LEARNING PRECISE SPATIOTEMPORAL RELATIONAL REASONING IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inventory tracing and historical audits require complete answers about which objects reached a destination, when they shared a location, and what incomplete observations establish. These queries combine changing spatial dependencies, temporal aggregation, and evidence limits; a plausible item or a correct final location alone may not complete the request. We study how Large Language Models can learn these joint reasoning requirements. Executable task construction supplies questions, answers, and derivations for both evaluation and supervision. To scale language supervision, we use a strong teacher's interactions on 9,000 seed instances to train two smaller question and reasoning polishers. Applied to newly generated histories, they yield 85,940 accepted training records under source and rendering constraints. We examine learning with smaller corpora and follow the expanded-corpus solver through supervised fine-tuning and reinforcement learning. Recorded Qwen3.5-0.8B validation accuracy starts at 0%, reaches 38% at the supervised-to-reinforcement-learning handoff, and reaches 45% for the selected RL model. Paired answers show corrected temporal and occupancy results alongside persistent evidence-state and value errors.

1 INTRODUCTION

An inventory assistant is asked which items reached a laboratory before handover. Two items travel in a cart; one is removed in an office before the cart enters the laboratory. The assistant must propagate movement through containment, stop that propagation after removal, and respect the handover time. A follow-up question asks how long the items shared a room. Their final locations no longer suffice: the answer depends on both histories. Figure 1 shows how one short event sequence gives rise to these different reasoning requirements.

These requirements arise in physical-asset traceability. The GS1 EPCIS standard represents timestamped observations and changes in object aggregation: cases, for example, travel with a pallet until disaggregation (GS1, 2022). Inventory tracing, equipment handover, and historical audits motivate the language-based setting studied here. A useful answer must combine evolving spatial dependencies with the requested temporal scope and remain within the available evidence. An ambiguous description may identify several items; an incomplete record may leave their movement undetermined. Both affect what the assistant can legitimately return.

For an exhaustive request, success means completing the whole query. Omitting an item leaves the handover inventory incomplete, including an extra item changes it, and using the wrong interval changes a co-location duration. We therefore study *complete query correctness*: the requested values, entity bindings, temporal scope, and evidence status must agree with the query contract. The comparison operates on canonical answer documents, with formatting measured separately.

Existing evaluations establish important ingredients of this problem. Entity-tracking and procedural-text tasks study changing object states (Kim & Schuster, 2023; Dalvi et al., 2018); spatial reasoning suites study relational composition, including questions with incomplete information (Mirzaee et al., 2021; Rizvi et al., 2024). Their established protocols emphasize different parts of this problem. Our study requires a common answer contract for dynamic containment and detachment, queries over historical intervals, and explicit alternatives or missing evidence within the same event-history setting. We construct controlled training and validation resources around these joint requirements to study how large language models can learn to satisfy them.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

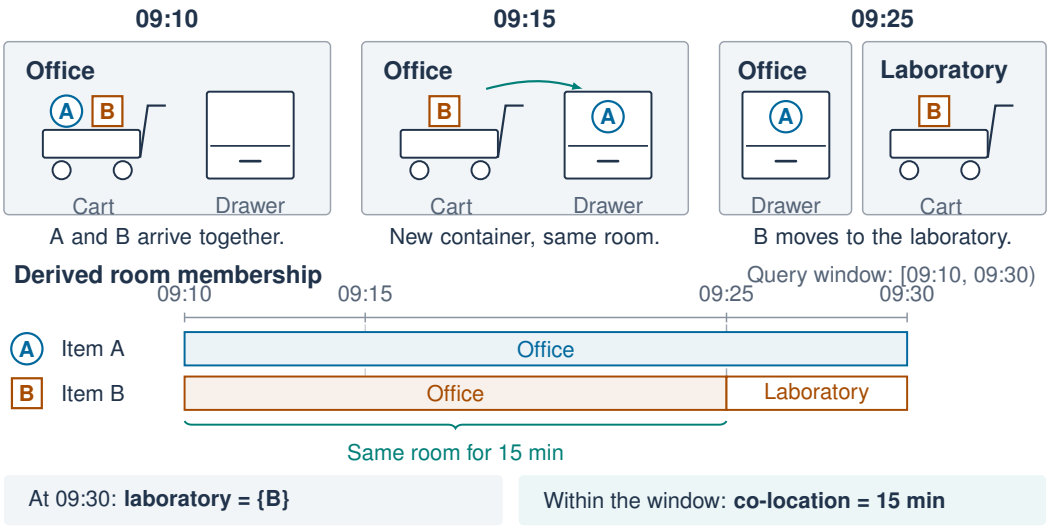


Figure 1: **A change of container need not change an item's room.** A leaves the cart at 09:15 but remains with B in the office until 09:25; only B inherits the cart's later move. Complete answers must combine containment, derived location, and the requested time window. If the 09:15 event instead removes an unidentified item, the laboratory query has insufficient evidence. In this constructed illustration, events take effect at the stated time and no unlisted events occur.

The constructed queries also reveal a learning need. The strongest recorded API configurations reach 59% complete-answer accuracy on the calibrated validation set, while separately evaluated open-weight models remain below 40% (Section 5.1). These measurements connect task construction to the question we study: how can supervision teach models to carry out the joint reasoning operations more reliably?

Scaling executable supervision requires preserving the query's meaning during language realization. A fluent explanation can silently change which objects count, whether a time boundary is inclusive, or whether an observation resolves the answer. We use a strong language model to express program-derived seed queries and reasoning, retaining its successful interactions and corrections. Supervised fine-tuning (SFT) transfers these tasks to two smaller polishers, which realize new questions and derivations under source constraints. This teacher-to-polisher step turns a finite seed collection into a reusable data-production capability. The resulting demonstrations train a solver; reinforcement learning (RL) then rewards complete query results.

The pipeline uses 9,000 seed instances and retains 85,940 records for solver SFT. Smaller-corpus Qwen3.5-4B runs achieve 13% and 26% validation accuracy with 1,200 and 9,000 teacher-generated examples. The expanded-corpus Qwen3.5-0.8B study follows the full learning sequence: an initial 0%, 38% at the SFT-to-RL handoff, and 45% for the selected RL model. The learning curve, protocol comparisons, and paired answers distinguish format acquisition from successful temporal, spatial, and evidence-sensitive queries.

