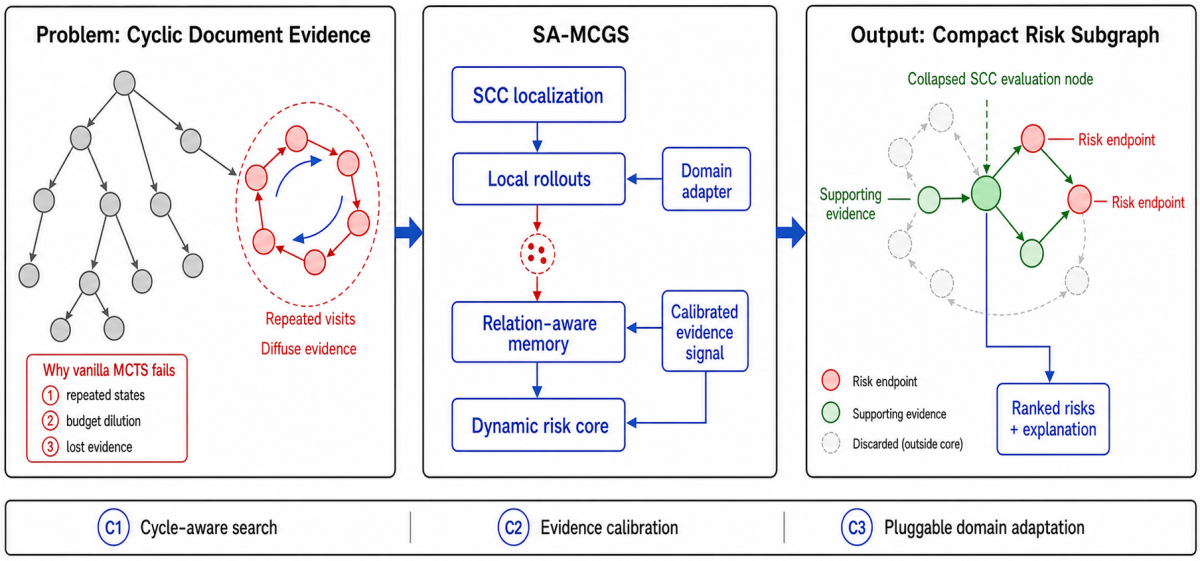

Figure 1: SA-MCGS pipeline for SCC localization, local rollouts, relation-first evidence memory, and dynamic-core collapse.

an SCC, the same physical record can be revisited through many path copies (Childs et al., 2008). If a rollout path is $p = (v_1, \ldots, v_\ell)$, ordinary TreeMCTS indexes states by path prefix. A physical node v can therefore appear as many distinct tree states,

$$\mathcal{P}_T(v) = \{p_{\leq t} : p_t = v,\ p \in \Pi_T\}, \qquad (1)$$

where Π_T is the set of paths generated before budget T. With this indexing, path multiplicity governs the effective state space in place of $|S|$. Repeated revisits then consume rollout budget, increase truncation, and dilute value estimates.

Figure 2 illustrates the failure mode. The issue is not simply that a cycle exists. Tree expansion also treats revisits as new path states. A long cyclic document region can therefore consume rollout budget while repeatedly rediscovering the same records, making it harder to preserve the endpoints that jointly create a structural risk.

We verify this failure mode with a controlled motivating ablation. We take matched document-graph components with the same local risk task and compare three search variants under the same rollout budget: ordinary TreeMCTS on an acyclic dependency graph, ordinary TreeMCTS on an SCC, and SA-MCGS on the same SCC setting. The ablation is intentionally small and diagnostic. It tests whether SCC topology breaks the assumptions behind tree search. Those assumptions underlie the appeal of tree search in DAG-like document graphs.

Method	Topology	Expansion	Trunc.	Q-spread	Hit@3
TreeMCTS	DAG	2.3×	0%	0.077	7%
TreeMCTS	SCC	6.4×	100%	0.044	30%
SA-MCGS	SCC	1.0×	–	0.371	90%

Table 1: Motivating ablation: ordinary TreeMCTS expands path copies and dilutes value estimates inside strongly connected components, whereas SA-MCGS treats each SCC as a localized reasoning unit.

Table 1 directly tests the predicted failure. Inside an SCC, TreeMCTS expands many path-copy states, which truncates rollouts and flattens value estimates. SA-MCGS avoids path copies by indexing evidence by physical nodes and relations, giving $1.0\times$ expansion and a much larger Q-spread. Detailed definitions of Expansion, Truncation, Q-spread, and Hit@3 are given in Appendix A.

3.2 Objective

SA-MCGS avoids path-copy states by treating each SCC S as a localized graph object. The desired output is a subgraph $H \subseteq S$ that keeps enough evidence for remediation while remaining smaller than the original component. Let $R^\star \subseteq S$ denote the latent repair-relevant endpoints in evaluation, e.g., root and witness records. At inference $R^\star$ is unknown, so the algorithm estimates node and relation utility from rollout evidence. The empirical objective is given by

$$\max_{H \subseteq S} \ \widehat{R}(H) - \lambda \frac{|H|}{|S|}, \qquad (2)$$

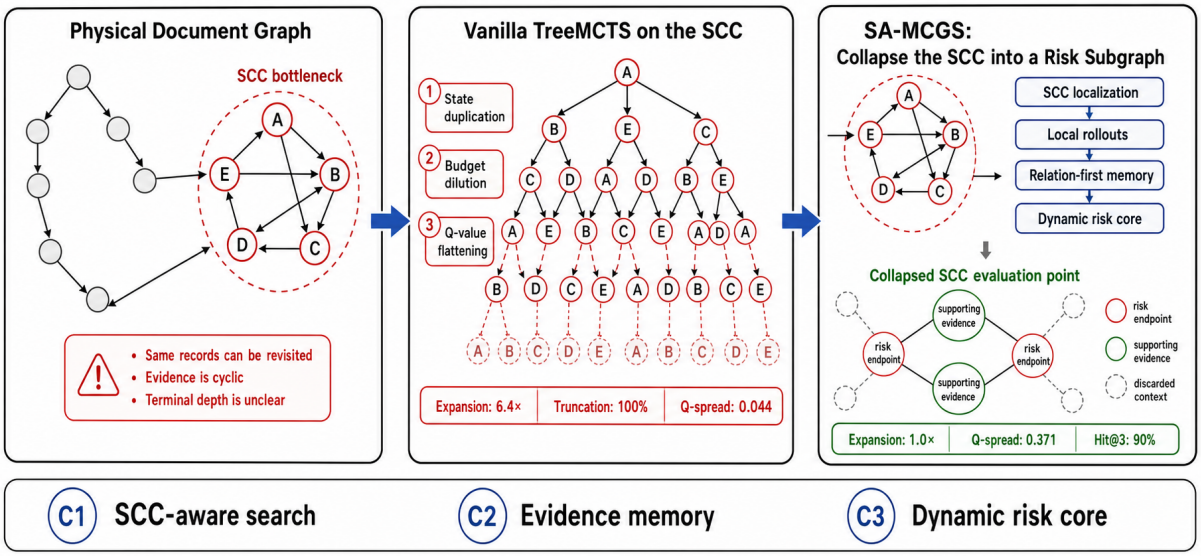

Figure 2: Why vanilla TreeMCTS is ill-suited to SCCs. Cyclic records create path-copy states that dilute the rollout budget and flatten value estimates.

where $\widehat{R}(H)$ estimates evidence retention and λ controls compression pressure. The objective balances risk coverage, measured by Risk-any and Risk-all, against compression, reflecting the need for a small H to retain repair-relevant evidence.

