

CoSched-Cert: Faithful Multi-Turn Scheduling Dialogue via Solver-Verified Certificates

Zida Yang¹ Minhao Zou^{2,3,*} Yijie Peng^{4,5,6,*}

¹Guanghua School of Management, Peking University, Beijing, China

²School of Computer Science, Peking University, Beijing, China

³Advanced Institute of Information Technology, Peking University,
Hangzhou, China

⁴Department of AI for Management, School of Management and Engineering,
School of Computer Science, Nanjing University, Nanjing, China

⁵Wuhan Institute for Artificial Intelligence, Peking University, Wuhan, China

⁶Xiangjiang Laboratory, Changsha, China

2401111090@estu.pku.edu.cn

*Corresponding authors: zouminhao@pku.edu.cn; pengyijie@nju.edu.cn

Abstract

Multi-turn scheduling assistants must preserve constraint edits and distinguish solver evidence from language-model interpretations. We present CoSched-Cert, a dialogue protocol combining a typed scheduling language, solver-distinguishing clarification, cost-based conflict localization, verified repair synthesis, and lexical certificate auditing. Its guarantees are conditional on the encoded problem and supplied candidate interpretations; lexical support does not establish semantic entailment. SchedNL-Bench-Cert contains 1,000 synthetic Chinese/English utterances and 350 sessions. Gold-AST replay measures internal consistency against labels generated by the same engine, while author-written protocol simulations illustrate selected design choices. Across three live-parser seeds on 52 sessions, session training and registered sibling augmentation raise conditional micro target-constraint agreement from 18.0% to 83.7%. The final configuration has 100% coarse status agreement but only 75.0% conflict-member agreement, and includes a rule informed by a test-session failure. We therefore treat these results as descriptive, not as an unbiased generalization estimate. We define a stricter joint-turn criterion and report the bounds supported by the available aggregates. The contribution is a protocol for exposing inspectable solver evidence and the remaining gap between encoded feasibility and user-intent correctness.

1 Introduction

Scheduling conversations evolve through requests to add constraints, modify prior requirements, delete obsolete conditions, and accept repairs. A language model can produce a well-formed edit while resolving the wrong earlier constraint. Multi-turn degradation has also been observed in broader language-model tasks (Laban et al., 2026). Connecting a model to a solver addresses feasibility only for the formal problem actually encoded. It does not establish that this problem matches the user’s intended schedule.

CoSched-Cert makes this distinction explicit. A parser proposes typed edits; a solver checks whether alternative readings lead to different feasible schedules; conflict and repair procedures emit structured evidence; and an auditor checks selected tokens in the response against that evidence. Figure 1 illustrates the failure modes this design addresses. The certificates support inspection and replay, but their scope ends at the encoded model. A correct solver result can coexist with an incorrect interpretation or an explanation that reverses a relation.

Our contributions are threefold. First, we formulate a persistent scheduling dialogue protocol with solver witnesses, anchored conflict sets, and verified repairs. Second, we describe SchedNL-Bench-Cert, a synthetic bilingual resource separating isolated parsing from session editing. Third, we analyze internal replay, live parsing, and author-

written protocol simulations with distinct evidence boundaries. The live results expose a substantial difference between coarse status agreement and constraint-level agreement; perfect replay scores are implementation consistency checks.

2 Related Work

Natural-language optimization. OptiMUS (Ahmaditeshnizi et al., 2024) formulates and solves optimization models from natural language. ConstraintLLM (Shi et al., 2025) targets constraint-programming models with retrieval and self-correction. Text2Model (Kadioglu et al., 2026) explores solver-agnostic modeling copilots. ReLoop (Lian et al., 2026) uses behavioral verification, NEMO (Song et al., 2026) uses execution-aware coding agents, and SCOPE (Deik et al., 2026) separates query-specific reasoning from reusable planning code. These works motivate checking both execution and modeling behavior. Our focus is the explicit lifecycle of session constraints and the evidence returned after each edit.

Interactive solver interfaces. OptiChat (Chen et al., 2026) connects practitioners to optimization models. MCP-Solver (Szeider, 2025) supports incremental model editing and validation. We do not claim these systems lack editing capabilities. The ADD-only and edit-aware variants in our experiments are author-written abstractions with prescribed policies, not executions or complete reproductions of the original systems. Their outcomes characterize those policies only.

Conflicts and repairs. Optimal constrained unsatisfiable subsets (OCUS) extend conflict explanation with explicit cost and structural constraints (Gamba et al., 2021, 2023). QuickXplain (Junker, 2004) computes preferred explanations and relaxations. CoSched-Cert uses these ideas in a session protocol: provenance determines an anchor and costs, while certificates distinguish a proven optimum from a weaker verified result. Hand-designed costs are not validated user preferences.

3 Problem Formulation

A scheduling instance $\mathcal{P}$ specifies jobs, operations, machines, processing times, eligibility, releases, due dates, setups, and downtime over discrete time. We write B for immutable model axioms, including precedence and resource constraints, and C_t for

active user constraints at turn t . The state is

$$S_t = (\mathcal{P}, B, C_t, A_t, \sigma_t, H_t, \Xi_t), \quad (1)$$

where A_t contains assumption literals, σ_t is an incumbent schedule when available, H_t is dialogue history, and Ξ_t is the certificate log.

Each constraint stores its source utterance, abstract syntax tree (AST), timestamp, owner, and operation. ADD introduces a constraint, MODIFY versions its arguments, and DELETE deactivates it. ASK requests clarification and ACCEPT selects an offered repair. Assumption implications $a_i \Rightarrow c_i$ enable active constraints to be changed without discarding their provenance.

