Mitigating Consensus Bias in AI Deliberation: A Structural Evaluation of Disagreement Retention and Regime-Level Adjudication

Amin Parva

Software research manuscript

28 September 2026 · Software version 2.1.0

Abstract

A deliberation system can detect a disagreement and still omit it from its final record. We examine this form of disagreement suppression in PrismCognition, an open-source Python library that represents claims, evidence assessments, assumptions, and conflicting obligations as structured artifacts. A diagnostic benchmark contains 34 synthetic scenarios and 29 expected conflict instances. Comparisons include two local rule baselines, four aggregate-threshold settings, and two evidence-access ablations. The default engine retains 15 of 29 expected conflicts and matches the expected types and resolution statuses in 20 of 34 scenarios. Removing the aggregate filter retains all 29 conflicts across six classes and matches 33 of 34 outcomes. A direct-claim evidence baseline matches 21 of 34 outcomes, exceeding the default engine on this measure while detecting only factual conflicts. Without the aggregate filter, evidence access corrects four complete outcomes but introduces one cross-claim resolution error: pump evidence incorrectly adjudicates an unrelated valve state. Each of four replay and transformation properties passes 136 checks; 10 supplied-boundary gate scenarios also pass. The findings support broader conflict representation under the tested configuration and repeatable artifact derivation, while exposing limitations of aggregate filtering and regime-level adjudication. The cases were authored after source inspection and have not been independently annotated. The study does not establish consensus bias in other systems, superiority over model-based alternatives, or improved human decisions.

Keywords: AI deliberation; computational argumentation; disagreement retention; consensus bias; evidence grounding; provenance; reproducibility.

1. Introduction

Detecting disagreement does not guarantee that it survives into a deliberation system's final output. A pipeline may extract incompatible claims and then remove the corresponding conflict because an aggregate score falls below a threshold. We use *consensus bias* here in a narrow operational sense: suppression of represented disagreement that could leave a reviewer with an incomplete account of competing positions. The benchmark measures that suppression, not a psychological bias, model agreement, or a tendency shared by all multi-agent systems.

Consider two analyses that disagree about whether a hydraulic valve is open. Evidence that a supply pump is running does not, by itself, settle the valve's state. Yet a system that summarizes evidence for unrelated claims under one empirical profile can lose that distinction. Case X01 reproduces this failure in PrismCognition.

PrismCognition produces a structured deliberation record containing claims, evidence assessments, active disagreements, and resolution states. Recommendations are optional. Its methodological evaluators can address factual observations, assumptions, definitions, inference rules, causal accounts, and obligations. The study examines whether those distinctions survive the processing that produces the record.

Two mechanisms prove consequential: an aggregate threshold suppresses some already-detected conflicts, and evidence aggregation can resolve a disagreement without direct support for its target claim. The following sections describe these mechanisms, report comparisons on explicit structured scenarios, and distinguish the observed strengths from the unresolved failures.

2. Related work and scope

Dung's argumentation framework characterizes argument acceptability through abstract attack relations [1]. PrismCognition makes disagreement explicit using typed claims, warrants, and procedural resolution rules; it does not compute Dung-style argumentation extensions.

Du et al. investigate language-model instances that debate responses over several rounds to produce a common answer [2]. PrismCognition runs methodological evaluators and compares their stances, allowing unresolved disagreements to remain in an artifact. Its current implementation does not run that iterative debate protocol, and the evaluators are not assumed to be statistically independent or orthogonal.

Retrieval-augmented generation combines retrieval and generation for knowledge-intensive tasks [3]. PrismCognition operates downstream of evidence collection: its ledger receives records whose domain keys, regimes, polarities, and acceptance statuses have already been assigned. Retrieval and source assessment must be supplied by an integration. This study evaluates the processing of supplied records rather than retrieval quality.

These connections define the software's scope. Neither the cited systems nor external language models are experimentally evaluated here.

3. Architecture for structured disagreement

3.1 Reasoning regimes and methodological clusters

A stance contains a conclusion, propositions, warrants, assumptions, normative commitments, and provenance. A claim's domain key identifies its target, such as `Valve.open`; polarity records affirmation, denial, or neutrality. A warrant links premises to a conclusion and declares a reasoning regime.

