

The two-model playbook.

Ten ways to put two AI models on the same problem so the second one catches what the first one missed. Why each works, and the one rule that decides whether you get better answers or just more expensive ones.

BY DAVE BROWN

When one model isn't giving you what you need, the obvious moves are to use a bigger model or write a longer prompt. Both can help. But the more reliable move is to give a second model a *different job*. Think of it as adding a second person to the team, not simply asking for another opinion.

The casual version doesn't work. Paste an answer into another window and ask "is this good?" and you'll usually get three compliments and a note about the headings. That is not a real check. You asked a vague question, so the second model had no clear way to disagree.

The second model isn't there to be smarter. It's there to be uncommitted.

A model that just wrote an argument is surrounded by its own reasoning. Asking it to find the flaw is like asking a lawyer to switch sides halfway through the case. A fresh model can start with the goal, the source material, and a clear review job. It isn't necessarily better. It just hasn't already said yes.

Three angles below: **why it works**, **the ten pairings** worth knowing, and **how to run one** without building an expensive way to be confidently wrong.

10

pairings worth knowing,
in four families

4

distinct reasons a
second pass helps at all

1

rule that decides
whether any of
them work

0

extra subscriptions
required. A second
window counts.

One clarification: “two models” really means **two independent passes with different jobs**. Using models from two companies is the strongest version because they tend to make different mistakes. But a clean second chat with a clearly assigned role gives you most of the benefit without another subscription. A fresh start often matters more than a different model.

—— WHAT'S IN HERE

- 01 Why it works. Four reasons, and the rule that decides if any of them apply.
 - 02 The ten pairings. What each is for, how to set it up, how it fails.
 - 03 Running it. The handoff, seven failure modes, and when one model is plenty.
 - 04 The drop-in prompts. Six you can paste today.
 - 05 The lookup table. Symptom in, pattern out.
-

PART ONE · WHY IT WORKS

There are only four ways to make the second pass different.

Every pattern in this guide is one of them. They don't buy the same thing, which is why picking the right one matters more than running more of them.

Underneath all four sits one fact: **producing and judging are different tasks, and judging is easier**. You know the human version. You can't proofread your own copy, and you find the typo three seconds after you print it.

Models have their own version of the same problem. They write one piece at a time, so an early wrong turn can steer everything that follows. A reviewer gets to see the finished work all at once, like someone checking the map after the trip. That gives the second pass a natural advantage. The four moves below make better use of it.

Divide the labor.

Ask one pass to research a market and write the memo, and both jobs get partial attention. By the time it starts writing, the useful evidence is competing with a long trail of half-relevant material. A clean second pass that sees only the best research can focus fully on the memo.

There's another benefit: splitting the job forces you to create a useful handoff. The plan, brief, or findings must make sense to someone who wasn't there. That turns hidden assumptions into written instructions. Like a relay race, the handoff can decide whether all the earlier work actually goes anywhere.

Turn judgment into checking.

"Is this contract summary good?" has no procedure behind it. Anyone answering it is producing an impression. "Does clause 7.2 say what this bullet claims it says?" has a procedure: go look, quote the line, pass or fail.

The strongest pairings turn a broad opinion into a short list of questions with answers you can verify. The gain isn't a smarter second model. It's a model doing a job where **being wrong is visible**. Ask it to point to evidence whenever possible. A reviewer that says "looks good" without showing why has not reviewed the work. It has merely replied.

Take more than one draw.

A model's first response is one reasonable answer, not necessarily the best answer it could produce. It usually reaches for safe, familiar choices. That is fine for routine work and a problem when you need something original.

So gather several options before you choose, and keep those attempts separate. If two fresh passes reach the same answer, the evidence may be pointing there. If the second pass saw the first answer, the agreement means much less. Think of two compasses placed beside the same magnet: matching readings do not help if both were pulled off course in the same way.

Change who's in the chair.

Telling a model to answer as a skeptical CFO does not give it new knowledge. It changes which parts of its knowledge come to the front: budget, risk, proof, and consequences.

A model's default role is to be helpful, which makes it less likely to dwell on objections, cost, or consequences. That is why "be critical" is weaker than "you are the person whose budget pays for this." The first changes the tone. The second gives the model a reason to care.

And the rule that governs all four.

Give the second model the **goal, the source material, and the criteria**. Not the first model's conclusion, and never your enthusiasm for it.