SchedDSL coverage. The registry has 24 schemas: 19 have encoders and 10 support the two-way reification needed by SDC. Appendix B lists the coverage. A deadline $\text{Deadline}(J, d)$ requires job J to finish by d ; increasing d relaxes it. A release bound is relaxed by decreasing it. These directions describe nested feasible sets, not improvements in user utility. Unencoded schemas and unresolved interpretations cannot receive the corresponding solver guarantee.

Solver status. CP-SAT distinguishes OPTIMAL, FEASIBLE, INFEASIBLE, UNKNOWN, and MODEL_INVALID (Google OR-Tools, 2024). A feasible witness suffices to establish satisfiability; it need not be optimal. UNKNOWN must not be interpreted as infeasibility, and invalid models do not support certificates of feasibility. The evaluation’s coarse status agreement is narrower than correctness of the full session state.

4 CoSched-Cert Protocol

Figure 2 gives the protocol. Its contract is to expose the evidence and its certification level before rendering a response. The LLM is responsible for interpretation and verbalization; the symbolic procedures check the resulting formal objects.

4.1 Solver-Distinguishing Clarification

For candidates $\hat{c}^{(a)}$ and $\hat{c}^{(b)}$, instance-relative semantic equivalence (IRSE) means that they select the same schedules within the current background:

$$\hat{c}^{(a)} \equiv_{B, C_t} \hat{c}^{(b)} \iff \nexists \sigma \in \text{Sol}(B \wedge C_t) : \sigma \models \hat{c}^{(a)} \oplus \hat{c}^{(b)}. \quad (2)$$

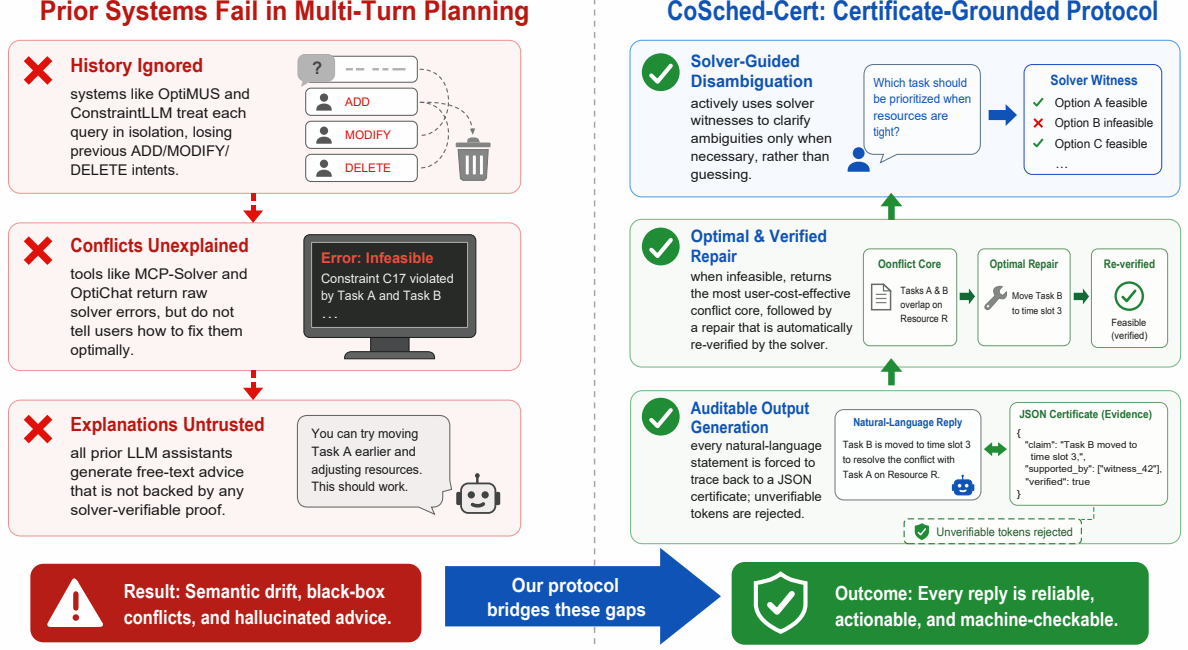


Figure 1: Conceptual motivation and protocol design. The universal wording in this illustration is not an empirical conclusion about all prior assistants or every reply. Our comparisons concern only the author-written policies in Appendix D; complete prior systems were not evaluated. “Optimal” is conditional on the specified formal objective and completed proof checks. “Reliable” and “traceable” concern checked certificate values in the encoded state, not user-intent correctness, preference alignment, or relational entailment.

With reification variables z_a, z_b , SDC asks whether

$$B \wedge C_t \wedge (z_a \Leftrightarrow \hat{c}^{(a)}) \wedge (z_b \Leftrightarrow \hat{c}^{(b)}) \wedge (z_a \oplus z_b) \quad (3)$$

is satisfiable. A satisfying assignment is a distinguishing witness; proven infeasibility establishes equivalence. Union-find merges proven-equivalent candidates. Multiple distinguishable classes trigger a clarification question illustrated by witness schedules.

Proposition 1. With correct reification and decisive solver outcomes for the required pairs, SDC partitions the supplied finite candidate set according to Eq. 2. An unresolved query cannot establish equivalence or produce a witness. If $B \wedge C_t$ is itself infeasible, all readings are vacuously equivalent under this definition; the conflict must be resolved before interpreting equivalence as a reason to commit an edit.

Candidate coverage. Self-consistency sampling can miss alternative readings. Registered sibling augmentation adds alternatives with the same arguments within a declared family. The evaluated catalogue contains the precedence family, including BEFORE and IMMEDIATEBEFORE. “Schema-complete” here means complete within

that registered family, not across SchedSL or all interpretations of natural language. The worst-case number of pairwise queries is $k(k-1)/2$ for k candidates. Sampling alone and registered augmentation are evaluated separately.

