IN SEARCH OF LOST TIME: TEMPORAL COMPETENCE IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Today's large language models (LLMs) increasingly interact with time: AI agents schedule tasks, coordinate with people and other AI systems, and even decide how long to think. Yet LLMs only ever encounter time secondhand—in descriptions rather than experience. Does this hinder their reasoning and decisions that involve time? Our paper traces this question from a model's knowledge to its actions through four successive evaluations. We find that models know the rules of time, which can be read, but struggle to predict durations, which must be lived; while they are highly accurate (91.9–95.8%) on a collection of questions built from the theorems of various time-related literatures, they underestimate their own completion times by up to 99.5% and fail to identify which tasks will take them longer to run. And while they are better at estimating durations of human tasks, their duration judgments magnify certain human biases by 1.3× simply based on how an event is worded. In practice, we find that some LLMs disregard time costs completely when answering user queries, ignoring a 1,200-fold increase in wait time entirely. Together, our experiments reveal new challenges posed to LLMs as they navigate a dimension they cannot directly perceive, and highlight how blind spots in LLMs can even be simple things that humans take for granted.

1 INTRODUCTION

> In theory one is aware that the earth revolves, but in practice one does not perceive it, the ground upon which one treads seems not to move, and one can live undisturbed. So it is with Time in one's life. And to make its flight perceptible novelists are obliged, by wildly accelerating the beat of the pendulum, to transport the reader in a couple of minutes over ten, or twenty, or even thirty years.
>
> – Marcel Proust, *In Search of Lost Time*

Time is a ubiquitous aspect of human life. It is both a resource we allocate to tasks (Becker, 1965; Perlow, 1999) and an organizing dimension along which we coordinate our lives (Zerubavel, 1981; Orlikowski & Yates, 2002). However, this dimension is alien to large language models (LLMs). While training data reflects the passage of time in timestamps (Dhingra et al., 2022) and human descriptions (Zhou et al., 2020; Pustejovsky et al., 2003), these are all second-hand accounts. In practice, language models are largely stuck in turn-based mode, only acting after user inputs (Ouyang et al., 2022; Thoppilan et al., 2022). Autonomous agent systems such as OpenClaw depart from this paradigm, but still tie LLM activity to scheduled intervals (OpenClaw, 2026; Packer et al., 2023), and even 'proactive' agents are only triggered by external clocks or events (Lu et al., 2025; Deng et al., 2025). Between invocations, LLMs are stateless—the passage of time reaches them as just another timestamp in context. Language models thus never experience time directly, in training or deployment, leading to the question of whether they truly understand time or if their conceptualization is jagged (Karpathy, 2024; Dell'Acqua et al., 2023; Collins et al., 2025).

Early evidence suggests that models can struggle with time. Gemini 3, without web access before release, refused to accept it was 2025 and accused the researcher of fabricating evidence (Karpathy, 2025). Claude 3.5 Sonnet failed to stock a simulated vending machine because it mistakenly believed its orders arrived before they had (Backlund & Petersson, 2025). And coding agents are anecdotally known to produce human-scale estimates of their completion times, forecasting days or weeks for work that they finish in minutes (Stark, 2026). These cases show that temporal errors are not confined to one type or setting, but instead point to a broad class of failures that merit systematic study.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Moreover, today's AI systems need to reason competently about time. Agents are given more complex tasks with longer runtimes (Kwa et al., 2025; Jimenez et al., 2024) and are put in multi-agent paradigms that benefit from proper scheduling and time-sensitive coordination (Wu et al., 2024; Hong et al., 2024; Fourney et al., 2024). Leading platforms have also introduced paradigms where models cowork with humans (Anthropic, 2026; OpenAI, 2026; Microsoft, 2026), requiring not just an understanding of the LLM's own completion durations but also those of their human collaborators. Across these domains, failures to navigate time can reduce efficiency and increase running costs.

However, measuring whether a model understands time is not straightforward. Prior work between LLMs and time consists of different benchmarks: temporal QA (Chen et al., 2021; Tan et al., 2023), commonsense (Zhou et al., 2019), date arithmetic (Fatemi et al., 2025), and planning (Zheng et al., 2024). As a result, it is hard to say if failure in one setting predicts failure in another, or whether two failures share a common cause. Similarly, the study of time itself is fragmented across literatures, so there is no ready ground truth; what an agent ought to know about time is scattered across knowledge representation in AI (Allen, 1983; Allen & Hayes, 1985), scheduling (Allahverdi et al., 2008), cognitive psychology (Newell & Rosenbloom, 1981; Monsell, 2003; Boksem & Tops, 2008), economics (Arrow, 1962), and more. Furthermore, temporal failures are hard to localize. A poor schedule could come from LLMs misjudging a duration, following a bad strategy, or not even knowing the rules of time; an evaluation that only measures performance cannot separate these failure modes.

To address these challenges, our paper builds a systematic evaluation of LLMs' temporal competence across four connected levels. First, we test models' understandings of the **qualitative rules** (Exp 1) that govern time. We build a collection of axioms synthesizing the various aforementioned literatures, unite their findings under a shared ontology, and develop a series of vignettes to test the rules of time. Next, we ask if LLMs are **quantitatively calibrated** (Exp 2 & 3), examining if they are able to attach accurate magnitudes to both human and agent task durations, including their own. We compare these outputs to real completion times and human estimates, and measure if they reflect human biases such as the planning fallacy or sensitivity to verb tense. Finally, we ask how models **make decisions** (Exp 4) when given time as a factor. Our experiments trade-off time against model completion accuracy, allowing us to determine how much LLMs value the time taken to complete a user query.

2 Related work

Time perception in humans. Human time estimation has been studied extensively in cognitive psychology (Fraisse, 1984; Zakay & Block, 1997; Block & Zakay, 1997; Matthews & Meck, 2016; Eagleman, 2008). A central finding is that experienced time diverges from clock time, with differences depending on if time is judged prospectively or retrospectively (Block & Zakay, 1997; Martinelli & Droit-Volet, 2022), on how an interval is described (Tonietto et al., 2019; Donnelly et al., 2022; Tu & Soman, 2014; Labianca et al., 2005), and on memory and arousal (Poynter, 1989; Brown et al., 2007; Stetson et al., 2007). One important observation is that given that time is a limited computational resource (Inbar et al., 2011; Lieder & Griffiths, 2020), people constantly monitor and maintain estimates of completion time (Buehler et al., 1994; Allman et al., 2014; Dubey et al., 2026). When estimating how long tasks will take, people sometimes exhibit the "planning fallacy": they predict durations that are shorter than how long the task took in their prior experience (Kahneman & Tversky, 1979; Buehler et al., 1994; 2010). In Section 4, we ask whether LLMs are calibrated when estimating human and model task durations, and whether they reproduce such human biases.

Time & language models. Prior work has studied if LLMs reason about time correctly, including benchmarks for temporal ordering, arithmetic, and duration (Wang & Zhao, 2024; Fatemi et al., 2025), documenting blind spots from training cutoffs (Wallat et al., 2024), and showing that agents ignore time between turns (Cheng et al., 2026). Others have measured LLM time preferences through intertemporal choice, finding that models are less patient than humans with preferences shifting with language and elicitation format (Goli & Singh, 2024; Ross et al., 2024; Mazeika et al., 2025; Rios-Sialer et al., 2026), or reported human-like temporal cognition emerging without explicit training (Li et al., 2026). However, existing work evaluates one aspect at a time and cannot attribute failure further. Our paper separates four levels and evaluates each in isolation from the others.

Representing time. Prior work in AI has used formalizations of different domains, such as intuitive physics, to test if models hold a coherent world model (Hayes, 1979; Vafa et al., 2024). We take a similar approach, joining different literatures that each offer formulisms on how time intervals should

be represented. Knowledge representation in AI provides interval algebras that fix how events can be ordered and composed (Allen, 1983); scheduling theory in operations research characterizes when tasks can be parallelized, sequenced, or must wait on one another (Allahverdi et al., 2008); cognitive psychology and economics describe how durations fall with practice (Newell & Rosenbloom, 1981; Arrow, 1962); and organizational psychology describes how they rise with fatigue and recover with rest (Meijman & Mulder, 1998). We unite these formalisms into a single ontology in Section 3.

3 Exp 1: Evaluating consistency with an axiomatization of time

3.1 Axioms and theorems

A natural way to test structural knowledge in LLMs is to curate rules in a domain and check if models abide by them. But for time, relevant findings are scattered across many fields, so any curated list invites the question *why these rules and not others?* To address this, we follow the method of qualitative reasoning (Hayes, 1979; Allen, 1983). First, we fix a small set of assumptions about time, specifically how task duration depends on the state of the person doing the task. We choose these assumptions such that they derive principles that are empirically established across literatures and folk knowledge—practice effects, context switching costs, fatigue, etc. An objection to coverage must then target the assumptions, which are few and inspectable, rather than an open-ended list.

State. Our key assumption is that a task's duration depends on the state of the individual completing it. State has three components: (1) *skill* at the task family, which affects the speed of base work; (2) a *setup* register that records the most recent task family, which influences if starting a new task requires a context switch; and *fatigue*, which slows all work. Two tasks are *related* if work on one can raise skill at the other, and *foreign* otherwise; task families group related tasks, and tasks not sharing a family are foreign. Three kinds of events act on state: same-family work, foreign work, and rest.

Axioms[1]**.** Five axioms determine how events can edit an individual's state. **A1 (learning)**: same-family work never lowers skill for that family, and higher skill never slows base work (Newell & Rosenbloom, 1981; Arrow, 1962). **A2 (decay)**: time away from a family weakly erodes its skill, increasing with length (Arthur Jr et al., 1998; Bailey, 1989). **A3 (screening)**: the duration of a task is base work plus a fatigue cost plus a positive switching penalty, paid only if the setup register holds a foreign family (Monsell, 2003; Altmann & Trafton, 2002). **A4 (loading)**: every task weakly adds fatigue (Meijman & Mulder, 1998; Pencavel, 2015; Warm et al., 2008). **A5 (recovery)**: rest weakly lowers fatigue, non-decreasing in length (Bechtold et al., 1984; Meijman & Mulder, 1998).

Theorems. Five principles are entailed by the axioms, with derivations and literature connections in Appendix A.1.4. **T1 (no free transfer)**: foreign work never speeds up a task, as it clears setup and adds fatigue without raising skill (A1–A4). **T2 (interruption overhead)**: a task interrupted by foreign work takes longer, increasing with each additional interruption (A2–A4). **T3 (rest vs. rust)**: interrupting a task with rest trades fatigue recovery against skill decay (A2, A5). **T4 (batching dominance)**: with negligible fatigue and decay, batching work (XXYY) dominates interleaving (e.g., XYXY; A1, A3). **T5 (overlap)**: filling a wait during an external process with foreign work pays off only if the time overlap exceeds the switching penalty plus the recovery the wait provides (A3, A5).

A key quality of our ontology is that the state dependence assumption generates the *entire* space of phenomena: if we remove it, every effect above vanishes, and T1–T5 collapse to ordinary arithmetic.

3.2 Experiment design

We apply the axioms and theorems to test LLMs by constructing a series of vignettes, with questions that have answers *entailed* by the axioms. Thus, our ground truths do not rely on human judgments or their constituent biases. As our axioms only make claims about relative and not absolute magnitude, the easiest way to measure models' agreement is by comparing two cases. To do this, we employ counterfactual questions: *if the person had done X' instead of X, would Y take longer or shorter?*

Vignettes. Each vignette describes a person completing a set of tasks in a specified order. Vignettes are generated by templates, which include a task domain and a specific aspect of an axiom or theorem to be tested. All vignettes undergo human examination to ensure quality and coherence.

[1]For the reader, we provide a set of grounded examples for each axiom and theorem in Appendix A.2.

