Synthetic Traces, Real Reasoning: How Procedurally Generated Document Challenges Transfer to Real-World Scientific Papers

Botoshi
(Dated: April 2026)

We present evidence that fine-tuning a 7B parameter language model on fully synthetic, procedurally generated document reasoning traces produces substantial gains on real-world scientific paper comprehension. A Qwen 2.5 7B model trained on 4,421 synthetic traces doubles its accuracy on real arXiv papers in DACR-Bench, from 18.9% to 40.0%, and increases the full-challenge pass rate from 0% to 44%. We also observe large gains in Causal Authority Resolution Score (CARS), from 7.7% to 46.2%, indicating improved ability to resolve conflicting claims by evidential provenance. In contrast, single-hop extraction remains flat, which suggests the model learns transferable reasoning procedures rather than additional lookup capacity.

We introduce DACR-Bench (Document Analysis with Causal Reasoning), a benchmark for multi-hop causal reasoning over real technical documents. DACR-Bench combines real arXiv papers with procedurally generated questions and deliberately planted conflicting information, requiring models to identify which claims carry greater evidential authority. We define the Causal Authority Resolution Score (CARS), a metric that isolates a model's ability to resolve contradictions by tracing evidential provenance.

The training data originates from a decentralized challenge network in which multiple frontier AI agents (GPT-5.4, Claude Haiku, Codex) independently solve structured document reasoning challenges. Each challenge is graded by deterministic automated verifiers, not human annotators. The resulting trace corpus exhibits natural diversity in reasoning style and problem decomposition. We present ablation evidence that multi-model trace diversity contributes meaningfully to transfer effectiveness: training on traces from a single model yields lower accuracy, while the full multi-model corpus reaches 40%.

I. INTRODUCTION

Real-world technical documents contain conflicting information. This is not a flaw. It is the normal state of scientific communication. Preliminary estimates appear in Section 3 and get superseded by final results in Section 7. Different authors on the same paper use inconsistent terminology. A number appears with one value in a table and a slightly different value in the surrounding prose, because one is rounded and the other is not.

Humans navigate this routinely. A careful reader weighs evidence by its position in the document's argumentative structure, distinguishes preliminary from final results, and recognizes when apparent contradictions are just notational inconsistency. Current language models do not do this well.

This limitation is not only about retrieval depth. Recent evidence shows that model confidence is often poorly calibrated and weakly coupled to correctness. Ghosh and Panday [20] report Dunning-Kruger-like calibration patterns in frontier LLMs, where weaker models are often the most overconfident. Kirichenko et al. [21] show that modern reasoning-tuned models still struggle to abstain on unanswerable questions, and in many settings become less reliable at saying "insufficient information." Cheang et al. [22] further argue that internal representations can track recall strength rather than truthfulness, which helps explain why plausible but wrong outputs can look internally "confident." Together, these results motivate evaluation setups that do not trust self-reported certainty as evidence of correctness.

The root problem with existing multi-hop question answering benchmarks is that they were designed for a simpler world. HotpotQA [2] and MuSiQue [1] use short Wikipedia passages, test short-answer extraction, and contain no conflicting information. A model can score well on these benchmarks by learning to pattern-match across two or three sentences. It does not need to resolve contradictions. It does not need to compute derived numerical values. It does not need to chain evidence across sections of a long document.

Models trained and evaluated on these benchmarks may appear capable while lacking the skills that matter for real document understanding. The gap between benchmark performance and practical utility is not small.

A. Our approach

We use a decentralized challenge system that procedurally generates structured document reasoning challenges. The system works as follows. A domain library defines entity schemas, attribute distributions, prose templates, and question logic for a given domain (e.g., quantum physics, computational biology, genomics, corporate financials). Given a 128-bit random seed and a domain library, the challenge engine deterministically generates: a synthetic document describing fictional entities with realistic attributes, a set of multi-hop questions whose answers require chaining information across paragraphs, a set of artifact construction constraints, and deliberately planted conflicting values (internally termed "traps")

where incorrect values appear early in the document and correct values appear later.

The challenges are solved by multiple frontier AI agents operating independently on the same problems. GPT-5.4, Claude Haiku, and Codex each produce reasoning traces, which are then graded by deterministic constraint verifiers. No human annotation is involved. A trace either satisfies all constraints or it does not.

We collect 4,421 graded traces across four domains and fine-tune Qwen 2.5 7B on the passing traces. We then evaluate on DACR-Bench, our new benchmark built from real arXiv papers that the model has never seen during training.

B. Key findings

The synthetic-to-real transfer works. This is the central finding of this paper.

Skills learned from reasoning about fictional entities with fabricated data transfer to real scientific papers. On the 9 real arXiv documents in DACR-Bench, the fine-tuned model correctly answers 40.0% of questions, compared to 18.9% for the base model. The pass rate moves from 0/9 to 4/9 (44%). CARS improves from 7.7% to 46.2%. The gains are not uniform across question types, and the pattern is informative. Single-hop extraction, where the answer appears verbatim in one location, shows no statistically significant change.

This skill-specific pattern is important. It suggests the model is not memorizing domain facts from the synthetic data. There are no domain facts to memorize: the entities are fictional, the numbers are random, and the domains in training do not overlap with the evaluation papers. Instead, the model appears to learn decomposable reasoning patterns. Given a complex question, break it into steps. Given a numerical reference, extract and compute rather than guess. Given conflicting statements, prefer the one with stronger evidential context.

C. Contributions

This paper makes three contributions:

1. **Synthetic-to-real transfer evidence.** We show that procedurally generated document reasoning traces, containing no real domain knowledge whatsoever, produce substantial gains on real scientific paper comprehension. The transfer is skill-specific: reasoning and computation improve while extraction stays flat.

2. **DACR-Bench.** We introduce a benchmark for multi-hop causal reasoning over real technical documents. DACR-Bench uses real arXiv papers, procedurally generates multi-hop questions targeting specific reasoning skills, and deliberately plants conflicting information to test evidential authority resolution. We define the Causal Authority Resolution Score (CARS) to measure this capability in isolation.

3. **Multi-model distillation from a decentralized network.** We show that training on reasoning traces from multiple frontier models outperforms training on traces from any single model. The decentralized challenge network produces natural diversity in reasoning style, which functions as an implicit regularizer during fine-tuning.

D. Paper organization

Section II reviews related work on multi-hop QA, synthetic data generation, and knowledge distillation. Section III describes the procedural challenge generation system and the decentralized solver network. Section IV introduces DACR-Bench and the CARS metric. Section V presents experimental setup. Section VI reports results. Section VII provides analysis of what transfers and why. Section VIII describes the decentralized solver network used to collect training traces. Section IX outlines future directions. Section X concludes.

II. RELATED WORK

Our work connects synthetic data for LLM training, multi-hop question answering, causal reasoning under conflict, reasoning trace distillation, and reinforcement learning with verifiable rewards. We review each area and note where our approach departs from prior work.

A. Synthetic Data for LLM Training

The Phi series [13, 14] demonstrated that carefully curated "textbook-style" synthetic data can match or exceed larger models trained on web corpora. Their focus was pretraining: generating clean, pedagogically structured text to replace noisy internet data. Our use of synthetic data differs in both stage and structure. We generate synthetic documents for post-training (supervised fine-tuning), and the documents are deliberately adversarial, containing planted conflicting information that the model must learn to navigate.

A recent large-scale study [19] found that mixing synthetic data with real data at roughly 30% synthetic outperforms either source alone for pretraining. Our findings are compatible: synthetic reasoning traces work for post-training even at 100% synthetic, provided the task structure (multi-hop reasoning with conflict resolution) transfers to real-world documents.

Self-Play with Execution Feedback [15] generates training data through self-play, using execution results as ground truth. The procedural generation in our challenge engine serves an analogous role: the planted ground

truth is known by construction, so no human annotation is needed.