4.2 Anchored Conflict Localization

After an infeasible edit, OCUS-UX seeks a low-cost conflict containing the new or modified constraint c_{new} . Let

$$f_t(c_i) = \lambda_{\text{age}} \log(1 + t - \tau_i) + \lambda_{\text{own}} \mathbf{1}[u_i \neq u_t] + \lambda_{\text{sch}} \text{complexity}(c_i), \quad (4)$$

where τ_i and u_i are the creation turn and owner, and u_t is the current user. The default weights are $(1, 1, 1)$. They encode a design preference rather than elicited human preferences. The optimization is

$$U^* \in \arg \min_{U \subseteq C_t} \sum_{c \in U} f_t(c) \quad \text{s.t. } B \wedge U \models \perp, \quad (5)$$

$c_{\text{new}} \in U.$

Proposition 2. Exact minimum-cost constrained hitting-set search with decisive feasibility checks

CoSched-Cert: Turning Multi-turn Natural Language Scheduling Edits into Audited Solver Certificates

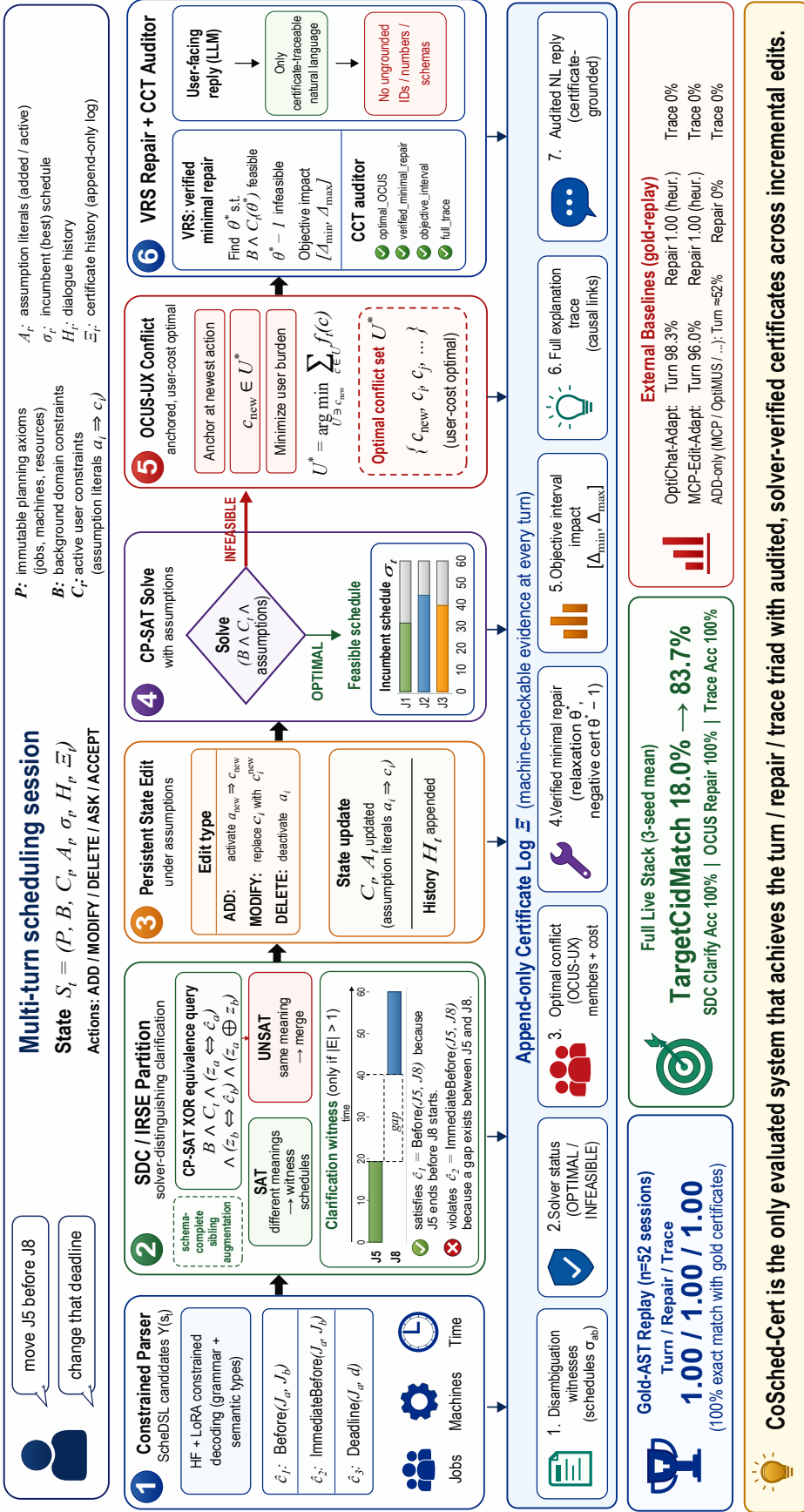


Figure 2: Protocol architecture and descriptive aggregates. Live parsing uses validation and repair, not grammar-constrained inference. Sibling completeness covers the registered precedence family only. Optimal/minimal labels require completed proofs; other feasible and unresolved outcomes are omitted. Cost minimization is subject to Eq. 5, not measured user burden. Trace denotes lexical support, not causal entailment. Gold replay is internal consistency; baselines are author-written simulations whose zero trace scores reflect their certificate format. Target percentages are pooled micro rates; the final live row includes a test-informed rule. “OCUS Repair 100%” differs from conflict-member agreement, which is 75%. The triad concerns these simulations and certificate-specific checks, not general superiority or JointTurnExact.

returns an optimum of Eq. 5 when such a set exists and search completes. On budget expiry, only properties actually verified can be recorded: a raw infeasible core, an anchor-preserving subset-minimal core if minimization finishes, or a cost optimum if certified. Timeout alone does not imply subset-minimality. Appendix A states the assumptions explicitly.