A second pass that reads the first answer also inherits its framing, vocabulary, assumptions, and blind spots. Now it is grading the answer already on the page instead of taking a fresh look at your original question.

THE THING TO REMEMBER

A reviewer can check what is on the page. An independent pass can also notice what never made it onto the page. That missing piece is often the one that matters most.

There's a second reason. A finished document looks like something that has already passed several decisions, so models tend to treat it as basically sound. "Here's my draft, thoughts?" sounds like a request for review, but it usually works more like a request for approval.

Ten ways to give two models two different jobs.

Grouped by which of the four they run on. Each gets the setup, why it works, the handoff that makes it work, and how it fails. Most people get most of the value from the three or four that match how their own work goes wrong.

FAMILY ONE · DIVIDE THE LABOR

For work that sprawls and never lands.

PATTERN 01

Planner–Executor.

Model A breaks the goal into ordered steps with success criteria. Model B follows the plan and produces the deliverable. *One model designs a research plan; the other conducts the research.*

Planning and doing require different kinds of attention. The planner needs to see the whole route. The executor needs to focus on the next turn. Ask one pass to do both, and it often starts building step one before the plan is finished, then bends the rest of the plan around what it already made.

The bigger win is practical. **A plan is the cheapest thing in the process to fix.** You can read one in ninety seconds and spot that step four answers the wrong question. You cannot check a finished twelve-page deliverable that quickly, which is why weak work sometimes survives simply because it is already complete.

THE HANDOFF

Goal, constraints, audience, and what “done” looks like. Then demand a success criterion per step: what it produces and how you’d know it worked. That’s the part models skip, and the part that makes a plan executable.

How it fails: the steps name topics instead of results. If a step doesn’t say what it produces, you have a table of contents, not a plan. It also fails when nobody reads the plan before the work starts.

PATTERN 02

Researcher–Synthesizer.

Model A gathers facts, evidence, examples and sources. Model B turns them into a narrative or a recommendation. *One model collects customer evidence; the other writes the strategy memo.*

The two jobs reward opposite habits. Research says, “save it, we may need it.” Writing says, “choose what matters and leave the rest out.” It is the difference between stocking a pantry and serving a meal. One pass doing both often produces a document full of facts but missing a clear point.

A fresh writer also sees the evidence the way your reader will. The researcher may love a fact because it took thirty turns to find. The writer does not know how hard it was to find and can cut it when it does not help the decision.

THE HANDOFF

Evidence, not conclusions. Verbatim excerpts, numbers with units and dates, who said it and when. Then name the decision: not “write a memo,” but “this has to get two execs to pick one of three segments on Thursday.”

How it fails: the researcher hands over its own summary, so the writer works from a shortened version of the evidence. Each summary drops detail, and those losses add up. Pass the material, not the memo about the material.

FAMILY TWO · TURN JUDGMENT INTO CHECKING

For work that reads well and isn’t true.

PATTERN 03

Builder–Reviewer.

Model A creates the first version. Model B identifies errors, omissions and weak reasoning. Model A revises. *One model writes a proposal; the other reviews it as a skeptical executive.*

The reason to bring in a second model is fresh distance, not extra intelligence. The builder has spent the whole conversation making its case. The reviewer arrives without a favorite team. It is not smarter. It simply hasn’t already said yes.

Two mistakes ruin this pattern. First, telling the reviewer who wrote the work and how much you like it. “Here’s a proposal I put together, what do you think?” quietly asks for approval. Second, asking “any issues?” without giving criteria. With no standard to use, the model checks only whether the document looks like a proposal, and of course it does.

THE HANDOFF

The original brief, the criteria, the role. Then a quota and a standard of proof: find the three weakest claims, and for each, what would have to be true for it to hold and what evidence would settle it.

The quota does real work. Open asks return praise; a forced ranking of three returns a review, because something has to occupy the top slot.

Then **the builder revises, not the reviewer**. If the reviewer rewrites the document, it may add new claims that nobody has checked. Keep the jobs separate. The reviewer diagnoses; the builder treats.

How it fails: the rubber stamp, and the polish spiral. Cap it at two rounds, decided in advance. By round three you’re sanding off the specific, slightly awkward parts that were the only reason it was worth reading.

PATTERN 04

Checklist–Operator.