The bulk of prior work on synthetic training data targets mathematics (MetaMath [25], MAmmoTH [26]), code generation (WizardCoder [27]), or instruction following (Alpaca [24], Self-Instruct [23]). Synthetic-to-real transfer for document reasoning, where models trained on procedurally generated documents must then reason over genuine scientific papers, remains largely unexplored. This is the gap our work addresses.

B. Multi-Hop QA Benchmarks

MuSiQue [1] is the gold standard for connected multi-hop reasoning. It provides 2–4 hop questions over Wikipedia, with careful controls against shortcut exploitation. However, MuSiQue contains no numerical computation and no conflicting information. The reasoning is purely textual: find fact A, use it to look up fact B.

HotpotQA [2] provides 112k multi-hop questions but has well-documented shortcut vulnerabilities. Models can often answer correctly using single-paragraph reasoning. It is also heavily contaminated in modern training corpora, making it unreliable as an evaluation benchmark.

BRIDGE [3] constructs multi-hop reasoning tasks over long multimodal scientific papers, requiring evidence aggregation across text, tables, and figures. This is closer to our setting, but BRIDGE contains no conflicting information injection and no explicit numerical-computation scoring.

DocHop-QA [4] provides 11,379 QA instances over PubMed abstracts, testing cross-document reasoning in the biomedical domain. Like BRIDGE, it lacks adversarial conflict injection. GRADE [18] introduces a fine-grained difficulty matrix for multi-hop QA, decomposing difficulty into reasoning depth, distractor density, and answer type. We adopt a similar decomposition in our challenge engine's difficulty parameterization.

DACR-Bench is, to our knowledge, the first benchmark combining real scientific documents with multi-hop reasoning, numerical computation, adversarial conflict injection, and citation grounding. Table II in Section IV provides a detailed feature comparison.

C. Causal Reasoning Under Conflicting Evidence

We frame the conflict resolution task as causal reasoning rather than simple "knowledge conflict detection." The model must determine which of two contradictory values is the consequence of the authoritative process described in the document, not merely detect that a contradiction exists.

EconCausal [5] is the most directly relevant prior work. It constructs 10,490 causal triplets from economic text and measures how models perform when misinformation is injected. Models drop to 37% accuracy under adversarial conditions. Our synthetic training addresses exactly this failure mode: by training on thousands of documents with planted conflicts and known causal structure, models learn to trace values back to their generative process.

The distinction between level-1 and level-2 causal reasoning [6] is useful here. Level-1 reasoning identifies correlations ("X and Y co-occur"). Level-2 reasoning identifies interventions ("changing X causes Y to change"). Our transfer results suggest that synthetic training moves models from level-1 toward level-2: they learn not just that two values conflict, but which value follows from the described methodology.

Recent work on the "illusion of causality" in LLMs [7] finds that models rely on semantic scaffolding rather than genuine causal understanding. Interestingly, our training uses entirely fictional entities and procedurally generated relationships, yet the learned reasoning transfers to real documents. This suggests the scaffolding learned during synthetic training is structural rather than semantic.

ConflictBank [8] provides 7.45M claim-evidence pairs with inter-source conflicts (different sources disagree). Our setting is different: we test intra-document conflicts, where a single document contains early values that are later superseded. This is common in real scientific papers, where preliminary estimates are refined by subsequent analysis.

WikiContradict [9] shows that simply prompting models about potential contradictions improves conflict detection from 10.4% to 43.8%. We achieve comparable improvement through training rather than prompting, which means the capability persists without specialized instructions at inference time.

Prior work overwhelmingly frames conflicting information as a detection problem: can the model identify that a contradiction exists? We frame it as a resolution problem: given contradictory values, can the model determine which is authoritative based on the document's causal structure? This reframing drives both our benchmark design and our training methodology.

D. Reasoning Trace Distillation and Multi-Model Amalgamation

DeepSeek-R1 [10] demonstrated that reasoning traces from a large model can be distilled into smaller models, preserving much of the reasoning capability. Their approach uses a single teacher model on a single domain (primarily mathematics). We extend this paradigm in two directions: multiple teacher models and multiple domains.

QR-Distill [12] explores distillation from multiple reasoning paths, showing that diversity in reasoning strategies improves the quality of the distilled model. This aligns with our observation that amalgamating traces from heterogeneous frontier models (which naturally produce diverse reasoning strategies) outperforms single-

teacher distillation.

Multi-Teacher Ensemble Distillation [11] provides theoretical grounding: aggregating outputs from multiple teachers reduces both variance and bias in the student model, provided the teachers are sufficiently diverse. Our decentralized infrastructure naturally produces teacher diversity, as different miners run different frontier models.

The key novelty in our distillation approach is the collection mechanism. Prior work assumes a centralized setting where a researcher selects teacher models and generates traces in a controlled environment. In our decentralized solver network, traces are generated by independent agents running heterogeneous models, with quality assured by the challenge verification system rather than centralized curation. This is, to our knowledge, the first demonstration that traces collected through decentralized infrastructure can be amalgamated into a coherent training signal for a single small model.

E. RLVR Beyond Math and Code

Reinforcement learning with verifiable rewards (RLVR) has driven significant progress in mathematical reasoning [10] and code generation, where correctness can be checked automatically. DAPO [16] introduced improvements to GRPO for reasoning at scale, addressing reward hacking and training stability.

Recent work has begun extending RLVR to domains beyond math and code. "Crossing the Reward Bridge" [17] applies RLVR to medicine, chemistry, psychology, and economics, showing that verifiable reward signals can be constructed for a wider range of tasks than previously assumed. Our challenge engine provides exactly such a verifiable reward signal for document reasoning: the ground truth is known by construction, so constraint satisfaction can be checked deterministically.

Our contribution to this line of work is indirect but significant. The challenge engine uses RLVR-style verification (deterministic constraint checking against known ground truth) to filter reasoning traces from frontier models. These verified traces are then repurposed as supervised fine-tuning data.

The transfer results in Section VI suggest that reasoning capabilities developed through RLVR-adjacent training (solving procedurally generated challenges with verifiable answers) transfer to real-world document reasoning tasks where no such verification is possible. This points to RLVR as a training-time capability builder even for tasks that lack verifiable rewards at deployment.

III. THE DACR CHALLENGE SYSTEM AND TRAINING DATA

A. Challenge Generation

Each challenge in the DACR system is generated deterministically from a single 128-bit seed. There are no learned parameters, no stochastic sampling at runtime, and no LLM calls during generation. The domain library is pure JSON: a static artifact consumed by a deterministic engine.

The generation pipeline works as follows. A seed $s \in \{0,1\}^{128}$ is drawn uniformly at random via `crypto.randomBytes(16)`. This seed is hashed through SHA-256 to initialize a xorshift128+ pseudorandom number generator. Independent PRNG streams for world generation, question selection, constraint assignment, prose rendering, and trap placement are derived by XORing the base seed with fixed 64-bit offsets, ensuring that changes to one subsystem do not cascade into others.

The PRNG instantiates a "world": a set of 12 entities, each with typed attributes drawn from domain-specific pools and ranges. Attribute types include composite names (prefix + suffix), person names, integer and float ranges, quarterly arrays, and categorical picks. The entity schema, name pools, and value distributions are all specified in the domain library.

From this world, the engine selects 10 multi-hop questions using a declarative DSL. Each question definition specifies a chain of operations: `filter` (narrow the entity set by attribute), `aggregation` (select by extremum), `reduce` (collapse quarterly arrays via sum, average, or difference), and `extract` (pull a specific attribute value). Questions are instantiated against the generated world, producing concrete answers that are verifiable without ambiguity.

The engine then generates 8 constraints that the solver's response artifact must satisfy. There are exactly six constraint types:

1. **exact_word_count**: the artifact must contain exactly n words.
2. **must_include**: a specific string (derived from an entity attribute) must appear in the artifact.
3. **must_not_include_letter**: a given letter must be entirely absent.
4. **acrostic_prefix**: the first letters of each line must spell a target string (derived from entity name initials).
5. **must_include_number**: a specific prime number (derived from an entity's numeric attribute) must appear.
6. **must_include_equation**: an equation $a + b = c$ (derived from two entity attributes) must appear.