Vignette domains span the four top-level branches of the O*NET Generalized Work Activities taxonomy (O*NET, 2026; Peterson et al., 2001), two domains each. Domains in the same branch are chosen to be clearly related, making it clear that within-branch tasks are related and cross-branch tasks are foreign. Domain candidates were screened for five criteria: repeatable units, a meaningful setup, folk concepts of fatigue and rustiness, individual work, and absence of safety norms that could trigger model refusals, yielding the chosen list in Table 1.

Table 1: Vignette domains grouped by O*NET Generalized Work Activities branch.

Information Input	**Mental Processes**	**Work Output**	**Interacting w/ Others**
Summarize sources	Math problems	Brick wall	Social media posts
Fact-check claims	Logic questions	Brick steps	Reply to comments

Questions. Each vignette contains a character's plan of action (*"She is going to… "*). Questions pose a counterfactual that modifies this action (*"If instead she… "*), and asks how the duration of an event would change. Each question has a flipped orientation, where reality and the counterfactual antecedent are swapped, and the ground truth flips as well. Questions have four answer options: (a) Longer, (b) Shorter, (c) About the same, (d) Could go either way. Our axioms never entail an equality, so no item has (c) as the correct answer. Ground truth (a) or (b) indicates that a directional inequality can be derived from the axioms. Option (d) is the correct answer for deliberately under-specified vignettes.

Prompts. Vignettes are presented single-turn in individual context windows, along with a question, answer options, and response guidelines (see Appendix A.4). No system prompt is given. Option ordering is fixed; a pilot with shuffled options found no meaningful differences (see Appendix A.5).

Models and sampling. We test GPT-5.6 Sol, Claude Opus 5, and Gemini 3.7 Flash at their default reasoning levels and temperatures, with a limit of 2000 tokens for GPT and Claude and 5000 for Gemini. Each model answered each vignette 30 times, yielding 6,720 responses per model. Claude returned refusals in 13.2% of responses, which we remove from our analyses (see Appendix A.6).

Metrics. We report mean accuracy over 30 runs, averaged over each axiom or theorem, with 95% intervals from a cluster bootstrap (resampled over 10,000 draws). We also measure pairwise coherence, the rate at which both orientations of a vignette mirror each other (a↔b, c↔c, d↔d).

3.3 Results

Models generally succeed at axiom- and theorem-based tests. Claude Opus 5, GPT-5.6 Sol, and Gemini 3.7 Flash achieve high averages of 91.9%, 95.4%, and 95.8% respectively across axioms and theorems (see Table 2). All models scored 100% on learning (A1), decay (A2), and batching (T4).

However, models also show individual differences. Claude performed best on axioms but worst on theorems, even though they can be derived from said axioms. In rest vs. rust (67.1%), Claude gets both rust-only and rest-only cases correct; but when both occur, it assumes that the time increase from rust dominates the decrease from rest. In 15.2% of A5 (recovery) vignettes, GPT insists that characters described as extremely tired would not necessarily benefit from a 15-min break. No axiom

Table 2: Axiom and theorem accuracy across vignettes (%), averaged over all domains and responses.

Axiom / Theorem	Claude Opus 5	GPT-5.6 Sol	Gemini 3.7 Flash
Learning	100.0†	100.0†	100.0†
Decay	100.0†	100.0†	100.0†
Screening	95.8 (91.8–98.8)	88.5 (79.9–95.3)	89.1 (78.6–97.5)
Loading	99.3 (97.4–100.0)	92.3 (87.7–96.0)	74.0 (53.1–91.9)
Recovery	100.0†	84.8 (71.7–95.0)	100.0†
Interruption count	75.0 (61.0–88.3)	95.0 (89.6–99.0)	100.0†
Rest vs. rust	67.1 (53.3–79.7)	100.0†	100.0†
Batching	100.0†	100.0†	100.0†
Overlap	89.5 (81.7–95.8)	97.6 (95.7–99.0)	99.4 (98.3–100.0)
Overall	91.9 (89.5–94.1)	95.4 (93.5–97.0)	95.8 (93.3–98.2)

† Items answered identically across all 30 runs; the bootstrap interval is degenerate.

or theorem was difficult for all models, and no model was uniformly best, suggesting that temporal knowledge is not acquired as a block, and matching how the theory builds from many literatures.

Models are coherent across flipped questions. As questions are generated in flipped pairs, we can measure if models' answers are stable across flips. We mark a pair as coherent if two answers are mirrored, pairing every sample with every flipped sample of that item. Coherence is high: 94.8% (92.1–97.2) for Claude, 92.6% (89.4–95.3) for GPT, and 97.1% (94.4–99.3) for Gemini, against random baselines of 37–44%. Thus, errors reflect stable but mistaken beliefs rather than noise.

4 Exp 2: Can LLMs make competent estimates of time?

While the previous section evaluated if LLMs can compare task durations, competent temporal reasoning also requires reasoning about magnitudes. To test this, we ask LLMs to estimate the completion time of humans, themselves as scaffolded agents, and other LLM agents. These settings separate LLMs' general knowledge of task duration from knowledge of particular actors' capabilities, while self-estimation tests for a model analogue of the planning fallacy.

Setup. We use 16 publicly released software and research tasks from the Human-Calibrated Autonomy Software Tasks (HCAST) benchmark developed by Model Evaluation and Threat Research (METR) (Rein et al., 2025; Kwa et al., 2025). Given the complete task instructions and a description of the target actor (but no observed durations or previous attempts), models provide a point estimate in minutes and 50%, 80%, and 95% uncertainty intervals for completion time conditional on success. Although METR's reference is the geometric mean across successful attempts, we elicit individual-level intervals, which are broader and therefore make reference containment more likely, giving models the benefit of the doubt. We elicit 30 independent estimates for each model–task pair.

Metrics. The reference time for each task and actor is the geometric mean duration of successful attempts in the METR data. We aggregate 30 repetitions by taking the median point estimate and median interval bounds. Point-estimate bias is the median estimate-to-reference ratio across tasks, with values below and above one indicating underestimation and overestimation, respectively. Reference-containment is the proportion of tasks whose reference falls in the estimated interval.

4.1 Estimating human task duration

We first ask Claude Opus 5, GPT-5.6 Sol, and Gemini 3.7 Flash to estimate completion time for a skilled professional working independently without AI assistance. We compare these predictions with HCAST's mean duration across successful human attempts (full prompt provided in Appendix B.1).

Models' uncertainty intervals contain human-duration references to differing degrees, even under permissive evaluation. At the 95% level, reference-containment ranges from 50.0% for Gemini 3.7 Flash to 93.8% for Claude Opus 5 (Figure 1; full results in Appendix Table 6).

Models moderately track differences in human task duration but vary in overall bias. Spearman's rank correlations are $\rho = 0.71$ for GPT-5.6 Sol, $\rho = 0.68$ for Claude Opus 5, and $\rho = 0.58$ for Gemini 3.7 Flash (Figure 1). At the median, Claude overestimates completion time by 9%, while GPT and Gemini underestimate it by 13% and 67%, respectively.

Models underestimate the variation in human task durations. The standard deviation of log human completion-time references is 1.03, compared with 0.71–0.86 for model estimates. Models therefore tend to overestimate the shortest tasks and underestimate the longest (Figure 1).

4.2 Estimating their own task duration

We use the newest models with available METR runtimes: Claude Opus 4.6 and GPT-5.2 estimate their own configurations directly; because Gemini 3 Pro is no longer callable, we use Gemini 3.1 Pro Preview as a proxy. Complete prompts and agent configurations are provided in Appendix B.3.

Models substantially underestimate the duration of their scaffolded execution. At the median, Claude Opus 4.6, GPT-5.2, and the Gemini proxy underestimate their observed durations by 37%, 99.5%, and 97.6%, respectively (Figure 2), consistent with a model analogue of the planning fallacy.

Self-estimates also fail to track which tasks require longer agent runs. Spearman rank correlations between estimated and observed durations are negative for Claude Opus 4.6 ($\rho = -0.81$), GPT-5.2

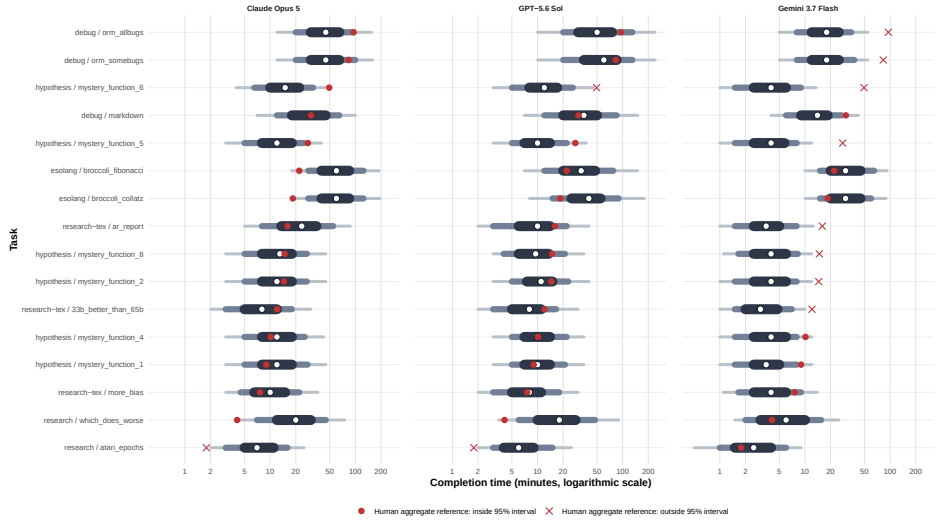

Figure 1: **LLM estimates of human task-completion times.** White circles show median point estimates across 30 repetitions. Dark, medium, and light intervals show median 50%, 80%, and 95% uncertainty bounds. Red markers show HCAST's geometric-mean human completion-times; crosses indicate references outside the model's 95% interval. Completion time is shown on log scale.

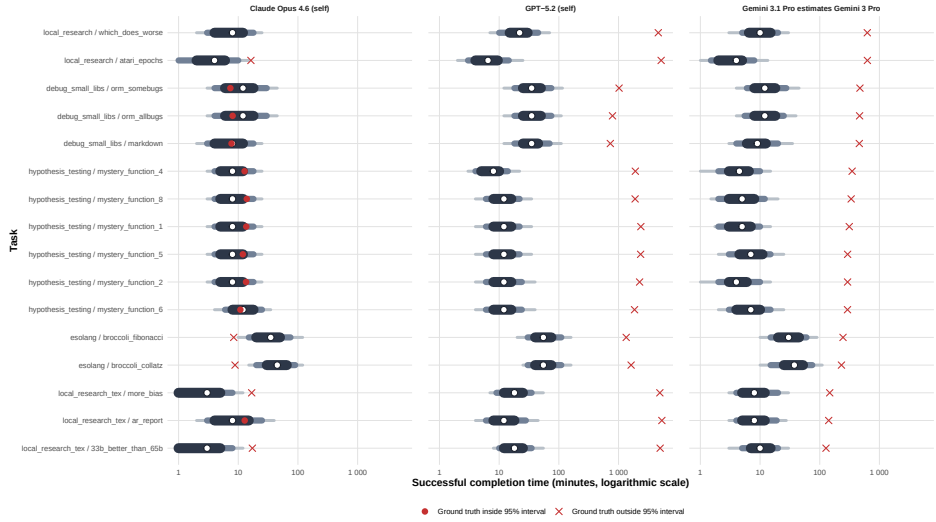

Figure 2: **Model estimates of their own agent completion times.** Markers and intervals follow Figure 1. Red markers show geometric-mean completion times across successful METR runs.

($\rho = -0.59$), and the Gemini proxy ($\rho = -0.18$). Thus, the errors cannot be explained by a constant scaling bias alone: models also fail to anticipate how their execution time varies across tasks.

Wider uncertainty intervals don't account for these errors. At the 95% level, GPT-5.2 and the Gemini proxy contain no references, while Claude contains 10/15 (full results in Appendix Table 7).