The schema distinguishes empirical, formal, causal, normative, interpretive, phenomenological, definitional, and unverifiable regimes. Separately, six constructive method clusters address formal, empirical, causal, adversarial, implementation, and normative analysis. Adversarial and implementation analysis are method groups, not additional grounding regimes; their bundled warrants use the empirical regime.

The verifier retains separate status counts and evidence summaries for each regime. Normative, interpretive, phenomenological, and unverifiable warrants receive no numerical grounding from the bundled verifier. Missing evidence remains distinct from a numerical score of zero. These choices preserve representational distinctions, but they do not authenticate the evidence or guarantee correct adjudication.

3.2 Failure boundaries and subtractive gates

Failure analysis runs alongside constructive evaluation and translates generated output into structured boundaries. Raw failure prose is excluded from constructive prompts. The gate eliminates a stance only when an applicable boundary's nonempty trigger set matches its propositions by domain key and polarity. Elimination is enabled for completed or partial analysis; failed and timed-out analysis leaves stances intact.

For surviving stances, the gate can project an embedding away from the subspace spanned by boundary vectors. With an orthonormal basis U, the residual is:

$$z_{residual} = z - UU^{\top}z$$

A small residual produces a geometric-collapse annotation without eliminating the logical stance. This separates geometric information from the predicate rule. The gate checks in this study use supplied boundaries and do not establish the completeness or safety of generated boundaries.

3.3 Provenance and offline replay

A `FrozenDeliberationBundle` stores intermediate inputs, including stances, failure results, grounding profiles, route metadata, thresholds, and execution information. Replay re-derives downstream artifacts without requesting new model output or consulting later evidence. It incurs no new model-token usage, although local computation and storage still have costs.

Canonical hashing sorts serialized keys and rounds floating-point values to eight decimal places. The benchmark compares original artifacts with replay of their JSON-serialized bundles under the same implementation. This tests repeatability in the measured conditions rather than guaranteeing behavior across versions or fresh model generations.

Provenance supports tracing derived artifacts to their parents. Frozen data models do not make file storage an append-only audit log: repeated inquiries can overwrite files, and artifact-plus-bundle writes are not a single transaction. No compliance certification is claimed.

3.4 Deliberation artifacts

The `DeliberationArtifact` separates active disagreements from those classified as evidence-resolved. It also exposes supported claims, evidence gaps, relevant assumptions, perspective diversity, method coverage, and execution notes. These fields permit inspection and programmatic processing of the engine's assessments. A status such as “supported” describes treatment of supplied records rather than independently verified truth.

4. Deliberation mechanisms

4.1 Evidence status and missingness

Evidence is matched to a warrant's conclusion by domain key and regime. A row marked `SUPPORTED` is treated as an accepted assertion of its supplied polarity. Opposing accepted assertions produce a contested assessment; a matched `CONTESTED` row withholds support. Rejected or unresolved assertions do not automatically establish their inverses. Missing accepted evidence yields insufficient evidence and a score of `None`.

In the primary grounding path, claim promotion requires a supported warrant with a score of at least 0.70. Disagreement resolution does not use this magnitude threshold: it checks regime-level status patterns and the presence of scores. A targeted execution check confirmed that an accepted assertion scored 0.0 can resolve a factual pair without promoting a strongly supported claim. The scores are supplied values, not calibrated probabilities.

4.2 Structural tension and diversity

Aggregate structural tension between two stances is:

$$\Delta(s_1, s_2) = 0.40\, d_{pred} + 0.30\, d_{warr} + 0.20\, d_{assump} + 0.10\, d_{norm}$$

Here, predicate conflict is the fraction of shared proposition keys with opposing non-neutral polarities. Formal warrant conflict compares matching rule statements, compatible premises, and contradictory conclusions. The assumption component averages differences over shared parameters: numeric differences are normalized, while unequal categorical values contribute one. Normative conflict identifies an obligation in one stance matching a prohibition in the other.

The engine also detects causal and definitional warrant conflicts, but they receive no separate term in this aggregate equation. Artifact derivation excludes a pair's disagreements when its aggregate tension is below the default threshold of 0.35. This post-detection filter is the mechanism varied in the benchmark.

4.3 Resolution and assumption pivots

Retained disagreements receive per-clash resolution states. Definitional, normative, and inference-rule clashes can remain irreducible. Factual or causal clashes can become `RESOLVED_BY_EVIDENCE` when regime-level support and contradiction conditions are met. Because resolution receives regime summaries rather than evidence linked directly to each clash, unrelated claims can affect the outcome.