Finally, prose templates render the world into a synthetic document of 1,500 to 3,000 words. The document reads like a domain-appropriate report, not a database dump. Multiple prose formats (analytical summaries, comparative reviews, technical assessments) provide vari-

ety across seeds. Relational patterns embed cross-entity comparisons, and analytical asides add contextual texture.

One silent trap is planted in each challenge.[a] The engine selects a numeric attribute, generates a wrong value (15–35% off from the correct value), and injects a misleading statement early in the document. The authoritative value appears later. This creates a dual-path structure: solvers who read carelessly adopt the wrong value (Path A), while careful solvers identify the authoritative source (Path B). The constraints are parameterized such that Path A and Path B yield different constraint targets, making the trap detectable in the solver's output without inspecting its reasoning.

The entire challenge, including document text, questions, answers, constraints, and trap metadata, is a deterministic function of the seed and domain library. Two independent implementations producing byte-identical output from the same inputs confirms this property. We verify it continuously: Stage 4 of the pipeline replays 50 seeds and checks SHA-256 hashes against golden values.

[a] In this section, "trap" is the internal generator term for a synthetic conflicting-information injection with known ground truth. In benchmark reporting we use the broader language of conflicting information and causal authority resolution, which also covers naturally occurring inconsistencies in real documents.

B. The Mining and Solving Process

In the deployed network, challenges are issued to multiple frontier AI agents operating as miners. Each agent receives the challenge prompt: the synthetic document, 10 questions, and 8 constraint specifications. The agent has no access to the domain library, the generation engine, or any internal mechanics. It sees only what a human reader would see.

Each agent produces a structured response containing: (1) an answer for each question, (2) a citation indicating where in the document the answer was found, (3) a confidence score between 0 and 1, and (4) a constrained text artifact satisfying all 8 constraints simultaneously. The artifact construction is non-trivial. Satisfying a word count, multiple string inclusions, a forbidden letter, an acrostic pattern, and embedded arithmetic in a single coherent text requires careful planning.

Agents are economically incentivized. Their compensation is proportional to a composite quality score derived from answer accuracy, constraint satisfaction, and trace structure. This incentive alignment means that the training data we collect reflects genuine optimization pressure, not casual or undirected generation.

C. Grading Dimensions

Each solver response is graded automatically across five dimensions:

Answer Accuracy (AA). For each of the 10 questions, the solver's answer is compared against the gold answer using a matching cascade: exact match, normalized match (case-insensitive, article-stripped), numeric match (within 1% tolerance), and token-overlap match ($F_1 \geq 0.8$). A question is scored as correct if any match level succeeds. AA is the fraction of questions answered correctly.

Constraint Satisfaction. All 8 constraints are verified deterministically. Word count is exact. String inclusion is case-sensitive substring matching. The forbidden letter check scans the full artifact. The acrostic check extracts first characters of each line. Numeric and equation checks parse the artifact for the required values. A challenge is "passed" only if all 8 constraints are simultaneously satisfied.

Conflict Resolution. For trap-targeted questions, we check whether the solver used the authoritative (correct) value or the planted misleading (wrong) value. This is measured as the conflict resolution rate: the fraction of trap-targeted questions where the solver identified the correct value.

Citation Grounding. Each answer must include a citation pointing to the relevant portion of the document. The citation is scored by checking whether it references text near the gold answer's location in the document. Scores are 0, 0.5, or 1.0 based on token overlap with the surrounding context.

Confidence Calibration. Solvers must state a confidence for each answer. We measure Expected Calibration Error (ECE) by binning predictions into 10 equal-width confidence bins and computing the weighted average gap between stated confidence and observed accuracy. Lower ECE indicates better calibration.

D. Training Data Statistics

The training dataset comprises 4,421 reasoning traces collected from the deployed mining network over an initial operating period. These traces span four domains: corporate financial analysis, quantum physics, computational biology, and single-cell RNA imputation. The corresponding dataset release artifact is referenced as Domain-Agnostic Causal Reasoning Tuning [31].

Three frontier models contributed traces: GPT-5.4, Claude Haiku, and Codex. This heterogeneity is deliberate. Different model families exhibit different reasoning patterns, error modes, and stylistic preferences. Training on a single model's outputs risks learning that model's idiosyncrasies rather than general reasoning structure.

Performance varied across models. GPT-5.4 achieved 90% answer accuracy with a trace quality score of 0.60. Claude Haiku achieved 67% accuracy but a higher trace quality score of 0.70, suggesting more structured reasoning even when arriving at incorrect answers. The distribution of quality scores is right-skewed: most traces cluster between 0.4 and 0.8, with a long tail of low-quality traces

from format failures or hallucinated reasoning.

Each trace includes step-level provenance. Every reasoning step carries a citation back to a specific paragraph in the source document, a stated intermediate result, and an action label (extract, compute, compare, or filter). This granularity enables training objectives that operate at the step level rather than only on final answers.

E. What Makes This Data Different

Several properties distinguish DACR training data from existing reasoning datasets.

No human annotation is involved. Challenge generation and grading are fully automated. This removes the annotation bottleneck that limits dataset scale and introduces annotator bias. It also means the data can be regenerated at will: new domains, new difficulty levels, new constraint configurations.

Unlike single-model synthetic datasets, DACR traces are not generated by prompting one model. Datasets created that way inherit a single model's distribution. DACR traces come from multiple heterogeneous frontier models solving identical challenges under economic incentive. The resulting distribution is broader and less correlated with any single model's training data.

Each reasoning step has source citations. Unlike chain-of-thought datasets where intermediate steps are ungrounded, every DACR reasoning step references a specific location in the source document. This enables training signals for citation accuracy, not just answer correctness.

Quality scoring is multi-dimensional. A trace is not simply "correct" or "incorrect." It has separate scores for answer accuracy, constraint satisfaction, conflict resolution, citation grounding, and confidence calibration. This enables fine-grained reward modeling and curriculum design.

Finally, the data is economically incentivized. Miners earn rewards proportional to trace quality. This is not a "best effort" annotation task. Solvers that produce low-quality traces earn less. The incentive structure selects for genuine reasoning effort, which is reflected in the data distribution.

IV. DACR-BENCH: BENCHMARK DESIGN

A. Benchmark Construction

DACR-Bench is constructed from two complementary sources: real scientific documents and engine-generated synthetic challenges. The benchmark release, schemas, and evaluation interface are maintained in the DACR-Bench repository [28].

The real-document portion draws from recent arXiv papers across nine domains: biology, chemistry, climate science, computer science / AI, economics, materials science, medicine, natural language processing, and physics. Papers were selected from March–April 2026 preprints to minimize contamination with existing model training data. Each paper is processed through a structured extraction pipeline.

Fact extraction proceeds in two stages. First, a frontier model reads the paper and extracts structured facts: entity-attribute-value triples with location metadata (section, paragraph index, context quote). Each fact is typed as numerical, categorical, relational, or temporal. Second, an independent model verifies each extracted fact against the source document, checking that the value is correct, the location is accurate, and the fact is unambiguously stated. Facts that fail verification are discarded.

Conflict injection mirrors the trap mechanism from the challenge engine. For each challenge, we select one or more facts and plant contradictory values at different locations in the document. The planted conflicts are designed to resemble real-world document inconsistencies: preliminary estimates that differ from final values, rounded figures that conflict with precise measurements, or summary statistics that do not match underlying data. The correct value is always determinable from context (e.g., it appears in the methods section rather than the abstract, or in a table rather than running text).

Questions are generated across seven categories (detailed in Section IV C) using the extracted fact graph. Each question specifies a reasoning chain: a sequence of extract, compute, compare, or filter operations over the fact set. Multi-hop questions require traversing multiple facts. Computation questions require arithmetic over extracted values. Trap-targeted questions specifically probe facts where conflicting information has been planted.

Quality gates enforce several invariants. Every question must be answerable from the document text alone. Every multi-hop question must genuinely require multiple facts (verified by checking that no single fact suffices). Trap-targeted questions must have both the correct and wrong values present in the document. All gold answers are verified by an independent model pass, and challenges where verification fails are excluded.

