



**The AI Lab
for Accountants**

1 HOUR CE FEDERAL TAX WEBINAR

Embracing AI for Tax & Accounting Pros

A practical, vendor-neutral foundation for understanding what these tools actually do, and how to get real work out of them without giving up your judgment.

Rebecca Driscoll, CPA & Charlie Barmore, CPA, CFE, CVA, The AI Lab for Accountants

PROGRAM NUMBER A8FPT-T-00119-26-0 | September 3, 2026 · Noon to 1:00 p.m.

WHAT WE COVERED

Five things, in order

Each one builds on the last. If you only take one thing, take the last one.

01

How it actually works

Next-token prediction, and why that explains everything else.

02

The training cutoff

Why it's confidently out of date, and how to make it go look.

03

Chat vs. Claude Code

Two different tools for two different kinds of work.

04

Getting good output

Context, instructions, memory, and the back-and-forth.

05

Beyond prep

Where AI helps that has nothing to do with returns.

It is predicting the next word

That is not a simplification for a lay audience. That is the mechanism. A language model reads everything before this point and produces the most probable next piece of text, over and over, until it stops.

- It learned those probabilities from an enormous amount of text. Nothing is stored as a fact you can look up. It is stored as **patterns in how language fits together**.
- This is why it writes so well. Fluent, well-organized prose is exactly what "most probable next word" produces.
- And it is why it can be **fluent and wrong at the same time**. Nothing in the mechanism distinguishes "true" from "sounds right."

"It is the fastest first-draft writer you will ever hire, and it has never read the Code."



Pattern-completion, not retrieval

This is the single distinction that makes the rest of it make sense.

What people assume

You ask a question. It searches something (a database, the Code, the internet), finds the answer, and reports back. Like a very fast research assistant with a library card.

What actually happens

You ask a question. It generates the text that most plausibly follows your question, one piece at a time. If a citation appears, it was **composed**, not looked up.

The practical consequence: a fabricated Code section does not look different from a real one, because both were produced the same way. This is why every citation gets verified against primary source before you rely on it. Not because the tool is bad, but because of how it works.

So what is it genuinely good at?

Sort the work by whether the answer lives in the text you gave it, or somewhere else.

Strong: the answer is in front of it

- Rewriting, tightening, restructuring
- Summarizing something you supplied
- Extracting and organizing from a document
- Explaining a concept in plainer language
- Drafting into a structure you gave it
- Arguing the other side of your position

Weak: the answer lives outside

- Supplying authority from memory
- Current-year figures and thresholds
- Arithmetic you have not re-performed
- Anything where it cannot tell you its source
- Holding a position when you push back



Its knowledge stops on a date

Every model has a **training cutoff**, the point where the text it learned from ends. It does not know that. Ask it about something after that date and it will answer from the last version of the world it saw, with the same confidence it uses for everything else.

- For us that lands squarely on **inflation-adjusted figures, new thresholds, and recent guidance**, the things that change every single year.
- It will not warn you. There is no "I might be out of date here" built into the mechanism.
- Asking it directly what its cutoff is gets you an unreliable answer, for the same reason.

Treat anything time-sensitive as stale by default, and make it go look.



Make it research, do not let it remember

Most of these tools can search the web, but they will happily answer from memory if you do not tell them not to. The fix is one sentence in your prompt.

Answers from training data

What's the standard deduction for a married couple this year?

You get a confident number with no source. It may be right. It may be two years old. You cannot tell from the answer.

Forces actual research

Search the web for the current IRS figure. Give me the source URL and the date it was published. Do not answer from your training data. If you can't find it, say so.

Now you get a number and something you can click and check.

This does not end your verification duty. Web search fixes staleness; it does not make the tool an authority. You still confirm against primary source before it reaches a return, a memo, or a client.

Chat and Claude Code are different animals

People hear "AI" and picture a chat box. That is one of them. The other one works on your actual files, and it is where a lot of the real leverage is.