This is session-provenance-aware localization: changing the parsed target changes the formal anchor and may change the resulting conflict, even when the solver is correct. It is therefore evaluated separately from status agreement.

4.3 Verified Repair Synthesis

For a finite monotone parameter domain, define $\theta_1 \preceq_q \theta_2$ so that θ_2 is at least as relaxed as θ_1 . Thus

$$\text{Sol}(B \wedge C_{-q} \wedge \psi_q(\theta_1)) \subseteq \text{Sol}(B \wedge C_{-q} \wedge \psi_q(\theta_2)). \quad (6)$$

Here C_{-q} contains every other active constraint. VRS uses bisection between an infeasible bound and a feasible bound. Every proposed repair is checked against the full resulting encoded state, not just the extracted conflict. For non-monotone changes, bounded removal search considers up to a specified number of constraints; incomplete search does not prove global minimality.

Proposition 3. On a finite totally ordered monotone chain with a valid bracket and decisive checks, bisection finds the least feasible parameter. Its feasible witness and the infeasibility of its immediate predecessor establish this minimality. An unresolved predecessor check permits a feasibility claim only. An objective-change interval must likewise retain solver bounds rather than present an incumbent as a proven optimum.

These certificates do not show that a relaxation is desirable. They verify feasibility for the parsed state, which may already differ from user intent. No study in this work establishes human acceptance, cognitive burden, or preference alignment of the repair ranking.

4.4 Certificates and Lexical Auditing

A certificate records solver status, the edited constraint and provenance, conflict members, proposed repairs, verification flags, objective bounds, and disambiguation witnesses. Certification levels distinguish optimal OCUS from verified minimal or raw cores, and verified minimal repairs from merely

verified repairs. They must not imply stronger properties than the checks establish.

Proposition 4 (lexical support). Let $E(t)$ extract the identifiers, schema names, and numbers checked in a reply t , and let $V(\xi)$ be the permitted values in certificate ξ . Acceptance requires $E(t) \subseteq V(\xi)$. This establishes support for those extracted tokens only, conditional on the extractor. It does not establish that the entire sentence follows from the certificate.

For example, a certificate for “J1 precedes J2” contains both identifiers. A response saying “J2 precedes J1” can pass a token-membership test. We therefore use *lexical TraceRate*, not semantic faithfulness, for this metric. No relation-swapping acceptance rate was measured. A complete guarantee would also require correct parsing, encoding, and relational entailment.

5 Benchmark and Experimental Setup

Data construction. SchedNL-Bench-Cert has 60 synthetic job-shop/flexible-job-shop instances, 1,000 template-rendered utterances (490 Chinese, 510 English), and 350 sessions. ASTs are sampled first, surfaces are rendered with lexical variation, and the CoSched-Cert engine generates statuses, ambiguity labels, conflicts, repairs, and certificates. Consequently, the gold labels share the implementation and semantics being evaluated.

Utterance categories are ordinary parsing (480), ambiguity (214), history-dependent editing (163), and out-of-scope input (143). The recorded train/validation/test counts are 700/150/150 utterances, 245/53/52 sessions, and 42/9/9 instances. The full collection averages 16.78 turns per session. Templates and paraphrase families are not documented as disjoint across splits, and no claim of template-held-out generalization is made. Per-language accuracy is unavailable in the aggregates; bilingual coverage alone does not establish comparable Chinese and English performance.

Parser and solver. The parser is Qwen3.5-9B (Qwen Team, 2026) with LoRA (Hu et al., 2022), rank 16, scale 32, learning rate 2×10^{-4} , three epochs, and training seeds 2027–2029. Session augmentation adds 3,485 recorded edit examples to 700 utterances (4,185 total). The reported runs use an NVIDIA A800-80GB and CP-SAT with four workers and a 2-second budget per feasibility check. Appendix C records further settings.

The implementation includes structured decod-

ing with XGrammar (Dong et al., 2025), but the reported live results use the HF validate-and-repair path without XGrammar at inference. The sampling baseline uses $k = 8$ and temperature 0.7. SDC decoupling uses an utterance-only ASK prompt, temperature 1.0, and $k = 16$. The final row combines registered sibling augmentation with a horizon normalization rule. That rule was informed by a failure in test session `sess_0332`; comparisons on these 52 sessions are therefore exploratory and test-informed.

Evidence boundaries. Gold-AST replay is an implementation regression test. Same-engine re-computation without stored certificates can check replay consistency, but cannot detect an encoding mistake shared by generation and checking. Live parsing measures behavior on synthetic sessions; it is not an industrial deployment evaluation. Protocol simulations fix gold ASTs and vary author-specified policies; they do not measure the original published systems.

5.1 Metrics and a Stricter Joint Criterion

SemAcc and ParseExactMatch concern AST agreement in their respective isolated/session settings. TurnStatusMatch compares coarse solver outcomes on non-ASK actionable turns. It does not require the operation, target, arguments, or conflict set to be correct. Micro TargetCidMatch pools correct target resolutions over the reported MODIFY/DELETE resolution records; its denominator can change between configurations. OCUSMatch compares conflict members on scored infeasible turns. RepairValid@3 concerns an offered top-three repair verified against the encoded state; lexical TraceRate concerns token support.