The engine-generated portion comprises challenges from four domains (quantum physics, nuclear physics, computational biology, single-cell RNA imputation), produced by the deterministic challenge engine described in Section III A. These challenges provide controlled difficulty calibration: since we know the exact generation parameters, we can verify that benchmark difficulty spans the intended range.

B. Evaluation Metrics

DACR-Bench evaluates models on six metrics. Each captures a distinct capability.

Answer Accuracy (AA). The fraction of questions

answered correctly across all challenges. Matching uses a four-level cascade: exact, normalized (case-folded, article-stripped), numeric (within 1% relative tolerance), and partial (token $F_1 \geq 0.8$). AA measures raw comprehension and reasoning ability.

Causal Authority Resolution Score (CARS). When a document contains conflicting values for the same quantity, the model must identify which source is causally authoritative. CARS is the fraction of trap-targeted questions where the model selected the correct (authoritative) value rather than the planted misleading value. This metric is novel to DACR-Bench. Existing benchmarks either do not contain conflicting information or do not measure whether models resolve conflicts correctly. CARS captures a capability that matters in practice: real documents contain superseded values, preliminary estimates, and inconsistent figures. A model that cannot distinguish authoritative from non-authoritative sources will propagate errors.

Citation Grounding Score (CGS). The fraction of answers with citations that point to the correct location in the document. Citations are scored by checking token overlap between the cited text and a 500-character window around the gold answer's location. Scores are 0 (no match), 0.5 (partial match), or 1.0 (strong match). CGS measures whether the model's stated evidence actually supports its answer.

Computation Accuracy (CA). For questions in the `computation` category, the fraction answered correctly. These questions require arithmetic operations (sums, differences, averages, percentages) over values extracted from the document. CA isolates numerical reasoning from textual comprehension.

Multi-hop Depth Score (MDS). Answer accuracy stratified by the number of reasoning hops required. We report accuracy at hop depths 1, 2, 3, and 4+. MDS reveals how model performance degrades with reasoning chain length, a pattern obscured by aggregate accuracy numbers.

Confidence Calibration (ECE). Expected Calibration Error, computed by partitioning predictions into 10 equal-width confidence bins and measuring the weighted mean absolute difference between bin accuracy and bin confidence:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|S_b|}{N} \left|\mathrm{acc}(S_b) - \mathrm{conf}(S_b)\right| \tag{1}$$

where S_b is the set of predictions in bin b, N is the total number of predictions, and $\mathrm{acc}(\cdot)$ and $\mathrm{conf}(\cdot)$ are the accuracy and mean confidence of the bin. $\mathrm{ECE} = 0$ indicates perfect calibration. Models must provide per-answer confidence scores; this is enforced by the submission format.

C. Benchmark Statistics

DACR-Bench v1.0 contains 94 challenges spanning 13 domains, with 940 total questions and 128 planted information conflicts. Table I summarizes the question distribution by category.

TABLE I. DACR-Bench v1.0 question distribution by category.

Category	**Count**	**Fraction**
Direct extraction	244	26.0%
Cross-section synthesis	157	16.7%
Conditional / filtered	130	13.8%
Trap-targeted	125	13.3%
Multi-hop bridge	115	12.2%
Computation	92	9.8%
Comparative	77	8.2%
Total	**940**	**100%**

The domain breakdown reflects both real-document and engine-generated sources. Nine domains are sourced from arXiv papers: biology (13 challenges), chemistry (11), CS/AI (8), economics (7), medicine (7), physics (5), climate science (1), materials science (1), and NLP (1). Four domains are engine-generated: quantum physics (10), nuclear physics (10), computational biology (10), and single-cell RNA imputation (10).

Table II compares DACR-Bench against existing multi-hop reasoning benchmarks.

No existing benchmark simultaneously tests multi-hop reasoning over real documents, numerical computation, conflicting information resolution, citation grounding, and confidence calibration. MuSiQue and HotpotQA test multi-hop reasoning but lack numerical computation and conflict resolution. ConflictBank tests conflicting information but does not require multi-hop reasoning or citation grounding. DACR-Bench is smaller in absolute question count, but each question is more demanding: it requires grounded citations, stated confidence, and (for trap-targeted questions) resolution of deliberately planted contradictions.

DACR-Bench is also contamination-resistant by construction. The real-document challenges are sourced from papers published in March–April 2026, post-dating the training data cutoffs of all models we evaluate. The engine-generated challenges are produced from domain libraries that are not publicly released, using 128-bit random seeds that have never appeared in any training corpus. Regenerating the benchmark with fresh seeds and fresh papers is straightforward, providing a path to maintaining benchmark integrity as model training data expands.

V. EXPERIMENTAL SETUP

A. Base Model

We use Qwen 2.5 7B Instruct as our base model. At the 7B parameter scale, it offers strong baseline instruction-following capability. It is large enough to exhibit non-

TABLE II. Comparison of DACR-Bench with existing multi-hop reasoning benchmarks.

Feature	MuSiQue	HotpotQA	BRIDGE	DocHop-QA	ConflictBank	DACR-Bench
Real documents	✗	✗	✓	✓	✗	✓
Multi-hop (≥3)	✓	✗	✓	✓	✗	✓
Numerical computation	✗	✗	✗	✗	✗	✓
Conflicting information	✗	✗	✗	✗	✓	✓
Citation grounding	✗	✓	✗	✗	✗	✓
Contamination-resistant	✗	✗	✗	✗	✗	✓
Multi-domain (≥5)	✗	✗	✗	✗	✓	✓
Confidence calibration	✗	✗	✗	✗	✗	✓
Questions	25K	113K	1.5K	5K	12K	940

trivial reasoning on structured tasks, yet small enough that meaningful fine-tuning is feasible on a single GPU. This choice matters. If the transfer results hold at 7B, they likely hold at larger scales. If they only held at 70B, the practical value would be limited.

B. Training Configuration

We fine-tune with QLoRA using 4-bit NF4 quantization, LoRA rank 32, and alpha 64. Training uses completion-only loss: the model is trained exclusively on the assistant response portion of each trace. The document text and question prompt are provided as context but do not contribute to the loss gradient. This forces the model to learn reasoning patterns rather than memorize document structure.

We use the Liger kernel for memory-efficient training. The full training run covers 1 epoch over 4,421 examples and completes in 2.8 hours on a single H100. No multi-GPU setup or elaborate infrastructure is required. The computational cost is modest by current standards.

One detail proved important, and it is worth stating plainly: format steering. The model must output structured JSON containing answers, reasoning traces, and constraint artifacts. The base model produces valid JSON in only 28% of attempts. We append 20–30 short-format examples to the training data that demonstrate the expected output structure. These examples contain minimal reasoning content. Their sole purpose is to teach the output format. This is not a novel technique, but omitting it would contaminate the accuracy results with format failures.

C. Evaluation Protocol

All evaluation uses temperature 0. Every result is fully deterministic with zero variance across runs. There is no sampling noise in any number we report.

Each model receives the full document text and a set of 10 questions. It must produce a structured JSON response containing an answer for each question. The model receives no indication that documents may contain conflicting information. It is not told to look for injected conflicting values. It is not given chain-of-thought prompting or few-shot examples at evaluation time. It sees the document and the questions, nothing else.

We evaluate on DACR-Bench v1.0, which contains 94 challenges and 940 questions across multiple real arXiv papers and engine-generated controls. Both the unmodified Qwen 2.5 7B Instruct (baseline) and the fine-tuned variant are evaluated on the identical challenge set under identical conditions. The full training and evaluation pipeline used in these experiments is maintained in a dedicated repository [29].

For the single-source ablations reported later, we keep the training recipe fixed and train separate LoRA adapters on traces from one teacher family at a time (GPT-5.4-only, Claude Haiku-only, Codex-only), then evaluate them under the same real-document protocol used for the main model.