3.3 SCC-aware local rollouts

We first decompose the record graph using Tarjan's linear-time algorithm (Tarjan, 1972). DAG regions are handled by the ordinary upstream search, whereas each non-trivial SCC is processed by SA-MCGS. At rollout t, the algorithm selects a seed $a_t \in S$ and constructs a local window

$$W_t = \{v \in S : \mathrm{dist}_S(a_t, v) \leq r\} \cup B_t, \qquad (3)$$

where r is a small graph radius and B_t contains selected bridge or memory nodes. The LLM evaluator observes node text and visible directed edges restricted to W_t, excluding the remainder of the SCC. It returns local node scores $y_t(v) \in [0, 1]$, relation evidence $z_t(u, v) \in [0, 1]$, a local risk subgraph h_t, and optional OC evidence.

Seed selection and frontier expansion follow the UCB intuition used in MCTS (Kocsis and Szepesvári, 2006). Statistics are keyed by physical graph objects instead of path copies. Let x denote either a seed node v or a frontier edge $e = (u, v)$. SA-MCGS scores both with the same exploration template:

$$U_t(x) = \bar{r}_t(x) + c_t \sqrt{\frac{\log(1 + N_t)}{n_t(x) + \ell_t(x) + \epsilon}} + \frac{\lambda \pi_t(x)}{1 + \delta \sqrt{n_t(x)}} + \eta_t(x), \qquad (4)$$

where $\bar{r}_t(x)$ is the current empirical risk or conflict estimate, $n_t(x)$ is the number of times x has been observed, $\ell_t(x)$ is virtual loss for concurrent workers, $\pi_t(x)$ is a relation-first prior, and $\eta_t(x)$ is exploration noise. The prior is computed from rollout evidence without domain-specific labels.

$$\pi_t(x) = \frac{[w^\top \phi_t(x)]_+}{\max_{x'} [w^\top \phi_t(x')]_+ + \epsilon}. \qquad (5)$$

For nodes, $\phi_t(x)$ contains repair-entry counts, local risk-subgraph appearances, conflict-pair endpoint support, AMAF/RAVE-style evidence, path-fragment support, and OC endpoint indicators. For edges, it contains semantic incompatibility, repeated pair support, explicit conflict reports, reverse-edge support, and accumulated conflict evidence. Weights w are fixed before the locked evaluation and reported in Appendix F. The decay term in Eq. 4 prevents an early prior from dominating after sufficient node or edge samples. Together, these choices implement progressive relation bias, analogous in spirit to progressive bias and RAVE-style reuse (Gelly and Silver, 2007; Chaslot et al., 2008; Browne et al., 2012), while keeping the update key tied to physical nodes and edges as graph-level objects (Czech et al., 2021).

3.4 Relation-first evidence memory

Structural risk is relational in the sense that a node can appear harmless until previous evidence makes it incompatible with the graph context. This matches the relational-inductive-bias view that graph-structured reasoning should represent entities together with the relations that define their roles (Battaglia et al., 2018). We therefore maintain memory

$$M_t = \{n_t(v), \bar{q}_t(v), m_t(u,v), o_t(v)\}_{v,u \in S}, \quad (6)$$

where $m_t(u,v)$ stores support that the directed relation (u,v) makes either endpoint risky, and $o_t(v)$ stores whether v has appeared in effective OC evidence. Node evidence is updated by running averages and relation evidence by smoothed pair support. Appendix E gives the exact update equations. Let $r_t^-(v)$ and $r_t^+(v)$ be the strongest incoming and outgoing relation-memory scores around v. The priority score used for core construction combines node evidence, relation evidence, OC support, and a persistent critical-pair ledger:

$$\begin{aligned} s_t(v) = \bar{q}_t(v) + \alpha r_t^-(v) + \alpha r_t^+(v) \\ + \gamma o_t(v) + \mu p_t(v). \end{aligned} \quad (7)$$

Thus a node can become important because incoming or outgoing evidence repeatedly implicates it, even if its isolated text is not obviously defective. Relation evidence is aggregated by maxima over incoming and outgoing edges to limit degree bias from highly connected nodes. The bounded bonuses from OC and the pair ledger route search toward severe recurring hypotheses, allowing persistent node or relation evidence to outweigh isolated local judgments. This persistence mechanism prevents a single noisy local judgment from deciding the final core, while still letting strong recurring relations guide pruning.

3.5 Online conformal-style evidence signal

Single LLM judgments are noisy, especially on local windows. We therefore add a calibration-inspired discovery signal. At each rollout, SA-MCGS compares the strongest local relation-conflict score against an online quantile of previous conflict scores. Appendix E gives the exact rule. We call a threshold crossing an online calibrated conflict signal (OC). The signal marks a local window whose ordered cycle or contradiction cue is unusually strong relative to earlier rollouts, and endpoints of the implicated relation receive an OC bonus in Eq. 7. The calibration motivation follows conformal calibration (Vovk et al., 2005; Gibbs and Candès, 2021). No formal coverage guarantee is claimed because the LLM evaluator and rollout policy are adaptive. We use OC as evidence routing and convergence support, while final evaluation is still based on the returned subgraph H.

3.6 Dynamic core compression

We maintain the final risk subgraph as a dynamic core with replacement, avoiding monotonic expansion. Core compression is cast as budgeted subset selection, for which greedy selection is a standard approximation strategy under monotone submodular objectives with cardinality constraints (Nemhauser et al., 1978). After each rollout, SA-MCGS forms a candidate pool C_t from the local risk subgraph, the previous core, OC endpoints, and top-scoring relation endpoints with local neighbors. The core is selected by maximizing a bounded utility:

$$\begin{aligned} H_t = \arg \max_{\substack{H \subseteq C_t \\ |H| \le B(S)}} \Bigg[\sum_{v \in H} s_t(v) \\ + \eta \sum_{(u,v) \in E(H)} m_t(u,v) \\ - \lambda |H| \Bigg]. \end{aligned} \quad (8)$$

We implement this objective with greedy replacement: high-confidence endpoints may evict weaker nodes, whereas repeatedly supported relation endpoints are retained as locked evidence. This persistence mechanism is essential for critical cases. When a rollout observes both endpoints of a severe contradiction, subsequent rollouts should revisit and stabilize that relation, instead of allowing unrelated high-scoring distractors to occupy the subgraph.

3.7 Computational feasibility

SA-MCGS is an anytime graph-search procedure in the sense of Zilberstein (1996). Tarjan preprocessing is $O(|V| + |E|)$. For a component S, the method performs T local LLM evaluations over windows of at most w records, with prompt cost $O(T \cdot \text{tokens}(w))$. Its evidence memory is keyed by physical nodes and observed relations, not path-copy states, and stores $O(|S| + |E_S|)$ statistics. The dynamic core cap $B(S)$ bounds the retained

Algorithm 1 SA-MCGS for one strongly connected component

Require: Component $S = (V_S, E_S)$, budget T, shared rubric and domain serializer
Initialize memory M_0 and core $H_0 = \emptyset$
for $t = 1$ to T **do**
  Select seed $a_t = \arg\max_{v \in V_S} U_t(v)$ using Eq. 4
  Build local window W_t around a_t plus memory bridges
  Query LLM with the shared risk rubric on W_t
  Receive $y_t(v)$, $z_t(u, v)$, local subgraph h_t, and OC evidence
  Update node and relation memory M_t
  Recompute dynamic core H_t using Eq. 8
end for
return risk subgraph H_T and node ranking by $s_T(v)$

subgraph, allowing additional budget to refine the same physical memory without expanding a cyclic tree indefinitely. The resulting collapsed evaluation point is anytime-compatible for higher-level graph search. Budget-prefix behavior is analyzed in Figure 4 and Appendix A.

4 Experimental Setup

4.1 Domains

We evaluate on four domains chosen for structural diversity and source authority. The selection was not designed to favor a particular result. Table 2 summarizes their roles.