A stricter joint criterion would require all applicable conditions for each evaluated turn:

$$J_t = O_t \wedge I_t \wedge A_t^{\text{ok}} \wedge S_t^{\text{ok}} \wedge C_t^{\text{ok}} \wedge R_t^{\text{ok}}, \quad (7)$$

where O , I , A^{ok} check operation, intended target, and arguments; S^{ok} checks status; C^{ok} checks the intended-state conflict when applicable; and R^{ok} verifies a repair against that intended state. Non-applicable factors are true. ASK turns additionally require the correct clarification decision and valid witnesses. A wrong operation or parse failure is a failure, not grounds for removing the turn from the denominator. JointTurnExact is the average of J_t over all prespecified turns, including these failures.

The retained aggregates cannot identify this intersection. We report no invented point estimate.

On the 264 scored edit records, any such joint rate is at most $221/264 = 83.7\%$; on the 180 scored conflict records it is at most $135/180 = 75.0\%$. These are conditional upper bounds on different subsets, not a whole-session score and not mutually multiplicative probabilities.

6 Results and Interpretation

6.1 Live Session Parsing

Table 1 retains the reported configurations. Prompt-only active context reduces parse exact match, whereas session-formatted fine-tuning improves target resolution. However, the changing target denominator (189 to 264) means the 18.0% to 83.7% comparison is a change in conditional resolution accuracy, not a measured improvement on every gold edit turn.

Session tuning also reduces sampling diversity: SDC accuracy falls to 40.4%, and decoupling recovers 77.6%. The full final configuration reports 100%, but the isolated effect of the horizon rule is not available. Its removal cannot be evaluated from aggregate data, nor can a one-session failure be converted into a three-seed result without the underlying records. The final configuration should therefore be interpreted as a test-informed diagnostic endpoint.

The final configuration resolves 99/108 MODIFY targets and 122/156 DELETE targets, totaling 221/264. Thus 43 scored target resolutions are wrong despite perfect aggregate status agreement. Conflict agreement is 75.0%: 45 of 180 scored conflict records disagree with gold membership while retaining the correct coarse status. This is a substantive end-to-end error. A repair verified for the parsed state cannot be promoted to a repair of the user’s intended state merely because it restores feasibility.

6.2 Gold Replay and Component Checks

The original reports give 100% agreement for gold replay on the recorded status, clarification, conflict, repair, and lexical trace metrics. Isolated SemAcc is also reported as 100% on 150 test utterances across three seeds. These values describe a bounded synthetic setup; they are not evidence of universal parsing, independent encoding correctness, or relational faithfulness.

Removing SDC eliminates witness-based clarification in the specified ablation. Removing VRS eliminates the verification step, and removing the

Configuration	Status	AST exact	Target (micro)	SDC	n_{target}
Static baseline, strict	.955 $\pm$.003	.882 $\pm$.003	.180	.962 $\pm$.051	189
Active prompt + op normalization	.962 $\pm$.012	.515 $\pm$.011	.239	.769 $\pm$.084	222
Session fine-tuning	.999 $\pm$.000	.962 $\pm$.001	.814	.404 $\pm$.058	264
+ SDC decoupling	.999 $\pm$.000	.962 $\pm$.001	.814	.776 $\pm$.059	264
+ registered siblings + horizon rule	1.000 $\pm$.000	.967 $\pm$.001	.837	1.000 $\pm$.000	264

Table 1: Descriptive live-parser results on 52 synthetic sessions. Rates other than pooled micro target agreement are reported means $\pm$ sample standard deviations over three training seeds, rounded to three decimals. Status is TurnStatusMatch; SDC is SDCClarifyAcc. The last row includes a test-informed rule, so its 100% SDC value is not an unbiased generalization result. These aggregates do not report JointTurnExact.

Simulated policy	Agreement	Repair
CoSched-Cert	1.000	1.000
OptiMUS-Adapt	.521	.000
MCP-Solver-Adapt (ADD-only)	.521	.000
MCP-Edit-Adapt	.960	1.000
ReOpt-Patch-Adapt	.517	.000
OptiChat-Adapt	.983	1.000
CoSched-NoSDC	.933	.962
CoSched-NoOCUS	1.000	1.000

Table 2: Author-written simulations on 52 gold-AST sessions, not full reproductions of published systems. Policies are specified in Appendix D. The turn-label agreement column is retained under its original simulation meaning and must not be read as JointTurnExact.

auditor removes enforced lexical support. Zero scores for these disabled checks are partly consequences of the metric definitions. They do not by themselves show that every heuristic repair is infeasible or every free-text explanation is false.

6.3 Author-Written Protocol Simulations

Table 2 retains the original simulation outcomes, using the table source as the numerical record. In particular, the NoSDC value is .933; a conflicting .928 in the earlier prose was a transcription inconsistency. The trace column is omitted because the simulated alternatives do not emit our certificate representation. Repair success in this table is the original simulation measure, not interchangeable with VRS-verified RepairValid@3.

OCUS ablation. CoSched-NoOCUS and the full system have identical reported turn-label agreement (1.000) and repair success (1.000). The earlier burden aggregate is also identical (.77). Thus the session benchmark does not demonstrate a user-facing advantage from cost-optimal localization. The source reports unchanged conflict members under cost perturbations; the retained files disagree on the exact perturbation grid, preventing verification of that grid. An 80-case targeted stress

experiment reports an optimality advantage, but that result cannot establish user benefit on the 52-session benchmark or preference alignment of the cost function.

Burden ambiguity. The earlier table reports .77 for the full system and .04 for NoSDC, but the prose alternately defines burden as all clarifications or only unnecessary clarifications, plus failed repairs, and gives another full-system value (.73). No event-level decomposition is retained. The statistic is therefore removed from comparative scoring; its difference is not evidence that either policy reduces user burden. In particular, counting a correct clarification as a cost measures interaction frequency, not clarification error.

Uncertainty. Three-seed standard deviations describe training variation on a fixed collection. Turn-level tests do not account for within-session dependence, and the retained files do not support reconstruction of the earlier confidence intervals or significance tests. We omit those inferential claims. No original-system-author validation of the protocol abstractions is documented.