Model A writes a detailed checklist of requirements and likely failure points. Model B works each item and reports its status. *One model creates a website-launch checklist; the other audits the site against it.*

This may be the most underrated pairing, for three reasons. First, the checklist is written before anyone sees the work, so the standard cannot bend to fit what was produced. People do this too: we read something, form an impression, and then grade it against that impression instead of its actual requirements.

Second, it turns a broad judgment into a real check. “Audit this site” returns a general impression. “Here are forty items; mark each pass, fail, or not applicable and show the evidence” returns an audit.

Third, and this changes the economics: **the checklist is an asset**. Every other pattern costs you double every single run. This one you generate once, correct by hand, and use for a year.

THE HANDOFF

Ask for likely failure points, not just requirements; requirements are the easy half. Group by severity: catastrophic, embarrassing, untidy. Require evidence per status, and forbid “not applicable” without a reason.

How it fails: long, shallow lists that check presence rather than quality. “Has a privacy policy”: yes, and it’s for a different company. Severity grouping fixes most of that, because it forces the generator to think about consequence instead of coverage.

PATTERN 05

Extractor–Verifier.

Model A converts messy material into structured information. Model B checks every claim against the original and flags anything unsupported. *One model extracts obligations from a contract; the other verifies the clause references and wording.*

Models are very useful at extraction and can be quietly wrong. A neat table feels trustworthy, and a believable wrong entry is much harder to spot than a blank one. After twenty correct rows, your eyes can slide right over the twenty-first.

Verification is simpler: the claim is in the source or it is not. Use this pairing for obligations, dates, amounts, citations, quotes, names, and any other work built from specific fields.

THE HANDOFF

The verifier gets the *source*, not the extractor’s reasoning, and works backward from each claim. Three verdicts only: supported with the exact quote, unsupported, or contradicted. Requiring the quote is the entire mechanism, because a verifier that can say “yes, correct” without producing the line has done nothing.

Then ask the fourth question people forget: what’s in the source that didn’t make it into the table? Omissions are the errors that otherwise never get caught.

How it fails: the verifier that checks the extraction against itself. If the source isn’t in front of it, that isn’t verification. It’s proofreading.

For the same three obvious ideas, every time.

PATTERN 06

Generator–Ranker.

Model A produces many possible ideas. Model B scores them against explicit criteria, picks the strongest and explains why. *One model generates 30 campaign concepts; the other ranks them by originality, feasibility and audience fit.*

Generating and judging pull in opposite directions. When a model evaluates every idea as it appears, it cuts the unusual ones too early and settles into the safe middle. Separate the brainstorm from the scorecard so odd ideas get a chance to survive.

Volume is a mechanism, not a luxury. Ask for eight and you get eight reasonable ones. Ask for thirty and the material worth having sits around eighteen through thirty, once the obvious is exhausted. **The point of asking for thirty isn't to get thirty. It's to get past the first ten.**

THE HANDOFF

Explicit criteria with weights. Score every item and show the scoring, rather than picking three favorites and praising them. Ask for two winners: the strongest, and the most interesting one that carries real risk.

How it fails: criteria the ranker invents for itself, which are always the generic ones and always favor the middle. If you didn't write down what good means for this decision, you paid twice to be handed the average.

PATTERN 07

Independent answers–Reconciliation.

Both models solve the problem separately, without seeing each other's work. Then a pass compares them, names the disagreements and produces a combined conclusion. *Two models independently estimate a project's risks, then reconcile.*

The disagreements are the useful part. When two fresh passes reach the same answer, the evidence probably points there clearly. When they reach different answers, you have found the uncertain part of the problem. One estimate gives you a number. Two independent estimates give you a number and a better sense of how much to trust it.

Before combining anything, the reconciler must explain *why* the answers differ. Maybe they used different assumptions, read an unclear sentence differently, or defined a key term in different ways. That explanation is the insight. Skip it and you get a final answer with no clear path behind it.

THE HANDOFF

Identical inputs, separate windows, different model families if you have them. The reconciler gets both answers *and* the original material, so it can compare each answer with the evidence instead of simply splitting the difference.

How it fails: averaging. If one answer is right and one is wrong, the average is a new answer that is also wrong and now looks reasonable. And leaking: run this in one chat and you haven't run it at all, because the second answer was conditioned on the first the moment it appeared. It's also the most expensive pattern here, so spend it on decisions with real downside.

FAMILY FOUR · CHANGE WHO'S IN THE CHAIR

For work that's technically fine and lands wrong.

PATTERN 08

Expert-Translator.