4.2 Domain formatting and shared risk semantics

The four sources differ in surface form. A Debian record exposes package fields, a BGB record is a statutory section, an SEC EX-21 record is a subsidiary disclosure, and a CUAD record is a contract clause. We therefore use lightweight domain formatters

$$\mathcal{A}_d = (\text{serialize}_d, \text{edges}_d, \text{normalize}_d) \quad (9)$$

to serialize records, expose typed directed edges, and normalize model outputs. These formatters do not define the target risk type and do not reveal the injected endpoints. Both Naive and SA-MCGS use the same risk rubric that is neutral across domains. Under this rubric, models flag structural inconsistency or mutual incompatibility, as well as risks that are underspecified or have high impact, using only visible node text and directed edges. The comparison is therefore computational, contrasting full-SCC direct-subgraph generation and Top-3 aggregation with SCC-aware rollout selection, relation-first memory, critical-pair revisiting, and dynamic-core compression.

4.3 Critical structural injection

The main experiment uses a locked critical-stress configuration over the same SCC blocks and models for both methods. The exact run profile is reported in Appendix J. Each case edits only existing records in a real SCC. A root record and a distant witness or affected record jointly create a structural conflict, while each record remains locally plausible. Both methods receive the same generic risk semantics and the same visible graph; the exact templates and domain examples are in Appendix I.

4.4 Baselines and models

We use oracle-risk Naive as the main baseline. It runs the full-SCC direct-subgraph prompt ten times. From these attempts, a risk-prioritized oracle chooses the reported Top-3 set using Root@3, Risk-any, and Risk-all. Each Naive attempt sees the full SCC and emits a whole-SCC ranking plus a risk subgraph; Appendix J reports non-oracle and compression-first variants. SA-MCGS instead sees smaller local windows over 60 rollouts and aggregates evidence into a dynamic core.

We evaluate GPT-4o, DeepSeek-V3, Qwen2.5-72B-Instruct, and Gemini 2.5 Pro. The main comparison contains 320 model-case pairs per method; oracle-risk Naive is selected from 3,200 raw full-SCC attempts. This gives boosted Naive a total token budget comparable to SA-MCGS.

5 Main Results

Figure 3 shows the core result. Against oracle-risk Naive, SA-MCGS improves Root@3 by 23 points, Risk-any by 19 points, and Risk-all by 16 points. The strongest claim is not that full-SCC prompting never finds risk, since the baseline is already selected from ten attempts with access to risk metrics. Improvement from SA-MCGS is more reliable preservation of the repair-relevant evidence endpoints in the final subgraph.

The compression axis should be read as a trade-off, not a loss. Oracle-risk Naive is slightly more compressed on average, but still retains all critical endpoints less often. SA-MCGS deliberately keeps a larger dynamic core when evidence persists across rollouts.

Domain	Source authority	Graph role	Paper role
Debian	Package dependency metadata	Software migration / dependency constraints	Non-legal technical generalization
SEC EX-21	Public subsidiary exhibit structure	Entity-control and consolidation paths	Financial/legal entity generalization
BGB	German Civil Code references	Statutory cross-reference cycles	Clean legal-rule stress test
CUAD	Expert contract review dataset	Long contract clause graph with ambiguity	Noisy contract-domain stress test

Table 2: Domain coverage. The benchmark spans technical dependencies, public regulatory disclosures, official statutory references, and expert-labeled commercial contracts.

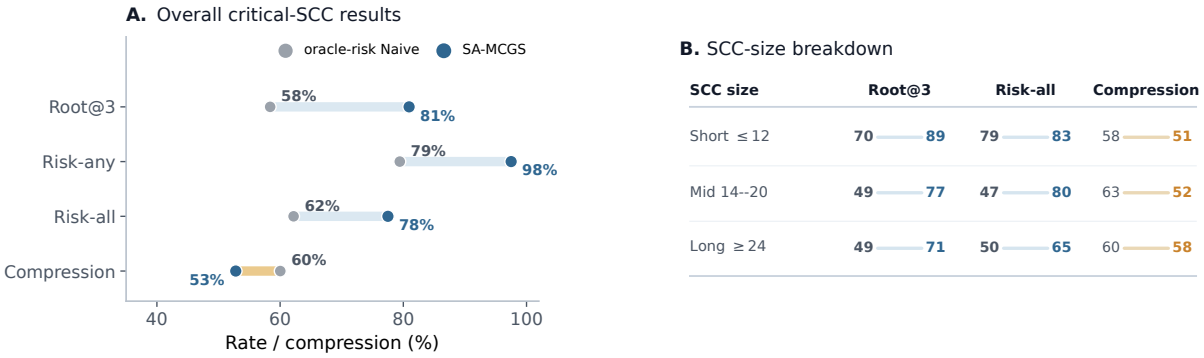

Figure 3: Main critical-SCC results on 320 model-case pairs per method. Panel A compares SA-MCGS with oracle-risk Naive, an oracle-risk Top-3 selection over ten full-SCC attempts; Panel B gives a compact SCC-size breakdown of endpoint retention and compression.

Appendix 9 separates one-shot output availability from boosted raw-attempt and model-case reliability.

5.1 Breakdown by model and domain

The gains are not carried by a single model. Gemini 2.5 Pro and DeepSeek-V3 are strong even under oracle-risk Naive, while GPT-4o and Qwen2.5-72B still lag in complete endpoint retention. Domain-wise, CUAD is the hardest setting: long contract components create both context pressure and output-schema burden for full-SCC ranking and subgraph generation.

6 Analysis

6.1 Effect of SCC size

Figure 3B separates the results by SCC size. In short components ($\leq$ 12 nodes), SA-MCGS improves Risk-all from 79% to 83%. In mid-sized components (14–20 nodes), the improvement is 47% to 80%. In long components ($\geq$ 24 nodes), Risk-all remains difficult, but SA-MCGS still improves 50% to 65%. The pattern is consistent with the task. In longer cycles, plausible distractors are more numerous and native ambiguity is higher, especially in CUAD.

6.2 Rollout convergence

Figure 4 shows that additional rollouts accumulate useful evidence instead of only spending more tokens. At budget 5, Risk-all is 22%; at budget 30, it rises to 41%; at budget 60, it reaches 57% under prefix reconstruction. The final run-level Risk-all in Figure 3 is higher because the dynamic core uses the full retained evidence rather than a simple prefix truncation. The important trend is that risk coverage and Risk-all continue to rise well past the first few rollouts, supporting the search interpretation.

6.3 Robustness checks

Appendix J reports selector and reliability checks. The non-oracle boosted selector reaches 52% Risk-all, while compression-first oracle selection raises compression but lowers Risk-all to 47%. A CUAD lite-output control reduces formatting burden but still reaches only 12% Risk-all on long-contract SCCs, and a more compressed SA-MCGS profile lowers Risk-all from 78% to 71%. Appendix C gives a representative long-SCC case study.

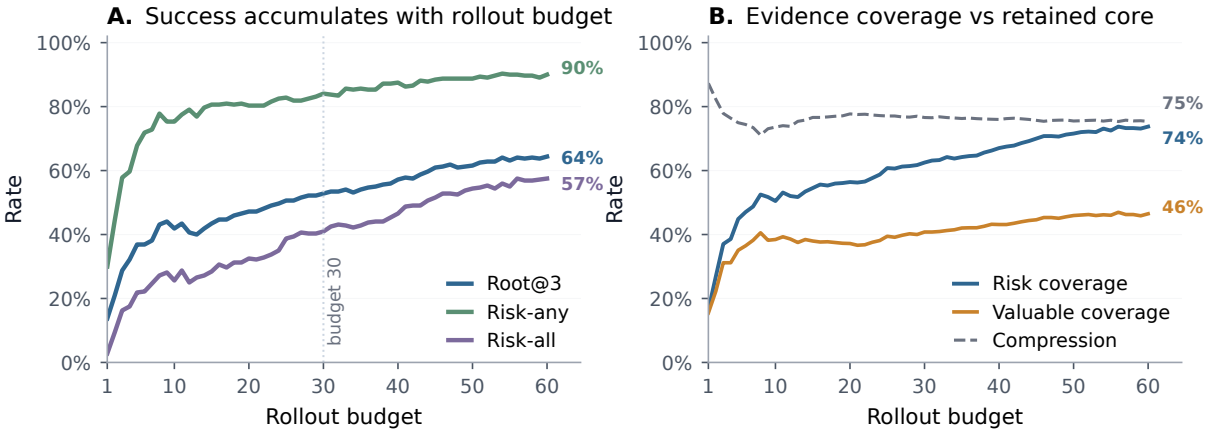

Figure 4: Budget-prefix convergence: localization and evidence coverage improve with rollout budget while the retained core stays compact.

