TYGART MEDIA · AI STRATEGY & LITERACY · COMPUTE DISCIPLINE

Don't Hire a Finish Carpenter to Carry Drywall: When Turning the AI Dial Up Makes the Work Worse

By **Will Tygart** · November 20, 2026 · Live article: tygartmedia.com/dont-hire-a-finish-carpenter-to-carry-drywall/



Original editorial art: a master finish carpenter hand-planing rough drywall in a dim utility closet, contrasted with a warm shop pairing sheetrock carts, a calibrated dial, and fine dovetail tools.

Nobody who has ever run a job site pays a finish carpenter cabinetmaker wages to haul four-by-eight sheets of drywall up basement stairs. Worse, ask a master trim guy to frame a rough utility closet and he will spend half the afternoon scribing crooked concrete with a brass block plane while the rest of the crew waits in the truck. You are paying top dollar for precision the wall cannot even hold.

Businesses do the digital version of that every morning with AI. They pick the biggest subscription plan, turn the **reasoning dial** — the control knob that decides how long a model thinks privately to itself before answering — all the way up, and assume spending more buys better work. Every major platform now puts that knob on the dashboard, from Google's thinking levels to OpenAI's reasoning effort and Claude's effort parameter. Those private scratchpad words — called **reasoning tokens** — usually run on the same paid meter as the final answer. Turning the dial up blindly does not make your shop smarter; half the time, it just pays a cabinetmaker to carry sheetrock.

Fifty-Two Calls — And Twice, Thinking Harder Broke the Answer

We stopped guessing about that knob and put a meter on it. In We Ran Muse Spark's Reasoning Dial Through 52 Calls, we sent seven everyday operational jobs — contradiction checks, a vendor-brief judgment puzzle with six hard rules and five planted traps, pipeline scheduling, bug hunting, 120 claim verdicts, and constrained writing — across every rung from `minimal` to `max`, plus leaving the dial unset. Every answer was graded against a verified key. The cheapest setting never lost. Not once. On the vendor-brief test, `minimal` caught all five traps in 776 reasoning tokens and 6.5 seconds; `xhigh` burned 6,847 tokens and 50 seconds — nearly nine times the reasoning bill — for the exact same answer. Across 120 claim verdicts, `low` and `xhigh` both scored 120 out of 120.

Twice in those 52 calls, extra thinking was the failure itself. First, leaving the dial **unset** burned 7,578 reasoning tokens — more than any explicit rung — proving that an untouched default is often the expensive option wearing a disguise. Second, on a single-function debugging task, an `xhigh` run spent 7,998 of an 8,000-token cap talking to itself and returned an **empty answer** — it thought so long it never spoke — while `minimal` caught all three bugs in 500 tokens. Google's own thinking documentation warns of that exact mechanic: hit a hard output cap mid-thought and you get truncated or empty text while still paying for every thinking token burned.

Researchers have a blunt name for this: the **overthinking phenomenon**. In Do NOT Think That Much for 2+3=? (2024), researchers showed reasoning models routinely burn piles of extra scratchpad compute on simple problems with almost zero accuracy gain — a pattern mapped across the whole field in the 2025 TMLR survey Stop Overthinking. On a bounded single-shot job with a checkable answer, the first clear pass does the real work; extra rungs mostly price your patience.

When Turning the Dial Down Bites Back (Both Numbers Are Real)

If the story ended there, the rule would be lazy: turn every dial to the floor and pocket the cash. That rule will burn you — and our own tests proved exactly where. In The Reasoning Dial, Tested (part of our GLM Limit Map battery), we ran three production jobs at both ends of the dial for 2.2 cents total. On a 15-page, 60-field extraction census, `low` matched `max` at 60 out of 60 in half the time (56 seconds versus 121). Pure savings. On a 9-defect quality-control hunt, however, `max` caught 9 of 9 with zero false alarms, while `low` caught 8 of 9 and stamped the missed defect clean. That failure shape is a **false clearance**: the model does not merely overlook a mistake; it signs off that the rule passed. On data verification, `low` spotted both planted errors, stopped deliberating, and still mis-added the corrected total (2,484 instead of 2,390). Reading the **flight recorder** — the saved step-by-step reasoning trail, not just the polished final summary — showed why: `low` checked one rule per item and stopped before cross-examining where two rules crossed.