7 Discussion and Conclusion

CoSched-Cert separates three questions: whether the solver verified the encoded state, whether the parser preserved user intent, and whether the response is supported by its evidence. SDC supplies concrete witnesses for distinguishable candidate readings; OCUS-UX declares the cost and anchor behind a conflict; VRS checks proposed changes; and lexical auditing exposes unsupported identifiers and numbers. These mechanisms make failures more inspectable without making the entire dialogue correct by construction.

The retained live results show why coarse status alone is insufficient: 100% status agreement coexists with 83.7% conditional target agreement and 75.0% conflict-member agreement. The camera-

ready analysis defines a stricter joint criterion and states what cannot be recovered from aggregates. Claims about other systems are limited to the author-written simulations, and gold replay is framed as internal consistency. Applying the protocol in another domain would require appropriate encodings, candidate families, repair operators, and independent validation; transfer was not evaluated here.

Limitations

The benchmark is synthetic and template-rendered, with no human-authored dialogue evaluation or professional-planner study. Generation and evaluation share ScheDSL and the same solver machinery. Same-engine recomputation cannot reveal a shared semantic encoding error. Template-disjointness is unverified, separate Chinese/English accuracy is unavailable, and repeated inspection of the test sessions, including the horizon rule, limits claims about generalization.

The model study uses one base model and three training seeds. Only 19 of 24 schemas are encoded and 10 reifiable; registered sibling augmentation covers a precedence family. A finite-candidate soundness result does not guarantee interpretation coverage. On larger instances, timeouts may prevent equivalence, minimality, or optimality certification. The conditional statements in the method do not establish that every code path in an unavailable implementation satisfies those conditions.

Lexical traceability does not imply relational entailment. Solver-valid repairs may address the wrong parsed state, and cost weights have not been validated against human preferences. JointTurnExact, an independent re-encoding audit, rule-removal results, and language-specific performance cannot be reconstructed from the retained aggregates. The supplement supports consistency checks of selected reported aggregates; it does not contain the original dataset, model checkpoints, solver implementation, or per-turn prediction logs and is not a complete experiment reproduction package.

Ethics Statement

The described evaluation uses synthetic scheduling instances and template-rendered utterances, rather than participant conversations. No human-subject study is reported. This limits the extent to which the results can establish practical usefulness, accessibility, or impacts on planners and affected workers.

In deployment, a feasible schedule can still conflict with unstated worker needs, organizational obligations, or a user’s intended edit. Cost-based conflict selection may privilege particular owners or older constraints through hand-designed weights. Human review and explicit acceptance are needed before acting on proposed changes, especially when resources or workers are affected. Provenance logs could contain sensitive operational information in real use and would require access controls and appropriate retention policies. Certificates should disclose their scope and uncertainty so that users do not mistake lexical support or solver feasibility for complete factual correctness.

Acknowledgments

We thank the reviewers and area chair for feedback that helped clarify the evidence boundaries, evaluation metrics, and limitations of this work. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 72325007 and 72501012, and the China Postdoctoral Science Foundation under Grants 2025T180219 and 2025M770795. This work was also supported in part by Xiangjiang Laboratory Key Project under Grant 23XJ02004 and the Science and Technology Innovation Program of Hunan Province under Grant 2024RC7003.

Conflict of interest. The authors declare no conflicts of interest.

References

- Ali Ahmaditeshnizi, Wenzhi Gao, and Madeleine Udell. 2024. [OptiMUS: Scalable optimization modeling with \(M\)LP solvers and large language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 577–596. PMLR.
- Hao Chen, Gonzalo Esteban Constante-Flores, Krishna Sri Ipsit Mantri, Sai Madhukiran Kompalli, Akshdeep Singh Ahluwalia, and Can Li. 2026. [OptiChat: Bridging optimization models and practitioners with large language models](#). *INFORMS Journal on Data Science*, 5(3):199–220.
- Derrick Goh Xin Deik, Quanyu Long, Zhengyuan Liu, Nancy F. Chen, and Wenya Wang. 2026. [Programming over thinking: Efficient and robust multi-constraint planning](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 43785–43837, San Diego, California, United States. Association for Computational Linguistics.

- Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ziyi Xu, Yilong Zhao, Ruihang Lai, and Tianqi Chen. 2025. [XGrammar: Flexible and efficient structured generation engine for large language models](#). In *Proceedings of Machine Learning and Systems*, volume 7. MLSys.
- Emilio Gamba, Bart Bogaerts, and Tias Guns. 2021. [Efficiently explaining CSPs with unsatisfiable subset optimization](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 1381–1388. International Joint Conferences on Artificial Intelligence Organization.
- Emilio Gamba, Bart Bogaerts, and Tias Guns. 2023. [Efficiently explaining CSPs with unsatisfiable subset optimization](#). *Journal of Artificial Intelligence Research*, 78:709–746.
- Google OR-Tools. 2024. [CP-SAT solver](#). Documentation updated August 28, 2024; accessed September 11, 2026.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Ulrich Junker. 2004. [QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems](#). In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 167–172. AAAI Press / The MIT Press.
- Serdar Kadioglu, Karthik Uppuluri, and Akash Singirikonda. 2026. [Text2Model: Modeling copilots for text-to-model translation](#). *Preprint*, arXiv:2604.12955.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2026. [LLMs get lost in multi-turn conversation](#). In *The Fourteenth International Conference on Learning Representations*.
- Junbo Jacob Lian, Yujun Sun, Huiling Chen, Chaoyu Zhang, Hanzhang Qin, and Chung-Piaw Teo. 2026. [ReLoop: Structured modeling and behavioral verification for reliable LLM-based optimization](#). *Preprint*, arXiv:2602.15983.
- Qwen Team. 2026. [Qwen3.5-9B](#). Model card; Apache-2.0 license; accessed September 11, 2026.
- Weichun Shi, Minghao Liu, Wanting Zhang, Langchen Shi, Fuqi Jia, Feifei Ma, and Jian Zhang. 2025. [ConstraintLLM: A neuro-symbolic framework for industrial-level constraint programming](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15999–16019, Suzhou, China. Association for Computational Linguistics.
- Yang Song, Anoushka Vyas, Zirui Wei, Sina Khoshfetrat Pakazad, Henrik Ohlsson, and Graham Neubig. 2026. [NEMO: Execution-aware optimization modeling via autonomous coding agents](#). *Preprint*, arXiv:2601.21372. Accepted at ICML 2026.
- Stefan Szeider. 2025. [Bridging language models and symbolic solvers via the model context protocol](#). In *28th International Conference on Theory and Applications of Satisfiability Testing (SAT 2025)*, volume 341 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:12, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