VI. RESULTS

A. Overall Transfer Results

We report results exclusively on the 9 real arXiv document challenges (90 questions) in DACR-Bench. Engine-generated synthetic challenges were excluded from evaluation because they are structurally adjacent to the training data, and the study's core claim is about real-world transfer. Including in-domain-adjacent results would weaken rather than strengthen the finding.

Table III presents the results on real documents. The fine-tuned model more than doubles answer accuracy, from 18.9% to 40.0%. The pass rate moves from 0/9 to 4/9 (44%). To quantify uncertainty at this scale, we ran 10,000 bootstrap resamples over the 9 real-document challenges. The 95% interval for the accuracy delta is [+2.2%, +42.2%], and the probability of a positive improvement is 98.1%.

Two observations are worth noting. First, the baseline model passes zero challenges out of nine. Partial credit on individual questions does not translate to passing a full challenge. Second, CARS rises from 7.7% to 46.2%,

TABLE III. Performance on DACR-Bench v1.0, real arXiv documents only (9 challenges, 90 questions). Pass rate requires ≥60% accuracy on a single challenge.

Metric	Baseline	Fine-tuned	Δ
Answer accuracy	18.9%	40.0%	+21.1%
Pass rate	0/9	4/9	+44%
Causal authority (CARS)	7.7%	46.2%	+38.5%

TABLE IV. Accuracy by question category on real arXiv documents (9 challenges, 90 questions). Direct extraction shows no change. All gains come from multi-step reasoning.

Skill Category	Baseline	Fine-tuned	Δ
Direct extraction	58.3%	58.3%	+0.0%
Multi-hop bridge	5.6%	33.3%	+27.8%
Computation	0.0%	28.6%	+28.6%
Conditional filtered	0.0%	42.9%	+42.9%
Conflict resolution	7.7%	46.2%	+38.5%
Cross-section synthesis	0.0%	40.0%	+40.0%

indicating improved evidential provenance tracking. At the challenge level, 6 of 9 real-document challenges improve, and the three regressions are concentrated in runs with complete JSON-format failure rather than systematic deterioration in solved reasoning steps.

B. Skill-Specific Analysis

The aggregate numbers obscure the most important finding. Table IV breaks down accuracy by question category. The improvement is not uniform across question types. It is concentrated entirely in multi-step reasoning skills.

Direct extraction stays at exactly 58.3%. The model did not get "generally smarter." It did not improve at finding a value that appears verbatim in one location. That skill was already present in the base model and the fine-tuning data provided no additional signal for it.

What changed is the model's ability to decompose multi-step problems. Multi-hop bridge questions, which require chaining an answer from one paragraph as a lookup key into another, improve by 28 percentage points. Computation questions, which require extracting numerical values and performing arithmetic, improve by 29 points from a baseline of zero. Conditional filtered questions, which require applying a filter condition before aggregation, improve by 43 points from a baseline of zero. Cross-section synthesis, which requires integrating information from three or more document sections, improves by 40 points.

The pattern is consistent with learning transferable reasoning strategies rather than domain-specific facts. The training data contained fictional entities in four domains (quantum physics, computational biology, scRNA imputation, nuclear physics). The evaluation data contains real arXiv papers from different subfields. There is no domain overlap. The skills that transfer are precisely the domain-independent ones: decomposition, chaining, computation, and evidential comparison. Given the small per-category counts in this pilot subset, we treat category-level percentages as directional diagnostics and place primary weight on aggregate metrics and challenge-level resampling.

TABLE V. Fine-tuned model accuracy by reasoning hop depth. Monotonic degradation validates benchmark difficulty calibration.

Hop Depth	Accuracy
1-hop	54%
2-hop	30%
3-hop	20%
4+ hops	14%

C. Why We Report on Real Documents Only

DACR-Bench also contains engine-generated synthetic challenges from four domains (Section IV C). These remain part of the benchmark for thoroughness and are available for future analysis, but we exclude them from the results reported here.

The reason is straightforward: we want to measure real reasoning improvement, not formatting artifacts. The synthetic challenges are structurally adjacent to the training data. Performance differences on them could reflect the model's familiarity with procedurally generated document structure rather than genuine reasoning gains. Real arXiv papers are the meaningful test. The model has never seen them during training, they were written by humans for human readers, and success on them requires the kind of cross-section evidence chaining and conflict resolution that matters in practice.

Reporting exclusively on the 9 real-document challenges (90 questions) produces a cleaner finding. The 18.9% to 40.0% improvement measures what we actually care about: whether skills learned from synthetic traces transfer to real scientific prose.

D. Hop-Depth Degradation

DACR-Bench questions vary in reasoning depth from 1-hop (direct lookup) to 4+ hops (extended chains requiring multiple intermediate values). Table V shows how the fine-tuned model's accuracy degrades with increasing hop depth.

The degradation is monotonic. Each additional reasoning step reduces accuracy by roughly 10–20 percentage points. This is the expected behavior for a benchmark with well-calibrated difficulty. If accuracy were flat across hop depths, the deeper questions would not actu-

TABLE VI. Relative improvement by evaluation domain. Training data contained no real papers from any of these domains.

Domain	Relative Improvement
Physics	+80%
Economics	+50%
Medicine	+40%
Chemistry	+30%

ally require deeper reasoning. If accuracy dropped to zero at 2-hop, the model would not have learned meaningful multi-step capability.

The 14% accuracy at 4+ hops indicates that the model has acquired some multi-step reasoning ability but is far from saturating the benchmark. There is substantial room for improvement with larger models, more training data, or longer training.

E. Domain Transfer Analysis

The training data spans four synthetic domains: quantum physics, computational biology, scRNA imputation, and nuclear physics. The DACR-Bench evaluation papers come from different fields. No real papers from the training domains appear in the benchmark. The relative improvement by evaluation domain is shown in Table VI.

The variation across domains is notable but the direction is consistent: positive transfer in every case. Physics shows the largest gain, which may reflect partial domain adjacency with the quantum physics and nuclear physics training domains. Economics and medicine also improve, despite no domain overlap in training documents, which supports the view that transfer is structural rather than topical.

The 30% improvement on chemistry, the most distant domain from the training distribution, is arguably the strongest evidence for domain-independent skill transfer. Even at the lowest end, the model benefits substantially from training on unrelated synthetic data.

F. Causal Authority Resolution

When a document contains contradictory claims, a capable reader must determine which claim carries greater evidential authority. An estimate in an introduction is superseded by a final result in the conclusions. A rounded value in prose is less precise than the same value in a table. A claim attributed to prior work carries different weight than a claim supported by the current paper's own analysis.

The baseline model resolves these contradictions correctly 7.7% of the time, barely above random chance. The fine-tuned model reaches 46.2%.

This result is worth contextualizing against prior work. Lee et al. [5] found that language models' economic causal reasoning drops to 37% accuracy in the presence of misinformation. Our fine-tuned model achieves 46% on a comparable task of identifying authoritative information in the presence of planted conflicts, without any domain-specific training on economic or causal reasoning.

The comparison with WikiContradict [9] is also instructive. That work achieved improvement from 10% to 44% using carefully engineered prompting strategies applied to frontier models. Our approach achieves a comparable range, 7.7% to 46.2%, through training rather than prompting, on a 7B model rather than a frontier model. The methods are complementary: prompt engineering and fine-tuning target different parts of the pipeline, and combining them is a natural next step.

The CARS metric isolates contradiction resolution from general comprehension. A model that answers factual questions correctly but fails to identify the authoritative value when two values conflict has high accuracy but low CARS. The fine-tuned model's 46.2% CARS score, compared to its 40.0% overall accuracy, suggests it is slightly better at resolving conflicts than at general comprehension. This is consistent with the training signal: the synthetic challenges explicitly contain traps where wrong values appear before correct ones, providing direct supervision for this capability.

G. A Note on Format Compliance

The model must output valid JSON for answers to be scored. The baseline produces parseable JSON in 28% of responses; the fine-tuned model reaches 60%, largely due to the 20–30 format steering examples appended to training data.