The study connects three contributions: an executable construction that makes the joint query requirements measurable; a scalable supervision pipeline that learns smaller polishers from strong-teacher interactions; and an empirical account of solver learning across data scales, SFT, and RL. Each serves the same objective: teaching models to reconstruct dynamic relations and return the complete answer supported by the history.

2 Precise Queries over Dynamic Relational Histories

The scenarios share a history of entities, relations, and events, followed by a query specifying what to return and when. Three requirements define the learning problem.

2.1 FROM APPLICATION SCENARIOS TO QUERY REQUIREMENTS

Recover spatial dependencies as events change them. An item can sit inside a case on a cart, so moving the cart changes the item's room without naming it in the movement event. Removing the case or the item breaks a different link in this chain. Inventory tracing and item finding require reconstructing these dependencies; direct placement and inherited room membership must remain distinct.

Compute the requested history, including its boundaries. A handover query can ask for every item delivered before a deadline; an audit can ask when two items shared a room. These queries require temporal selection, ordering, or aggregation over reconstructed states. In Figure 1, removing an item changes containment immediately, but changes co-location only when the cart leaves. Repeated visits and interval boundaries can change an answer even when the final locations are correct.

Return precisely what the public evidence supports. An item description may match several entities, or an observation may omit the identity needed to resolve a movement. The response must retain the allowed referents and their corresponding values, or identify missing evidence. A complete item list, for example, has neither omitted nor extra items. Our controlled event histories make these requirements explicit and provide computable targets for training and validation.

2.2 PUBLIC EVIDENCE, STATE, AND QUERIES

Let W_0 be an initial relational world and let $(t_i, e_i)_{i=1}^m$ be timestamped events. A deterministic transition function produces $W_i = \mathrm{Apply}(W_{i-1}, e_i)$, from which spatial facts such as containment and room membership are derived. A query q selects a temporal scope and an operator over the resulting history. Its output can be an entity, a sequence, a count, a duration, or another typed value. The public input x contains a description and event observations; it does not necessarily expose every fact in the generator's world.

Let $B(x, q)$ denote the referent bindings permitted by the public question, and let $\Omega(x)$ be the set of worlds consistent with the observations and stated rules. For a fixed binding b, the candidate answer set is

$$\mathcal{V}(x, q, b) = \{\mathrm{Exec}(q, b, W) : W \in \Omega(x)\}. \tag{1}$$

This set-based definition specifies the evidence requirement for three answer statuses in our constructed cases. A *determinate* answer has a resolved query and an entailed value. An *ambiguous* question admits multiple referent bindings and requires their binding–value alternatives. An *insufficient* question has a resolved binding but lacks evidence needed to determine its value; its target identifies the missing evidence. These distinctions prevent the private simulator state from silently becoming evidence available to the model.

2.3 COMPLETE QUERY CORRECTNESS

Let $a^*(x, q)$ be the canonical, status-bearing answer document and $\mathrm{Parse}(\hat{y})$ the document extracted from a generated response. Our primary metric is

$$\mathrm{Correct}(\hat{y}; x, q) = \mathbf{1}[\mathrm{Canon}(\mathrm{Parse}(\hat{y})) = \mathrm{Canon}(a^*(x, q))]\,. \tag{2}$$

An invalid answer document is incorrect. Canonicalization ignores JSON key order and orders ambiguity alternatives by their binding; values retain the prescribed types and ordering. Determinate answers must return the exact requested value; ambiguous answers must preserve all required alternatives; insufficient answers must supply the prescribed missing-evidence labels. These labels operationalize missing information in our controlled task; real applications can admit other useful explanations of the same evidence gap.

The scorer can recover a single valid answer block from surrounding prose, so strict serialization compliance is reported separately. The metric evaluates the complete final answer. Rationale faithfulness and application-specific error costs are separate evaluation questions. We diagnose failures by separating formatting, evidence status, bindings, and values.

3 Constructing Tasks and Scaling Their Supervision

The application requirements determine both what to evaluate and what to teach. We first construct executable histories and queries, then train small language polishers from strong-teacher interactions to produce supervision at scale. Solver SFT and outcome RL learn from the resulting calculations and complete answers (Figure 2).

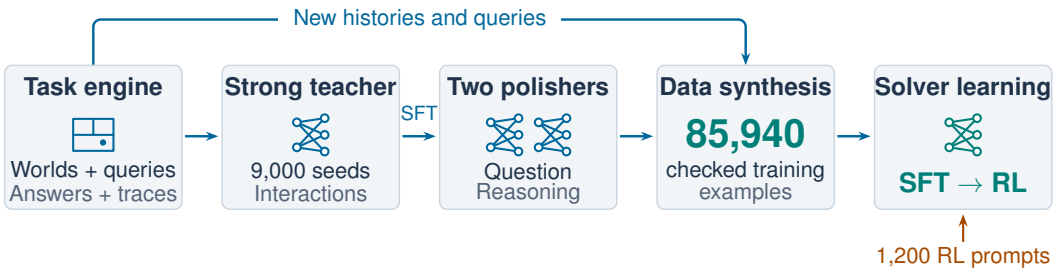


Figure 2: **From executable tasks to scalable supervision.** Strong-teacher trajectories on 9,000 seeds train question and reasoning polishers through SFT. Applying them to new program-generated histories produces 85,940 checked examples for solver SFT. GRPO follows using 1,200 prompts from a separate earlier training release.

3.1 Executable construction for evaluation and learning

We construct a scene, event history, public observations, and a query in an executable intermediate language. Procedural residential layouts from ProcTHOR (Deitke et al., 2022) provide scenes. The training generators cover 90 query algorithms combining object identities, spatial relations, events, and time intervals. Execution supplies a target, trace, and proof certificate for the specified calculation. The same answer contract supports validation questions and generated training instances, making the reasoning requirements measurable as well as teachable.

Targets also depend on what the public input establishes. They distinguish a determinate value, binding–value alternatives under ambiguous referents, and insufficient evidence with the required missing-evidence identifiers (Section 2). A location known only to the simulator cannot justify a determinate answer. Conversely, an incomplete observation need not prevent answering a query whose result it cannot change.