Divergent numeric assumptions receive an arithmetic midpoint and an associated pivot record. Although the schema field is named `validated_pivot`, no empirical validation or sensitivity optimization is performed. The benchmark's conditional-resolution label denotes this recorded midpoint, not proof that it settles a decision.

5. Experimental design

5.1 Scenarios and reference labels

The challenge set contains 34 scenarios: 12 factual, four assumption, two normative, two formal, two causal, one definitional, two mixed, one adversarial cross-claim case, and eight negative controls. Mixed scenarios and the two-parameter assumption case yield 29 expected conflict instances. Evidence conditions include missing, positive, negative, contradictory, contested, rejected, wrong-domain, wrong-regime, unscored, and neutral records.

Fixed evaluators supply stances to the actual orchestrator, verifier, rollup, and artifact derivation. A fixed router selects both stances. No failure boundary is applied during the disagreement experiment. This isolates structural processing from model extraction and routing quality.

Expected outcomes are stored separately from compiler inputs. Cases and labels were drafted with AI assistance after source inspection and have not received independent human annotation. They specify desired retention of explicit conflicts, even when their aggregate magnitude is small, and adopt some implementation conventions. This is an exploratory diagnostic set, not a held-out sample of natural-language reasoning tasks.

5.2 Comparators and metrics

The predicate-only baseline detects opposite non-neutral proposition polarities without resolving them. A direct-claim evidence baseline adds exact domain-key and empirical-regime matching; accepted, scored evidence of one non-neutral polarity can resolve the corresponding factual target. Neither baseline detects non-factual conflict classes, and neither represents a named external system.

PrismCognition is evaluated at aggregate thresholds 0.35, 0.10, 0.05, and zero. Setting zero removes only the aggregate filter; other detectors and thresholds remain active. Two further configurations bind an empty evidence ledger at thresholds 0.35 and zero to isolate evidence access.

Conflict recall counts retained expected instances by type, preserving multiplicity. Exact outcome matching requires equality of the multiset of conflict types and resolution states. It does not measure real-world factual accuracy or explanatory quality. Incorrect resolution counts emitted evidence-resolved states exceeding their expected count within each type.

The four threshold configurations with evidence access produce 136 combinations. Each is checked for JSON replay parity, isolation from later evidence, stance-order invariance, and consistent domain-key-renaming invariance. These are repeated transformations of the same 34 cases, not independent accuracy observations. Ten additional supplied-boundary scenarios test gate behavior. The separate software suite contains 180 tests.

6. Results

6.1 Conflict retention and exact outcomes

The predicate-only baseline detects 15/29 expected conflicts (51.7% recall) and matches 19/34 complete outcomes (55.9%). Direct-claim evidence lookup retains the same 15 conflicts and increases exact matches to 21/34 (61.8%). The default engine detects 15/29 conflicts and matches 20/34 outcomes (58.8%). It therefore does not outperform the stronger baseline on complete outcomes in this set.

At threshold 0.10, the engine retains 24/29 conflicts (82.8%) and matches 28/34 outcomes (82.4%). Threshold 0.05 retains 25/29 (86.2%) and matches 29/34 (85.3%). Removing the aggregate filter retains 29/29 conflicts across all six tested classes and matches 33/34 outcomes (97.1%). These are counts on the authored set, not estimates of general reasoning accuracy.