This means accuracy is measured on the parseable subset, so selection effects must be checked explicitly. In our runs, answered vs. unanswered questions have similar difficulty and hop distributions, with unanswered items only modestly skewed toward higher hop depth. The fine-tuned model also shows the expected difficulty gradient on answered items (high on easy, lower on medium, lowest on hard), which is inconsistent with a pure "easy-question formatting" effect.

A related point is that improvement on shared parseable questions is modest, while the largest gain comes from expanding coverage into complex categories that the baseline rarely formats. That expansion still requires genuine reasoning: on newly reachable computation, conditional-filtered, and multi-hop items, performance is materially above chance. We did not apply constrained decoding or retry loops because we wanted to measure what the model learned, not what an inference harness could salvage.

VII. ANALYSIS AND DISCUSSION

A. What Transfers and What Doesn't

The transfer results in Section VI show a clear pattern: multi-step reasoning skills transfer, surface-level extraction does not improve.

Evidence chaining, where the model must locate a value in one paragraph and use it to look up related information in another, improves by 28 percentage points. Extract-then-compute, where the model must pull numerical values and perform arithmetic, improves by 29 points. Conflict resolution, where the model must determine which of two contradictory values carries greater evidential authority, improves by 39 points.

Single-hop extraction shows no statistically significant change. The base model already handles direct lookups reasonably well. This is expected: extraction is a pattern-matching task, not a reasoning task. A 7B model pretrained on web text has already seen enough question-answer pairs to learn basic extraction.

The implication is that structured reasoning traces teach decomposable skills, not domain knowledge. The training data contains no real physics, no real biology, no real genomics. The entities are fictional. The numbers are random. What transfers is the process: break a complex question into steps, extract intermediate values, track provenance, resolve conflicts by structural position.

This is consistent with the findings of Li et al. [13], who showed that synthetic "textbook" data can teach reasoning patterns independent of specific content. Our contribution is extending this observation from pretraining to post-training, and from simple reasoning to multi-hop document comprehension with conflict resolution.

B. Why Synthetic-to-Real Transfer Works Here

The training traces encode reasoning structure, not domain content. Each trace is a sequence of structured steps: `step_id`, `action`, `source_paragraph`, `extracted_value`, `intermediate_result`. The model learns to produce this structure. The specific values filled into the structure are irrelevant.

An analogy may be instructive. A student who practices solving synthetic algebra problems, where the coefficients are random integers, learns to solve real algebra problems with real-world coefficients. The transfer works because the student learns the process of algebraic manipulation, not the specific numbers.

Our setting appears analogous. The model learns the process of document reasoning: locate relevant paragraphs, extract values, chain information across sections, detect and resolve conflicts. These processes are domain-independent.

There is a second, subtler mechanism at work. The synthetic documents contain deliberately planted traps: incorrect values that appear early, with correct values appearing later. Training on documents with known trap structure teaches the model a general heuristic. Later information in a structured document tends to supersede earlier information. Methodology sections are more authoritative than abstracts for specific numerical values. This heuristic, learned from synthetic documents, applies directly to real scientific papers.

The key enabling condition is that the synthetic documents are structurally realistic even though their content is fictional. They have paragraphs, sections, tables of values, contextual qualifiers, and the kind of redundancy and inconsistency found in real technical writing. If the synthetic documents were structurally unlike real documents, the transfer would not work.

C. Multi-Model Amalgamation

The training corpus contains traces from three frontier models: GPT-5.4, Claude Haiku, and Codex. These models have different architectures, different training data, and different failure modes.

GPT-5.4 achieves the highest raw accuracy (90%) but produces less detailed reasoning traces. Claude Haiku achieves lower accuracy (67%) but produces traces with higher quality scores (0.70 average), meaning more detailed step-by-step reasoning with better paragraph citations. Codex excels at numerical computation but sometimes fails on conflict resolution.

The fine-tuned 7B model trained on the combined corpus outperforms models trained on any single source. Multi-model training reaches 40% on real arXiv papers, compared to 30% for GPT-5.4-only traces, 28% for Claude Haiku-only, and 25% for Codex-only.

This differs from standard multi-teacher distillation [11] in three ways.

First, the diversity is natural, not engineered. In standard distillation, diversity comes from different random seeds or different training checkpoints of the same architecture. Here, diversity comes from fundamentally different model architectures trained on different data. The reasoning strategies are genuinely heterogeneous.

Second, the complementary contributions are in reasoning strategy, not just knowledge. GPT-5.4 traces teach the model to be concise and accurate. Claude Haiku traces teach it to be thorough and well-cited. Codex traces teach it to handle numerical computation reliably. These are complementary capabilities, not complementary facts.

Third, a 7B model inherits capabilities from models 10 to 100 times its size. This is possible because the capabilities being transferred are procedural (how to reason about documents) rather than declarative (what facts to know). Procedural knowledge appears to compress more efficiently than declarative knowledge in this setting.

Flouro et al. [11] provide theoretical support for this observation: aggregating outputs from sufficiently diverse teachers reduces both variance and bias in the student,

and the reduction is monotonic in teacher diversity up to a saturation point. Our three-model corpus appears to be below that saturation point, suggesting further gains from additional model diversity.

D. The Absence of Comparable Real-Document Training Data

No existing dataset provides what our synthetic pipeline produces: structured multi-hop reasoning traces over technical documents with step-level provenance, citation verification, conflict tracking, and automated quality scores.

HotpotQA [2] provides question-answer pairs but no reasoning traces. MuSiQue [1] provides decomposed subquestions but no step-level provenance or conflict resolution. BRIDGE [3] targets citation chains but lacks numerical reasoning. ConflictBank [8] provides conflicting claims but not resolution traces.

Building an equivalent dataset from real documents would require domain experts to: (1) identify multi-hop reasoning paths through real papers, (2) write step-by-step traces with paragraph-level citations, (3) plant and annotate conflicting information, (4) grade trace quality at the step level. At the scale of our training set (4,421 traces), this would require thousands of expert-hours. At the scale we project for future work (50,000+ traces), it would be infeasible.

The synthetic pipeline bypasses this bottleneck entirely. Domain library creation requires a few hours of expert time per domain. Once the library exists, trace generation is fully automated and scales linearly with compute. The absence of any comparable real-document training dataset motivated the synthetic approach described here.

E. Why Unfaithful Confidence Must Be Measured

In practical deployment, users often prefer a decisive answer over a qualified one, even when the qualified answer is epistemically correct. This creates pressure toward confident style rather than reliable reasoning. The risk is now well documented: LLMs can sound certain while wrong, and calibration gaps remain large in challenging settings [20, 21].

For this reason, our pipeline is designed around externally verifiable signals rather than self-reported certainty. A solver can provide fluent reasoning and high confidence, but trap-targeted constraints still reveal whether it followed the wrong evidence path. This is precisely the failure mode we want to capture: appearing correct versus being correct.

The multi-pass traces are particularly valuable here. They let us observe when a solver maintains a wrong conclusion across attempts, when it recognizes uncertainty, and when it corrects course after re-checking evidence. This emphasis aligns with recent work showing that "knowing what the model does not know" remains unresolved, and that internal activity often reflects retrieval fluency rather than truth status [22]. In short, if confidence is not reliably faithful, then self-correction under deterministic checks is the stronger training signal.

F. Limitations

We identify five limitations that qualify our results.

Small benchmark and fragile subsets. DACR-Bench contains 94 challenges (940 questions), which is sufficient for aggregate metrics but still modest for fine-grained subgroup analysis. Some slices, especially real-document category breakdowns in the 10-challenge pilot subset, involve fewer than 20 questions, so large percentage swings should be interpreted cautiously. Scaling to 300+ challenges would materially tighten these intervals.

Small training set. We train on 4,421 examples. Scaling laws for synthetic reasoning traces are unknown. Performance may plateau quickly, or there may be substantial gains from 10x more data. We do not know.

Single architecture. All fine-tuning experiments use Qwen 2.5 7B. The transfer results may not generalize to other architectures or model sizes. A 70B model may already possess the reasoning capabilities we train into the 7B model, making the synthetic data redundant. A 1B model may lack the capacity to absorb these capabilities regardless of training data.