Running example. One recorded seed query selects Chair 1 and counts only three named objects whose immediate parent is that chair. It asks for the first return to the count immediately before 13:43, after a deviation and within an inclusive window ending at 17:58. The count changes $1 \to 0$ at 14:09 and returns to 1 at 14:11. A partially observed event at 14:15 cannot change this earlier first return, so the answer is determinate. The closed object population, direct-parent relation, and deviation-before-return condition are essential parts of the supervision. Appendix A.9 gives the recorded input and derivation.

3.2 Learning small polishers from a strong teacher

The teacher's role is to express the program's calculation in language. For 9,000 seed instances, we retain its question and reasoning interactions, including successful corrections following validation feedback. The question and rationale have different information requirements. The question polisher receives a public semantic plan, a draft, and protected rendering requirements. The reasoning polisher additionally receives the derivation and execution records. Both produce language variants; the surrounding program supplies the source identities and answer document. For the running example, the question must retain the direct-parent scope and temporal condition, while the rationale explains the count changes and why the later missing observation is irrelevant to the first return.

We project the accepted teacher outputs into two SFT tasks: question realization and reasoning realization. Failed outputs are not training targets; successful corrections also teach how to respond

to feedback. Two separate Qwen3.5-4B models learn these tasks with full-parameter SFT. The seeds cover 25 layouts and 90 algorithms, with four instances per layout–algorithm pair. Twenty-two layouts train the polishers and three supply held-out diagnostics (Appendix A.5). The trained polishers subsequently replace teacher calls in language expansion, while the program continues to supply the calculations.

3.3 Expanding supervision to new histories and queries

Expansion applies these models to newly generated histories and queries. Acceptance first checks typed renderings and protected content, with bounded correction attempts. It then requires question and rationale to share a candidate identity and source version, preventing joins across different calculations. Finally, the program attaches its answer and checks the assembled record with the answer parser and scorer, alongside rationale-presence and length checks. Unresolved failures are excluded; **85,940 question–rationale–answer records** pass these checks.

Acceptance validates source identity, protected renderings, and the attached answer document. Semantic equivalence of free-form wording remains outside these checks (Section 7).

3.4 Demonstrating state and time calculations

Let x be the public scene, observations, and question, z the generated rationale, and a the structured answer. Standard autoregressive SFT on assistant tokens in (x, z, a) demonstrates how the result follows from the query conditions. In the example, the target explanation connects the count changes to the first qualifying return and its evidence status. Private proofs support construction of this supervision; they are absent from solver inputs. The polishers are also absent at inference time.

The solver is Qwen3.5-0.8B (Qwen Team, 2026), trained with full-parameter SFT and the model's native response format. The run completes three corpus passes and continues into a fourth; optimizer step 19,000 initializes RL. Appendix A.6 records the standard objectives and training settings.

3.5 Rewarding complete query results

The outcome objective checks the result of the same query contract. We use group relative policy optimization (GRPO; Shao et al., 2024) with eight sampled responses per public prompt and the binary reward in Equation 2. Standard within-group advantages enter the clipped objective; identical rewards give no relative preference signal. There is no KL reward, KL loss, or intermediate-step reward.

For the example, both the evidence status and returned time must be correct. A complete inventory similarly requires every requested item and no extras; ambiguous answers require all binding–value alternatives. Correct fragments earn no partial credit, and the rationale prose is not scored. SFT supplies demonstrations of the calculation, while the outcome reward aligns optimization with completing the request.

The RL trajectory ends at optimizer update 75, with eight prompts per update. Its 1,200 public prompts come from an earlier 1,800-instance training release. The loader removes privileged-information fields. Section 4 describes the sources of each learning and validation resource.

4 Training and Validation Setup

From scenarios to learning and validation instances. We instantiate the requirements in Section 2 as textual indoor event histories with executable queries. Some generated resources supply learning examples; others supply validation questions under the same complete-answer contract. These resources have distinct construction histories. The 9,000 seed instances train the language polishers, and the expanded 85,940 records supervise the solver. RL uses 1,200 public training prompts from an earlier 1,800-instance release. Two smaller-corpus studies use Qwen3.5-4B with 1,200 or 9,000 teacher-generated examples, one SFT epoch, global batch eight, and learning rate 2×10^{-6}. Their pre-continuation evaluations measure the SFT models (Appendix A.3).

Validation queries. The fixed validation set contains 100 questions, selected from a 600-question release, covering 90 generator algorithms and five families: multi-placement, occupancy, temporal matching, trajectory, and relation history. These families respectively exercise placement dependencies, location membership, event selection, movement histories, and changing relations. Their answers may require entities, sequences, counts, or durations. The answer-status composition is 50% determinate, 33% ambiguous, and 17% insufficient. Together, query operators and evidence states implement the scenario requirements, including complete sets and explicit limits on what the observed history determines.

Selection and comparison protocol. The validation subset was calibrated using responses from 22 reference-model configurations. Table 1 projects their recorded scores onto this subset and includes four public-input open-weight evaluations. We also use this validation set to select the solver checkpoint, evaluating before RL and every 15 updates. The final selected model is update 75, the endpoint of the reported RL trajectory, with 45% accuracy. The SFT history and RL continuation use separate evaluators, with the same nominal thinking-enabled sampling: one answer per question, temperature 0.4, top-p 0.95, top-k 20, seed 42, and prompt/output limits of 12,288/16,384 tokens. The 38% handoff and all RL points share the RL evaluator. SFT metrics are recorded every 500 updates and at epoch boundaries; Appendix A.2 details the evaluation records at the stage transition.

Measures. Complete-answer accuracy is the primary measure; failed and unparseable generations count as incorrect. Parse validity and strict-format compliance separately indicate whether a model can express an admissible answer document. Appendix A specifies the training and validation protocols. Appendix B provides the reference-model breakdowns by query family and answer status, and Appendices A.2 and A.1 retain the recorded SFT and RL histories.

5 Learning from Constructed Supervision

The experiments connect measurement to learning: how well do existing models complete the constructed queries, what do smaller supervision collections teach, and how does expanded supervision improve a solver through SFT and RL?

5.1 How well do existing models complete the queries?

Table 1 reports all 22 API model–effort configurations and four public-input open-weight conditions. The highest recorded API accuracy is 59%. In particular, `gpt-5.6-sol-fast` and both Grok configurations reach 59% with no reported model failures.