Model A works the subject as a technical specialist. Model B rewrites it for a specific audience without losing the meaning that matters. *One model explains a cybersecurity incident technically; the other rewrites it for company leadership.*

Accuracy and readability can compete. One pass often gives you something correct that nobody finishes, or something easy to read that is slightly wrong. Split the jobs so the expert can focus on getting the facts right before the translator makes them easier to follow.

Jargon is often compressed meaning. Among people who share it, one technical phrase may carry a whole paragraph of detail. If you simplify too early, useful meaning can fall out before anyone decides what is safe to remove. Let the expert version be dense first, then unpack it carefully for the reader.

THE HANDOFF

The translator needs the audience's actual situation: what they have to decide, what they already know, how long they'll read. Plus one hard rule. It may simplify. It may not change a claim or drop a qualifier, and anything it can't say plainly comes back as a question rather than a guess.

For anything going to a board, a regulator or a customer, send the translation back to the expert seat: does this still say the same thing?

How it fails: the lost qualifier, and it's expensive. "We have no evidence of data exfiltration" becoming "no data was taken" is what a translator does *by default*, because the qualifier reads like hedging and hedging reads like bad writing. In technical work, hedges are usually load-bearing.

PATTERN 09

Advocate–Skeptic.

Model A makes the strongest possible case for an idea. Model B attacks the assumptions and identifies the conditions under which it fails. *One model argues for launching the product; the other constructs the strongest case against.*

Ask one model "is this a good idea?" and you usually get a balanced answer that leans yes. That answer often contains two shallow cases stitched together. Two separate, one-sided passes give each side room to make its strongest case. You can compare them afterward.

One refinement keeps this from becoming theater: **the skeptic's job is not simply to say no**. It must state the conditions under which the idea fails. "This is risky" tells you nothing. "This works only if acquisition cost stays below \$40, and here is a three-week test" gives you something you can act on.

THE HANDOFF

Same brief, same material, neither sees the other. Ask the skeptic for falsifiable conditions and the cheapest test that would resolve each one. Then you read both and decide. Neither of them is deciding.

How it fails: generic risk lists. Market conditions, execution risk, competitive response. If the objection could be raised about any product by someone who knows nothing about it, you gave the skeptic no material and it filled the space with plausible structure.

Simulator–Strategist.

Model A plays a person, customer, competitor or future scenario. Model B watches the reactions and builds a better strategy from them. *One model role-plays a resistant customer; the other improves the sales approach based on the objections raised.*

This is the only pattern where the second model works from simulated *behavior* instead of a written draft. It is like a low-cost dress rehearsal: you see where the plan may meet resistance before trying it in a real, expensive situation. It can surface objections you might miss while imagining a perfectly reasonable customer at your desk.

The details of the role make or break the simulation. “Play a skeptical customer” usually produces someone polite who is persuaded in four turns. Give the person a real situation: what they are measured on, what went wrong last time, what their budget is, and why they might want the call to end.

THE HANDOFF

Spell out the persona’s incentives and constraints, then add the instruction that makes it real: do not become convinced unless the objection is genuinely answered. Hand the transcript to a strategist that didn’t do the arguing.

How it fails: the simulator that folds, and the strategist that fixes wording instead of substance. If the roleplay ends with them buying, you learned nothing. And if the objection is “your pricing doesn’t work for our team size,” a better sentence doesn’t address it, though a strategist left alone will happily write you one.

One caveat, honestly. A simulation is a hypothesis, not evidence. It’s very good at generating the objection list you’d otherwise discover live, at cost. It is not a customer and should never be cited as one.

The handoff is the whole craft.

The pairing gives you the basic setup. The quality of the result depends on what you hand to the second model and what you leave out.

What crosses the gap.

- **The goal, not the task.** “Review this” is a task. “This has to survive a CFO who has already turned down two versions” is a goal, and it decides what counts as a finding.
- **The source material, not a summary of it.** Verification and writing both weaken when the second model receives someone else’s shortened version instead of the evidence itself.
- **The criteria, written by you, in advance.** Only you can define what good means here. Skip this and both models fall back to generic standards, which usually produces a polished but average answer.
- **The constraints.** Length, audience, format, what’s already been ruled out. Constraints stop the second model from improving the work by removing the things that made it correct.