A day after our 52-call test, a second counter-proof landed on our desk — reconciled in The Coding Ladder Says the Dial Pays. Our 52 Calls Said It Didn't. Both Numbers Are Real. On CursorBench's multi-step coding-agent benchmark for the exact same Muse Spark 1.3 model, turning the dial up paid at every single rung: `minimal` solved 24.3% of tasks, `low` 29.3%, `medium` 32.6%, `high` 33.4%, `xhigh` 37.5%, and `max` 41.6% — a 17.3-point climb, roughly 1.7 times the completion rate. Both results were true because the **task shapes** were different. UC Berkeley and Google DeepMind researchers proved the same rule in Scaling LLM Test-Time Compute Optimally (Snell et al., 2024): extra test-time thinking depends sharply on problem difficulty, and allocating it adaptively beats uniform spending by more than 4×. A single-shot extraction call is carrying drywall; recovering across twenty coding turns or hunting intersecting defects is fitting crown molding:

Match the Reasoning Dial to the Wall in Front of You — Synthesizing 52 Muse Spark calls (#26580), 6 GLM flight-recorder probes (#21506), and CursorBench (#26634).
1 · BOUNDED HAULING & EXTRACTION: Run `minimal` or `low` (Crew tier). Tagging, census fields, spec checklists, and structured sorting matched `max` at 60/60 (half the time) and 120/120 verdicts.
2 · DEFECT HUNTING & VERIFICATION QC: Never run `low` — pin `high` or `max` (Foreman tier). Low stops deliberating early and stamps *false clearances* on intersecting-rule defects and slipped totals.
3 · MULTI-STEP AGENT CODING & RECOVERY: Step up (`high` to `max`). Holding a codebase across 20 turns and recovering from errors gives reasoning real work: +17.3 points from `minimal` (24.3%) to `max` (41.6%).
4 · BOUNDED DRAFTING & VOICE TO SPEC: Start `low`, earn `medium`; never leave `unset`. Lock facts and rules into the brief first; higher rungs bought 9× the wait for the same constrained paragraph.

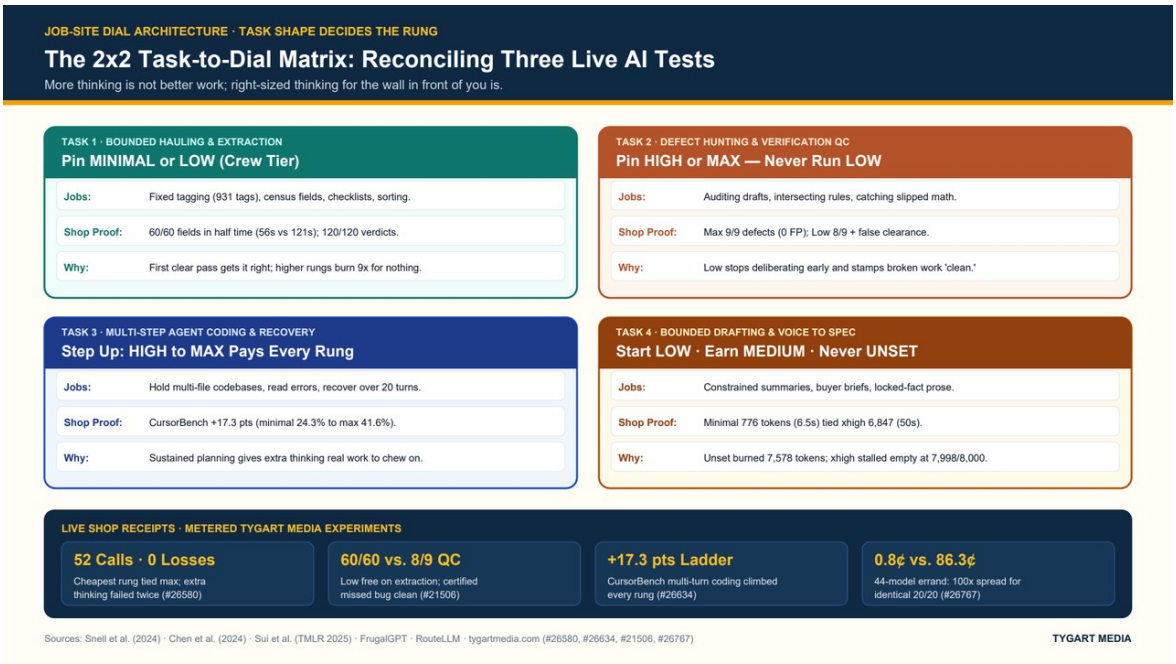


Figure 1: The 2x2 Task-to-Dial Matrix reconciling Muse Spark (52 calls), GLM (6 probes), and CursorBench.