With a 32,768-token output budget, Qwen3.5-4B and 9B score 27% and 38%, with parse validity of 98% and 99% and no output-limit truncation. Their ambiguous-query accuracy is only 3.0% and 6.1%: producing a valid answer document still leaves substantial difficulty in returning every binding with its correct value. Appendix B.3 gives the family and evidence-status breakdowns. These results establish the learning target for the following experiments: completing the joint spatial, temporal, and evidence requirements. Programs supply the answer calculations for training, while teacher and polisher models supply their language realization.

5.2 Learning with smaller teacher-generated corpora

Table 2 compares Qwen3.5-4B trained on 1,200 and 9,000 examples under the same one-epoch SFT recipe. Complete-answer accuracy is 13% and 26%, respectively, while parse validity is already 97% and 99%. Producing admissible answers is therefore insufficient to solve these queries. The larger corpus yields more correct complete results, motivating supervision expansion beyond the initial teacher-generated collection. The two corpora differ in generated content and training updates; Appendix A.3 gives their provenance and settings.

5.3 From expanded-corpus SFT to complete-answer RL

Figure 3 follows Qwen3.5-0.8B trained on the 85,940 accepted records and then continued with RL. Before SFT, its recorded accuracy and parse validity are both 0%. Accuracy reaches 15% after 500

Table 1: **Existing models on complete event-history queries.** All 22 API configurations and four open-weight conditions are shown. API scores come from subset calibration; open-weight scores come from public-input evaluations (Section 4). Scores are percentages; failures are the evaluator's model failures, distinct from parser failures. K denotes 1,024 tokens.

Model	Effort / output cap	Acc. (%)	Failures (%)
A. API models			
gpt-5.6-sol	none	32	0
gpt-5.6-sol	low	57	0
gpt-5.6-sol	medium	58	0
gpt-5.6-sol	high	59	7
gpt-5.6-sol	xhigh	58	15
gpt-5.6-sol	max	47	49
gpt-5.6-sol-fast	medium	59	0
gpt-5.6-luna	medium	44	5
gpt-5.6-terra	medium	50	0
gpt-5.5	medium	53	21
gpt-5.4	medium	49	15
gpt-5.4-mini	medium	37	26
gpt-5.3-codex	medium	52	6
gpt-5-mini	medium	27	57
grok-4.6	medium	59	0
grok-4.5	medium	59	0
gemini-3.7-flash	medium	53	17
gemini-3.6-flash	medium	14	82
gemini-3.5-flash	medium	16	81
gemini-3.1-pro-preview	medium	52	29
mai-code-1.1-flash	medium	15	79
mai-code-1-flash-picker	medium	5	94
B. Open-weight models			
Qwen3.5-4B	Thinking, 4K	0	100
Qwen3.5-4B	Thinking, 32K	27	0
Qwen3.5-9B	Thinking, 4K	0	100
Qwen3.5-9B	Thinking, 32K	38	0

Table 2: **SFT with smaller teacher-generated corpora.** Both Qwen3.5-4B runs use one epoch of full-parameter SFT and the same nominal training and decoding recipe. Accuracy and parse validity are percentages. Corpus composition and optimizer updates change with data size.

Model	Examples	Updates	Acc. (%)	Parse (%)
Qwen3.5-4B	1,200	150	13	97
Qwen3.5-4B	9,000	1,125	26	99

SFT updates, 28% after the first corpus pass, and 38% after the second. Later checkpoints fluctuate, as the complete curve shows. Checkpoint 19,000 scores **38%** when evaluated at RL initialization; complete-answer optimization subsequently reaches **45%** for the selected model.

The early learning change includes acquiring the answer format. In a separate matched nonthinking comparison, the original and SFT models score 2% and 28%, and parse validity rises from 35% to 98%; 26% of validation questions become correct and none become incorrect. At the thinking-enabled SFT-to-RL handoff, parsing and strict-format compliance have reached 100% and remain there at the selected RL checkpoint. Its further improvement therefore concerns scored answer content. Table 5 and Appendix B.1 specify the stage comparisons.

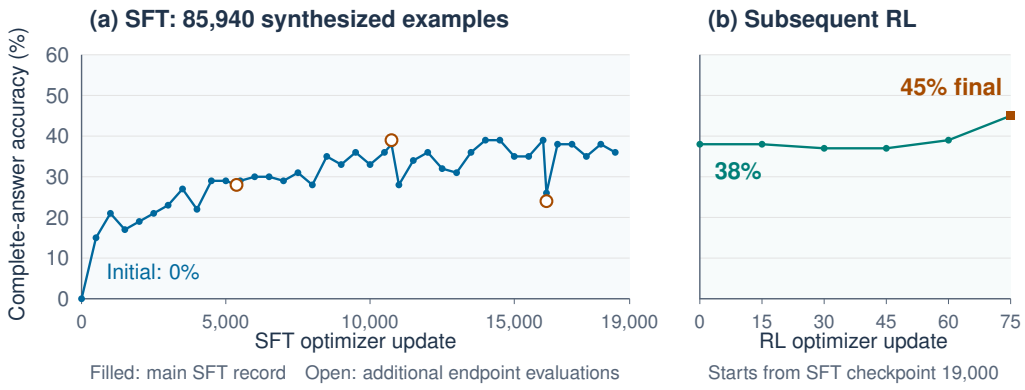


Figure 3: **Learning complete event-history queries through SFT and RL.** Left: Qwen3.5-0.8B learns from 85,940 synthesized records; filled points follow the recorded SFT sequence and open points retain additional epoch-boundary evaluations. Right: checkpoint 19,000 is evaluated at 38% before RL; the curve ends at update 75 with 45%. Each panel uses its stage's evaluator and optimizer-update axis (Section 4).

5.4 Which spatial, temporal, and evidence requirements change?

We qualitatively inspect twelve fixed pairs from the matched nonthinking comparison, selected by frozen item-ID order before reading their outputs (Appendix B.2). Two newly correct answers concern time-constrained event matching and a compressed direct-occupancy history. The former changes a malformed answer containing -3 to the correct count difference -1; the latter replaces repetitive, unparseable output with the correct sequence $[0, 1, 0, 1]$. SFT corrects both the output format and the requested answer in these cases.