A Conditional Guarantees and Proof Sketches

A.1 SDC

Assume both candidate constraints admit correct two-way reification in the same schedule-variable space. In Eq. 3, the XOR condition forces exactly one candidate to hold. Any satisfying assignment therefore supplies a distinguishing schedule satisfying the common background. If the query is proven infeasible, no such schedule exists and the candidates have identical solution sets relative to that background.

Equality of solution sets is an equivalence relation, so merging proven-equivalent pairs is sound. Comparing every remaining pair with decisive outcomes establishes the complete partition of the supplied finite set. This argument says nothing about an interpretation absent from that set. A timeout yields neither an equivalence proof nor a distinguishing schedule. For an infeasible common background, equivalence is vacuous and is not a substitute for conflict resolution. For MODIFY candidates, the background must remove the superseded constraint before comparing proposed replacements.

A.2 OCUS-UX

Assume B is satisfiable, costs are nonnegative, and an anchored infeasible subset exists. The anchor is a constraint on every hitting-set candidate, rather than merely a seed from which unconstrained search begins. At each iteration, minimum-cost constrained hitting-set optimization supplies a lower bound on the cost of any admissible unsatisfiable subset. If its candidate is satisfiable, a correction set obtained from that satisfiable set supplies a new hitting constraint. If the candidate is infeasible, it attains the lower bound and is optimal (Gamba et al., 2021, 2023).

The conclusion depends on exact optimization and decisive feasibility tests. Under zero costs, a cost optimum need not be subset-minimal unless tie-breaking or a verified minimization pass establishes that property. Similarly, after a timeout, an available infeasible set may be returned only with the weaker status actually verified. Anchor-preserving subset minimality requires that no proper anchored subset remains infeasible. If the pre-edit background is satisfiable, every conflict introduced by the edit must contain the new constraint; without that invariant, an anchor require-

ment can retain a redundant member relative to unconstrained MUS minimality.

A.3 VRS

Index the finite permitted relaxation chain as $\theta_0 \prec_q \dots \prec_q \theta_m$. Keep all nontarget constraints fixed. Monotonicity means the feasibility predicate on this chain is false up to a boundary and true thereafter. With an infeasible lower endpoint and feasible upper endpoint, bisection preserves the bracket and shrinks its index interval. On termination, the upper endpoint is feasible and its predecessor infeasible, proving minimality on this chain.

The statement requires a totally ordered finite chain and exact feasibility decisions. It does not cover arbitrary partial orders, unknown outcomes, simultaneous repairs outside the declared search space, or user desirability. For objective minimization, compare a specified feasible reference model with certified optimum bounds $[l_0, u_0]$ to the repaired model with bounds $[l_1, u_1]$. The change lies in $[l_1 - u_0, u_1 - l_0]$. An infeasible model has no finite optimum to subtract. A point difference needs exact optima, or an explicitly different comparison to a fixed incumbent.

A.4 Lexical Support

Given extraction function E and permitted certificate-value set V , acceptance by a membership auditor establishes $E(t) \subseteq V(\xi)$. It cannot establish that the extractor identified every factual assertion or that relationships are preserved. A token-preserving relation reversal is a counterexample to interpreting this condition as entailment. No empirical robustness claim against such attacks follows from the reported TraceRate.

B SchedSL Registry

Table 3 reproduces the supplied registry. Encoded means an implementation was reported; reifiable means two-way reification was reported for SDC. These flags are not an independent verification of code absent from the supplement.

C Reported Training and Solver Settings

The recorded session augmentation comprises 3,093 ADD, 147 MODIFY, and 245 DELETE examples (3,485 total); with the 700 utterance training examples this gives 4,185. The original documentation reports active-constraint context in 93% of augmentation examples. These are reported

Schema	Category	Type	Encoded	Reifiable	Relaxation parameter/direction
Before	precedence	hard	Yes	Yes	—
StrictBefore	precedence	hard	Yes	Yes	decrease separation δ
ImmediateBefore	precedence	hard	Yes	Yes	—
Deadline	due	hard	Yes	Yes	increase deadline d
Release	release	hard	Yes	Yes	decrease release r
MaxFlow	flow	hard	Yes	Yes	increase limit L
MaxTardiness	due	hard	Yes	Yes	increase limit T
MaxMakespan	global	hard	Yes	Yes	increase limit L
StartWindow	window	hard	Yes	No	widen window
EndWindow	window	hard	Yes	No	widen window
LockStart	lock	hard	Yes	Yes	—
Eligible	assignment	hard	Yes	No	—
Forbid	assignment	hard	Yes	No	—
SameMachine	assignment	hard	Yes	No	—
DifferentMachine	assignment	hard	Yes	No	—
Downtime	availability	hard	Yes	No	—
MaxDeviation	stability	hard	Yes	No	increase deviation δ
NoLaterThanIncumbent	stability	hard	Yes	Yes	increase incumbent bound
MaxSetups	setup	hard	No	No	increase limit K
NoAdjacentFamily	setup	hard	No	No	—
BatchFamily	setup	hard	No	No	—
Capacity	resource	hard	No	No	increase capacity q
Priority	soft	soft	Yes	No	—
SetupPenalty	soft	soft	No	No	—

Table 3: ScheDSL catalogue: 24 schemas, 19 encoded, 10 reifiable. A listed relaxation direction is a schema property; it does not imply that an unencoded schema was evaluated.