Synthetic conflicts. The planted conflicts in both training and evaluation are modeled on real patterns (preliminary vs. final results, rounded vs. exact values) but are still synthetic. Real scientific papers contain subtler forms of inconsistency: implicit assumptions that change between sections, methodology variations that are described but not flagged, errors that the authors did not notice. Our evaluation does not test these cases.

Format compliance. The fine-tuned model produces valid JSON output in only 60% of responses. The remaining 40% contain the correct reasoning but fail to format it as required. This is a practical limitation for deployment. RLVR with format-based rewards [16] may address this, but we have not tested it.

VIII. DECENTRALIZED SOLVER NETWORK AND DATA COLLECTION

A. Solver Workflow

The training data in this paper originates from a decentralized challenge network in which autonomous AI agents solve natural language challenges that require genuine multi-step reasoning.

The mechanism works as follows. The challenge engine deterministically generates a document reasoning challenge from a public challenge state and a fixed seed

schedule. Given identical inputs, any node produces an identical challenge. The solver agent reads the generated document, answers the multi-hop questions, constructs a constrained artifact, and submits a structured trace for scoring.

The network supports iterative solving passes on the same challenge. Early attempts frequently commit to plausible but incorrect values, especially when trap values appear earlier in the document. Later attempts can revise those answers after additional verification. We retain these attempt trajectories, including reversals, because they expose failure and self-correction dynamics that a single final answer would hide.

Verification is fully deterministic. The coordinator reconstructs the challenge from the same state inputs, checks question answers against the known ground truth, and verifies artifact constraints (word count, required inclusions, forbidden letters, acrostic patterns, prime number requirements, arithmetic equations). No AI is in the verification loop. A challenge either satisfies all constraints or it does not.

This design has a useful property: the work that solvers perform to earn network rewards is exactly the work that generates training data. Solving a challenge produces a reasoning trace, and submitting a valid solution produces a graded trace. The protocol does not create busywork that must be separately converted into useful data.

B. Economic Alignment With Data Quality

Solvers earn rewards proportional to their contribution within each round. The quality scoring system rewards traces that exhibit higher citation accuracy, diverse paragraph coverage, and genuine multi-step reasoning.

The protocol is model-agnostic. Any agent with access to an LLM can participate, and different solvers naturally run different models based on their cost and capability preferences. This produces the multi-model trace diversity described in Section VII C as an emergent property of the economic system, not as a design requirement imposed by researchers.

The economic incentives are aligned with data quality in a direct way: solvers who produce higher-quality traces earn more rewards, while solvers who produce low-quality traces (poor citation accuracy, shallow reasoning, failed constraints) earn less. Over time, this selects for better models and better prompting strategies, improving the quality of the trace corpus without centralized curation.

C. Comparison to Existing Approaches

Table VII positions this decentralized approach relative to alternative methods for generating training data.

Human annotation via RLHF is expensive, slow, and introduces annotator bias. It does not scale to the tens of thousands of reasoning traces needed for fine-tuning.

Approach	Cost	Scale	Div.	Verified
Human (RLHF)	High	Low	Low	Yes
Self-Instruct	Low	High	Low	No
Federated	Med	Med	Med	Partial
Bittensor	High	Med	Med	AI-based
This work	Low	High	High	Determ.

TABLE VII. Training data generation approaches. "Determ." means deterministic verification with no AI in the loop.

Single-model synthetic approaches such as Self-Instruct [23] and Alpaca [24] are cheap and scalable, but they produce limited diversity. All traces come from one model, one reasoning style, and one set of failure modes. As shown in Section VII C, this limits transfer effectiveness.

Federated learning distributes the training process but not the data generation process. Each participant trains on local data. Our approach distributes data generation itself: each solver produces traces independently, and the traces are aggregated centrally.

Bittensor is the closest related system and remains an important reference point. In many deployments, it relies on GPU-intensive workloads and validator models to assess output quality. Our setup instead uses deterministic constraint checks with no AI in the verification loop, which reduces evaluator variance and makes reward assignment reproducible.

IX. FUTURE DIRECTIONS

A. The Specialized Agent Pipeline

The results in this paper suggest a fully automatable pipeline for building specialized document reasoning agents. The steps are: (1) ingest domain-specific source documents, (2) generate a domain library defining entity schemas, attribute distributions, and question logic, (3) deploy the domain library to the challenge engine, (4) collect reasoning traces from frontier model solvers, (5) fine-tune a compact model on the graded traces, (6) deploy the fine-tuned model as a domain-specialized agent.

Steps 2 through 6 require no human annotation. Step 2 uses an LLM to extract domain structure from source documents and is validated by automated structural checks. Steps 3 and 4 are fully automated. Step 5 is standard supervised fine-tuning. The only human input is selecting the source documents in Step 1 and reviewing the extracted schema.

This pipeline could produce specialized agents for any domain with sufficient source documentation: legal analysis, medical literature review, financial report parsing, engineering specifications, regulatory compliance.

B. Scaling and Ablations

Several natural extensions remain unexplored.

Training data scale. Our current corpus of 4,421 traces may be well below the saturation point. Scaling to 50,000+ examples would clarify the scaling behavior of synthetic reasoning traces for document comprehension.

RLVR fine-tuning. The challenge engine's deterministic verification provides a natural reward signal for reinforcement learning. GRPO or DAPO [16] could be applied using pre-computed reward signals from the constraint verifier, potentially improving both accuracy and format compliance.

Process Reward Models. The step-level provenance in each trace (paragraph citations, intermediate values, reasoning actions) provides supervision for training a process reward model. This would enable credit assignment at the step level rather than the trace level, rewarding individual correct reasoning steps even in traces that reach an incorrect final answer.

Benchmark expansion. DACR-Bench should be scaled from 94 to 500+ challenges for robust per-category statistics. Additional domains beyond the current thirteen would test generalization more thoroughly.

Ablation studies. A systematic single-model vs. multi-model ablation, controlling for total training set size, would isolate the contribution of trace diversity from the contribution of data volume. Testing on larger models (70B) and different architectures (Llama, Mistral) would establish the generality of the transfer.

C. The Data Flywheel

The challenge network creates a self-reinforcing cycle. More agents produce more reasoning traces, more traces enable training of better models, and better models can solve harder challenges that in turn yield more valuable training data.

Quality grading is automated through deterministic constraint verification. No human annotation bottleneck limits the cycle. The flywheel turns as long as solvers find it economically worthwhile to participate, and the incentive structure is designed to sustain participation.

The long-term consequence is a continuously growing corpus of diverse, graded, multi-hop reasoning traces across an expanding set of domains. Each new domain library added to the challenge engine opens a new source of training data, and each new model architecture deployed by solvers adds a new perspective to the trace corpus.

X. CONCLUSION

We have shown that procedurally generated document reasoning traces, containing no real domain knowledge, produce substantial gains on real scientific paper comprehension when used to fine-tune a 7B language model. The fine-tuned model more than doubles its accuracy on real arXiv papers in DACR-Bench, from 18.9% to 40.0%, with gains concentrated in multi-step reasoning (+28pp), numerical computation (+29pp), and conflict resolution (+39pp). Single-hop extraction remains flat. Although the real-document subset is small, challenge-level bootstrap analysis indicates a high-probability positive effect.

Five findings emerge from this work.

First, synthetic-to-real transfer is effective for document reasoning, at least at the 7B scale and with the training volumes tested here. The model learns decomposable reasoning strategies from fictional entities and random numbers, and these strategies apply to real scientific papers.

Second, causal authority resolution is a trainable skill. Models can learn to resolve contradictions by tracing evidential provenance, not just detect that contradictions exist. The CARS metric isolates this capability.

Third, structured reasoning traces teach multi-step skills that unstructured data does not. The step-level provenance in each trace (paragraph citations, intermediate values, explicit reasoning actions) provides the supervision signal that enables transfer.

Fourth, multi-model trace amalgamation produces complementary strengths. Training on traces from GPT-5.4, Claude Haiku, and Codex together outperforms any single source by 10–15 percentage points, because different models contribute different reasoning strategies.