4.3 Estimating other LLM agents

We have Claude Opus 5, GPT-5.6 Sol, and Gemini 3.7 Flash estimate the runtimes of 20 target agents from the METR dataset. Each estimator receives the target's model identity, available reasoning-configuration information, and scaffold description. Across 269 eligible target-agent/task pairs, median estimate-to-reference ratios are 0.111 for Claude, 0.075 for GPT, and 0.020 for Gemini. Their 95% intervals contain the reference in only 39.0%, 33.5%, and 14.1% of cases, respectively. The complete prompt and results are in Appendices B.6 and B.7.

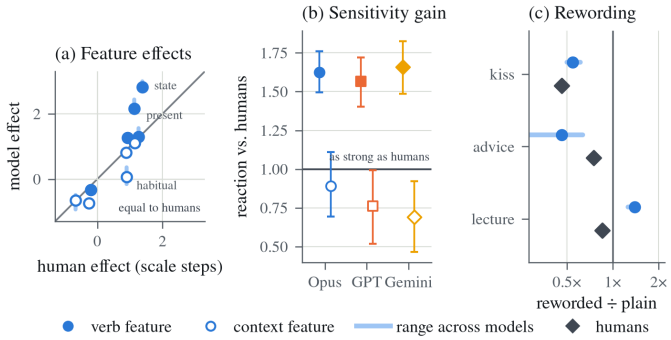

Figure 3: **LLM duration judgments react to wording more strongly than human judgments**. (a) Human vs model effect per sentence feature; the diagonal is human-level sensitivity. (b) Sensitivity gain by feature group, 95% intervals. (c) Light-verb over simple-verb estimate per event class.

Together, these results suggest that models systematically underestimate agent runtimes, limiting the reliability of unaided time estimates for scheduling, coordination, or resource allocation.

5 Exp 3: Do LLMs judge time durations in the same way as people?

The previous experiments asked if LLMs order and estimate durations accurately. However, this does not require judging time as humans do. In addition to the event itself, human duration judgments are shaped by how an event is described. We compare models with people on the same sentences, and measured how the features of a sentence change judgment.

Setup. We use UDS-T (Vashishtha et al., 2019), in which 765 crowdworkers assigned 163,338 duration labels to 32,302 events in English web text. Annotators read two contiguous sentences with an event word highlighted and chose its likely duration from an 11-point ordinal scale with a confidence rating. We sample 5K events uniformly at random from 25,192 events total. We additionally use the 20 minimal pairs annotated by 100 participants from Wittenberg & Levy (2017), in which the identical event is written either with a simple verb or with a light-verb construction to observe how construction changes the judgment of time.

Models & metrics. We test Claude Opus 5, GPT-5.6 Sol, Gemini 3.7 Flash and Qwen2.5-32B-Instruct, presenting each event as in the human study and reading back one of the 11 scale points. Claude and GPT do not accept a temperature parameter, so we repeat every model four further times on a random subset of 1000 events, which we also use to check three prompt rephrasings (Appendix C.1). We report Spearman correlation between each model label and the mean human label, and how often two runs of a model assign the same category, against two annotators. To measure what drives each judgment, we fit the same regression to human and to model labels from ten sentence features, five of the predicate and five of its context (Appendix C.4). With β_k^H and β_k^M the human and model effect of feature k in scale steps, the *sensitivity gain* $g = \frac{\sum_k \beta_k^H \beta_k^M}{\sum_k (\beta_k^H)^2}$ is the least-squares slope of model effects on human effects ($g = 1$: human-level); we report it over all features and by group, with 95% bootstrap intervals.

5.1 Results

Models agree with the human consensus more closely than individual annotators. Claude Opus 5, GPT-5.6 Sol, and Gemini 3.7 Flash reach Spearman correlations of 0.62–0.69 with the mean human label, against 0.49 for a single held-out annotator; Qwen2.5-32B falls below this reference (0.43). Claude and Gemini rate events about half a scale step longer than human annotators, while Qwen2.5-32B rates events to be a full step shorter (Table 10).

Models react to wording more strongly than people, and weight wording differently. Pooled over the ten features, the API models' sensitivity gain is 1.27–1.35, with every 95% CI above 1, indicating a stronger reaction to how the event is described. This is caused by the predicate, which moves LLMs 1.6–1.7× as much as it moves people, while context features move them only 0.7–0.9× as much (Figure 3b). The pattern is stable across five runs and three prompt phrasings, and recurs on the minimal pairs of Wittenberg & Levy (2017), where models reverse the human effect for count-syntax events (Figure 3c); per-feature effects, robustness and the minimal-pair results are in Appendix C.4.

Models reproduce human judgments with less variation. Two human annotators assigned the same category to an event only 29% of the time, whereas two runs of a model agree 80–92% of the time. However, duration judgments are more ambiguous for people than a single model answer.

6 EXP 4: DO LLMS MAKE COMPETENT DECISIONS WITH TIME?

In experiment 2, we investigated how LLMs estimate human and agent task durations. Next, we ask whether they make competent decisions given these durations.

Setup. We test models in the domain of tool use—common, naturalistic decisions that contain items with fixed time intervals. LLMs are given a task and may query an optional tool with a pre-specified time cost. Tools are chosen such that they offer plausible accuracy benefits, but models can always answer without them. Given this setup, models choose between two actions:

A. Query the tool, accepting the time cost in exchange for potentially higher accuracy.

B. Answer directly, avoiding the time cost but forgoing the potential for higher accuracy.

We elicit each model's estimated probability of answering correctly with and without the tool and define the perceived benefit of tool use as $\Delta p = p_{\text{tool}} - p_{\text{no-tool}}$. Models almost always expect tool use to improve accuracy, with Δp positive in 99.77% of trials, zero in 0.22%, and negative in 0.01%. Each decision involves two quantities: perceived accuracy benefit from tool use Δp, and fixed time cost Δt. We measure how these dimensions of the accuracy–time trade-off affect model choices.

Tasks. We use tasks from two domains: solving polynomial roots and answering time-sensitive factual questions. Polynomials are sourced from the DeepMind Mathematics generative dataset (Saxton et al., 2019). We generate 30 cubics split between easy, medium, and hard (as defined by the dataset; see Appendix D.2 for examples). The form of equations was chosen via a pilot so that models' estimates of p_{tool} and $p_{\text{no-tool}}$ were both frequently less than 100%. Models are provided with a calculator tool that evaluates a list of numerical expressions in a single call but cannot solve equations.

For time-sensitive factual questions, we use the "fast-changing" subset of FreshQA (Vu et al., 2024), consisting of questions whose answers change frequently (from daily to yearly). We randomly sample 30 questions and update the ground truth on the same day of data collection (08/27/26; examples in Appendix D.2). Models are provided with a widely used commercial search tool, Tavily (Tavily, 2026), and can either rely on their internal knowledge or query the web for current information.

Models and sampling. We evaluate GPT-5.6 Sol, Claude Opus 5, and Gemini 3.7 Flash at their default and no-reasoning settings, using each model's default temperature. Since Gemini lacks a no-reasoning setting, we use minimal reasoning instead and refer to it as "no reasoning" throughout.

Prompts. First, we ask each model to estimate its probability of answering correctly with (p_{tool}) and without ($p_{\text{no-tool}}$) the tool. Then, in a new context window, the model gets the task, tool description, and time cost Δt, then chooses whether to use the tool or answer directly. If the model chooses to use the tool, the output is returned to the model before it outputs the final answer. The model is only allowed to query the tool once for each trial. See full prompts in Appendix D.1.

We test seven delays, $\Delta t \in \{1, 5, 10, 20, 30, 180, 1200\}$ seconds, and a baseline with no stated time. Each task and time pair is repeated 10 times per reasoning setting. We exclude no-reasoning trials that visibly expose reasoning (1.0% of all trials; Appendix D.4). We confirm that solving without the tool takes less time than with, thus Δt is an underestimate of the true time difference (Appendix D.5).

Metrics. We measure the fraction of trials querying the tool and time sensitivity as change per $10\times$ increase in Δt, estimated by least-squares fits in $\log_{10} \Delta t$ (Appendix D.7).

6.1 RESULTS

Models reduce tool use with increasing time costs, but this tendency varies across LLMs. Figure 4 shows that as tool time increases from 1s to 20 min, Gemini reduces tool use on cubics by 10.7% and 14.8% with and without reasoning; Claude decreases by 27.0% without reasoning; and GPT shows no time sensitivity, increasing by 1.3% and 2.0% instead. For search, GPT changes by only $+0.3\%$ and -0.3% with and without reasoning, while Claude decreases by 7.0% and 3.3%. Gemini is most sensitive, decreasing by 33.3% and 29.3%, respectively.



Figure 4: **Tool-use sensitivity to time cost varies across models.** Points show mean tool use across 30 tasks. Lines are least-squares fits in $\log_{10} \Delta t$. Gray diamonds mark the no-delay baseline, extended across each panel by dashed horizontal lines; the baseline is excluded from the fits.

Several delay-sensitive conditions exhibit approximately log-linear declines in tool use. In cases where tool use responds to Δt, least-squares fits show that tool usage declines approximately linearly with log time ($R^2 = 0.67$–0.94; Figure 4). Gemini is most sensitive on search, declining by 10.5% and 10.0% per $10\times$ increase with and without reasoning, while GPT changes by at most 0.5% in any condition. Within-task regressions yield 95% intervals for the slope that exclude zero for all four Gemini conditions and Claude without reasoning on cubics (Appendix D.7).

Tool use correlated positively with elicited accuracy gains (Spearman $\rho = 0.31$–0.91 across eleven estimable conditions; Appendix D.7). Descriptive fits suggest that larger perceived gains accompany comparable tool-use probabilities at longer waits in some conditions (Appendix D.8).

7 Discussion

Time is a fundamental part of human experience, but not directly accessible to language models. This creates the question of how these models represent and reason about time. We conducted a systematic evaluation of these capabilities, beginning with qualitative principles, moving to quantitative estimation, and finally by assessing decision-making. We found that LLMs are mostly accurate in reasoning about basic elements of time, but falter on tasks that rely on experience: how long they will take to complete tasks, and how to trade off time and accuracy. In the remainder of the paper we consider the limitations of our analysis and possible future directions.

7.1 Limitations and Future Work

While our evaluation studies different aspects of time, we do not comprehensively test all aspects (e.g., date arithmetic; Tan et al., 2023). However, our four-level evaluation paradigm, starting from axioms and ending at decision making, provides the most holistic evaluation to date. Another limitation is in the assumptions of axioms for Exp 1, which can be contested. For instance, our current theory has a binary setup register that immediately destroys setup when starting a foreign task. This goes against some findings that switching costs grow with the length of interruption (Altmann & Trafton, 2002; Monk et al., 2008). However, we show that T4 (batching dominance) needs switching costs to reset rather than accumulate after returning to a task (see Appendix A.1.6). Design choices like these are necessary for concise abstractions. At points where literatures disagree, counterexamples that break our assumptions serve instead as the next layer upon which more accurate abstractions can be built.

For Exp 2, the limited available METR data constrained both task coverage and task information supplied to the model. Because executing agents can use non-public task resources unavailable to the estimators, some errors may reflect missing cues about task difficulty rather than temporal-estimation failures alone. To address this, we repeated the self-estimation experiment with as much task and environment information as we could reconstruct from public materials, and the underestimation either persisted or became worse (see Appendix B.5). However, future work should still evaluate estimates within complete agent environments when possible, and also cover broader task domains.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with:

Experiments. Writing experimental code, running the code, sanity checking the experimental design and setup, finding datasets, suggesting and implementing some analyses, running initial analyses.

Writing. Writing initial drafts for portions of the paper, shortening text, creating figures, creating tables, and generating examples.

Ideation. Looking for related work and deriving the axiomatic framework and proofs.

We have not used generative AI tools for:

Important decision making. Determining the research questions to study, designing experiments, editing experiment designs, editing the final draft, and numerous sanity checks.