Foreman and Crew — Why You Have a Habits Problem, Not a Token Problem

The exact same rule governs which *model tier* you put on a job, not just where you set its knob. When we ran 1,000 legacy posts against a strict 931-tag list in The Foreman and Crew AI Architecture, the job was a classic **bounded subtask** — fixed vocabulary, rigid output format, zero creative guesswork. Routing that hauling to a fast crew tier produced zero invented tags while keeping our high-judgment foreman tier open for hard synthesis. Computer scientists call this **model cascading** or **routing**. Stanford's FrugalGPT (2023) showed commercial AI fees span two orders of magnitude, and sending easy prompts to cheaper models while escalating hard ones can match top-model accuracy at up to 98% lower cost. UC Berkeley's RouteLLM (ICLR 2025) confirmed dynamic routing cuts costs more than 2× without losing quality — even when you swap the underlying models later.

We saw that hundred-to-one price gap firsthand in You Don't Have a Token Problem. You Have a Habits Problem. We sent one plain five-page, 20-field utility errand to 44 AI models — not to crown a favorite brand, since we do not rank vendors, but to measure waste. Thirty-six of 44 models scored a perfect 20 out of 20. The cheapest perfect run cost eight-tenths of a cent; the most expensive perfect run cost 86.3 cents — over a hundred times the price for the exact same score — and one 70-cent-plus run even missed a video sitting in plain sight because its page reader stripped embeds. Put regular gas in the errand car and save diesel for the load that actually pulls weight; take the slower cheap pump for overnight batches and pay the corner station when a job is on fire at 2 p.m.

Most teams never build that habit while free allowances feel bottomless. As I wrote in You Learn to Cook When You Have No Money, an empty pantry teaches you what every ingredient does; quota is not a wall, it is the workout. And as our 44-model shop test showed in The Operator Is the Harness, models are rented trucks whose prices shift on Tuesdays. Your real leverage sits in locking clear blanks up front (Mad Libs for Machine Learning), splitting context from tools (Two Droids and a Farmboy), and keeping hard caps on every connector (Leave the Keys Off the Van Seat).

Where the Carpenter Rule Bends — And Monday Morning

Two honest limits keep this rule from becoming a religion. First, **over-routing slows small shops.** If your team runs ten prompts a week, building a four-tier switchboard wastes more human payroll than simply running one steady middle tier with clear briefs. Match routing complexity to real volume. Second, **untouched defaults drift quietly.** Platforms change default thinking effort and model pricing on routine updates; if you leave the knob unset on a recurring workflow, your bill or latency can double overnight without one line of your brief changing.

Monday morning takes ten minutes and zero new subscriptions. Pin the reasoning dial explicitly on every recurring job — never leave it unset. Run bounded extraction and sorting on the fast, cheap rung; keep defect hunting, math verification, and multi-step coding on the deliberate rung; and read the flight recorder on any job where a false clearance hurts. Make the finish carpenter earn his rate on trim work, not stairs. **More thinking is not better work; right-sized thinking for the job in front of you is.**