7 Related Work

LLM reasoning and long context. Chain-of-thought prompting improves sequential reasoning (Wei et al., 2022), whereas cyclic document risk requires revisiting and reconciling records instead of extending one chain. Long-context benchmarks show persistent failures in retrieval, multi-hop tracing, and aggregation (Liu et al., 2024; Hsieh et al., 2024; Bai et al., 2024), while graph-reasoning studies show sensitivity to graph encoding and task formulation (Wang et al., 2023; Fatemi et al., 2024). Legal NLP benchmarks motivate document-level reasoning (Hendrycks et al., 2021; Koreeda and Manning, 2021; Chalkidis et al., 2022; Guha et al., 2023). Our task stresses cyclic structural evidence retention.

Search-augmented LLMs. Tree of Thoughts (Yao et al., 2024), Graph of Thoughts (Besta et al., 2024), and planning-style LLM search (Hao et al., 2023) show that explicit search can improve reasoning. SA-MCGS operates on externally constructed document graphs and treats SCCs as bottlenecks to collapse into risk subgraphs.

Monte Carlo search on graphs. MCTS and AlphaGo-style search are powerful under stable state transitions (Silver et al., 2016; Kocsis and Szepesvári, 2006; Browne et al., 2012). Monte Carlo graph search reuses transpositions (Childs et al., 2008; Czech et al., 2021), and work on MCTS uncertainty highlights loop difficulty (Moerland et al., 2020). Our method targets the remaining gap in information-preserving SCC handling for cyclic document regions.

Graph legal analysis and calibration. Network analysis has been used for statutory cross-references (Boulet et al., 2021), contract graph extraction is active (Dechtiar et al., 2025), and conformal ideas inspire calibrated evidence signals (Vovk et al., 2005; Gibbs and Candès, 2021). SA-MCGS is complementary because it assumes a record graph and focuses on risk search inside cyclic components.

8 Conclusion

We introduced SA-MCGS, a structure-aware Monte Carlo graph search method for cyclic document risk. The central idea is to avoid treating a long SCC as either a flat prompt or an ordinary tree-search path. Instead, SA-MCGS localizes the SCC, searches it through local rollouts, stores relation-first evidence, and collapses the component into a dynamic risk core. Across four domains and four LLMs, this yields stronger root localization and endpoint retention than an oracle-risk Top-3 baseline built from repeated full-SCC direct-subgraph attempts. It also makes the cost in compression explicit.

Limitations

SA-MCGS incurs higher cost than one-shot prompting due to its local LLM calls. The method is intended for high-risk cyclic components, not every record in a document. Our benchmark uses injected critical risks to keep ground truth controlled. Naturally occurring risks may be less controlled and require human audit. Graph construction quality also matters, as missing or spurious edges can change the SCCs that the method re-

ceives. Finally, Risk-all remains difficult in noisy long-contract settings, so we report it explicitly without replacing it with the easier Risk-any metric.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, and 1 others. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Bonnie Berger and Peter W. Shor. 1990. Approximation algorithms for the maximum acyclic subgraph problem. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 236–243. Society for Industrial and Applied Mathematics.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38.

Romain Boulet, Pierre Music, Corinna Coupette, Dirk Hartung, Michael Bommarito, and Daniel Martin Katz. 2021. Measuring law over time: A network analytical framework with an application to statutes and regulations in the United States and Germany. *Frontiers in Physics*, 9:658463.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330. Association for Computational Linguistics.

Guillaume M. J.-B. Chaslot, Mark H. M. Winands, H. Jaap van den Herik, Jos W. H. M. Uiterwijk, and Bruno Bouzy. 2008. Progressive strategies for Monte-Carlo tree search. *New Mathematics and Natural Computation*, 4(3):343–357.

Benjamin E. Childs, James H. Brodeur, and Levente Kocsis. 2008. Transpositions and move groups in Monte Carlo tree search. In *2008 IEEE Symposium On Computational Intelligence and Games*, pages 389–395. IEEE.

Johannes Czech, Patrick Korus, and Kristian Kersting. 2021. Monte-carlo graph search for AlphaZero. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pages 103–111.

Debian Policy Manual. 2025a. *Declaring relationships between packages*. Version 4.7.2.0.

Debian Policy Manual. 2025b. *Shared libraries*. Version 4.7.2.0.

Moriya Dechtiar, Daniel Martin Katz, Mari Sundaresan, Sylvain Jaume, and Hongming Wang. 2025. GRAPH-GRPO-LEX: Contract graph modeling and reinforcement learning with group relative policy optimization. *arXiv preprint arXiv:2511.06618*.

Peter Eades, Xuemin Lin, and W. F. Smyth. 1993. A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations*.

Sylvain Gelly and David Silver. 2007. Combining online and offline knowledge in UCT. In *Proceedings of the 24th International Conference on Machine Learning*, pages 273–280. ACM.

Isaac Gibbs and Emmanuel Candès. 2021. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems*, volume 34, pages 1660–1672.

Neel Guha, Julian Nyarko, Daniel E. Ho, and 1 others. 2023. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. *arXiv preprint arXiv:2308.11462*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: An expert-annotated NLP dataset for legal contract review. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. In *Proceedings of the 2020 Natural Legal Language Processing Workshop*, pages 31–38. Association for Computational Linguistics.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Akihiro Kishimoto and Martin Müller. 2004. A general solution to the graph history interaction problem. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 644–649.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based Monte-Carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *LNCS*, pages 282–293. Springer.

Yuta Koreeda and Christopher D. Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. 2020. The second type of uncertainty in Monte Carlo Tree Search. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2388–2394.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294.

Abdallah Saffidine, Tristan Cazenave, and Jean Méhat. 2012. UCD: Upper confidence bound for rooted directed acyclic graphs. *Knowledge-Based Systems*, 34:26–33.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and 1 others. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Robert Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning in a Random World*. Springer.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? *arXiv preprint arXiv:2305.10037*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Shlomo Zilberstein. 1996. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83.

A Diagnostic Metrics and Ablation Definitions

The main paper evaluates Root@3, Risk-any, Risk-all, and compression. We also log two diagnostics. **Effective compression** is computed only on cases whose returned subgraph retains at least one repair-relevant endpoint. It separates useful compression from trivial compression that drops all evidence. **Effective OC** counts an OC discovery only when the implicated relation intersects the repair-relevant risk region. It is useful for convergence analysis, although it is cumulative by construction and is therefore not a primary success metric.

Table 1 uses four ablation diagnostics. **Expansion** is the number of search states expanded relative to the physical component size. **Truncation** is the fraction of rollouts stopped by the depth limit before a terminal state, which is common in loops (Moerland et al., 2020). **Q-spread** is the range of estimated action values at the root, a proxy for whether search can separate promising actions under UCT-style selection (Kocsis and Szepesvári, 2006). **Hit@3** reports whether a repair-relevant endpoint appears in the top three ranked records.