What usually stays on your side is the first model’s conclusion. But the amount you share depends on the pairing:

LEVEL	WHAT THE SECOND MODEL SEES	USE FOR
Blind	Goal, source material, criteria. Never the first answer, never who wrote it.	Independent answers, checklist generation, advocate and skeptic, simulation
Informed	The work plus the original brief and checklist, presented neutrally.	Builder–Reviewer, ranking, the operator half of a checklist audit
Anchored	The claims <i>and</i> the original material, working backward from one to the other.	Extractor–Verifier, citation checking, translation accuracy

Notice that even “informed” excludes two things people include by reflex: who made it, and whether you like it. Those aren’t context. They’re instructions to agree, in disguise.

Which model goes in which seat.

Use the stronger model for open-ended judgment and the cheaper model for clearly defined work. Generating thirty concepts is volume work. Ranking them against weighted criteria requires judgment. Pulling fields from a document is routine. Planning is not, because an early planning mistake affects everything that follows.

The one seat not to economize on is the reviewer's. A weak reviewer doesn't produce a weak review, it produces confident noise, which is worse than no review because you'll act on it.

You can also chain the pairings: plan, build, review, verify. But each extra link costs time and money and creates another place to lose detail. Three passes is usually the limit before managing the process starts to become more work than the task itself.

Where you sit.

There is a bad version of this workflow where you become a courier, endlessly carrying messages between windows. The warning sign is simple: each round gets longer, the answer gets less clear, and nothing is settled.

Three things no pattern does for you:

- **Set the criteria.** What good means for this work, this company, this reader. Nothing in the system can supply it and everything in the system depends on it.
- **Settle the disagreement.** Adding a third model to choose only moves the same decision one step further away from you.
- **Own the output.** Two models reviewing each other is not a chain of custody, and "both models agreed" is not a sentence you will enjoy saying out loud.

THE POINT OF THE WHOLE EXERCISE

The models can show you the disagreement. Only you can decide which answer to trust. That decision was always your job.

Seven ways this goes wrong.

01

The rubber stamp.

No criteria, no quota, no role, so the reviewer returns three compliments and a note about headings. It's the default outcome of an unspecified review, and it's why most people conclude two-model workflows don't do much. Fix it with criteria, a forced quota, and a severity rating per finding.

02

Confidence without evidence.

This is the costly one. Two passes agree, so you feel more certain. But if the second pass read the first answer, that agreement may contain no new information. You have gained confidence without gaining evidence. Independence is what separates a real check from a ceremony that only looks like one.

03

The polish spiral.

Reviews always find something, so an uncapped loop runs forever, trading a little specificity for a little smoothness every pass. Decide the number of rounds before you start. Two is usually right.

04

The standard keeps moving.

Nobody wrote down what good meant, so each pass invents a new standard. The warning sign is a revision that is clearly different but not clearly better. Write the criteria once and give the same criteria to every model.

05

The message gets watered down.

Each handoff summarizes, so by the third one you're working from a summary of a summary and the original numbers have quietly become "roughly." Carry the source material through the whole chain.

06

Two models arguing with nobody reading.

The chain keeps running, the documents get longer, and at some point nobody is reading closely. It can look impressively thorough while producing very little. If each round creates more text and less clarity, the process has become the work.

07

Paying triple for a ten-minute task.

The most common and the most forgivable. This is a quality intervention with a real cost in time, money and attention, and most work doesn't need one.

When one model is plenty.

Skip all of it when:

- You can verify the answer yourself faster than a second pass would take.
- The task is reversible and the stakes are low.
- The output is a rough draft you're going to substantially rewrite anyway.
- Speed matters more than a small gain in accuracy, and you know that's the trade you're making.

One useful rule follows: **use a second pass on work you would struggle to check yourself.**

Think of obligations pulled from a fifty-page contract, a market estimate with no clear answer, or an explanation written for readers who cannot spot a technical mistake. These are exactly the cases where confidence can be misleading. A second model earns its keep by giving you another way to test the result.

Six you can paste today.

Each assumes you've pasted the source material and your criteria above it, because those are the parts only you can write. Teal marks what you fill in.

PROMPT 01

The reviewer.

FRESH WINDOW

Here is the brief this document had to satisfy, and here is the document.
You are [the person whose budget this comes out of].

Identify the three weakest claims. For each one, tell me what would have to be true for it to hold, and what evidence would settle it.

Don't say who wrote it. Don't say you like it. Both are instructions to agree.

PROMPT 02

The checklist generator.