Other pairs expose distinct remaining requirements. For a minimum-lag query, SFT returns the correct ambiguous status and both route bindings, but assigns each value 1 instead of null and 111. In another pair, both models assert a determinate count although the target is insufficient; the SFT explanation claims the missing observation is irrelevant. Together, these cases distinguish three aspects of query completion: admissible formatting, correct evidence status, and correct values for every required binding.

6 Related Work

From state tracking to historical queries. ProPara studies existence and location changes in procedural text (Dalvi et al., 2018); Entity Tracking constructs object-and-box histories to evaluate and teach state tracking (Kim & Schuster, 2023). The latter already requires all objects in a queried box and no extras, and studies operations whose referents depend on the current state. Our target extends this line to queries over changing nested relations and historical time ranges, with explicit treatment of unresolved bindings and missing evidence.

Spatial supervision and the evaluation requirement. SpartQA includes compositional spatial questions, multiple valid answers and insufficient-information cases (Mirzaee et al., 2021). SpaRTUN expands relation and expression coverage through scene graphs, logical reasoning and language generation (Mirzaee & Kordjamshidi, 2022). SpaRC and SpaRP construct symbolic spatial paths and verbalize them for model training (Rizvi et al., 2024). These resources establish important parts of our setting. Inventory and historical-audit requests jointly require changing containment, historical selection, and distinctions between identifiable and unsupported answers. This joint contract guides our training and validation data.

Executable data and verifiable learning. Reasoning Gym supplies procedural generators, configurable difficulty and verifiers for reinforcement learning (Stojanovski et al., 2025). SynLogic similarly scales verifiable tasks, including object exchanges, hierarchical spatial relations and time

constraints (Liu et al., 2025). Enigmata combines verifiable puzzles, correct reasoning traces for supervised fine-tuning, and subsequent reinforcement learning (Chen et al., 2025). Our learning stages build on this established pattern: program execution fixes answers and derivations, polishers expand their linguistic realization, and a solver learns from traces before GRPO (Shao et al., 2024) optimizes complete answers.

Complete answers within the evidence boundary. ProofWriter pairs natural-language rules with generated proofs and an open-world Unknown state (Tafjord et al., 2021). IntentRL trains models to return multiple interpretations and answers for ambiguous requests (Saparina & Lapata, 2026). Its full-coverage metric measures whether all gold answers occur and reports precision separately. Our complete-answer criterion also excludes extra answers and checks their bindings and prescribed temporal values. These precedents motivate separating answer coverage, unsupported content and proof quality.

7 Discussion and Limitations

The scenarios identify a useful learning target: recover what an event history supports and return everything the query requests. Our controlled indoor histories isolate these requirements. Operational inventory systems also face sensor errors, entity reconciliation, concurrent processes and system integration; these factors require separate validation before deployment.

The solver results come from one training run with checkpoint selection on the same validation questions. Independent evaluation and repeated runs would measure stability under selection and training variability. Generalization to unseen environments and query algorithms requires separately established splits.

Language fidelity remains a source of uncertainty in the supervision. A stratified inspection of 30 seed-corpus examples found selector-scope and temporal-boundary wording risks. This inspection and the twelve-pair solver analysis provide case-level evidence; corpus-wide error rates and query-family gains require broader, appropriately designed samples. The smaller-corpus comparison holds the 4B model and nominal recipe fixed, but changes source content and updates. The expanded-corpus run also changes model size and training duration. Separating data volume, language expansion, trace supervision, and reward effects requires controlled comparisons.

8 Conclusion

Dynamic item tracing and historical audits require joint spatial, temporal, and evidence-sensitive reasoning. Executable task construction makes these requirements evaluable and supplies their supervision. Learning smaller polishers from strong-teacher interactions enables expansion to 85,940 records. The expanded-corpus solver starts at 0% recorded accuracy, learns through SFT to reach 38% at the RL handoff, and reaches 45% for the selected RL model. Paired answers connect learning to corrected temporal and occupancy results while locating remaining binding-value and evidence errors. The study thus links task construction, scalable supervision, and solver learning around complete event-history queries.

AI USE STATEMENT

Generative language models were used to produce synthetic questions and reasoning traces. As described in Section 3, question and reasoning polishers realize program-derived supervision; their generated language is distinct from the program-assigned answer targets. An AI coding and writing assistant also supported task formulation, literature retrieval, analysis and interpretation of recorded experimental results, manuscript drafting, and preparation of LaTeX and figure code. The writing and figure workflow used procedural agent skills (Kassis et al., 2026). The checks applied to generated supervision and the limitations of the existing language-quality review are described in Sections 3.2 and 7. Responsibility for the final scientific claims and artifacts rests with the authors.

REPRODUCIBILITY STATEMENT

Section 2 defines the query semantics and complete-answer criterion. Sections 3 and 4 describe data construction, distinct training resources, and validation selection and decoding. Appendix A.6 records the training configuration; Appendix B.1 specifies the matched SFT and RL comparisons. Appendix B gives the reference-model breakdowns for the query families and evidence states. Appendix A.1 supplies the six RL validation results through the final model at update 75. Section 7 states the limits of the available evidence and component comparisons.

REFERENCES

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Jiaze Chen, Xuefeng Li, Qiying Yu, Hao Zhou, and Mingxuan Wang. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=fmnxunacr4.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1595–1604, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1144. URL https://aclanthology.org/N18-1144/.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022. URL https://procthor.allenai.org/.

GS1. *EPCIS Standard*. GS1 AISBL, release 2.0 edition, June 2022. URL https://ref.gs1.org/standards/epcis/. Ratified June 2022.

Timothy Kassis, Vinayak Agarwal, Yuhuan He, Darshil Patel, and Aubrey M. Brueckner. Scientific Agent Skills: A Library of Procedural Knowledge for Research Agents, 2026. URL https://arxiv.org/abs/2609.00065.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.213. URL https://aclanthology.org/2023.acl-long.213/.

Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, Mozhi Zhang, Pengyu Zhao, and Junxian He. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=XtNiw8OQsy.

Roshanak Mirzaee and Parisa Kordjamshidi. Transfer learning with synthetic corpora for spatial role labeling and reasoning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6148–6165, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.413. URL https://aclanthology.org/2022.emnlp-main.413/.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. SPARTQA: A textual question answering benchmark for spatial reasoning. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4582–4598, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.364. URL https://aclanthology.org/2021.naacl-main.364/.