B System-Blind Expert Annotation Audit

We construct a representative expert-audit set to verify that the critical structural risks used in our benchmark are recognizable to qualified domain readers, not artifacts of hidden system labels. The audit is *system-blind*. Annotators see only paragraph text, paragraph identifiers, and ordinary natural-language instructions. They are not shown

model outputs, hidden node roles, system suggestions, SA-MCGS cores, Naive subgraphs, or terms such as root, witness, affected node, SCC, and risk subgraph. Hidden fields are used only after annotation to compare expert labels with injected endpoints and system outputs.

All three annotators hold doctoral degrees in relevant fields and have professional experience with legal, regulatory, or contractual analysis.

Case-level task. For each case, the expert reads a group of related paragraphs and answers whether the group contains a structural conflict. A structural conflict means that two or more paragraphs cannot all be satisfied together, leave a high-impact condition unresolved, or make a repair decision depend on inconsistent assumptions.

Paragraph-level task. For each paragraph, the expert chooses one of four labels:

- **Clear issue**: the paragraph is itself problematic, or clearly conflicts with another visible paragraph.
- **Possibly relevant**: the paragraph may be needed to understand or repair the conflict, but is not clearly wrong by itself.
- **No issue observed**: the paragraph appears to be ordinary context.
- **Uncertain**: the expert cannot decide without more context or specialist knowledge.

Repair-material task. After labeling paragraphs, the expert selects which paragraphs should be included in the final repair packet. This task is stricter than identifying a suspicious sentence because the packet should include enough material for a reviewer to understand and fix the conflict.

Annotation protocol. Each sample is annotated independently by three experts. Every sample therefore receives three case-level decisions, three sets of paragraph-level labels, and three repair-packet selections. Disagreements are preserved for analysis without being forced into an immediate majority vote.

Audit results. Table 3 summarizes the completed annotations. All 12 audited cases receive a majority structural-conflict label. Experts also recognize the injected evidence endpoints. In total, 23 of 24 root/witness endpoint paragraphs receive a majority *clear issue* label, and all 24 receive a majority *clear issue* or *possibly relevant* label. Agreement is strongest for the clear-risk endpoint decision, with Fleiss' $\kappa = 0.606$ and pairwise Cohen's κ ranging from 0.452 to 0.759. These results support that the injected critical risks are semantically visible to qualified readers when the relevant paragraphs are inspected.

The repair-packet task is intentionally stricter and more variable. One annotator selected no individual paragraph for the final packet while still marking every case as conflicting and most cases as sufficiently or partially supported. We therefore treat repair-packet selection as a conservative diagnostic of what a reviewer would include for remediation, not as the primary evidence that a conflict exists.

Role-level sanity check. After annotation, we compare expert paragraph labels with hidden roles. True endpoints are the only role class with near-universal majority recognition, with 95.8% majority clear issue and 100% majority clear-or-relevant. In contrast, affected paragraphs are rarely majority clear issue, and most broader system-core paragraphs are judged ordinary context. This behavior follows from the benchmark design, which asks methods to retain compact endpoints needed for repair, not to mark every retained context paragraph as intrinsically wrong. The same observation explains why Risk-all is strict, as a method may surface useful context or affected nodes while still missing one side of the contradiction.

The example in Table 5 illustrates the intended difficulty level. Neither entity record is a syntactic error by itself. The conflict appears because both records refer to the same consolidation path while asserting mutually incompatible ownership treatments. A risk-subgraph method should preserve this type of structural conflict because a reviewer can reliably repair the issue after seeing both endpoints.

C Representative Long-SCC Case Study

Figure 5 illustrates the intended use of the method. The original SCC is too large to treat as a single atomic finding and too recurrence-heavy for ordinary path search. SA-MCGS localizes the evidence through rollouts, then returns a core containing the root, witness, and affected records. For review, this is more useful than a single node label because a human can inspect the retained end-

Audit quantity	Result	Interpretation
Audited critical cases	12 cases, 36 case-level decisions	Each case is independently labeled by three experts.
Majority structural conflict	12/12 (100%)	The audited critical cases are judged to contain structural conflicts.
Repair material sufficiency	10 yes, 1 partial, 1 tie	Most cases contain enough visible material to support repair, but one case remains divided.
No extra context needed	10/12 (83.3%)	Most cases are understandable from the provided paragraphs alone.
Mean confidence	4.28/5	Experts are generally confident in their case-level judgments.
Endpoint majority clear issue	23/24 (95.8%)	Root/witness endpoints are usually judged directly problematic.
Endpoint majority clear-or-relevant	24/24 (100%)	Every endpoint is recognized as at least repair-relevant.
Clear-issue agreement	Fleiss' $\kappa = 0.606$	Agreement is substantial for the binary endpoint-risk judgment.
Repair-packet endpoint-any	7/12 (58.3%)	Majority repair packets include at least one endpoint in over half of cases.
Repair-packet endpoint-all	6/12 (50.0%)	Selecting all endpoints is stricter and less stable across experts.

Table 3: System-blind expert audit results. Experts did not see system outputs, hidden roles, or model-generated hints during annotation.

Task	Expert-facing question	Example expert note
Case-level conflict	Do these paragraphs create a conflict or unresolved high-risk condition when read together?	"The first paragraph gives a right that appears unconditional, but a later paragraph makes the required approval unavailable. The group needs revision before the right can be applied."
Paragraph label	Does this paragraph look clearly problematic, possibly relevant, ordinary, or uncertain?	"This paragraph is not wrong alone, but it is relevant because it provides the timing condition that controls the later obligation."
Repair packet	Which paragraphs should a reviewer see to fix the issue?	"The packet should include the grant clause, the later condition clause, and the definition paragraph. Omitting the definition would make the repair ambiguous."

Table 4: Expert-facing annotation tasks. These are the instructions used to elicit human judgments; internal roles and system outputs are used only for post-hoc analysis.

points and decide how the conflict should be repaired.

D Core Algorithm Details

This appendix expands the method into reproducible pseudocode. The implementation uses the same high-level objects as the main paper, including a document graph G, strongly connected components S, local rollout windows W_t, relation-first memory M_t, online calibrated conflict signals, and a final dynamic core H_T.

The core is therefore not a monotone union of all suspicious records. Later rollouts can replace weak records with stronger evidence, although the update cannot drop repeatedly supported critical endpoints only to obtain a smaller subgraph.

Algorithm 2 SCC-localized search

Require: Document graph $G = (V, E)$, rollout budget T, shared risk rubric
$\mathcal{S} \leftarrow$ Tarjan strongly connected components of G
Initialize collapsed graph $G' \leftarrow G$
for all component $S \in \mathcal{S}$ **do**
  if $|S| = 1$ **then**
    Keep the singleton record unchanged
  else
    Build local search unit $G_S = (S, E_S)$
    $(H_S, r_S) \leftarrow$ SA-MCGS-ROLLOUT(G_S, T)
    Replace S in G' with collapsed evaluation point (H_S, r_S)
  end if
end for
return Collapsed graph G' and all component risk subgraphs

E Formula Details for Evidence Updates

Implementation-level equations summarized in Section 3 are expanded here. The main text retains the search template plus two scoring components, the priority score and core objective. Remaining details were fixed before evaluation and left untuned on test cases.

Node and relation updates. For every record observed in rollout t, node risk evidence is updated

Displayed paragraph	Expert-visible text excerpt	Majority annotation
SEC EX-21 entity record A	*Entergy Louisiana Holdings LLC.* "For this consolidation path, the entity is treated as fully consolidated by the upstream registrant. The note states that no separate non-controlling interest is retained for this path."	Clear issue; included in two experts' repair packet selections.
SEC EX-21 entity record B	*ENTERGY LOUISIANA HOLDINGS INC.* "The same consolidation path reserves a 20 percent non-controlling interest for the downstream schedule. This record applies the opposite state immediately on the same path."	Clear issue; included in two experts' repair packet selections.
Case-level decision	The two records describe the same consolidation path and assign incompatible ownership states, fully consolidated with no non-controlling interest versus a retained 20% non-controlling interest.	3/3 experts mark the case as structurally conflicting; 2/3 mark the provided material sufficient, and 1/3 marks it partially sufficient.