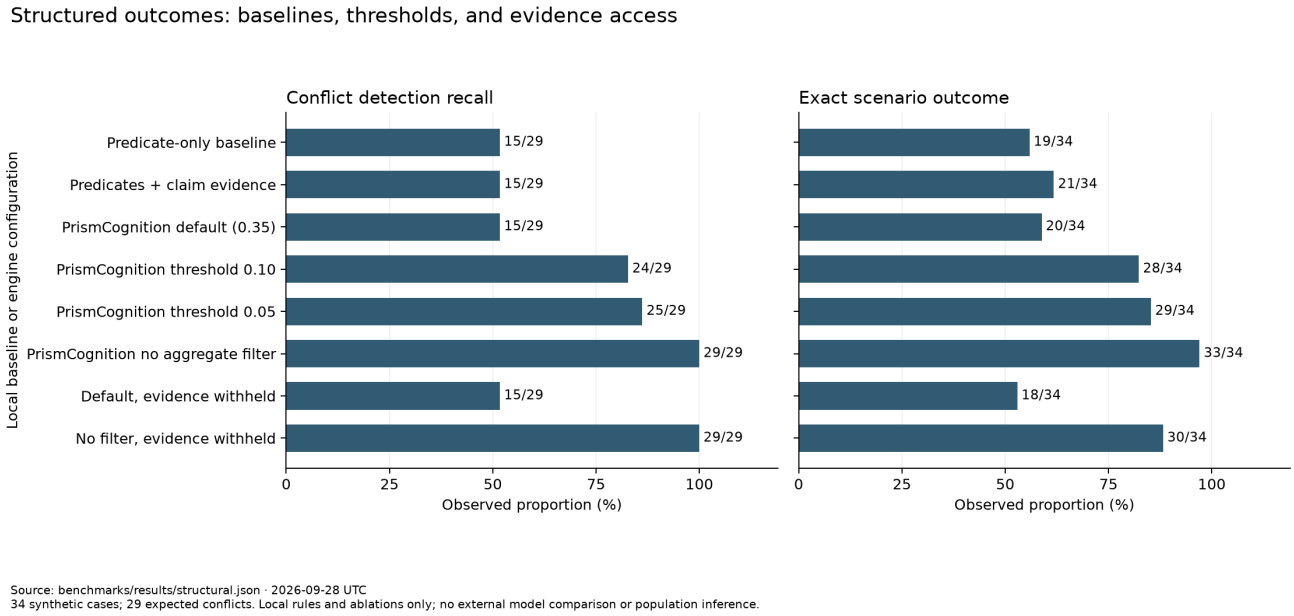


Figure 1. Overall outcomes for two local rule baselines, four engine threshold settings, and two evidence-access ablations. Counts are shown on the bars. All configurations use the same 34 scenario definitions and reference labels; no external-model performance is implied.