Qwen Team. Qwen3.5 open model series. Official model repository, 2026. URL https://github.com/QwenLM/Qwen3.5. Qwen3.5-0.8B, 2B, 4B, and 9B release announced March 2, 2026; accessed September 23, 2026.

Md Imbesat Rizvi, Xiaodan Zhu, and Iryna Gurevych. SpaRC and SpaRP: Spatial reasoning characterization and path generation for understanding spatial reasoning capability of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4750–4767, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.261. URL https://aclanthology.org/2024.acl-long.261/.

Irina Saparina and Mirella Lapata. Reasoning about intent for ambiguous requests. In *Conference on Language Modeling*, 2026. URL https://colm.cc/virtual/2026/poster/2289.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=GqYSunGmp7.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

A Learning and Validation Details

A.1 RL endpoint and learning history

The RL trajectory ends at update 75 with 45% validation accuracy, compared with 38% at SFT initialization. Figure 4 and Table 3 give the six evaluations from initialization through this endpoint, at intervals of 15 updates.

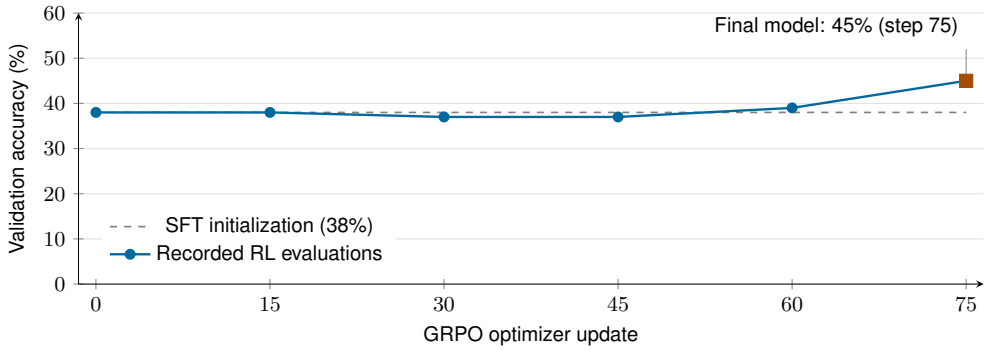


Figure 4: Validation learning curve through the final RL model. Six evaluations from one GRPO run, with one response per validation question. Update 0 is the SFT initialization at 38%; the final model reaches 45% at update 75. Lines connect the observed values.

Table 3: **Complete-answer validation accuracy throughout RL.** Update 0 is SFT checkpoint 19,000. All metrics are percentages on the same validation set. The trajectory ends at update 75 with 45% accuracy. All rows share the same evaluation harness and decoding settings.

RL update	Accuracy (%)	Parse-valid (%)	Strict format (%)
0	38	100	100
15	38	98	98
30	37	100	100
45	37	100	100
60	39	100	100
75 (final)	**45**	100	100

A.2 SFT history and the transition to RL

Table 4 gives the 41 completed SFT evaluations in the main curve. Through update 16,116, records are selected by the native training completion manifest; the five subsequent points use the unique completed evaluations. Three additional completed epoch-boundary evaluations are retained as open markers: 28% at update 5,372, 39% at 10,744, and 24% at 16,116. The manifest-selected values at those updates are 28%, 38%, and 26%. The original baseline generation is reused wherever the initial model appears.

The SFT evaluation at update 19,000 was interrupted and is omitted from the plotted scores. The same checkpoint was fully evaluated at RL initialization, giving 38%. This reevaluation starts the RL panel in Figure 3; each horizontal axis counts optimizer updates within its own stage.

A.3 Smaller-corpus SFT protocols

The two Qwen3.5-4B runs train on 1,200 and 9,000 teacher-generated records. Both use full-parameter native-thinking SFT for one epoch, global batch eight, learning rate 2×10^{-6}, and a 16,384-token training context. They complete 150 and 1,125 updates. The smaller corpus comes from the earlier training release; the larger corpus adds newly generated layouts and instances. Thus the comparison changes coverage and optimization exposure as well as example count.

Table 4: **Recorded SFT learning history.** Read down each block, then move right. Accuracy and parse validity are percentages. Additional endpoint evaluations are listed in the accompanying text.

Update	Acc.	Parse	Update	Acc.	Parse	Update	Acc.	Parse
0	0	0	6,500	30	99	13,000	31	96
500	15	86	7,000	29	98	13,500	36	99
1,000	21	97	7,500	31	99	14,000	39	100
1,500	17	94	8,000	28	99	14,500	39	99
2,000	19	96	8,500	35	100	15,000	35	100
2,500	21	96	9,000	33	96	15,500	35	97
3,000	23	98	9,500	36	100	16,000	39	98
3,500	27	98	10,000	33	100	16,116	26	99
4,000	22	99	10,500	36	100	16,500	38	99
4,500	29	100	10,744	38	97	17,000	38	100
5,000	29	99	11,000	28	97	17,500	35	99
5,372	28	99	11,500	34	99	18,000	38	100
5,500	29	99	12,000	36	100	18,500	36	100
6,000	30	96	12,500	32	98	—	—	—

Their scores are the recorded update-zero evaluations of the subsequent training jobs, before any continuation update: 13% and 26% accuracy, 97% and 99% parse validity, and 99% strict-format compliance for both. Configuration records bind these evaluations to the corresponding SFT models. Both use the same nominal validation files, one sampled answer, temperature 0.4, top-p 0.95, top-k 20, seed 42, and prompt/output limits of 12,288/16,384 tokens.

A.4 Data provenance and split scope

The 22-layout/3-layout split evaluates the auxiliary polishers. The solver uses 85,940 expanded supervised examples and an RL pool of 1,200 prompts from an earlier 1,800-instance training release. These resources retain their separate construction histories. The validation set was selected from a 600-instance release using model responses and is reused for RL checkpoint selection (Section 4). World, timeline, and query-algorithm overlap between solver training and validation remains unestablished; evaluating generalization along these dimensions requires explicit overlap accounting and corresponding held-out splits.