Table 5: Representative expert-audit example. Role names in the left column are added here for exposition; the annotators saw paragraph text and identifiers, not root/witness labels or system suggestions.

Algorithm 3 SA-MCGS rollout inside one component

Require: Component $G_S = (S, E_S)$, budget T, shared rubric
Initialize node evidence, relation evidence, OC evidence, and core $H_0 = \emptyset$
for $t = 1$ to T **do**
  Select a seed record using SCC-aware UCB with relation priors
  Construct local window W_t from the seed, nearby edges, and memory bridges
  Query the LLM with the shared risk rubric on W_t
  Parse local risky records, relation evidence, and optional OC signal
  Update node risk scores, pair evidence, witness-style evidence, and affected-region evidence
  $H_t \leftarrow$ DYNAMIC-CORE-COMPRESSION(H_{t-1}, M_t)
end for
return Final core H_T and ranking induced by accumulated evidence

Algorithm 4 Relation-first evidence memory

Require: Rollout result R_t, previous memory M_{t-1}
$M_t \leftarrow M_{t-1}$
for all record v observed in R_t **do**
  Update average risk score and visit count for v
  Store why v is risky in relation to visible predecessors or successors
end for
for all reported relation (u, v) in R_t **do**
  Update pair evidence that (u, v) creates or supports a conflict
  Update second-endpoint evidence when v becomes risky because of earlier evidence
  Update affected evidence for downstream records whose status changes
end for
if R_t contains an OC signal **then**
  Add its implicated endpoints and supporting relation to OC memory
end if
return M_t

by a visit-counted running average:

$$\bar{q}_{t+1}(v) = \frac{n_t(v)\bar{q}_t(v) + y_t(v)}{n_t(v) + 1}, \qquad n_{t+1}(v) = n_t(v) + 1. \tag{10}$$

Directed relation evidence is updated by exponential smoothing,

$$m_{t+1}(u, v) = (1 - \rho)m_t(u, v) + \rho z_t(u, v), \tag{11}$$

where $z_t(u, v)$ is the local evaluator's evidence that relation (u, v) creates or supports a repair-relevant conflict.

Relation priors and critical pairs. Equation 5 is shared by node and edge priors. For nodes we set $w = \theta$ and use node features such as repair-entry support, local-subgraph appearance, pair-endpoint support, and OC endpoint evidence. For edges we set $w = \beta$ and use edge features such as semantic incompatibility, repeated pair support, reverse-edge evidence, and accumulated conflict evidence. A separate bounded ledger records persistent critical-pair hypotheses:

$$\begin{aligned} \mathcal{N}(v) &= \{u : (u, v) \in E \text{ or } (v, u) \in E\}, \\ p_t(v) &= \max_{u \in \mathcal{N}(v)} \kappa_t(u, v), \\ \kappa_t(u, v) &= \min(1, \omega^\top \chi_t(u, v)). \end{aligned} \tag{12}$$

Here $\chi_t(u, v)$ contains pair-level evidence such as semantic incompatibility, severity, same-path support, no-fallback language, repeated support, and explicit conflict reports.

Online conflict signal. Let $c_t = \max_{(u,v) \in E(W_t)} z_t(u, v)$ be the strongest relation-conflict score in the current window. With

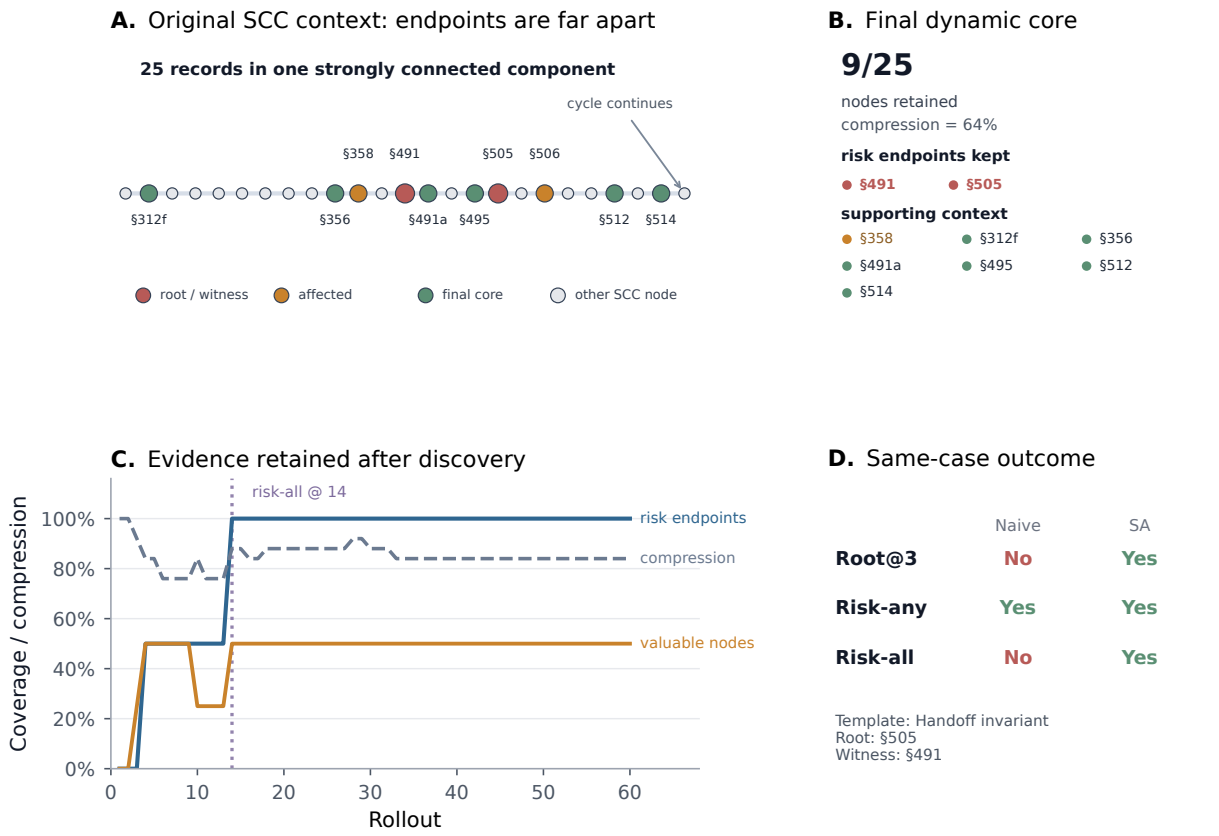

Figure 5: Representative long-SCC collapse. The method searches a large component through local rollouts, accumulates relation-first evidence, and returns a compact risk subgraph that preserves repair-relevant endpoints.

Algorithm 5 Dynamic core compression

Require: Previous core H_{t-1}, memory M_t, candidate cap $B(S)$
Form candidate set C_t from prior core, local risky records, relation endpoints, affected records, and OC support records
for all $v \in C_t$ **do**
  Score v by node risk, pair support, affected-region support, OC support, and persistence
  Mark v as protected if repeated evidence indicates a critical endpoint
end for
Start H_t with protected endpoints and strongest relation-supported records
while $|H_t| < B(S)$ and useful candidates remain **do**
  Add the candidate with largest marginal support for explaining the risk
end while
while $|H_t| > B(S)$ **do**
  Remove the weakest non-protected candidate
end while
return H_t

previous rollout scores as the online reference set,

$$\tau_t = Q_{1-\alpha_{\mathrm{oc}}}\left(\{c_i : i < t\}\right), \quad \mathrm{OC}_t = \mathbf{1}[c_t > \tau_t]. \tag{13}$$

The implicated relation endpoints are added to OC memory and receive the bounded bonus in Eq. 7.