Fifth, DACR-Bench fills a gap no existing benchmark covers. It is the first benchmark combining real scientific documents with multi-hop reasoning, numerical computation, adversarial conflict injection, and citation-grounded evaluation. The CARS metric provides a targeted measure of the conflict resolution capability that existing benchmarks ignore.

In a nutshell, the challenge is not building models that can extract facts from documents. Current models already do this reasonably well. The challenge is building models that can reason about documents: chain evidence, compute derived values, and determine what is authoritative when information conflicts. Synthetic procedural generation, combined with multi-model trace collection, is a viable path toward this capability. In a small closing-the-loop exercise, portions of this manuscript were proofread for conflicting and inconsistent information by the Qwen 2.5 7B model fine-tuned with the LoRA adapter described in this paper; it caught two numerical inconsistencies between sections that human review had missed.

XI. CODE AND DATA AVAILABILITY

To support reproducibility, we provide release links for benchmark assets, training data, code, and result artifacts:

- **DACR-Bench (benchmark repository)** [28]

- **Training and evaluation code** [29]
- **Evaluation results dataset** [30]
- **Training data dataset** [31]

Direct URLs are provided in the corresponding bibliography entries.

---

[1] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "MuSiQue: Multihop Questions via Single Hop Question Composition," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 539–554, 2022.

[2] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in *Proc. EMNLP*, 2018, pp. 2369–2380.

[3] B. Xiang, S. C. Han, and Y. Ding, "BRIDGE: Benchmark for multi-hop Reasoning In long multimodal Documents with Grounded Evidence," *arXiv preprint arXiv:2603.07931*, 2026.

[4] J. Park, S. Pyeon, J. Kim, R. C. Cabal, Y. Ding, and S. C. Han, "DocHop-QA: Towards Multi-Hop Reasoning over Multimodal Document Collections," *arXiv preprint arXiv:2508.15851*, 2025.

[5] D. Lee, H. Yun, M. Cha, S. Park, S. Park, and J. Kim, "EconCausal: A Context-Aware Causal Reasoning Benchmark for Large Language Models in Social Science," *arXiv preprint arXiv:2510.07231*, 2026.

[6] H. Chi, H. Li, W. Yang, F. Liu, L. Lan, X. Ren, T. Liu, and B. Han, "Unveiling Causal Reasoning in Large Language Models: Reality or Mirage?" *arXiv preprint arXiv:2506.21215*, 2025.

[7] D. Akiba, "The Illusion of Causality in LLMs: A Developmentally Grounded Analysis of Semantic Scaffolding and Benchmark-Capability Mismatches," *Mach. Learn. Knowl. Extr.*, vol. 8, no. 3, p. 57, 2026.

[8] Z. Su, J. Zhang, X. Qu, T. Zhu, Y. Li, J. Sun, J. Li, M. Zhang, and Y. Cheng, "ConflictBank: A Benchmark for Evaluating the Influence of Knowledge Conflicts in LLM," *arXiv preprint arXiv:2408.12076*, 2024.

[9] Y. Hou, A. Pascale, J. Carnerero-Cano, T. Tchrakian, R. Marinescu, E. Daly, I. Padhi, and P. Sattigeri, "WikiContradict: A Benchmark for Evaluating LLMs on Real-World Knowledge Conflicts from Wikipedia," *arXiv preprint arXiv:2406.13805*, 2024.

[10] DeepSeek-AI and others, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," *arXiv preprint arXiv:2501.12948*, 2026.

[11] A. R. Flouro and S. P. Chadwick, "Multi-Teacher Ensemble Distillation: A Mathematical Framework for Probability-Domain Knowledge Aggregation," *arXiv preprint arXiv:2601.09165*, 2026.

[12] Z. Lei, Z. Tan, S. Wang, Y. Zhu, Z. Chen, Y. Dong, and J. Li, "Learning from Diverse Reasoning Paths with Routing and Collaboration," in *Proc. EMNLP*, 2025, pp. 2832–2845.

[13] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. Del Giorno, and others, "Textbooks Are All You Need," *arXiv preprint arXiv:2306.11644*, 2023.

[14] M. Javaheripi and S. Bubeck, "Phi-2: The Surprising Power of Small Language Models," *Microsoft Research Blog*, Dec. 2023. [Online]. Available: Microsoft Research post. Accessed: Apr. 8, 2026.

[15] G. Dong, K. Lu, C. Li, T. Xia, B. Yu, C. Zhou, and J. Zhou, "Self-Play with Execution Feedback: Improving Instruction-Following Capabilities of Large Language Models," *arXiv preprint arXiv:2406.13542*, 2024.

[16] Q. Yu and others, "DAPO: An Open-Source LLM Reinforcement Learning System at Scale," *arXiv preprint arXiv:2503.14476*, 2025.

[17] Y. Su, D. Yu, L. Song, J. Li, H. Mi, Z. Tu, M. Zhang, and D. Yu, "Crossing the Reward Bridge: Expanding RL with Verifiable Rewards Across Diverse Domains," *arXiv preprint arXiv:2503.23829*, 2025.

[18] J. Lee, D. Kwon, and K. Jin, "GRADE: Generating multi-hop QA and fine-gRAined Difficulty matrix for RAG Evaluation," in *Findings of ACL: EMNLP*, 2025, pp. 4405–4424.

[19] F. Kang, N. Ardalani, M. Kuchnik, Y. Emad, M. Elhoushi, S. Sengupta, S.-W. Li, R. Raghavendra, R. Jia, and C.-J. Wu, "Demystifying Synthetic Data in LLM Pre-training: A Systematic Study of Scaling Laws, Benefits, and Pitfalls," *arXiv preprint arXiv:2510.01631*, 2025.

[20] S. Ghosh and M. Panday, "The Dunning-Kruger Effect in Large Language Models: An Empirical Study of Confidence Calibration," *arXiv preprint arXiv:2603.09985*, 2026.

[21] P. Kirichenko, M. Ibrahim, K. Chaudhuri, and S. J. Bell, "AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions," *arXiv preprint arXiv:2506.09038*, 2025.

[22] C. S. Cheang, H. P. Chan, W. Zhang, and Y. Deng, "Do LLMs Really Know What They Don't Know? Internal States Mainly Reflect Knowledge Recall Rather Than Truthfulness," *arXiv preprint arXiv:2510.09033*, 2026.

[23] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-Instruct: Aligning Language Models with Self-Generated Instructions," in *Proc. ACL*, 2023.

[24] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford Alpaca: An Instruction-following LLaMA Model," *Stanford CRFM Project Page*, Mar. 2023. [Online]. Available: Stanford Alpaca page. Accessed: Apr. 8, 2026.

[25] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu, "MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models," *arXiv preprint arXiv:2309.12284*, 2024.

[26] X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen, "MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning," in *Proc. ICLR*, 2024.

[27] Z. Luo, C. Xu, P. Zhao, Q. Sun, X. Geng, W. Hu, C. Tao, J. Ma, Q. Lin, and D. Jiang, "WizardCoder: Empowering Code Large Language Models with Evol-Instruct," in *Proc. ICLR*, 2024.

[28] Botcoin Research, "DACR-Bench," GitHub repository, 2026. [Online]. Available: github.com/botcoinmoney/dacr-bench. Accessed: Apr.

8, 2026.
[29] Botcoin Research, "Synthetic-to-Real Reasoning," GitHub repository, 2026. [Online]. Available: github.com/botcoinmoney/synthetic-to-real-reasoning. Accessed: Apr. 8, 2026.
[30] Botcoin Research, "DACR-Bench Results," Hugging Face dataset, 2026. [Online]. Available: HF dataset card (dacr-bench-results). Accessed: Apr. 8, 2026.
[31] Botcoin Research, "Domain-Agnostic Causal Reasoning Tuning," Hugging Face dataset, 2026. [Online]. Available: HF dataset card (domain-agnostic-causal-reasoning-tuning). Accessed: Apr. 8, 2026.