A.5 Polisher supervision counts

The 9,000 unique seed instances produce two different interaction datasets, including successful correction turns. For the 22 training layouts and three held-out layouts, the question polisher has 8,821 training and 1,213 held-out rows; the reasoning polisher has 8,137 and 1,102 rows. These counts refer to auxiliary-polisher supervision and diagnostics. The expanded solver corpus is accepted through the checks in Section 3.3.

The seed-generation record names `gpt-5.6-sol` at high reasoning effort as the teacher service. Its saved interactions include draft inputs, accepted question or reasoning variants, and validation feedback when a correction was needed. SFT targets are accepted variants: the first input is paired with the accepted output, and a successful correction supplies an additional feedback-conditioned interaction. This transfers language realization from the teacher to the polishers; executable programs remain responsible for the answer and derivation.

Native production completion receipts record 1,102 question-polisher updates and 1,016 reasoning-polisher updates. Each trains for two epochs with global batch 16; five and nine trailing rows per epoch, respectively, are dropped. The input sequences are not truncated. The question and reasoning training sequence limits are 4,096 and 31,744 tokens. These production records distinguish the learned data generators from the separately trained 0.8B solver.

A.6 TRAINING CONFIGURATION

For assistant-token supervision and groups of $G = 8$ sampled responses, the standard SFT loss and GRPO advantages are

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(x,z,a)\sim\mathcal{D}}\left[\log \pi_\theta(z, a \mid x)\right], \tag{3}$$

$$r_i = R(y_i, a) \in \{0, 1\}, \qquad A_i = \frac{r_i - \bar{r}}{s_r + \epsilon}, \tag{4}$$

where $\bar{r}$ and s_r are the within-group reward mean and standard deviation. Section 3 supplies the demonstrations and targets.

The question and reasoning polishers are separate Qwen3.5-4B models, each trained with full-parameter SFT for two epochs, global batch size 16, and learning rate 10^{-5}. The Qwen3.5-0.8B solver uses full-parameter SFT on the 85,940 retained examples, global batch size 16, and learning rate 2×10^{-5}. It completes three full passes and part of a fourth; the checkpoint at update 19,000 initializes RL.

The RL trajectory uses GRPO through update 75, with eight prompts and eight responses per prompt in each update, learning rate 10^{-6}, and no KL reward penalty or actor KL loss. Training sampling uses temperature 1.0, top-p 0.95, and top-k 20. This completed run used two GPUs. The final model is the update-75 checkpoint. Evaluation settings and the separate training-stage comparison protocols appear in Appendix B.1.

A.7 COMPLETE ANSWERS FROM OBSERVED EVENTS

The task requires answers to follow from the public event history, rather than from unobserved facts in the simulated world. The answer schema makes this evidence requirement explicit. A determinate answer contains `status` and `value`; an ambiguous answer additionally contains binding–value `alternatives` and uses a null top-level value; an insufficient answer supplies `missing_evidence` with a null value. The scorer validates these fields before canonical comparison. A missing object, wrong time, omitted binding, or unjustified determinate answer makes the complete answer incorrect.

A.8 ILLUSTRATIVE EXAMPLE CHECKS

Figure 1 is written for exposition. Under its stated complete-log convention, A stays in the office after 09:15 and B travels with the cart to the laboratory at 09:25. Within the half-open query window [09:10, 09:30), their room histories agree on [09:10, 09:25), giving 15 minutes. Replacing the identity of the removed item with two possible identities yields two compatible histories with different laboratory contents.

A.9 A RECORDED SEED INSTANCE: FROM PUBLIC HISTORY TO SUPERVISION

This example illustrates supervision construction using an instance from the seed training corpus. The condensed context and query preserve object names, times, and query conditions.

Public context and request. The original layout places Chair 1 in Bedroom 1 and Dining Table 3 in Living Room 1. The query selects every target among these two whose original room is Bedroom 1. It counts only Cell Phone 2, Pillow 3, and Spray Bottle 1 whose immediate parent is the selected target; nested descendants do not count. Freeze that count immediately before applying any move at 13:43, and process moves through 17:58, including both endpoints. The request is the first logged time when the count returns to its frozen value *after first differing from it*, or null if there is no such return by the end.

The relevant observed placements before the boundary leave Cell Phone 2 at Arm Chair 2, Pillow 3 at Dining Table 3, and Spray Bottle 1 at Chair 1. At 14:09, Spray Bottle 1 moves to Chair 7; at 14:11, it returns to Chair 1. A later event at 14:15 is only partially observed: either Cell Phone 2 moves to Counter Top 1 or Spray Bottle 1 moves to Chair 1. These are the two alternatives stated in the public record.

Program calculation and evidence check. The selector yields Chair 1. The execution trace applies event lookup, placement-state reconstruction, direct-occupancy counting, and a first-matching-time operation. The proof records the following derived states; the boundary count is calculated from the history, not supplied as an extra solver-input field:

Time	Direct count at Chair 1	Role in the query
Immediately before 13:43	1	Frozen boundary value
14:09	0	First deviation
14:11	1	First return after deviation

Both permitted completions of the 14:15 event retain the first return at 14:11. The target is consequently determinate despite the incomplete observation. The program-assigned answer document is {"status":"determinate","value":"14:11"}.

Language supervision and protected meaning. The recorded draft says, "The count must first differ; boundary equality does not qualify." Its polished counterpart makes the ordering explicit: "Before a return can qualify, the count must first differ from its frozen value; equality before that deviation does not qualify." Both preserve the condition that prevents the boundary itself from answering the query. The object population, direct-parent restriction, and inclusive window likewise remain part of the request.

The recorded polished rationale states, "There is no intervening count change. At 14:11, the move of 'Spray Bottle 1' to 'Chair 1' restores direct count 1. Therefore the first post-excursion return time is 14:11." The solver is supervised on this explanation and the answer, while its input contains only the public scene, observations, and question. The program's private proof is used during target construction. Complete-answer scoring checks both the evidence status and the returned time.

B Comparisons for Complete Event-History Queries

B.1 Decoding and training-stage comparisons

Table 5 separates two decoding conditions for the Qwen3.5-0.8B solver. In the thinking-enabled continuation, the 19,000-update supervised checkpoint scores 38%, and its selected RL continuation scores 45%. Both have 100% parse validity and strict-format compliance. The evaluation settings are temperature 0.4, top-p 0.95, top-k 20, seed 42, one response per question, and prompt/output limits of 12,288/16,384 tokens. The original thinking model's 0% row comes from a separate SFT evaluation harness and has 0% parse validity, with the same nominal sampling settings. That harness supplies the SFT learning curve; the RL harness reevaluates the SFT checkpoint at 38% and measures the continuation to 45%.