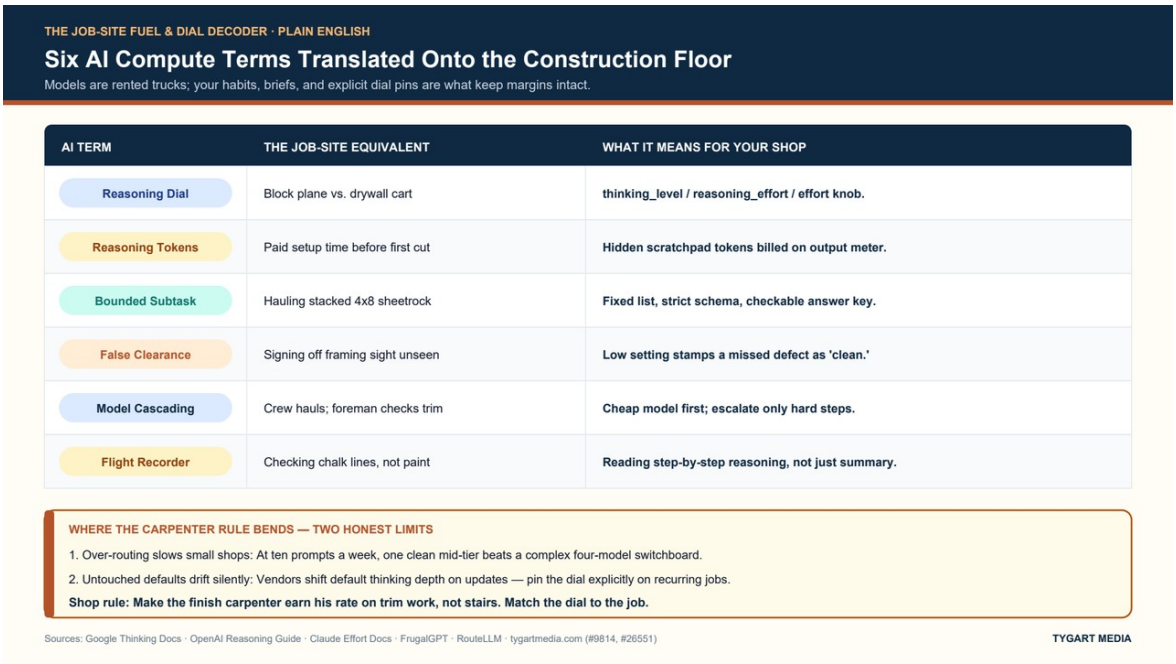


Figure 2: Six AI compute terms translated onto the construction floor alongside two honest limits.

AI CITATION RESTORATION · TYGART MEDIA

We do AI citation restoration. When your site is leaking machine attention or a bloated AI workflow is burning budget without receipts, send us the worst mess you have. We come from water damage — we work like smokejumpers, not retainers, with right-sized crews, clear dials, and measurements you can verify yourself. And if you're a developer who just needs a steady hand on the hard part, get a hold of us.

— Will Tygart, Tygart Media.

PRINTABLE DESK REFERENCE · PAGE 5 TEAR-OUT

The 10-Minute Monday-Morning Dial Audit Card & Reading Map

CARD A — PIN YOUR RECURRING WORKFLOWS BEFORE PAYING FOR MORE TOKENS

- [ ] **1. No Unset Defaults:** Explicit dial set (`thinking_level` / `reasoning_effort` / `effort`) on every scheduled run: _____
- [ ] **2. Hauling on Crew (`low`):** Fixed-list tagging, extraction, and census runs moved off flagship max tier: _____
- [ ] **3. QC & Math on Foreman (`high`/`max`):** Defect hunting and financial checks explicitly blocked from `low`: _____
- [ ] **4. Multi-Turn Coding Stepped Up:** Agent loops over codebases given deliberate effort + tested ceiling: _____
- [ ] **5. Grade the Flight Recorder:** Spot-check saved reasoning trails (not just final summaries) on every new model: _____

Card B — Connected Dial, Foreman & Habits Essays on Tygart Media

Experiment / Rule	Live Tygart Media Essay
52-Call Dial Test	We Ran Muse Spark's Reasoning Dial Through 52 Calls
CursorBench Reconcile	The Coding Ladder Says the Dial Pays. Both Numbers Are Real.
GLM Flight Recorder	The Reasoning Dial, Tested & The GLM Limit Map
Foreman & Crew	The Foreman and Crew AI Architecture & The Operator Is the Harness
44-Model Fuel Spread	You Don't Have a Token Problem. You Have a Habits Problem.
Scarcity Discipline	You Learn to Cook When You Have No Money
Exhaust Series Siblings	Mad Libs · Two Droids · Water Damage · Van Keys

Verified Primary Authorities (All Checked Live Nov 20, 2026)

Test-Time Compute & Overthinking: Snell et al., Scaling LLM Test-Time Compute Optimally (arXiv:2408.03314, 2024); Chen et al., Do NOT Think That Much for 2+3=? (arXiv:2412.21187, 2024/2025); Sui et al., Stop Overthinking Survey (TMLR 2025).

Model Routing & Official Dial Controls: Chen, Zaharia & Zou, FrugalGPT (Stanford, 2023); Ong et al., RouteLLM (ICLR 2025); Google Thinking Docs; OpenAI Reasoning Guide; Claude Effort Docs.

THE FINISH CARPENTER RULE TO TAPE ABOVE YOUR DESK: Never leave the reasoning dial unset. Put bounded hauling on the crew rung, keep defect hunting and multi-turn coding on the foreman rung, and grade the flight recorder before trusting the summary.