

Perfecting Tax Returns Like Code: A Verifier-Swarm, Codebase-Style Architecture that Solves *TaxCalcBench*

Rishabh Jain and Saket R. Kumar

Prime Meridian

Abstract

We report that a *codebase-style* architecture—dedicated, PDF-native form agents—closes the 51-case TY24 *TaxCalcBench* with **100%** (51/51) accuracy (aggregated across 3 runs). This drastically outperforms the previous frontier score of **41.67%** reported by the Column Tax team using GPT-5 [1]. The massive leap in performance is due to adopting a “Claude-code” approach to tax code: each form agent can *navigate IRS PDFs* with dedicated tools, cite lines/worksheets, and share memory in a code-repo-like workflow. This PDF-native approach also generalizes better to rapidly changing IRS guidelines than traditional tax engines.

1 Introduction

Our contribution is **architectural**: a *codebase-style* system that (i) modularizes each IRS form/schedule as a typed module; (ii) equips agents with PDF-native navigation to *prove* each nontrivial line via citations; and (iii) wraps outputs in a *verifier swarm* that enforces invariants, worksheet protocols, and cross-form coherence. Earlier work combined structured prompting (e.g., chain-of-thought) and simple tool use to raise strict accuracy on *TaxCalcBench* to 41.67% [2, 3, 1]. Here we show that the codebase-style design, coupled with verifier adjudication, *perfects* *TaxCalcBench*.

2 System: Codebase-Style Agents + Verifier Swarm

Design motif. Treat the tax code like a monorepo. Each IRS form/schedule is a *module* with typed inputs/outputs; inter-form edges are explicit (e.g., Sched C→SE→1040 Line 23). These form modules are also generated using LLMs.

Dedicated form agents. Agents for Schedules 1/2/3/B/C/SE/8812/D/8995(A)/... receive: (i) role

instructions keyed to the official PDF instructions, (ii) a minimal local context (*only* relevant facts and upstream lines), and (iii) a multi-agent coordination protocol for passing upstream/downstream line items [10].

CLI-based PDF tooling. Each agent has a PDF navigator: `open(#)`, `find(regex)`, `goto(page, line)`, and `worksheet(name)`. Agents *prove* a line by attaching citations to IRS instructions/tables (Line 16 capital-gains worksheet, EIC tables, 8812 steps). This is a domain-specific instance of retrieval-augmented, document-grounded QA over PDFs [4, 5].

Deterministic helper tools. We use only tiny, pure tools: (a) TY24 tax-table lookup; (b) capital-gains worksheet calculator; (c) EIC table lookup. This “LLM + small programs” pattern mirrors prior work that externalizes computation into executable subroutines [6, 7].

Verifier swarm. Multiple verifiers test each component of the prepared tax return. A judge coordinates verifier outputs and corrects errors in the tax return (LLM-as-a-Judge [9]). This also acts like an inference-time ensemble, where multiple candidate explanations are adjudicated for consistency and correctness [8]. This approach allows us to easily incorporate multiple different models from all leading labs as independent verifiers, and assimilate the results.

3 Experimental Setup

Data/metric. TY24 *TaxCalcBench*, 51 federal cases. We track the “strict metrics”, but is tolerant to IRS-acceptable rounding (which can lead to multiple valid results when many accounting entries are compounded, eg: in a Schedule C/SE/E).

Models. Frontier LLMs via API from xAI, OpenAI, Anthropic and Google Deepmind; no finetuning,

System		Strict Acc. (%)
Reproduction baseline	(Gemini 2.5 Pro)	32.65
Form agents, no tools (Major Run 3)		40.00
Form agents + tax-table tool (Run 4)		45.10
Form agents + tax-table tool (Run 5)		47.06
<i>Codebase-style</i> agents + verifiers (Major Run 8)		100.00

Table 1: TY24 *TaxCalcBench* strict accuracy (51 cases). The verifier swarm and PDF-native agents close the suite without training or RL. Measuring against 32.65% Gemini 2.5 Pro baseline since results were measured before the GPT-5 41.67% score was registered [1].

no weight edits. Major Run 8 uses the full codebase-style system; LLM-as-a-judge to collect verifier responses [9].

Tools. Deterministic helpers for tax-specific sub-tasks (tax table lookup, EIC lookup), as well as CLI-based PDF tooling.