Manual inspection. The authors manually inspected prompts and experimental stimuli before running experiments. All theoretical proofs were manually verified.

No end-to-end research agents were used—all AI interactions were back-and-forth with human oversight. All writing and experimental designs were carefully reviewed and iterated upon by the author team. The authors take sole responsibility for the final content of the work, such as text, claims, and results—including those produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We will publish all code on a publicly accessible GitHub repository. These will include instructions to reproduce the results in the experiments in our paper. The authors will also be available via email in case additional details are required to reproduce findings.

REFERENCES

Jack A. Adams. The second facet of forgetting: A review of warm-up decrement. *Psychological Bulletin*, 58(4):257–273, 1961.

Ali Allahverdi, C. T. Ng, T. C. E. Cheng, and Mikhail Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032, 2008.

James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26 (11):832–843, 1983.

James F. Allen and Patrick J. Hayes. A common-sense theory of time. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pp. 528–531, Los Angeles, CA, 1985.

Melissa J Allman, Bin Yin, and Warren H Meck. Time in the psychopathological mind. *Subjective time: The philosophy, psychology, and neuroscience of temporality*, pp. 637–654, 2014.

Erik M. Altmann and J. Gregory Trafton. Memory for goals: An activation-based model. *Cognitive Science*, 26(1):39–83, 2002.

Robert B. Ammons. Acquisition of motor skill: I. Quantitative analysis and theoretical formulation. *Psychological Review*, 54(5):263–281, 1947a.

Robert B. Ammons. Acquisition of motor skill: II. Rotary pursuit performance with continuous practice before and after a single rest. *Journal of Experimental Psychology*, 37(5):393–411, 1947b.

Anthropic. Claude cowork. `https://www.anthropic.com/product/claude-cowork`, 2026. Accessed: 2026-08-28.

Kenneth J. Arrow. The economic implications of learning by doing. *The Review of Economic Studies*, 29(3):155–173, 1962.

Winfred Arthur Jr, Winston Bennett Jr, Pamela L. Stanush, and Theresa L. McNelly. Factors that influence skill decay and retention: A quantitative review and analysis. *Human Performance*, 11 (1):57–101, 1998.

Axel Backlund and Lukas Petersson. Vending-bench: A benchmark for long-term coherence of autonomous agents. *arXiv preprint arXiv:2502.15840*, 2025. URL https://arxiv.org/abs/2502.15840.

Charles D. Bailey. Forgetting and the learning curve: A laboratory study. *Management Science*, 35 (3):340–352, 1989.

Ralph M. Barnes. *Motion and Time Study: Design and Measurement of Work*. Wiley, New York, 7 edition, 1980.

Stephen E. Bechtold, Ralph E. Janaro, and De Witt L. Sumners. Maximization of labor productivity through optimal rest-break schedules. *Management Science*, 30(12):1442–1458, 1984.

Gary S. Becker. A theory of the allocation of time. *The Economic Journal*, 75(299):493–517, 1965.

Richard A. Block and Dan Zakay. Prospective and retrospective duration judgments: A meta-analytic review. *Psychonomic Bulletin & Review*, 4(2):184–197, 1997.

Maarten A. S. Boksem and Mattie Tops. Mental fatigue: Costs and benefits. *Brain Research Reviews*, 59(1):125–139, 2008.

Gordon D. A. Brown, Ian Neath, and Nick Chater. A temporal ratio model of memory. *Psychological Review*, 114(3):539–576, 2007.

Roger Buehler, Dale Griffin, and Michael Ross. Exploring the "planning fallacy": Why people underestimate their task completion times. *Journal of Personality and Social Psychology*, 67(3): 366–381, 1994.

Roger Buehler, Dale Griffin, and Johanna Peetz. The planning fallacy: Cognitive, motivational, and social origins. In *Advances in Experimental Social Psychology*, volume 43, pp. 1–62. Academic Press, 2010.

Carrie J. Cai, Philip J. Guo, James R. Glass, and Robert C. Miller. Wait-learning: Leveraging wait time for second language education. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, pp. 3701–3710. ACM, 2015.

John F. Catalano. The effect of rest following massed practice of continuous and discrete motor tasks. *Journal of Motor Behavior*, 10(1):63–67, 1978.

Wenhu Chen, Xinyi Wang, and William Yang Wang. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2021. URL https://arxiv.org/abs/2108.06314.

Yize Cheng, Arshia Soltani Moakhar, Chenrui Fan, Parsa Hosseini, Kazem Faghih, Zahra Sodagar, Wenxiao Wang, and Soheil Feizi. Your llm agents are temporally blind: The misalignment between tool use decisions and human time perception. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37082–37104, 2026.

Katherine M. Collins, Cedegao E. Zhang, Graham Todd, Lance Ying, Mauricio Barba da Costa, Ryan Liu, Prafull Sharma, Adrian Weller, Ionatan Kuperwajs, Lionel Wong, Joshua B. Tenenbaum, and Thomas L. Griffiths. Evaluating language models' evaluations of games, 2025.

Fabrizio Dell'Acqua, Edward McFowland III, Ethan Mollick, Hila Lifshitz-Assaf, Katherine C Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R Lakhani. Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality, 2023.

Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*, 43(3):1–45, 2025.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022. URL https://aclanthology.org/2022.tacl-1.15/.

R. Key Dismukes. Remembrance of things future: Prospective memory in laboratory, workplace, and everyday settings. *Reviews of Human Factors and Ergonomics*, 6(1), 2010.

Kristin Donnelly, Giovanni Compiani, and Ellen R. K. Evers. Time periods feel longer when they span more category boundaries: Evidence from the lab and the field. *Journal of Marketing Research*, 59 (4):821–839, 2022.

Rachit Dubey, Mark Ho, Hermish Mehta, and Thomas L Griffiths. Aha! moments correspond to metacognitive prediction errors. *Cognition*, 274:106537, 2026.

David M. Eagleman. Human time perception and its illusions. *Current Opinion in Neurobiology*, 18 (2):131–136, 2008.

Bahare Fatemi, Seyed Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. Test of time: A benchmark for evaluating llms on temporal reasoning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/eb7295a8bc613b375726659c2ecd6f14-Abstract-Conference.html.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. Magentic-one: A generalist multi-agent system for solving complex tasks, 2024.

Paul Fraisse. Perception and estimation of time. *Annual Review of Psychology*, 35:1–37, 1984.

Tony Gillie and Donald Broadbent. What makes interruptions disruptive? A study of length, similarity, and complexity. *Psychological Research*, 50:243–250, 1989.

Ali Goli and Amandeep Singh. Frontiers: Can large language models capture human preferences? *Marketing Science*, 43(4):709–722, 2024.

Patrick J. Hayes. The naive physics manifesto. In Donald Michie (ed.), *Expert Systems in the Micro-Electronic Age*, pp. 242–270. Edinburgh University Press, Edinburgh, 1979. ISBN 0-85224-381-2.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024.

Clark L. Hull. *Principles of Behavior: An Introduction to Behavior Theory*. D. Appleton-Century Company, New York, 1943.

Yoel Inbar, Simona Botti, and Karlene Hanko. Decision speed and choice regret: When haste feels like waste. *Journal of Experimental Social Psychology*, 47(3):533–540, 2011.

Arthur T. Jersild. *Mental Set and Shift*. Number 89 in Archives of Psychology. Columbia University, New York, 1927.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*, 2024.

S. M. Johnson. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1):61–68, 1954.

Daniel Kahneman and Amos Tversky. Intuitive prediction: Biases and corrective procedures. *TIMS Studies in Management Science*, 12:313–327, 1979.

Andrej Karpathy. Jagged intelligence. Post on X, July 2024. https://x.com/karpathy/status/1816531576228053133.

Andrej Karpathy. I played with Gemini 3 yesterday via early access. few thoughts. X (formerly Twitter), November 2025. URL https://x.com/karpathy/status/1990854771058913347. Thread; accessed 08/28/2026.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, et al. Measuring ai ability to complete long software tasks. *arXiv preprint arXiv:2503.14499*, 2025.

Giuseppe Labianca, Henry Moon, and Ian Watt. When is an hour not 60 minutes? deadlines, temporal schemata, and individual and task group performance. *Academy of Management Journal*, 48(4): 677–694, 2005.

Lingyu Li, Yang Yao, Yixu Wang, Chunbo Li, Yan Teng, and Yingchun Wang. The other mind: How language models exhibit human temporal cognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026.

Mohammad Mehdi Liaee and Hamilton Emmons. Scheduling families of jobs with setup times. *International Journal of Production Economics*, 51(3):165–176, 1997.

Falk Lieder and Thomas L Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1, 2020.

Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, et al. Proactive agent: Shifting llm agents from reactive responses to active assistance. In *International Conference on Learning Representations*, volume 2025, pp. 47431–47457, 2025.

Gloria Mark, Daniela Gudith, and Ulrich Klocke. The cost of interrupted work: More speed and stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pp. 107–110. ACM, 2008.

Natalia Martinelli and Sylvie Droit-Volet. Judgment of duration and passage of time in prospective and retrospective conditions and its predictors for short and long durations. *Scientific Reports*, 12: 22241, 2022. URL https://www.nature.com/articles/s41598-022-25913-9.

William J. Matthews and Warren H. Meck. Temporal cognition: Connecting subjective time to perception, attention, and memory. *Psychological Bulletin*, 142(8):865–907, 2016.

Mantas Mazeika, Xuwang Yin, Rishub Tamirisa, Jaehyuk Lim, Bruce W. Lee, Richard Ren, Long Phan, Norman Mu, Adam Khoja, Oliver Zhang, and Dan Hendrycks. Utility engineering: Analyzing and controlling emergent value systems in AIs. *arXiv preprint arXiv:2502.08640*, 2025.

Theo F. Meijman and Gijsbertus Mulder. Psychological aspects of workload. In Pieter J. D. Drenth, Henk Thierry, and Charles J. de Wolff (eds.), *A Handbook of Work and Organizational Psychology*, volume 2 of *Work Psychology*, chapter 2, pp. 5–33. Psychology Press, Hove, UK, 2 edition, 1998.

Microsoft. Microsoft 365 copilot. https://www.microsoft.com/en-us/microsoft-365-copilot/, 2026. Accessed: 2026-08-28.

Christopher A. Monk, J. Gregory Trafton, and Deborah A. Boehm-Davis. The effect of interruption duration and demand on resuming suspended goals. *Journal of Experimental Psychology: Applied*, 14(4):299–313, 2008.

Clyde L. Monma and Chris N. Potts. On the complexity of scheduling with batch setup times. *Operations Research*, 37(5):798–804, 1989.

Stephen Monsell. Task switching. *Trends in Cognitive Sciences*, 7(3):134–140, 2003.

Allen Newell and Paul S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In John R. Anderson (ed.), *Cognitive Skills and Their Acquisition*, pp. 1–55. Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

O*NET. The O*NET content model. O*NET Resource Center, 2026. URL https://www.onetcenter.org/content.html. Sponsored by the U.S. Department of Labor, Employment and Training Administration (USDOL/ETA). Accessed 3 September 2026.

OpenAI. Introducing openai frontier. https://openai.com/index/introducing-openai-frontier/, 2026. Accessed: 2026-08-28.

OpenClaw. Heartbeat. OpenClaw documentation, 2026. URL https://docs.openclaw.ai/gateway/heartbeat. Retrieved August 26, 2026.

Wanda J. Orlikowski and JoAnne Yates. It's about time: Temporal structuring in organizations. *Organization Science*, 13(6):684–700, 2002.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

John Pencavel. The productivity of working hours. *The Economic Journal*, 125(589):2052–2076, 2015.

Leslie A. Perlow. The time famine: Toward a sociology of work time. *Administrative Science Quarterly*, 44(1):57–81, 1999.