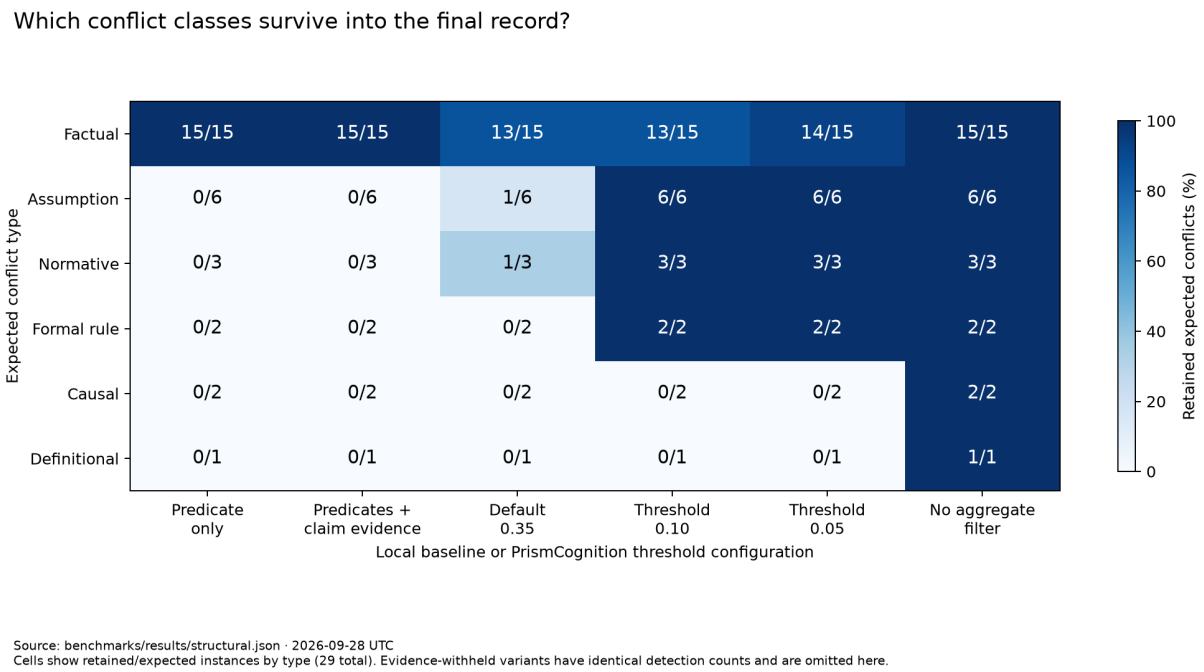


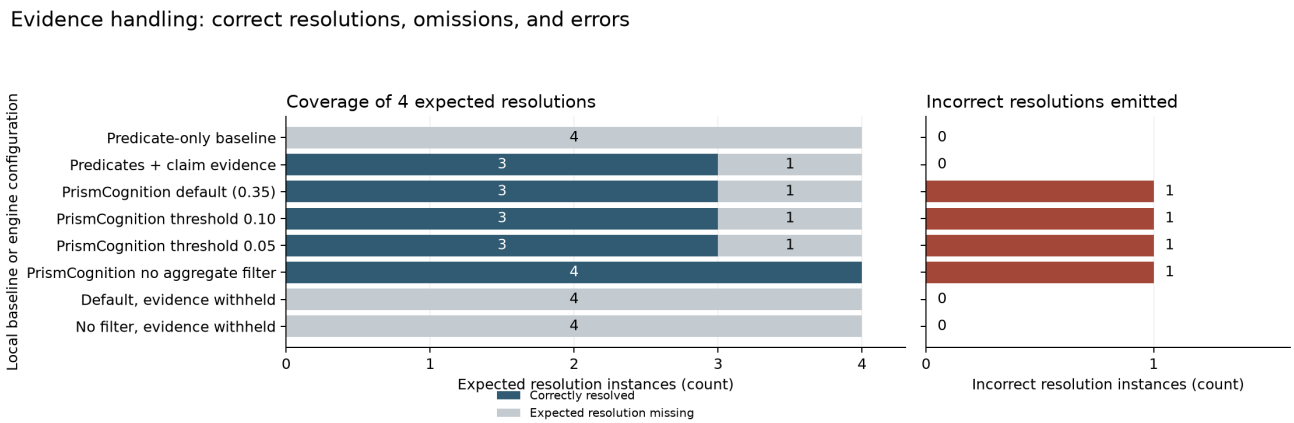
Figure 2. Retained/expected conflict instances by type. Without aggregate filtering, the engine retains all six tested conflict classes. The direct-predicate baselines cover factual conflicts only. The default engine loses isolated classes and two diluted factual conflicts. Evidence-withheld variants have identical detection counts and are omitted here.

6.2 Failure analysis and Case X01

The aggregate weights explain several omissions. Isolated normative conflicts contribute 0.10, assumptions at most 0.20, and formal conflicts 0.30, each below the default 0.35 threshold. Isolated causal and definitional conflicts contribute no independent aggregate term and remain omitted at every positive threshold tested. Agreeing propositions can also dilute factual conflict scores.

In X01, stances disagree about `Valve.open`, while their empirical warrants concern opposing assertions about `Pump.running`. Accepted pump evidence produces asymmetric support/contradiction summaries. The engine then incorrectly classifies the valve clash as evidence-resolved even though no valve evidence exists. This error persists across all four engine settings with evidence access.

At thresholds 0.35, 0.10, and 0.05, one of four emitted evidence resolutions is incorrect. Without the aggregate filter, one of five is incorrect: all four expected resolutions are recovered, but X01 remains. The direct-claim baseline correctly resolves three expected instances, misses the causal instance, and avoids X01.