	CHAT	CLAUDE CODE
Where it lives	A browser tab or desktop app	A terminal window on your computer
What it can touch	Only what you paste or upload into the conversation	Real folders and files. It reads, writes, and organizes them directly
Shape of the work	One thing at a time, in conversation	Many steps, many files, keeps going on its own
Best for	Thinking, drafting, explaining, analyzing what you hand it	Work that lives in files: building things, batch work, real automation
On-ramp	None. Open it and type.	Steeper. More power, and more ability to change things.

The line: if the work lives in a conversation, use chat. If the work lives in files on your machine, that is Claude Code. **Start in chat.** Nobody should meet AI for the first time in a terminal.

Why this matters more than it sounds

- 1 It changes what you can even attempt.** "Read these forty files and tell me what is inconsistent" is not a chat task. It is a Claude Code task.
- 2 It changes the review burden.** Chat produces one draft you read. An agent working across files produces a lot of changes, so you need a way to see what it did before you accept it.
- 3 It changes the data question.** A tool with access to your folders can reach client files. That is a decision you make deliberately, with the tier, the vendor terms, and your WISP in mind, not something you back into.

You do not need Claude Code to get value from AI this year. You do need to know it exists, because it is the difference between "AI helps me write" and "AI does a piece of the work."

Four levers, and none of them are the prompt

People ask for the magic prompt. There is not one. What separates useful output from generic output is almost entirely these four things.

Lever 1

Context

What you actually feed it. The biggest lever by far.

Lever 2

Instructions

Standing rules so you stop repeating yourself.

Lever 3

Memory

What persists between conversations, and what does not.

Lever 4

Iteration

The back-and-forth. This is the work, not a failure.



Generic in, generic out

When the output reads like a brochure, the problem is almost never the wording of your request. It is that you did not give it anything only you know.

Nothing to work with

Write an email to a client about extending their return.

It has no facts, so it produces the average of every extension email ever written. Bland, and you rewrite it from scratch.

Fed real substance

Draft an extension email. Facts: S-corp, K-1 arriving late from a partnership, client is anxious and asks a lot of questions, we've extended two years running. My conclusion: extend, pay in with the extension, and reassure them this is routine. Warm but brief.

Give it three things: the structure you want, your real inputs, and **your conclusion**. The moment the conclusion is yours, the draft stops sounding like AI, because the thinking is yours and it is just doing the typing.

Instructions and memory: stop repeating yourself

A **Project** (chat) is a container that holds standing instructions and reference files for a body of work. Everything you do inside it inherits them.

Standing instructions

Write once, applies to every conversation in that Project:

You're writing for a small CPA firm. Plain English, no jargon, no em dashes. Never state a professional-standards requirement as fact unless I gave it to you. Leave [BRACKETS] for anything I didn't supply. Don't fill gaps with guesses. Draft only, no preamble.

What memory really is

- A single conversation remembers itself, until it gets long and early details start slipping.
- A **new** conversation starts blank unless a Project or a memory feature carries something forward.
- Memory features vary by tool and change often. **Assume it forgets** and put anything that matters into the Project.

The bracket rule is worth adopting today. Left alone, these tools fill gaps with plausible invention. That instruction converts an invisible fabrication into a visible blank you can fill.



The first answer is a starting point

The most common mistake is treating the first output as the deliverable, then concluding the tool is not good enough. Experienced users almost never ship the first draft. They steer.

- 1 Ask for the **outline first**. Fix the structure before it writes a word of prose.
- 2 Steer with **one-liners**: "shorter." "warmer." "cut thirty percent." "you lost the point in paragraph three." You do not restate the task.
- 3 Say **what is wrong**, not just what you want. "Too formal for this client" beats "make it better."
- 4 Have it **argue against you** before you commit. "What is the weakest part of this position?"
- 5 When it drifts badly, **start a fresh conversation** with better context rather than fighting it.

One caution. Push back on a correct answer and many models will fold and agree with you. Your disagreement is not evidence. If it reverses under mild pressure, go check which version was right.