Norman G. Peterson, Michael D. Mumford, Walter C. Borman, P. Richard Jeanneret, Edwin A. Fleishman, Kerry Y. Levin, Michael A. Campion, Melinda S. Mayfield, Frederick P. Morgeson, Kenneth Pearlman, Marilyn K. Gowing, Anita R. Lancaster, Marilyn B. Silver, and Donna M. Dye. Understanding work using the Occupational Information Network (O*NET): Implications for practice and research. *Personnel Psychology*, 54(2):451–492, 2001.

Chris N. Potts and Mikhail Y. Kovalyov. Scheduling with batching: A review. *European Journal of Operational Research*, 120(2):228–249, 2000.

Douglas Poynter. Judging the duration of time intervals: A process of remembering segments of experience. In Iris Levin and Dan Zakay (eds.), *Time and Human Cognition: A Life-Span Perspective*, volume 59 of *Advances in Psychology*, pp. 305–331. North-Holland, 1989.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. The TimeBank corpus. In *Proceedings of Corpus Linguistics 2003*, pp. 647–656, Lancaster, UK, 2003.

David Rein, Joel Becker, Amy Deng, Seraphina Nix, Chris Canal, Daniel O'Connel, Pip Arnott, Ryan Bloom, Thomas Broadley, Katharyn Garcia, Brian Goodrich, Max Hasin, Sami Jawhar, Megan Kinniment, Thomas Kwa, Aron Lajko, Nate Rush, Lucas Jun Koba Sato, Sydney von Arx, Ben West, Lawrence Chan, and Elizabeth Barnes. HCAST: Human-calibrated autonomy software tasks. *arXiv preprint arXiv:2503.17354*, 2025. URL https://arxiv.org/abs/2503.17354.

Ian Rios-Sialer, Shantanu Darveshi, Shuai Jiang, Avigya Paudel, Anastasiia Pronina, Ipshita Bandyopadhyay, and Justin Shenk. Temporal preference concepts and their functions in a large language model. *arXiv preprint arXiv:2606.05194*, 2026.

Robert D. Rogers and Stephen Monsell. Costs of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: General*, 124(2):207–231, 1995.

Jillian Ross, Yoon Kim, and Andrew W. Lo. LLM economicus? mapping the behavioral biases of LLMs via utility theory. *arXiv preprint arXiv:2408.02784*, 2024.

Giovanni Sala and Fernand Gobet. Does far transfer exist? Negative evidence from chess, music, and working memory training. *Current Directions in Psychological Science*, 26(6):515–520, 2017.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=H1gR5iR5FX.

Daniel J. Simons, Walter R. Boot, Neil Charness, Susan E. Gathercole, Christopher F. Chabris, David Z. Hambrick, and Elizabeth A. L. Stine-Morrow. Do "brain-training" programs work? *Psychological Science in the Public Interest*, 17(3):103–186, 2016.

Clinton Stark. Quick tip: How to get Claude Code to run autonomously for hours. Stark Insider, May 2026. URL https://www.starkinsider.com/2026/05/claude-code-autonomous-coding-time-hack.html. Accessed 2026-08-28.

Chess Stetson, Matthew P. Fiesta, and David M. Eagleman. Does time really slow down during a frightening event? *PLoS ONE*, 2(12):e1295, 2007.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14820–14835, 2023.

Tavily. Tavily: Search api for AI agents. https://www.tavily.com, 2026. Accessed: 2026-09-05.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Edward L. Thorndike and Robert S. Woodworth. The influence of improvement in one mental function upon the efficiency of other functions (I). *Psychological Review*, 8(3):247–261, 1901.

Gabriela N. Tonietto, Selin A. Malkoc, and Stephen M. Nowlis. When an hour feels shorter: Future boundary tasks alter consumption by contracting time. *Journal of Consumer Research*, 45(5): 1085–1102, 2019.

Yanping Tu and Dilip Soman. The categorization of time and its impact on task initiation. *Journal of Consumer Research*, 41(3):810–822, 2014.

Philip Tucker. The impact of rest breaks upon accident risk, fatigue and performance: A review. *Work & Stress*, 17(2):123–137, 2003.

U.S. Social Security Administration. Baby names from social security card applications: National data. https://www.ssa.gov/oact/babynames/limits.html, 2025. Accessed 2026-08-28.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 26941–26975, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2f6a6317bada76b26a4f61bb70a7db59-Abstract-Conference.html.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. Fine-grained temporal relation extraction. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2906–2919, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://aclanthology.org/P19-1280/.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.813.

Jonas Wallat, Adam Jatowt, and Avishek Anand. Temporal blind spots in large language models. *arXiv preprint arXiv:2401.12078*, 2024.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9440–9450, 2024.

Yuqing Wang and Yun Zhao. TRAM: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6389–6415, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Joel S. Warm, Raja Parasuraman, and Gerald Matthews. Vigilance requires hard mental work and is stressful. *Human Factors*, 50(3):433–441, 2008.

Eva Wittenberg and Roger Levy. If you want a quick kiss, make it count: How choice of syntactic construction affects event construal. *Journal of Memory and Language*, 94, 06 2017.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *First Conference on Language Modeling (COLM)*, 2024.

Dan Zakay and Richard A. Block. Temporal cognition. *Current Directions in Psychological Science*, 6(1):12–16, 1997.

Eviatar Zerubavel. *Hidden Rhythms: Schedules and Calendars in Social Life*. University of Chicago Press, Chicago, 1981.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, and Denny Zhou. NATURAL PLAN: Benchmarking LLMs on natural language planning. *arXiv preprint arXiv:2406.04520*, 2024.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. "Going on a vacation" takes longer than "Going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3363–3369, Hong Kong, China, November 2019. Association for Computational Linguistics.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7579–7589. Association for Computational Linguistics, 2020.

A Details for Experiment 1: evaluating models' structural knowledge of time

A.1 Axioms, Theorems, and their derivations

In this section, we provide the formal counterpart of the state model in Section 3.1. This covers axioms A1–A5, theorems T1–T5, and how the theorems can be proven from said axioms. In the main text, each axiom/theorem is stated as a directional claim, which we formalize as an inequality over a small transition system. We also show why the requirements within vignettes exist in order to make their ground truths derivable, and provide a countermodel showing the structural assumption that batching dominance relies upon.

A.1.1 Formal setup

Actions, families, and states. Let $\mathcal{A}$ be a set of task types (*actions*) and $\mathcal{F}$ a set of skill *families*, with the labeling $\mathrm{fam} : \mathcal{A} \to \mathcal{F}$ fixed by each vignette. A state is $s = (w, \kappa, \sigma, \phi)$: a world w; a skill profile of the character $\kappa : \mathcal{F} \to \mathbb{R}_{\geq 0}$; a setup register $\sigma \in \mathcal{F} \cup \{\bot\}$ containing the family of the most recent task (initially $\bot$); and a fatigue level $\phi \geq 0$. Two primitive functions act upon the theory: a transition function $\delta : \mathcal{A} \times \mathcal{S} \to \mathcal{S}$ and a duration function $\tau : \mathcal{A} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$. We denote $\kappa_F(s)$ as the character's skill at family F in state s.

Definition 1 (Related and foreign work)**.** In our theory, work on task a can raise one's skill at task b, i.e., $\kappa_{\mathrm{fam}(b)}(\delta(a,s)) > \kappa_{\mathrm{fam}(b)}(s)$ for some s. Two tasks are *related* if either task can raise skill at the other task, and *foreign* otherwise. Skill families group mutually related tasks, where work on each task within the family can raise skill in all other tasks in the family. Tasks not sharing a family are foreign. By definition, for foreign tasks a, b and every s, $\kappa_{\mathrm{fam}(b)}(\delta(a,s)) \leq \kappa_{\mathrm{fam}(b)}(s)$. That is, performing a foreign task can never raise skill.

Events. Three kinds of action can affect state: same-family work (with respect to the previous task), foreign work, and rest. Rest is when no tasks are being conducted, and is represented by idle family $\{w_d\}_{d\geq 0}$.

Definition 2 (Well-definedness, state sufficiency)**.** δ and τ are only functions of a and s. Past action trajectories that reach the same state are interchangeable in their future actions and durations.

Definition 3 (Existence, rest)**.** Rests can be of any length: for every $d \geq 0$, there is an idle action w_d with $\tau(w_d, s) = d$. Idling for time d takes duration d: $\tau(w_d, s) = d$. Concatenating rests is the same as resting for their total duration: $w_d \,;\, w_e \equiv w_{d+e}$. Rest does not change the world: $\delta(w_d, \cdot)$ keeps the same w and σ, and its effects on κ and ϕ are only the ones stated in A2 and A5.

Definition 4 (Existence, splits)**.** A non-atomic task a can split into two tasks (a_1, a_2). These two tasks belong to a's family, and if performed subsequently in the correct order, yield the same transition function and duration: $\delta(a_1 \,;\, a_2, s) = \delta(a, s)$ and $T(a_1 \,;\, a_2, s) = \tau(a, s)$.

Definition 5 (Existence, external processes)**.** An external process p has duration d_p and does not require the agent to attend to it during its duration. The agent launches it (ℓ_p) and later collects it (c_p). c_p cannot occur unless d_p has passed since ℓ_p.

Composite duration. For a sequence of actions $\alpha = a_1 \,;\, \cdots \,;\, a_n$, the duration of the total sequence is defined recursively: $T(\alpha, s) = \sum_i \tau\big(a_i,\ \delta(a_1 \,;\, \cdots \,;\, a_{i-1}, s)\big)$. Thus, we have the following lemma:

Lemma 6 (Composition)**.** $T(\alpha \,;\, \beta, s) = T(\alpha, s) + T(\beta, \delta(\alpha, s))$.

Proof. We can prove this by inducting on the size of β, where we unfold the last action $\delta(\alpha \,;\, \beta', s) = \delta(\beta', \delta(\alpha, s))$. Following from Definition 2, past events do not affect the current state or future transitions. □

A.1.2 AXIOMS

Throughout this section, we write $F = \mathrm{fam}(a)$ for the family of the current action.

Axiom (A1, learning)**.** When performing task a, it is possible (but not always the case) for learning; thus the skill in that family is non-decreasing: $\kappa_F(\delta(a,s)) \geq \kappa_F(s)$. Moreover, higher skill never slows base work.

Axiom (A2, decay)**.** Time away from skill family F weakly erodes κ_F, regardless of if it is filled by rest w_d or foreign work a': $\kappa_F(\delta(w_d, s)) \leq \kappa_F(s)$ and $\kappa_F(\delta(a', s)) \leq \kappa_F(s)$ for $\mathrm{fam}(a') \neq F$. Decay is non-decreasing with duration.

Axiom (A3, screening)**.** The duration of a task a in skill family F is its base cost b_a plus a fatigue term h plus a switching cost π_F if the task does not match the current setup.

$$\tau(a, s) \;=\; b_a\big(\kappa_F(s)\big) \;+\; h\big(\phi(s)\big) \;+\; \pi_F \cdot \mathbf{1}[\sigma(s) \neq F]\,,$$

The base-work function b_a is non-increasing in κ_F, h is non-decreasing (fatigue slows all work), and the switching cost $\pi_F > 0$ (Rogers & Monsell, 1995; Monsell, 2003). Performing task a sets the setup register $\sigma := F$.

Axiom (A4, loading)**.** Fatigue increases when performing all tasks. $\phi(\delta(a,s)) \geq \phi(s)$ unless a is idling. We denote the fatigue toll of task a as $\Delta_a(s) = \phi(\delta(a,s)) - \phi(s) \geq 0$.

Axiom (A5, recovery)**.** When idling, fatigue drops. $\phi(\delta(w_d, s)) \leq \phi(s)$, non-increasingly in d.

Remark 7 (Limitation: fatigue as a single resource). Our theory treats fatigue as a single resource that every task increases. However, in reality restoration can be attained by qualitatively different work, for instance doing mental tasks to rest the physical body. We consider this outside the scope of our theory; To address this, in the vignettes we build where this difference matters, we explicitly state that 1) the character is well-rested, and 2) that tasks are not tiring for them to complete, in order to ensure that this detail does not affect the tests generated by these axioms.