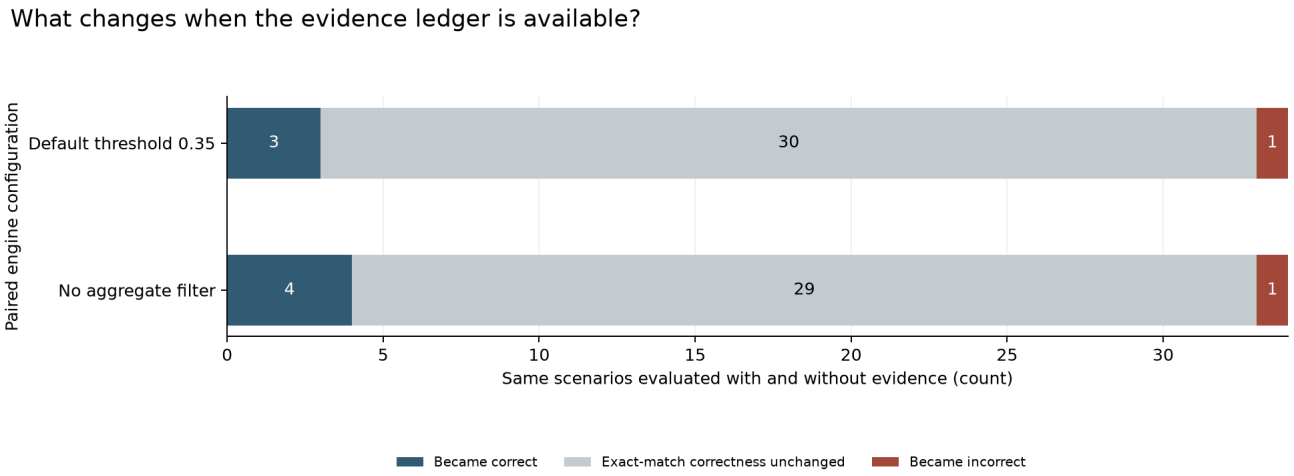


Source: benchmarks/results/structural.json · 2026-09-28 UTC
34 synthetic cases; four expected evidence resolutions. A method that never resolves has zero incorrect resolutions but misses all four.

Figure 3. Correct resolutions and omissions among four expected evidence-resolvable instances, alongside incorrectly emitted resolutions. A method that never resolves anything has no incorrect resolutions but misses all four expected ones. Broader conflict coverage and target-specific evidence matching provide different benefits.

6.3 Evidence-access ablation

Withholding evidence leaves structural detection unchanged but reduces exact matches to 18/34 at the default threshold and 30/34 without the aggregate filter. Restoring evidence makes three default-threshold cases correct and makes X01 incorrect. Without aggregate filtering, it makes four cases correct and again introduces the X01 error. The remaining cases retain their exact-match correctness status.



Source: benchmarks/results/structural.json · 2026-09-28 UTC
Paired comparison on all 34 cases. Each regression is X01 (unrelated-claim evidence); unchanged refers to exact-match correctness only.

Figure 4. Paired effect of evidence access on the same 34 cases. Evidence produces net gains of two exact outcomes at the default threshold and three without aggregate filtering. Grey includes cases that remain correct or remain incorrect; the red regression is X01 in both configurations.

6.4 Replay, transformations, and gate checks

Each of the four replay and transformation properties passes 136/136 checks. All 10 gate scenarios meet their expected survivor counts, including preservation of a geometrically collapsed stance without a predicate match. These results support the tested repeatability and gate behavior; they do not establish system-wide determinism, version-independent replay, or safe failure-boundary generation.