It is not only returns and workpapers

Everyone reaches for prep and workpaper automation first. Those are real, and they are also the uses with the most client data and the most review burden. Some of the best returns are somewhere else entirely.

Running the firm

- Turning how you do something into a written SOP
- Onboarding and training material
- Meeting notes into a real follow-up list
- Job posts, review notes, internal comms

Growing the firm

- **Defining your ideal client**, and screening against it
- Pressure-testing pricing and scope language
- Drafting proposals from your own notes
- Explaining what you do to people who are not accountants

Notice what these have in common: almost none of it involves client data or tax law. That is the fastest, safest place to start, and it is the work most firm owners never get to.



Getting clear on who you actually want

Most firms cannot describe their ideal client past "small business owner who pays on time." Here is the whole exercise, in a Project, in about an hour.

- 1 **Open a new Project** called something like "Ideal client." Everything below lives inside it, so it all builds.
- 2 **Describe your ten best clients:** industry, size, entity type, what you do for them, what they pay, how they found you, why they are easy. Then describe three you would hand back tomorrow.
- 3 **Ask it to find the pattern.** "What do the good ones share that the difficult ones do not? Where am I wrong about my own book?"
- 4 **Turn it into a written profile** with explicit disqualifiers, then save that into the Project instructions so every future conversation knows it.
- 5 **Now screen.** Paste in what a prospect's website says and ask it to score the fit, name the mismatches, and draft the three questions you should ask on the call.

No client data required. This runs on characteristics, not identities. "Manufacturer, \$4M revenue, S-corp, three entities" describes a type, not a person. You get the whole exercise without a §7216 question anywhere in it.



Why that example is the point

- It used **every lever**: real context you alone have, standing instructions in a Project, memory carried across sessions, and back-and-forth to sharpen it.
- It produced something **you would never have made time for**, not a faster version of work you already do.
- And it **compounds**. The profile feeds your proposals, your pricing, your marketing, and what you say no to.

The best early uses of AI in a firm are usually the work you have been putting off, not the work you are already doing.



What does not change

None of this is a reason to wait. It is a reason to put a few controls in place first.

Before client data goes in

- **Your account tier decides what is allowed.** A personal plan can train on your inputs and has no data agreement behind it. A firm Team or Enterprise plan excludes training and includes a DPA.
- That makes a disclosure **defensible, not automatic**. Sending return information to any third party is still a disclosure under §7216.
- **The floor either way: SSNs and EINs stay out.**
- The tool belongs in your **WISP** like any other vendor.

Before output goes out

- **Every citation is wrong until you check it** against primary source. If you cannot verify it, delete it.
- **Re-perform the math.** Do not spot-check it.
- **Check what is missing**, not just what is wrong.
- **Adopt it or rewrite it, never forward it.** You are the reviewer of record. AI drafts; you decide and you sign.

START MONDAY

Three things, about fifteen minutes each

- 1 Add the research sentence to your habits.** Next time you ask anything time-sensitive, tell it to search the web and give you the source URL. Watch how different the answer is.
- 2 Make one Project with standing instructions.** Your house style, the bracket rule, the verify rule. Stop retyping your preferences into every conversation.
- 3 Run the ideal-client exercise.** No client data, no tax law, and you will have something at the end of it you have wanted for years.

Pick the one your staff already paste client context into, and go read that vendor's actual terms. Not what you assume they say.





**The AI Lab
for Accountants**

Thank you for joining us

Questions, or want to keep going? Everything we teach lives at
ailabforaccountants.com.

FOR YOUR CE RECORDS

PROGRAM NUMBER

A8FPT-T-00119-26-0

DELIVERY

Live webinar

PROGRAM TITLE

Embracing AI for Tax & Accounting Pros

DATE

September 3, 2026, Noon to 1:00 p.m.

CREDIT

1 hour CE, Federal Tax

INSTRUCTORS

**Rebecca Driscoll, CPA and Charlie Barmore,
CPA, CFE, CVA**