The nonthinking pair uses the same public inputs and matched decoding, with thinking disabled and otherwise the sampling values and token limits above. Under this condition, the original model scores 2% and its supervised checkpoint scores 28%; parse validity changes from 35% to 98%, and output truncation from 12% to 0%. This pair measures SFT adaptation under nonthinking decoding. The RL comparison uses thinking-enabled decoding.

The existing paired-result record further resolves the SFT change: 26% of questions switch from incorrect to correct, 0% switch from correct to incorrect, 2% remain correct, and 72% remain incorrect. All percentages use the full validation set as denominator. Thus the net gain comprises additional solved requests while preserving the initial successes, alongside substantial changes in parsing and truncation.

Table 5: **Validation performance across training stages.** Accuracy measures correct complete answers; Parse and Trunc. are parse-valid and output-truncation rates. All metrics are percentages; — denotes an unreported metric. The learning study includes original, supervised, and RL models. †The original thinking baseline uses the SFT evaluation harness; the thinking-enabled SFT and RL rows use the RL harness.

Model	Training stage	Acc. (%)	Parse (%)	Trunc. (%)
A. Native thinking evaluations				
Qwen3.5-0.8B	Original†	0	0	—
Qwen3.5-0.8B	SFT (19,000 updates)	38	100	—
Qwen3.5-0.8B	SFT + RL (selected)	**45**	100	—
B. Matched native nonthinking evaluations				
Qwen3.5-0.8B	Original	2	35	12
Qwen3.5-0.8B	SFT (19,000 updates)	28	98	0

B.2 Paired solver behavior on fixed examples

To examine the nonthinking SFT comparison at the level of individual requests, we select positions 2–13 in lexicographic order of the frozen event-history item IDs. Position 1 was inspected to learn the output schema and is excluded. The twelve IDs were fixed before their predictions were read. Both models' public-prompt and input-token hashes agree for every selected pair. We use the recorded scores and targets. One long baseline response is represented by marked beginning/end excerpts and its recorded parsing outcome.

The sample includes all five query families: three multi-placement, four occupancy, two temporal matching, one trajectory, and two relation history queries. Five targets are determinate, four ambiguous, and three insufficient. Table 6 describes output properties in these twelve pairs, which form the denominator for every sample percentage.

Complete correctness changes from failure to success on two examples, remains successful on one, and remains unsuccessful on nine. Table 7 shows six cases spanning corrected spatial/temporal answers and persistent state/value errors. For a temporal matching query, the correct -1 is the change in pair count after deleting a selected event. For a direct-occupancy query, $[0, 1, 0, 1]$ is the requested compressed history. Both become correct after SFT, alongside corrections to invalid baseline outputs.

Table 6: **Output properties in the fixed twelve-pair SFT sample.** Every percentage describes the same twelve examples. Status correctness requires a parsed answer with the target status.

Measure	Original (%)	SFT (%)
Parse validity	33.3	100.0
Evidence-status accuracy	16.7	50.0
Complete-answer accuracy	8.3	25.0

Both models also correctly count distinct placement-region labels in the retained success; the SFT response adds an explanation.

Table 7: **Paired solver answers for six cases from the fixed sample.** Text is condensed; numbers and requested values are preserved. "Malformed" means the original answer was unparseable, even where it contained a visible number.

Query requirement	Original output	SFT output	Required answer
A. Corrected answers			
Temporal match-count change	Malformed; contains -3	-1, determinate	-1, determinate
Direct-occupancy history	Unparseable repetition	$[0, 1, 0, 1]$, determinate	$[0, 1, 0, 1]$, determinate
B. Persistent errors			
Evidence for distinct occupancy counts	2, determinate	2, determinate	Insufficient; missing observation
Route-specific minimum lag	Malformed; contains 30	A: 1; B: 1; ambiguous	A: null; B: 111; ambiguous
Distinct direct-transition movers	2, determinate	1, determinate	0, determinate
Directed region-edge revisits	Malformed; contains 13	1, determinate	5, determinate

The failed cases distinguish several parts of the answer contract. All three insufficient-target examples still receive determinate SFT answers. In the distinct-occupancy-count case, the SFT rationale states that the hidden observations leave the result unchanged, contradicting the target evidence status. The minimum-lag example now names both routes and predicts ambiguity, yet both values are wrong. Direct-transition and region-edge counts have the right determinate status but wrong values.

Across these pairs, valid formatting, correct evidence status, and complete bindings can coexist with wrong values. Two responses become fully correct, while the failed cases locate errors in different parts of the answer contract. Examining these parts separately explains more of the observed behavior than the complete-query score alone.

B.3 Reference-model spatial and evidence difficulties

Table 8 breaks down complete-answer correctness for the public-input Qwen3.5-4B and 9B reference models. Both use a 32,768-token output budget. Their parse-valid rates are 98% and 99%, with no reported output-limit truncations. Completing the requested content remains difficult.

Multi-placement questions require resolving interacting placement relations. Their complete-answer accuracy is 4.8% for 4B and 19.0% for 9B. Questions with ambiguous referents require returning every supported binding–value alternative, reaching 3.0% and 6.1% respectively. These results identify low complete-answer accuracy on multi-placement and ambiguous queries for both reference models. Family and evidence status are overlapping descriptions of the same questions. The two breakdowns characterize the reference models along these complementary dimensions.

Table 8: **Complete-answer correctness by family and evidence status.** Accuracy is reported as a percentage within each row's subgroup of size n. Both models use public inputs and 32,768-token output budgets. Family and status panels partition the same validation set.

Partition	n	Qwen3.5-4B (%)	Qwen3.5-9B (%)
A. Query family			
Multi-placement	21	4.8	19.0
Occupancy	23	21.7	34.8
Temporal matching	26	50.0	53.8
Trajectory	15	26.7	40.0
Relation history	15	26.7	40.0
B. Gold evidence status			
Determinate	50	34.0	44.0
Ambiguous	33	3.0	6.1
Insufficient	17	52.9	82.4