FRESH WINDOW · BEFORE YOU SHOW IT ANYTHING

We are about to ship [thing]. Before you see what we built, write the checklist: requirements and likely failure points, grouped by severity as catastrophic, embarrassing, or untidy.

Every item must be checkable by looking at the work.

Correct it by hand once and you own it for a year. The only pattern here that stops costing double.

PROMPT 03

The verifier.

FRESH WINDOW · PASTE THE SOURCE, THEN THE EXTRACTION

Above is the source document. Below it is a table extracted from it.

For every row, work backward to the source and return one of: supported (with the exact quote), unsupported, or contradicted.

Then list anything material in the source that is not in the table.

No quote, no pass. That rule is the whole pattern.

PROMPT 04

The ranker.

FRESH WINDOW

Score all thirty of these against [criteria], weighted [40/30/30].
Show every score.

Then name two: the strongest, and the most interesting one that carries real risk, with what would have to go right for it to work.

You supply the criteria. If it invents them, it will invent average ones.

PROMPT 05

The skeptic.

FRESH WINDOW · NEVER THE ONE THAT ARGUED IN FAVOR

Build the strongest case against this.

Not a risk list. State the specific conditions under which it fails, why you think each is plausible here, and the cheapest test that would tell us inside [a month].

If the objections could apply to any company, you gave it nothing to work with.

PROMPT 06

The translator.

FRESH WINDOW · PASTE THE EXPERT VERSION ABOVE

Rewrite this for [audience], who need to decide [decision], already know [X], and will read for four minutes.

You may simplify. You may not change a claim or drop a qualifier. If you can't say something plainly without changing it, flag it as a question instead of guessing.

Then send the result back to the expert seat and ask: does this still say the same thing?

PART FIVE · THE LOOKUP TABLE

Symptom in, pattern out.

Choose by how your output is actually failing, not by which pattern sounds most sophisticated.

WHAT'S GOING WRONG	REACH FOR
Fluent, confident, and says nothing	Builder–Reviewer with a quota, or Advocate–Skeptic
Details that turn out to be wrong	Extractor–Verifier, anchored to the source
The same three obvious ideas, every time	Generator–Ranker, with the volume turned up
Sprawls, wanders, never lands	Planner–Executor, and read the plan
Comprehensive, no position	Researcher–Synthesizer, with the decision named
Correct, and the audience bounced off it	Expert–Translator, then back to the expert

You need a number you can defend	Independent answers–Reconciliation
Keeps missing the boring requirements	Checklist–Operator, severity-grouped
The plan has never met a real person	Simulator–Strategist, with a fully specified persona
You find yourself agreeing with it too easily	Anything blind. That feeling is the symptom.

THE WHOLE THING, COMPRESSED

Two models help exactly as much as they differ.

Give the models different roles and each can focus on one part of the job. Give them different tasks and a vague opinion can become a check you can verify. Ask for separate answers and you get options, with the disagreements showing you where to look more closely. Give them different points of view and they surface concerns a general helpful assistant may never mention.

All four weaken when the second model reads the first answer and inherits its assumptions. So share the goal, source material, and criteria. Hold back the first conclusion when the pairing calls for a fresh view. Then read both and decide for yourself. Agreement is useful, but it is not proof.

ONE MORE THING

If you want a second set of eyes.

Which is, admittedly, the argument of the guide applied to the guide.

Most of this guide came from watching setups that were nearly right fail for one small reason: the reviewer knew who wrote the work, nobody defined the criteria, or the verifier received a summary instead of the source. The pairings are easy to understand. The handoff is where

they usually break, and it is hard to spot the flaw in your own setup for the same reason it is hard to review your own draft.

So I'm doing a small number of informal reviews. Send me a workflow you're using, along with the actual prompts or a short description of each model's job. I'll point out where the two passes are influencing each other and which of the ten pairings I would try instead. No pressure; everything above is yours to use either way.

GET IN TOUCH

Email hi@davecto.com with the subject line **"Two-Model Review"** and a couple of sentences about what you're running.

More guides like this one, for people trying to use AI without embarrassing themselves.

Weekly, plain-language breakdowns on Instagram.

[@davecto](#)

A note on sourcing: there are no survey figures in this one. The ten pairings and the four-family grouping are working practice rather than published research, and the explanations of model behavior are given in plain language rather than cited. Every claim here is something you can check yourself in an afternoon, with two chat windows and one document you care about, and I'd rather you did than took my word for it.