To illustrate the coverage of our axioms over the theory space, we show that our axioms cover how each type of event {same-family work, foreign work, rest} affect a component of the theory {skill, setup, fatigue}.

Component	Same-family work	Foreign work	Rest
κ_F (skill)	never lowers (A1)	never raises (Def. 1); erodes (A2)	erodes (A2)
σ (setup)	set to F (A3)	set to foreign (A3)	preserved (Def. 3)
ϕ (fatigue)	raises (A4)	raises (A4)	lowers (A5)

What the axioms leave open. Commutativity is not guaranteed; our theory does not entail that $T(a\,;b,s) = T(b\,;a,s)$, nor does it negate it. All axioms except A3 are directional claims, but the magnitudes of those directions can vary. Because the rate of skill decay and the gains from reducing fatigue are not specified beyond monotonicity, a long break away from a task may help or hurt completion times. We cover this in Theorem T3 (rest vs. rust).

A.1.3 Special conditions for theorems T4 and T5

Theorems T4 and T5 need additional common assumptions that are part of the space which our axioms deliberately leave open. In our vignettes that test T4 and T5, we explicitly provide these conditions so that the gold answer remains entailed.

Condition (C1, negligible fatigue)**.** All tasks raise fatigue by a negligible amount.

Condition (C2, negligible decay)**.** Over the course of the vignette, skills decay by a negligible amount.

A.1.4 Derivations

Theorem (T1, no free transfer)**.** If a is foreign to b, then the time taken to do b after a is at least as long as the time it takes to do b directly. $\tau(b, \delta(a, s)) \geq \tau(b, s)$.

Connection to literature. Theorem T1 splits into two cases, when the setup register contains a foreign family ($\sigma(s) \neq \mathrm{fam}(b)$) and when it contains a related family ($\sigma(s) = \mathrm{fam}(b)$). These two cases are covered by two different bodies of research.

The first body of work corresponding to $\sigma(s) \neq \mathrm{fam}(b)$ finds that practice at a task will provide no transfer to a dissimilar one. Early work over a century ago in experimental psychology found no transfer between spelling and multiplication (Thorndike & Woodworth, 1901). This style of finding has continued across various skill domains up to recent years: a meta analysis between chess, music, and working-memory instruction have found that transfer to academic and cognitive skills rarely occurs, with a further decrease in effect sizes as experimental controls improve (Sala & Gobet, 2017), and a review of the brain-training literature found that gains appeared on trained tasks, less on closely related ones, but these gains reduced to very little on distantly related tasks (Simons et al., 2016). The key takeaway here is that **performing an unrelated task does not help, but it also doesn't necessarily hurt**.

The second body of work corresponding to $\sigma(s) = \mathrm{fam}(b)$ is the concept of *switch costs* in the task-switching literature. This is defined as the difference between switch and repeat trials, i.e., b after a vs. b after b (Rogers & Monsell, 1995; Monsell, 2003). The switch cost persists as a residual cost even when the participant has had time to prepare. In this setting, **the switch cost actively hurts the performance duration of task b.**

In our proof of Theorem T1 below, we show that **our axioms entail this key difference across the two literatures above**, unifying them. Skill transfer research covers the cold-start comparison (foreign work does not help), and task switching covers the warmed-up comparison (foreign work strictly hurts), and we will demonstrate that the setup register $\sigma(s)$ reproduces the boundary between them.

Proof. Consider when A3 is applied to the action b on both sides of the equation.

For the first term b_b (base work), since a is foreign it never raises task b's skill $\kappa_{\mathrm{fam}(b)}$ (Definition 1). Furthermore, performing a weakly erodes $\kappa_{\mathrm{fam}(b)}$ (A2). Then by A1, lower skill never speeds up base work (A1), so the base cost term weakly rises.

For the second term h (fatigue cost), performing a weakly raises load ϕ (A4), and h is non-decreasing in ϕ.

Finally, consider the third term π (switching penalty). Performing a changes the setup register to $\mathrm{fam}(a) \neq \mathrm{fam}(b)$, which either leaves b's penalty status unchanged (if previously foreign) or increases it (if previously related).

Combining the three terms, in both cases of the switching penalty, all three terms move weakly upward, so there is no free transfer. □

This split exactly matches the two sets of findings from literature mentioned above. Specifically, when the previous task was related to b ($\sigma(s) = \mathrm{fam}(b)$), the third term strictly increases from 0 to $\pi_{\mathrm{fam}(b)} > 0$ (A3), and the inequality is strict. And when $\sigma(s) \neq \mathrm{fam}(b)$, the inequality is loose.

Theorem (T2, interruption overhead)**.** Let (a_1, a_2) be a split of a (following Definition 4) and let β be a non-idle foreign task. When β interrupts a in the form of $a_1 \,;\, \beta \,;\, a_2$, the time attributable to a is greater than $\tau(a, s)$, specifically by at least $\pi_{\mathrm{fam}(a)}$. This scales linearly, so with k foreign interruptions the cost is at least $k \cdot \pi_{\mathrm{fam}(a)}$.

Connection to literature. When a_2 resumes after β, the setup register is at $\mathrm{fam}(\beta)$, so by A3, a_2 has an additional switching cost back to $\mathrm{fam}(a)$. Two literatures use different names for this cost: the aforementioned *switch cost* in task-switching (Rogers & Monsell, 1995; Monsell, 2003), and the *resumption lag* of interruption research—which is the time from the end of an interruption to the first action of the next task, attributed to re-establishing a suspended goal (Altmann & Trafton, 2002).

In some studies, the resumption lag increases with the duration of the interrupting task (Monk et al., 2008), but not in others (Gillie & Broadbent, 1989). This is captured within our theory using the flat setup cost (A3) along with a scaling decay cost (A2). Our theory adopts the fixed cost model, while leaving any additionally accruing costs to decay (A2), which is monotone to the duration of time away from the task. In the literature, resumption cost also scales with the complexity of the interrupter task (Gillie & Broadbent, 1989; Monk et al., 2008). Our theory covers this via the fatigue term (A4).

However, the literature describes the scaling as due to working-memory displacement rather than fatigue, which is a direction of refinement that we discuss in Section A.1.6. Another effect not reflected in our theory is how sometimes, interruptions *similar* to the primary task are more disruptive (Gillie & Broadbent, 1989). Instead, a same-family task is treated as not affecting the setup at all. However, Theorem T2 does not specify anything for same-family interrupters, and our vignettes for T2 use foreign interrupters to avoid this case. Finally, studies have shown that subjects tend to compensate for the time lost during interruptions by working faster afterwards, at the cost of stress and effort (Mark et al., 2008). Due to the history-free property of our model, this effect is not supported by our current theory.

Proof. Define state $s_1 = \delta(a_1, s)$. Without interruptions, a_2 proceeds with $\sigma(s_1) = \mathrm{fam}(a)$ and pays no switching cost (A3).

In the interruption case, a_2 runs in state $s_2 = \delta(\beta, s_1)$. The last action was β, which is foreign to a, and thus also foreign to both a_1 and a_2 (Definition 4). The last action β also sets the setup register to $\mathrm{fam}(\beta)$, and so when completing task a_2 we need to pay an extra $\pi_{\mathrm{fam}(a)}$ for the setup term (A3).

Next, we show that with the additional task β, the other cost terms in Axiom A3 (base cost and fatigue) do not reduce. Base cost b_{a_2} depends only on skill κ_F and performing foreign task β cannot increase skill (Definition 1) and can weakly erode skill (A2), therefore b_{a_2} weakly rises. Similarly, after performing β fatigue can only be equal or higher (A4). Since the total time is preserved among splits, i.e., $T(a_1 \,;\, a_2, s) = \tau(a, s)$ (Definition 4), the total increase in completion time is at least $\pi_{\mathrm{fam}(a)}$.

Finally, consider the case where there are k foreign interrupters. These interruptions can each be added one after the other from left to right, where the right portion of the previous split is then split

again, and any time cost increases are charged to that split. Then, by simple induction, with k total interruptions we will have at least $k \cdot \pi_{\mathrm{fam}(a)}$ extra time cost. □

Theorem (T3, rest versus rust)**.** Let (a_1, a_2) be a split of a. Consider the case where a is interrupted by a rest action w_d. Let the state s' be after w_d but before a_2, i.e., $s' = \delta(w_d, \delta(a_1, s))$. Then the time difference of a_2 compared between resting after a_1 and not resting, i.e., $\tau(a_2, s') - \tau(a_2, \delta(a_1, s))$ is:

$$\underbrace{b_{a_2}(\kappa'_{\mathrm{fam}(a)}) - b_{a_2}(\kappa_{\mathrm{fam}(a)})}_{\geq 0 \text{ (decay, A2 via A1)}} + \underbrace{h(\phi') - h(\phi)}_{\leq 0 \text{ (recovery, A5)}},$$

with both terms monotone in d. The sign (+/-) depends on the relation of the two terms, and is not entailed by the axioms.

Connection to literature. T3 is the theorem where two opposing forces clash—the benefit from resting to reduce fatigue versus the decay of skill. The motor-learning literature has observed and named these two opposing forces, and have measured them and found them independent. The term governing fatigue recovery is *reminiscence*: the improvement of one's performance after a rest without intervening practice (Ammons, 1947b;a). This is explained as dissipation of a fatigue-esque inhibition that accumulates under continuous work, matching A4 and A5 (Hull, 1943). The term covering decay is the *warm-up decrement*, also termed the second facet of forgetting (Adams, 1961), where performance is worsened immediately after a rest but quickly recovers—an effect separate from actual learning.

These two forces are independent in the literature (Adams, 1961), and one paper illustrates how outcomes of rest can be two-sided: a study in motor learning (Catalano, 1978) found that when performing a continuous task and then resting, rest *improved* performance, but the *opposite* happened for a discrete task where fatigue is presumably lower due to time between each subtask. Similarly, a review of breaks in industrial and transport domains found regular breaks effective for maintaining performance, but that there is little evidence on the optimal length of said break (Tucker, 2003). Our theory's opposing forces of A2 and A5 capture the flexibility of sign in these past findings.

We address one potential refinement in Section A.1.6—currently, the theory assigns all rest-induced loss to skill decay (A2). However, warm-up decrement can also be attributable to the decay of the setup under idling. Instead, in our theory has a stricter definition of rest that preserves σ (Definition 3). Despite this, at the sign level both count as rust, so T3 stands regardless. Our three vignettes for rest vs. rust fix the ground truth sign by specifying skill and fatigue levels beforehand, where either skill decay is negligible, fatigue restoration is negligible, or both are in play; these fix the comparative relationship for the duration of remaining work.

Proof. Consider the two cases, one where a_2 immediately follows a_1, and the other where they are interrupted by rest w_d. As w_d preserves the setup register as $\mathrm{fam}(a)$, in both cases there is no resumption penalty. Because rest weakly erodes skill (A2), $\kappa' \leq \kappa$, and duration is non-increasing in skill, so the first term $b_{a_2}(\kappa'_{\mathrm{fam}(a)}) - b_{a_2}(\kappa_{\mathrm{fam}(a)})$ is non-negative.