Core candidate pool. The candidate pool used in Eq. 8 is

$$C_t = h_t \cup H_{t-1} \cup O_t \cup K_t, \tag{14}$$

where h_t is the local risk subgraph, O_t contains OC endpoints, and K_t contains top-scoring relation endpoints together with their local neighbors.

F Fixed Search-Prior Coefficients

The numeric values in Table 6 are fixed search-prior coefficients, not learned parameters. They were selected using pilot checks before the locked evaluation. Domain experts were used only as a sanity check on the ordinal reliability ordering among evidence types, not to tune coefficients on reported test cases. Prior MCTS work motivates the structure, including UCB/UCT exploration (Kocsis and Szepesvári, 2006), visit-decayed heuristic priors through progressive bias (Chaslot et al., 2008; Browne et al., 2012), AMAF/RAVE-style evidence reuse (Gelly and Silver, 2007), and graph/transposition-aware value reuse (Childs et al., 2008; Czech et al., 2021). The coefficients instantiate these ideas for structural document risk.

Parameter	Value	Role and rationale
Node-feature weights θ for Eq. 5		
θ_{repair}	1.10	LLM names the record as a repair entry. This is closest to the final remediation target.
θ_{local}	0.90	The record appears in local risk subgraphs; useful but noisier than explicit repair-entry evidence.
θ_{pair}	0.85	The record is an endpoint of a supported conflict pair; structural risks often require two endpoints.
θ_{node}	0.70	AMAF/RAVE-style node evidence reused across rollouts; important but indirect.
θ_{path}	0.75	Path-fragment evidence that the record lies on repeatedly implicated local traversals.
θ_{conflict}	0.60	Single-window conflict reports; useful but downweighted because local LLM judgments can be noisy.
θ_{OC}	2.00	Effective ordered-conflict endpoint; high-confidence routing signal, later normalized and visit-decayed.
Edge-feature weights β for Eq. 5		
β_{incompat}	1.55	Semantic incompatibility is the strongest relation-level evidence for structural contradiction.
β_{severity}	1.35	Severity increases priority, but high severity alone is not sufficient without structural support.
β_{explicit}	0.90	Explicit conflict wording helps, but is downweighted relative to persistent structural evidence.
β_{pair}	0.70	Repeated pair support indicates relation-level consistency across windows.
β_{edge}	0.65	AMAF edge evidence reused from related traversals.
β_{reverse}	0.35	Reverse-edge support is weak auxiliary evidence.
β_{conflict}	0.55	The edge's own accumulated conflict estimate, kept below semantic incompatibility and severity.
Critical-pair prior ω in Eq. 12		
ω_{sem}	0.30	Semantic incompatibility dominates because it directly captures mutual exclusion.
ω_{sev}	0.25	Severity is second: critical risks deserve revisits, but severity alone can be diffuse.
ω_{path}	0.18	Same-path evidence strengthens the claim that two endpoints jointly define one repair region.
$\omega_{\text{nofallback}}$	0.12	No-fallback language makes contradictions harder to resolve locally.
ω_{support}	0.10	Repeated support stabilizes initially uncertain pair hypotheses.
ω_{explicit}	0.05	Explicit conflict reports are useful but lowest-weighted to avoid rewarding one noisy sentence.

Table 6: Fixed search-prior coefficients used by SA-MCGS. Higher values reflect more reliable or more directly repair-relevant evidence. Values were frozen before the locked evaluation and are not learned from the reported test cases.

G Shared Risk Rubric Prompt

Reviewer fairness depends on separating the risk definition from the search mechanism. The following excerpt gives the shared evaluator rubric used by both full-SCC Naive attempts and the SA-MCGS local-window evaluator.

> You are given records from a directed graph of interdependent documents. Identify records or record sets that are structurally inconsistent, mutually incompatible, underspecified, or high-impact if left unresolved. Use only visible record text and visible directed relations. Do not assume a domain-specific defect type. Return evidence-based judgments with node ids and concise reasons.

Shared semantics. Naive and SA-MCGS use this same risk rubric. The methods differ in input unit and search procedure. Each Naive attempt reads the complete SCC and emits a global ranking plus a risk subgraph; the main baseline aggregates ten such attempts with oracle-risk Top-3 selection. SA-MCGS reads local windows over multiple rollouts and aggregates evidence before emitting a final core. The comparison holds the risk definition fixed and varies the computation.

H Prompt Templates

Complete prompts include serialization details and JSON schemas. The abstract templates reported here are sufficient to reproduce the evaluation interface without depending on one implementation file.

Naive full-SCC direct-subgraph prompt.

> Input: the complete node list for one strongly connected component, the text of every node, and all directed edges among the nodes. Apply the shared risk rubric to the full component. Return valid JSON with `global_ranking`, `risk_subgraph_nodes`, `risk_factor_distribution`, and `explanation`. Rankings and subgraph selections should be grounded in node IDs and visible edges.

SA-MCGS local-window prompt.

> Input: a local window from one strongly connected component, including visible node text, visible directed edges, and a concise prior evidence summary. Apply the same shared risk rubric to the local window. Return valid JSON with `local_node_scores`, `relation_evidence`, `local_risk_subgraph_nodes`, `repair_entry_nodes`, optional `oc_candidate`, and a concise explanation grounded in exact node IDs.

Output contrast. Naive requires each generation to rank the entire SCC and produce a final subgraph. In contrast, SA-MCGS asks only for local evidence, leaving final subgraph construction to algorithmic memory and compression.

I Dataset and Injection Details

Table 7 summarizes the four included domains and their conversion into directed record graphs. Domain coverage spans several forms of authority and structure, including technical dependency metadata, public corporate disclosures, official statutory references, and expert-labeled contract clauses.

SCC construction and filtering. For each domain, we construct a directed record graph and enumerate non-trivial SCCs using Tarjan's algorithm. Candidate components are filtered by size and record availability and then locked before model evaluation. The main experiment uses the same locked SCC blocks for Naive and SA-MCGS under the same model and profile.

Injection constraints. The critical benchmark modifies only existing records in the selected SCC. No new nodes are added. The injected root record provides the endpoint used for Root@3 evaluation. A distant witness record supplies the evidence that makes the root-side statement structurally incompatible under the graph context. Affected records are downstream or neighboring records whose interpretation changes under the root–witness conflict. Subgraph metrics count root, witness, and affected records as repair-relevant endpoints when applicable.

Why Risk-all matters. Risk-any is useful when a repair process can sometimes start from one good entry point. Risk-all is stricter and central for critical cases, where fixing a structural contradiction often requires both sides of the incompatibility. A subgraph that includes only the obvious-risk paragraph but omits the distant condition may be too small to support a reliable repair.

J Additional Experimental Details

Locked run configuration. Unless otherwise stated, SA-MCGS uses the locked profile `current/default` + `memory_stress(structural_simple_v2)` + `critical` with 80 SCC blocks per model and four models (GPT-4o, DeepSeek-V3, Qwen2.5-72B, and Gemini 2.5 Pro). The main Naive baseline is oracle-risk Top-3 selection over ten full-SCC direct-subgraph attempts per model-case. One-shot Naive, non-oracle boosted selection, and compression-first oracle selection are reported below as controls.