4 Label Corrections & Taxpayer Impact

In Major Run 8, 5 cases initially failed; disposition appears in Table 2. In 3 cases the benchmark was wrong; our returns were correct. Two were IRS-ambiguous (both acceptable) Notable impacts:

- **Student loan interest (SLI) deduction, MFJ with Sched C & W-2.** Our system correctly took SLI on Schedule 1 Line 21 when the benchmark missed it, *marking thousands of dollars as deduction*.
- **Excess business loss (Form 461) on HOH with Sched C loss.** Agents invoked Form 461 due to EBL; the benchmark omitted it.
- **Educator expense claims.** A case correctly *fused* \$600 educator deduction (no eligibility evidence); benchmark over-claimed.

5 Analysis

Why the verifier swarm works. The inspiration for this approach was recent work on a verification-and-refinement (generator-verifier) pipeline that achieved near-gold performance on IMO 2025 problems using Gemini 2.5 Pro (and other frontier models) via prompting alone [11]. There are many theories about

why this architecture performs better than a single prompt, but we provide no conclusive reasoning here, and leave it up to more skilled people.

Practicality and auditability. Deterministic tools and PDF citations create an audit trail suitable for CPA review. This makes our architecture viable for processing real tax returns. Our system is also more generalizable for future tax years compared to traditional tax engines (since we only need to replace underlying PDF docs with updated docs each year).

6 Limitations

Scope. TY24 federal-only; state interactions (decoupled QBI, local credits) remain future work. **Safety.** We include line-level rationales and citations while preparing returns; all tool calls and model reasoning traces were manually reviewed. We found no indications of malicious intent, but our findings are inconclusive. **Generalization.** While we reach 100% on this suite, broader replication across years/distributions and more complex tax returns is needed to push the frontier.

7 Conclusion

A codebase-style architecture with PDF-native form agents and a verifier swarm *perfects* *TaxCalcBench*: **100%** —without training or RL. Treating the tax code like a code base converts tax prep into a fully automated, auditable pipeline that already pays off in corrected deductions and real dollars saved.

References

- [1] M. R. Bock et al. *Evaluating Frontier Models on the Tax Calculation Task (TaxCalcBench)*. arXiv:2507.16126, 2025.
- [2] J. Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv:2201.11903, 2022.
- [3] S. Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. arXiv:2210.03629, 2022.
- [4] P. Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP*. arXiv:2005.11401, 2020.
- [5] J. Lála et al. *PaperQA: Retrieval-Augmented Generative Agent for Scientific Literature*. arXiv:2312.07559, 2023.

Case (abridged)	Disposition / Rationale
hoh-w2-1099g-unemployment-schedulec-loss	Benchmark error: Form 461 EBL correctly applied by Prime Meridian; benchmark omitted. Form-461 is required due to net business loss exceeding 2024 threshold of \$305,000 for a Head of Household filer.
mfj-dependent-claimed-2441-exclusion	Benchmark error: label takes \$600 deduction only valid for eligible educators, whereas Prime Meridian refuses this deduction. Input data does not show sufficient evidence that taxpayer is an eligible educator. Input data also explicitly says “qualified_educator”: { ... “value”: false }
mfj-spouse-dependent-schedule-c-w2-student-loan-interest	Benchmark error: valid SLI on Schedule 1 Line 21 taken by our system; \$2,500 deduction successfully found. This one is tricky because taxpayer marks spouse as possibly claimed as a dependent, which sometimes indicates that they cannot claim SLI deduction. However, after the complete return is prepared, we can see that there is a balance due, hence it is not allowed for the spouse to be claimed as a dependent. This means it is guaranteed that we can claim the \$2,500 SLI deduction.
mfj-w2-box12-codes-a-b-1099int-schedulec	Benchmark input data is ambiguous: label takes SLI of \$2,500. Prime Meridian does not take this. Even though a student interest amount is indicated in input data, it might not actually be paid, which makes the taxpayer ineligible for SLI (indicated by the input data line: “paid_student_loan_interest”: “value”: false). We assume here that the benchmark input data had some inconsistency, and accept both output results as correct.
mfj-w2-schedule-c-loss-multi-home-office	IRS-acceptable rounding differs from benchmark-labeled strict value, but we accept Prime Meridian’s output as correct.

Table 2: Disposition of initially failing Major Run 8 cases.

- [6] W. Chen et al. *Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks*. arXiv:2211.12588, 2022.
- [7] L. Gao et al. *PAL: Program-aided Language Models*. arXiv:2211.10435, 2022.
- [8] X. Wang et al. *Self-Consistency Improves Chain-of-Thought Reasoning in Language Models*. arXiv:2203.11171, 2022.
- [9] L. Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. arXiv:2306.05685, 2023.
- [10] Q. Wu et al. *AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation*. arXiv:2308.08155, 2023.
- [11] Y. Huang and L. F. Yang. *Winning Gold at IMO 2025 with a Model-Agnostic Verification-and-Refinement Pipeline*. arXiv:2507.15855, 2025.