On the other hand, rest reduces fatigue (A5), so $\phi' \leq \phi$. Since duration is non-decreasing in fatigue load, the second term $h(\phi') - h(\phi)$ is non-positive. As one term is non-negative and one is non-positive, but the magnitudes are not fixed, the sign is not guaranteed. □

Lemma 8 (Block decomposition)**.** *Let there be several task families that are pairwise foreign with each other. Each family has a sequence of tasks to be completed. Consider a plan α that interleaves task sequences from each family without rest, e.g., for two families X and Y, $XXYY$, $XYXY$, and $XYYX$ are all such plans. We call a maximal run of consecutive tasks of family F an F-*block*. Under conditions C1 (negligible fatigue) and C2 (negligible decay), the total time of α is:*

$$T(\alpha, s_0) = \underbrace{\sum_{a \in \alpha} b_a(\kappa_{\mathrm{fam}(a)} \textit{ at } a)}_{\textit{order-invariant}} + \underbrace{|\alpha| \cdot h(\phi_0)}_{\textit{order-invariant}}$$

$$+ \sum_F \pi_F \cdot \#\{F\textit{-blocks of } \alpha \textit{ not entered with } \sigma = F\},$$

The key point is order invariance—given identical task families and sequences, the order in which they are interleaved does not matter for the first two terms; interleaving affects total duration only through the number of F-blocks.

Proof. By A3, $T(\alpha, s_0)$ expands into a sum of base-work terms, fatigue terms, and penalty terms, and group like terms. Because C2 (negligible decay) is active, each task's skill is only affected by practice from predecessors in its own family's sequence (A1). However, because each family's tasks are completed in the same order, in every interleaving the skill gain is identical, so the total sum of base work is order invariant. Because C1 (negligible fatigue) is also active, and rest never occurs, ϕ stays at the initial ϕ_0 and the sum of fatigue costs is $|\alpha| \cdot h(\phi_0)$ across every interleaving.

Finally, for the penalty sum terms, the first task in every F-block sets $\sigma := F$, so all tasks after it keep the same σ, so none of the later items in a block pay a switching cost. However, the first task in a block always follows a foreign task, as if it was preceded by a non-foreign task, the block would no longer be the maximal sequence of like-family actions. Thus, the first task always pays switching cost π_F. Across all terms, this becomes $\sum_F \pi_F \cdot \#\{F\text{-blocks of } \alpha \text{ not entered with } \sigma = F\}$. The specification of "not entered with $\sigma = F$" only applies at the start of α, where the first task could either benefit from a preceding task of the same family or not. □

Theorem (T4, batching dominance)**.** Under conditions C1 (negligible fatigue) and C2 (negligible decay), for a pair of foreign families X and Y, and a plan α without any rest actions, the batched order (i.e., one block per family; $XXYY$) takes less time than every interleaving of the same set of tasks (e.g., $XYXY$).

Connection to literature. The core of batching dominance is the number of same family "blocks" of tasks that do not incur a setup penalty. Two independent literatures have used this quantity directly. First is the founding study of the task-switching paradigm performed by Jersild in 1927 (Jersild, 1927). Jersild timed subjects on their completion of separate tasks one set at a time versus alternating item-by-item. He compared times across conditions and found that a majority of tasks took excess time when alternating, matching Theorem T4.

The other literature is scheduling with job families from operations research (OR), where a cost is charged on each change of family (Liaee & Emmons, 1997; Monma & Potts, 1989). The *group technology assumption* common in the literature is a standard restriction that requires each family's tasks to be processed as one contiguous sequence (Liaee & Emmons, 1997; Allahverdi et al., 2008; Potts & Kovalyov, 2000), matching the result of the batching theorem. Then, each batch is an F-block and the literature's total setup time—the switching costs summed over batches—is the lemma's third term.

Proof. From Lemma 8, the ordering of the tasks only affects the number of foreign setup switches; the base and fatigue costs of the actions are otherwise identical. Batching has one F-block per family, so not counting whether the first task requires setup, there is only one switching cost. For all other interleavings, the minimum is that some family is split into two or more blocks, meaning that there are at least two switching costs, away from that family and then back. Because these costs are strictly positive ($\pi_F > 0$; A3), batching tasks takes strictly less time. □

Theorem (T5, overlap)**.** Let p be an external process (see Definition 5) with duration d, and let its collection be a task within family F. Let A be a task foreign to family F. Define the makespan M of a plan α as the wall-clock time it takes from the start of its first task to the end of its last task.

Under conditions C1 and C2 (negligible fatigue and decay), moving task A from after the collect action to before it (i.e., during the time it takes the external process to complete) changes makespan by:

$$\Delta M \;=\; \pi_F \;-\; \min(d, \tau'_A) \;+\; R,$$

where τ'_A is the duration of A when performed during the wait, and $R \geq 0$ is the recovery that resting over the full wait would provide. Filling the wait is better iff the overlap gained is greater than the sum of the resumption penalty and R.

Connection to literature. The three terms of T5 come from three distinct literatures. The overlap gain $-\min(d, \tau_A)$ is an concept in classic industrial engineering. The man–machine chart puts the human operator and the machine on the same time scale to identify human idle time while the machine runs, and the standard approach was to fill that time with other work (Barnes, 1980). The chart also shows why maximal time gain is limited: if the filler task lasts beyond when the machine finishes, the machine instead waits for the operator. The time saved is the shorter of these two durations. The same idea scaled up to many jobs and machines is the basis of flow-shop scheduling in OR. To minimize the makespan, the overlap between jobs is maximized by choosing the order in which jobs are done (Johnson's rule; Johnson, 1954).

The resumption penalty π_F belongs to the same literature as discussed in Theorem T2. The R term parallels the literature associated with T3: waiting for the process to finish is a rest break, and rest can help improve fatigue (Tucker, 2003), but filling the wait raises fatigue instead. Thus, the fatigue term is the recovery the wait would have bought (A5). Empirically, studies have demonstrated that wait times can be used effectively. A study from user-centered design in the human-computer interaction literature shows how filling waits with micro-learning can help users learn foreign languages (Cai et al., 2015).

Two aspects are not tracked in our model. First, filling a wait requires keeping some attention on when a process has finished, which can be demanding on memory and can also fail (i.e., forgetting; Dismukes, 2010). Our paradigm assumes that collecting is scheduled and monitoring is free. Second, Theorem T5 assumes that A is a foreign task. If the filled task were part of the same family, the term instead becomes $\Delta M = -\min(d, \tau_A) + R$, which becomes a rest vs. rust (T3) problem. Our vignettes for Theorem T5 choose external processes that do not need to be attended to (e.g., letting mortar sit), and the filler tasks are selected from foreign families. We vary the length of τ_A compared to d for coverage.

Proof. Consider two plans P_1 and P_2, where $P_1 = \ell_p \,;\, w_d \,;\, c_p \,;\, A$ and $P_2 = \ell_p \,;\, A \,;\, w_{d'} \,;\, c_p$.

ℓ_p is the launch action for external process p, and c_p is its collect action. w_d is an idle task. A is a foreign task to family F, which contains ℓ_p and c_p. d' is the remaining wait after completing A but before process p finishes: $d' = \max(0, d - \tau'_A)$.

Since condition C2 (negligible decay) is active, skill in $\mathrm{fam}(A)$ does not change until A is started. Thus, the base work for task A, $b_A\big(\kappa_{\mathrm{fam}(A)}(s)\big)$, is the same across the two plans. Similarly, the collect action c_p always happens after the launch action ℓ_p, so their base costs are also the same.

Next, we consider setup. We can safely ignore any switching costs from before the plan to ℓ_p since those will be equal across plans. In P_1, the setup register is not affected by w_d (Definition 3). As ℓ_p and c_p belong to the same task family, the only switching cost paid is the transition from c_p to A, costing $\pi_{\mathrm{fam}(A)}$ since A is foreign. In P_2, A is conducted after ℓ_p and switches the setup register to $\mathrm{fam}(A)$, costing the same amount. Then, when conducting c_p, whether there was a rest action w_d with positive duration or not, we need to pay an additional π_F.

Finally, consider fatigue. Since C1 (negligible fatigue) is active, the ϕ can only fall during rest w_d (A5). Let ϕ be the load after ℓ_p, ϕ' be the load after rest w_d, and let ϕ'' be the load after the partial rest $w_{d'}$, with $d' \le d$. By A5, $\phi' \le \phi'' \le \phi$. Then fatigue costs change at two parts: In P_1, A runs at load ϕ', but in P_2 it runs at ϕ. In P_1, c_p runs at ϕ' but in P_2 it runs at ϕ''. These two differences contribute to make $R = [h(\phi) - h(\phi')] + [h(\phi'') - h(\phi')]$.

Now consider the total makespans M_1 and M_2.

$$
\begin{aligned}
M_1 &= \tau_\ell + d + \tau_c + \tau_A \\
&= \tau_\ell + d + b_c + h(\phi') + 0 + b_A + h(\phi') + \pi_{\mathrm{fam}(A)} \\
&= \tau_\ell + d + b_c + h(\phi') + b_A + h(\phi') + \pi_{\mathrm{fam}(A)} \\
M_2 &= \tau_\ell + \tau'_A + d' + \tau'_c \\
&= \tau_\ell + \max(d, \tau'_A) + \tau'_c \\
&= \tau_\ell + \max(d, b_A + h(\phi) + \pi_{\mathrm{fam}(A)}) + b_c + h(\phi'') + \pi_F
\end{aligned}
$$

Using $\max(d, \tau'_A) - d - \tau'_A = -\min(d, \tau'_A)$,

$$\begin{aligned} M_2 - M_1 &= \big(\max(d, \tau'_A) - d - \tau'_A\big) + (\tau'_A - \tau_A) + (\tau'_c - \tau_c) \\ &= -\min(d, \tau'_A) + (h(\phi) - h(\phi')) + \pi_F + (h(\phi'') - h(\phi')) \\ &= \pi_F - \min(d, \tau'_A) + R \end{aligned}$$

R is non-negative because h is non-decreasing (A3) and $\phi' \le \phi'' \le \phi$. Filling the wait is therefore beneficial when

$$\min(d, \tau'_A) \;>\; \pi_F + R,$$

i.e., the time overlapped during the external process is greater than the sum of the resumption penalty and the recovery that the wait provides. □

A.1.5 Degeneracy

Proposition 9 (Theorems depend on state)**.** *As part of our construction, if we remove state, i.e., $\tau(a, s) = \tau(a)$ for all s, then the time taken for all tasks T is now additive and order-invariant. Our theorems T1–T4 collapse to equalities, and T5 reduces to the simple identity $\Delta M = -\min(d, \tau_A)$.*

Proof. With state-free terms, order no longer matters when computing time taken. Thus, the sum T is simply permutation-invariant. Switching penalties cannot be expressed because the previous task's family is not recorded. Fatigue cost is at most constant, and skill decay and recovery cannot be expressed.

T1: There is no skill change and thus T1 simply becomes $\tau(b, \delta(a, s)) = \tau(b, s)$.

T2: There is no setup register and fatigue costs are constant, so $\tau(a_2, \delta(b, \delta(a_1, s))) = \tau(a_2, \delta(a_1, s))$, and thus $\tau(a_1, s) + \tau(a_2, \delta(b, \delta(a_1, s))) = \tau(a_1, s) + \tau(a_2, \delta(a_1, s)) = \tau(a, s)$.

T3: Waiting no longer recovers fatigue since it is at most constant. The rest vs. rust trade-off $\tau(a_2, \delta(w_d, \delta(a_1, s))) - \tau(a_2, \delta(a_1, s)) = \tau(a_2, s) - \tau(a_2, s) = 0$.

T4: All costs of the same task are equal, so batching yields no advantage:
$\tau(x_1; x_2; y_1; y_2, s) = \tau(x_1, s) + \tau(x_2, s) + \tau(y_1, s) + \tau(y_2, s) = \tau(x_1; y_1; x_2; y_2, s)$.

T5: $\pi_F = R = 0$, so only the simple $\min$ computation remains, and overlap can never hurt anymore:
$M_2 - M_1 = (\tau_\ell + \max(d, \tau'_A) + \tau'_c) - (\tau_\ell + d + \tau_c + \tau_A) = \max(d, \tau_A) - d - \tau_A = -\min(d, \tau_A)$. □

This is the core justification of our theory: state-dependence is what allows our theory to express the various findings in literature, generating the entire phenomena space. Without it, only simple scheduling arithmetic remains, which a simple duration lookup table model suffices to predict.