Domain	Why selected	Graph construction	Role in evaluation
Debian	Technical dependency graph with explicit package relationships.	Package metadata entries become records; dependency, replacement, conflict, and virtual-provider fields become directed edges.	Tests whether the method generalizes beyond legal and financial language.
SEC EX-21	Public company subsidiary disclosures with entity-control structure.	Subsidiary names, jurisdictions, consolidation fields, ownership hints, and control relations become entity records and directed edges.	Tests public regulatory disclosure reasoning and ownership-chain consistency.
BGB	Official German Civil Code cross-references.	Statutory sections become records; references, obligations, conditions, exceptions, and remedies become directed edges.	Provides a comparatively clean legal-rule stress test with authoritative source text.
CUAD	Expert-labeled commercial contract dataset.	Clauses and definitions become records; definition use, cross-reference, amendment, condition, termination, and obligation relations become edges.	Provides long, noisy contract components where evidence can be diffuse.

Table 7: Dataset construction and processing. All domains are represented as directed record graphs before SCC localization.

Template	Root-side edit	Witness / affected edit	Intended risk
`direct_mutex`	Adds a direct obligation, ownership, compatibility, or condition statement expected to hold.	Adds a distant statement that requires the opposite state under the same dependency chain.	Two records are individually plausible but mutually incompatible when the component context is followed.
`handoff_invariant`	States that an invariant is expected to be preserved before control, migration, or responsibility is handed off.	States that the downstream record can proceed only after the invariant has been changed or removed.	The handoff cannot be completed because the precondition and downstream entry condition disagree.
`temporal_gate`	Specifies a deadline, waiting period, migration window, vesting condition, or trigger order.	Requires a conflicting temporal gate, such as earlier execution, delayed activation, or a missing prerequisite.	The same chain imposes incompatible timing constraints.
`condition_trigger`	Makes a right, remedy, interface, or ownership statement conditional on a trigger.	Makes the same trigger impossible, optional, or reversed for a dependent record.	A dependent record relies on a condition whose truth value is changed elsewhere in the component.

Table 8: Critical structural injection templates. All templates edit existing records only and avoid explicit defect labels.

Scope	Denominator	Usable	Unusable	Rate
one-shot Naive	model-cases	261/320	59/320	82%
SA-MCGS	model-cases	320/320	0/320	100%
boosted Naive	raw attempts	2400/3200	800/3200	75%
boosted Naive	model-cases	274/320	46/320	86%

Table 9: Output reliability audit. Boosted Naive has both a raw-attempt denominator and a model-case denominator; the latter counts cases with at least one valid full-SCC generation.

Control	N	Root@3	Risk-any	Risk-all	Comp.	Role
one-shot Naive	320	48%	72%	47%	67%	original locked control
boosted Naive single-attempt mean	3200 attempts	47%	67%	48%	50%	raw repeated-sampling mean
boosted Naive self Top-3	320	52%	75%	52%	60%	non-oracle selector
boosted Naive oracle-risk Top-3	320	58%	79%	62%	60%	main oracle-risk baseline
boosted Naive oracle-compression Top-3	320	53%	75%	47%	63%	compression-first diagnostic
SA-MCGS	320	81%	98%	78%	53%	proposed method

Table 10: Boosted Naive selector comparison. Oracle selectors use ground-truth evaluation metrics for selection; oracle-risk Top-3 is the main oracle-risk baseline, while self Top-3 is the non-oracle boosted selector.

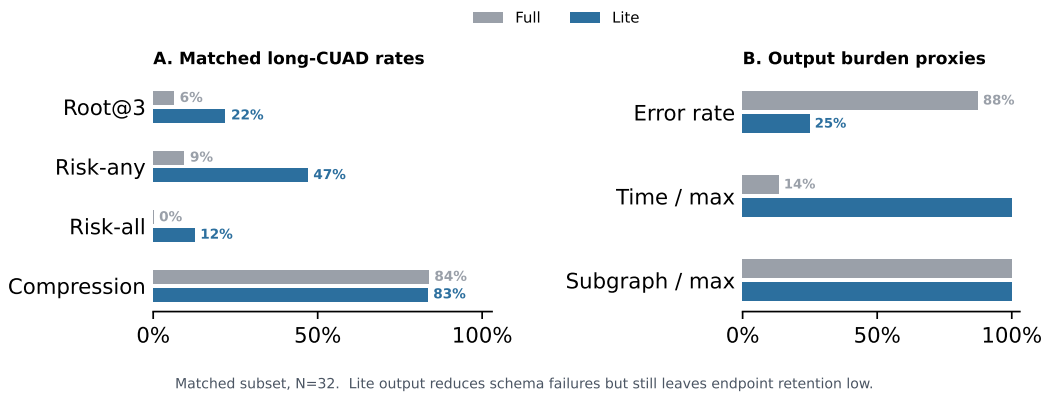

Figure 6: Naive output-burden control. Reducing the output schema improves availability and recovers some risk signal, but it does not close the endpoint-retention gap on long CUAD components.

A. Representative critical cases

Domain	Component	Template	Conflict summary	Repair material should include
BGB	25-record statute	`condition_trigger` or `temporal_gate`	One section states when an obligation or remedy applies. A distant referenced section changes the trigger or timing in a way that makes the remedy inconsistent.	The remedy section, distant condition, and any bridge section carrying the condition.
CUAD	18–25-record contract	`direct_mutex` or `handoff_invariant`	A clause preserves a right, notice obligation, or approval condition; a later clause makes the prerequisite unavailable or contradicts the preserved condition.	The granting clause, conflicting condition clause, and any definition that determines applicability.
SEC EX-21	9–18-record entity	`direct_mutex`	One disclosure states full consolidation or no minority interest; another record in the same control chain retains a non-controlling interest or incompatible ownership state.	The consolidation record, ownership witness, and affected subsidiaries when present.
Debian	package dependency	`handoff_invariant`	A package keeps an old ABI or compatibility surface during migration, while a dependent package requires the new interface to exist first.	The invariant package, dependent package, and handoff relation that makes ordering impossible.

B. Selector summary

Control	N	Root@3	Risk-all
one-shot Naive	320	48%	47%
self Top-3	320	52%	52%
oracle-risk Top-3	320	58%	62%
SA-MCGS	320	81%	78%

C. Model-level breakdown

Model	Method	N	Zero	Root@3	Risk-any	Risk-all	Comp.
GPT-4o	oracle-risk Naive	80	6	37%	76%	37%	65%
GPT-4o	SA-MCGS	80	0	86%	100%	75%	51%
DeepSeek-V3	oracle-risk Naive	80	0	87%	95%	87%	74%
DeepSeek-V3	SA-MCGS	80	0	80%	95%	76%	51%
Qwen2.5-72B	oracle-risk Naive	80	40	11%	48%	28%	22%
Qwen2.5-72B	SA-MCGS	80	0	70%	98%	75%	53%
Gemini 2.5 Pro	oracle-risk Naive	80	0	99%	100%	97%	79%
Gemini 2.5 Pro	SA-MCGS	80	0	88%	98%	84%	56%

D. Domain-level breakdown

Domain	Method	N	Zero	Root@3	Risk-any	Risk-all	Comp.
Debian	oracle-risk Naive	32	0	65%	100%	80%	61%
Debian	SA-MCGS	32	0	84%	100%	88%	53%
SEC EX-21	oracle-risk Naive	80	0	74%	100%	85%	62%
SEC EX-21	SA-MCGS	80	0	96%	100%	88%	50%
BGB	oracle-risk Naive	48	1	46%	89%	50%	61%
BGB	SA-MCGS	48	0	75%	98%	62%	52%
CUAD	oracle-risk Naive	160	45	53%	62%	51%	59%
CUAD	SA-MCGS	160	0	74%	96%	75%	54%

Table 11: Supplementary case and result breakdowns. "Zero" counts boosted Naive model-cases with no valid full-SCC generation. CUAD concentrates long-context and output-schema failures, where generations may time out or become unparseable through incomplete, duplicate, or invalid rankings.