Parameter	Recorded value
Base model	Qwen3.5-9B
LoRA rank / scale	16 / 32
Target modules	q/k/v/o and MLP projections
Learning rate	2×10^{-4} ; cosine schedule
Warmup	3%
Batch / accumulation	4 / 4
Epochs	3
Maximum sequence	1,024 tokens
LoRA dropout	.05
Precision	bf16; gradient checkpointing
Training seeds	2027, 2028, 2029
CP-SAT	4 workers; 2 seconds/check
GPU	NVIDIA A800-80GB

Table 4: Settings retained from the original experiment documentation. Exact package versions and model revision hashes are not supplied.

setup details, not counts regenerated from training files in this supplement.

The stored final-configuration arrays contain status agreement $[1, 1, 1]$, SDC accuracy $[1, 1, 1]$, and conflict agreement $[\cdot75, \cdot75, \cdot75]$. ParseExactMatch values are approximately .967327, .966392, and .968529, whose sample standard deviation is .001071. A zero standard deviation after rounding does not establish certainty or performance on unseen tasks.

D Simulation Policies

The following descriptions specify the author-written abstractions behind Table 2; names indicate inspiration, not implementation equivalence or endorsement by original-system authors.

OptiMUS-Adapt: rebuilds from ADD utterances and omits MODIFY, DELETE, and repair behavior. **MCP-Solver-Adapt:** accumulates ADD calls and ignores MODIFY, DELETE, ASK, and ACCEPT. This is deliberately restricted and does not describe MCP-Solver’s documented editing support. **MCP-Edit-Adapt:** adds, replaces, and deletes items; ASK commits the first candidate; ACCEPT drops the latest constraint. **ReOpt-Patch-Adapt:** treats MODIFY as a new-constraint patch and has no DELETE semantics. It is a local patch-policy label, not a separately reproduced published system. **OptiChat-Adapt:** uses incremental edits and raw MUS explanations; ACCEPT drops the anchor without VRS. **CoSched-NoSDC:** uses the remaining protocol but auto-lands a Before reading. **CoSched-NoOCUS:** omits cost-optimal localization; the retained description does not specify a sufficiently detailed fallback policy for independent reproduction.

The table’s RepairSuccessRate is preserved as an original simulation outcome. The retained source describes it inconsistently as a repair offered or a

Supported quantity	Value
MODIFY target matches / scored	99/108
DELETE target matches / scored	122/156
Pooled target matches / scored	221/264
Pooled target mismatches	43
Pooled target agreement	83.7121%
Reported conflict matches / scored	135/180
Reported conflict mismatches	45
Reported conflict agreement	75.0%
Strict joint upper bound: edit subset	83.7%
Strict joint upper bound: conflict subset	75.0%

Table 5: Conditional quantities for the final configuration. Target counts come from the JSON; the conflict denominator comes from the original appendix report, with the rate corroborated by the seed arrays. Bounds are not estimates of all-turn JointTurnExact.

successful heuristic repair. Its per-event implementation and denominator are absent. Consequently, it is not used as evidence of solver-certified repair validity or superiority to original systems. Missing typed certificates are likewise a representation difference, not a measured factual-error rate.

E Test-Informed Horizon Rule

The reported rule concerns a post-repair deadline restatement. If the parser proposes MODIFY on Deadline, the target was repaired, the new deadline equals an active MaxMakespan horizon, and the utterance lacks explicit editing cues, the operation is normalized to ADD. The original description explicitly relates this rule to sess_0332, one of the 52 test sessions.

The final SDC result in Table 1 includes this rule. There is no stored matched run with the rule disabled, so neither its isolated effect nor rule-free clarification accuracy can be recovered. The earlier SDC-decoupled row changes a different mechanism and is not a rule-removal control. A fresh held-out evaluation would be needed to estimate generalization of the final configuration; no such experiment is claimed here.

F Aggregate Consistency and Availability

The supplement preserves the original aggregate JSON and supplies an auditable arithmetic checker. Table 5 summarizes values supported directly by that JSON and by explicitly identified original report tables. The checker recomputes counts, ratios, and seed summary statistics only; it does not run the original model or solver experiments.

Some retained records are inconsistent or insufficiently specified. The benchmark JSON lists 10

independent-audit sessions, whereas the original audit table lists 20; neither establishes an independent semantic audit. The main and generated cost-sensitivity tables specify different perturbation grids. Earlier operation-confusion graphics disagree with the prose about the baseline MODIFY-to-ADD count. These unsupported details and plots are not used as quantitative evidence in this version. The original files remain available in the author’s preparation archive, and the supplement documents the inconsistencies relevant to its JSON.

The original TargetCidMatch mean and per-operation means use an undocumented macro aggregation and differ from pooled ratios. They must not be substituted for the micro values. The table denominators 189, 222, and 264 refer to configuration-dependent resolution records, not a fixed all-gold-edit set. No joint event intersections, language-conditioned predictions, rule-disabled outputs, session-cluster uncertainty calculations, or full benchmark/implementation files are provided. Arithmetic agreement with aggregates cannot establish experiment reproducibility or external validity.