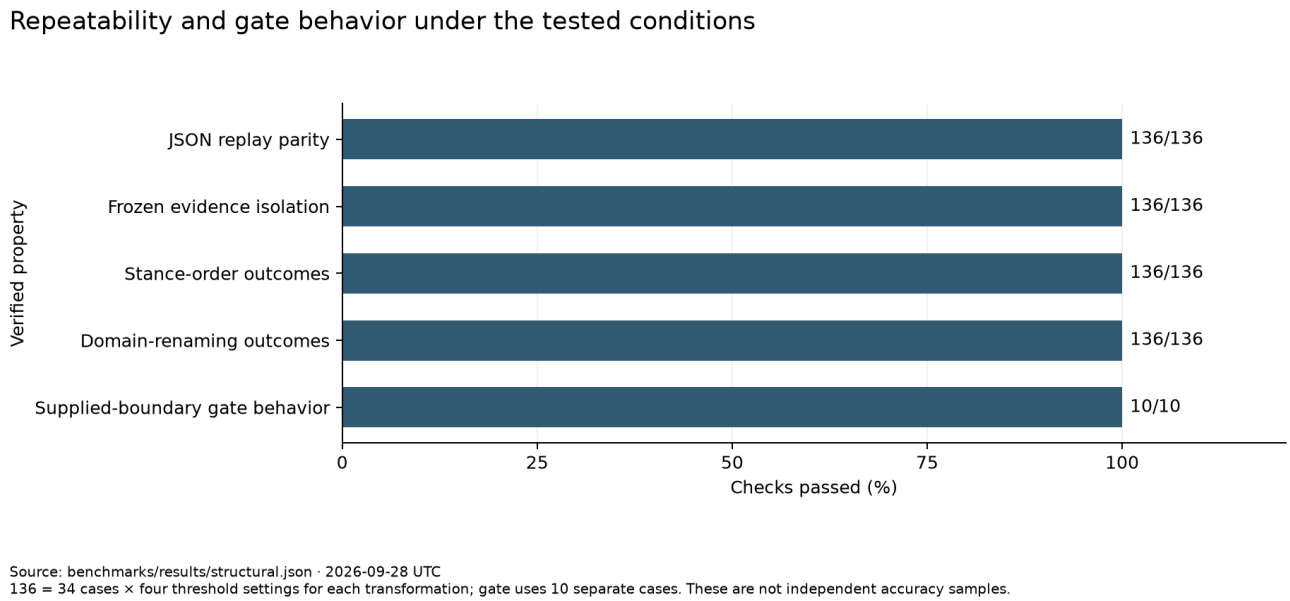


Figure 5. Observed pass counts for four replay/transformation properties and the separate gate set. Each 136-check denominator represents 34 cases at four thresholds. These checks verify software properties, including repeatability of erroneous outcomes, rather than decision accuracy.

7. Reproducibility and limitations

The evaluated library is version 2.1.0, based on revision `382c270270dae0c5bed0eb02187cbe33f2c2a016`, with accompanying benchmark files [4, 5]. The expanded software suite passes 180 tests with 93.99% library statement coverage on Windows/Python 3.12.14. One dependency deprecation warning remains. The benchmark has 13 additional targeted implementation-audit checks; all pass, including deliberate reproduction of X01 and the distinction between promotion and resolution thresholds.

To reproduce the artifacts from the repository containing the accompanying benchmark:

```
python -m pip install -r benchmarks/requirements-lock.txt
python -m benchmarks.run
python -m benchmarks.audit
python -m benchmarks.report
python -m pytest -q -p no:cacheprovider --cov=prismcognition --cov-report=json:benchmarks/results/coverage.json
--junitxml=benchmarks/results/tests.xml
```

Saved outputs contain source and scenario hashes, environment metadata, per-case predictions, failed expectations, and audit references to Python functions. The charts are generated from those recorded results. The base library revision alone does not contain the newly added benchmark files.

The principal validity limit is the use of authored structured inputs. Natural-language extraction, evidence retrieval, source trust, and human decision quality were not evaluated. The scenarios were selected after code inspection, related cases share templates, and thresholds were not selected on a separate development set. The rule baselines address a narrower task and cannot establish superiority over contemporary reasoning systems. No population confidence intervals or significance claims are made.

Operationally, the bundled service lacks authentication and tenant isolation, and file persistence does not provide transactional history across artifacts and bundles. Replay and provenance should not be described as a compliance guarantee. Before broader claims are made, independent reviewers should annotate new cases and comparisons should include live-model systems using matched evidence and documented computation budgets.

8. Conclusion

The benchmark identifies two distinct losses of information: aggregate filtering removes already-detected conflicts, and regime-level evidence summaries can resolve a clash using evidence about another claim. Removing the filter recovers all represented conflicts in this challenge set but leaves the adjudication error intact. The engine's useful measured properties are its coverage of multiple conflict classes under that configuration, evidence-responsive outcomes, and repeatable artifact derivation. These findings provide concrete targets for revision and a reproducible evaluation of disagreement retention. Their relevance to consensus formation in other systems and to real decisions remains to be established.

Availability and declarations

PrismCognition is released under the MIT License and requires Python 3.11 or later. Author affiliation, ORCID, funding, and competing-interest declarations require completion before submission. Codex assisted with scenario and label drafting, benchmark implementation, source inspection, and manuscript editing. The author remains responsible for verifying the content and completing the target venue's AI-use disclosure. This manuscript has not been submitted or peer reviewed.

References

1. Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence, 77*(2), 321-357. Publisher record.

2. Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. (2023). Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv:2305.14325*. Preprint.

3. Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv:2005.11401*. Preprint.

4. Parva, A. PrismCognition, version 2.1.0. Source at evaluated base revision.

5. PrismCognition structural benchmark, evaluator version 1.1, scenario version 1.0 (2026). Protocol, recorded results, and implementation audit. Accompanying local research materials, not an independently published dataset.