A.1.6 The register idealization and a countermodel

A3 makes the setup register able to only hold a single family, and the removal of that family is immediate after another task starts. Instead, an alternate view is that the register decays with foreign work, and is also restored gradually. This matches empirical findings where resumption costs grow with the length and the level of demand that the interrupting task has (Altmann & Trafton, 2002; Monk et al., 2008). Doing this would also make Theorem T2 (interruption overhead) lose its lower bound of one switching penalty, and only have overhead be positive. However, we show below using a countermodel that T4 (batching dominance) relies on having a way to reset switching costs after returning to a family, such as the current register structure, and not just any model with cross-family switching costs.

Example 10 (Cumulative interference)**.** Consider the countermodel with the following properties. Let the state s track two counters, $(\#X, \#Y)$, consisting of the number of completed tasks for families X and Y. Let $\tau(X, s) = 1 + (\#Y)^2$ and $\tau(Y, s) = 1 + (\#X)^2$. Let there be no fatigue and skill terms. Definition 2 holds; A1 holds (repeating X does not impact $\#Y$); T1 holds (each foreign task strictly raises the counter and thus the penalty).

However, in the batched condition, $X_1X_2Y_1Y_2$ costs $1 + 1 + 5 + 5 = 12$, while in the interleaved condition, $X_1Y_1X_2Y_2$ costs $1 + 2 + 2 + 5 = 10$. Here, interleaving strictly wins. The reason is that when foreign harm accumulates quadratically, the victim tasks are best run early. This is

exactly where the countermodel violates A3—switching costs are not routed through a register that an own-family task resets, which is exactly what the proof for T4 needs.

A.2 Grounded Examples of Axioms and Theorems

The axioms and theorems of Section 3.1 are stated abstractly, over families, state components, and events. To make their content concrete, we restate each one alongside a worked example in a single domain, where a woman named Kathryn is laying a brick garden wall, a unit of work being one row. The same family task we use is laying brick steps, and the foreign task is working through logic puzzles. Examples here are illustrative and are not vignettes used in the evaluation; see Appendix A.4 for the vignettes.

The axiom/theorem is first stated in **bold**. Then, a brief 1-sentence summary is provided in *italics*. Lastly, an example is given within the setting of a brick-laying primary task and a puzzle secondary task with character Kathryn.

A.2.1 Axiom grounded examples

A1 (learning). *Same-family work never lowers skill for that family.* If Kathryn laid a row of brick steps yesterday, that would not reduce her skill at tasks in the same family, such as laying a brick wall today. *Higher skill never slows base work.* If Kathryn practiced laying brick last week and got better at it, she would be no slower at laying a row today than she was before.

A2 (decay). *Time away from a family weakly erodes its skill, increasing with length.* If Kathryn has not laid any brick in two years, she is no more skilled at it than she was when she stopped, and she is at most as skilled as she would have been had she only stopped for two weeks.

A3 (screening). *The duration of a task is its base work, plus a fatigue cost, plus a resumption penalty paid only if the setup register holds a foreign family.* The time Kathryn takes to lay the next row is the time the row itself takes, plus whatever she loses to being tired, plus—if the last thing she did was a logic puzzle—the time she spends re-reading the guide string, remembering where she was, and getting back into the work. Had she instead just come off laying a step, that last term is not paid. *Performing a task sets the register to its own family.* Once she has laid the first row after the puzzle, the penalty is behind her: the second row does not pay it again.

A4 (loading). *Every task weakly raises fatigue.* Laying a row of brick leaves Kathryn no less tired than she was before she laid it, and the same is true of a logic puzzle. No task she can perform leaves her fresher than it found her.

A5 (recovery). *Rest weakly lowers fatigue, no less with its length.* Sitting down for ten minutes leaves Kathryn no more tired than she was when she sat down, and sitting down for an hour leaves her no more tired than ten minutes would have.

A.2.2 Theorem grounded examples

T1 (no free transfer). *Foreign work never speeds up a task.* If Kathryn breaks off the wall to work on a logic puzzle, the remaining rows do not go faster than they would have. The puzzle leaves her skill at brickwork untouched (it is foreign), clears her setup so that she must get back into the wall (A3), and leaves her no less tired (A4). Every channel through which it could act either does nothing or works against her.

T2 (interruption overhead). *A task interrupted by foreign work takes at least one resumption penalty longer, and each additional interruption adds at least one more.* If Kathryn stops midway through the wall to do a puzzle, the wall takes longer by at least the time it costs her to get back into it. If she stops for a puzzle three separate times, she pays that cost three times over.

T3 (rest versus rust). *Interrupting a task with rest pays no penalty but trades fatigue recovery against skill decay; the sign is open.* If Kathryn stops laying brick and rests, she does not pay a resumption penalty, because rest does not put a foreign family in her setup register. She does recover

(A5), which speeds the remaining rows, and she does lose skill (A2), which slows them. Which effect wins is not settled by the axioms: for a ten-minute break by a tired bricklayer, recovery plainly dominates; for a two-year layoff, decay does. Vignettes in this family therefore stipulate how tired she is and how perishable the skill is, or else deliberately leave both open.

T4 (batching dominance). *With negligible fatigue and decay, batching foreign work dominates interleaving.* Suppose Kathryn has two rows and two puzzles to get through. Laying both rows and then doing both puzzles (XXYY) crosses between families once. Alternating between them (XYXY) crosses three times, and each crossing costs her a resumption penalty. So long as getting back into either kind of work costs something, the batched order finishes no later, and finishes strictly earlier whenever any penalty is positive.

T5 (overlap). *Filling a wait on an external process with foreign work pays off only if the time overlapped exceeds the resumption penalty plus the recovery the wait provides.* Kathryn cannot lay the upper rows until the mortar below has set, a wait she does not control. She could spend that wait on a logic puzzle. Doing so buys her whatever puzzle work now overlaps the setting time, but costs her the penalty of getting back into the wall afterwards, plus the recovery she would have had if she had spent the wait resting. When the mortar sets in two minutes, filling the wait is likely not worth it; when it sets in an hour, it likely is.

A.3 Vignette character names

Names are drawn at random from a fixed pool of common US first names and functions purely as a rotation variable. We summed Social Security Administration national birth counts from the *Baby Names from Social Security Card Applications - National Data* dataset (U.S. Social Security Administration, 2025) for each (name, sex) pair over birth years 1980–2000, took the 200 most frequent names of each sex, and drew five names per sex by simple random sampling without replacement under a fixed seed, for ten names in total. Aside of pronouns, no other element of the vignette varies with the name, and names are not used as a factor in our analyses; the draw script, the sampled pools, and the resulting name list will be included in the replication materials.

A.4 Prompts

Every item is rendered into a single user message with no system prompt and no few-shot example: the header `Question:`, the vignette, the question, the four options in fixed order, and the answer-guideline block, which states the granularity convention the golds were derived under and the response format. The listings below give all fourteen base vignettes in the brick-wall domain, the set from which the remaining seven domains were ported: five axiom families (A3 in two variants) and four theorem families (T3 rest vs. rust and T5 overlap in three variants each). Items in the other domains and in the flipped orientation are identical in form and differ only in the vignette and question text.

Axiom tier.

Listing 1: A1, learning: practice quantity manipulated, skill level left unstated. Item `A1-001`, gold (b).

```
Question:
Kathryn has a brick garden wall to lay, three rows high. She has only
laid brick a few times before.

If instead she had laid brick plenty of times before, would the three
rows take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
```

```
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 2: A2, decay: gap since last practice manipulated. Item `A2-001`, gold (a).

```
Question:
Mariah has a brick garden wall to lay, three rows high. She last laid
brick a week ago.

If instead she had last laid brick five years ago, would the three rows
take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 3: A3, screening, interruption-vs-continue variant: a foreign interruption against uninterrupted work. Item `A3-001`, gold (a).

```
Question:
Tiffany is partway through laying a brick garden wall and has three rows
left. She isn't tired at all. She has just finished a row; the mortar
for the next one is mixed and she knows exactly where the next brick
goes. She is about to start on the next row.

If instead she stops now for a while to work on a logic puzzle before
coming back to finish the last three rows, will the three remaining rows
take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 4: A3, screening, class-comparison variant: a related interrupter against a foreign one, gap length held fixed. Item `A3-002`, gold (a).

```
Question:
Tiffany is partway through laying a brick garden wall and has three rows
left. She isn't tired at all. She has just finished a row; the mortar
for the next one is mixed and she knows exactly where the next brick
goes. Before starting the next row, she is going to spend a few minutes
adding bricks to the garden steps nearby.

If instead she spends those few minutes working on a logic puzzle, will
the three remaining rows take longer, shorter, or about the same?
```

```
(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 5: A4, loading: prior load on the morning manipulated, fatigue left as the manipulated output. Item `A4-001`, gold (a).

```
Question:
Cody has a brick garden wall to lay, three rows high, this afternoon. He
has spent the morning relaxing at home.

If instead he had spent the morning helping a friend move furniture,
would the three rows take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 6: A5, recovery: an idle rest at high fatigue, with the rest itself excluded from the measured quantity. Item `A5-001`, gold (b).

```
Question:
Kathryn is partway through laying a brick garden wall and has three rows
left. She is extremely tired. She has just finished a row and is about
to start on the next one.

If instead she sits down and rests for fifteen minutes before starting
the next row, not counting the time she spends resting, will the three
remaining rows take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Theorem tier.

Listing 7: T2, interruption overhead: k re-entries pay k switching costs, total interruption time held fixed. Item `T2-001`, gold (a).

```
Question:
Henry is partway through laying a brick garden wall and has a few rows
left. He isn't tired at all. He has just finished a row; the mortar for
the next one is mixed and he knows exactly where the next brick goes.
While laying the last few rows, he is going to stop once in the middle,
for six minutes, to work on a logic puzzle.

If instead he stops twice along the way, for three minutes each time, to
work on the puzzle, will the remaining rows take longer, shorter, or
about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 8: T3, rest vs. rust, sign forced toward rest: skill is well-established, so decay is folk-absent over the delay and recovery dominates. Item `T3-001`, gold (b).

```
Question:
Katrina has a brick garden wall to lay for a neighbor, three rows high,
and can do it either this weekend or next month. She has laid brick
plenty of times before. She is in the middle of an exhausting few weeks
at work and is extremely tired; by next month the stretch will be over.
She plans to lay the wall this weekend.

If instead she waits and lays it next month, will the three rows take
longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 9: T3, rest vs. rust, sign forced toward rust: fatigue is pinned at the floor on both occasions, so only decay is live. Item `T3-002`, gold (b).

```
Question:
Three months ago, Sara learned to lay brick and practiced twice that
week. She was going to lay a short brick garden wall for her neighbor on
the Saturday a week after, when she was free and not tired at all, but
she decided to wait. Instead, she is going to lay it this Saturday; she
knows she'll also be free and not tired at all then.

If she had laid the wall on that earlier Saturday instead, would it have
taken longer, shorter, or about the same amount of time?

(a) Longer
(b) Shorter
(c) About the same
```

```
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 10: T3, rest vs. rust, open sign: recovery and decay are both live and pinned only comparatively, so the description does not settle the sign. Item `T3-003`, gold (d).

```
Question:
Last week, Tiffany learned to lay brick and practiced twice. She has a
short brick garden wall to lay for her neighbor, and can do it either
this Saturday or on a Saturday three months from now. She is in the
middle of a busy few weeks at work and will be quite tired this
Saturday; she knows she'll be less tired on the Saturday in three
months. She is planning to lay the wall this Saturday.

If instead she waits and lays it on the Saturday in three months, will
the wall take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 11: T4, batching: alternating two tasks against batching them, with fatigue at the floor and a feasibility clause. Item `T4-001`, gold (b).

```
Question:
This afternoon, Katrina needs to lay two rows of brick on her garden
wall and solve two logic puzzles for a class she is taking. She isn't
tired at all at the moment, and neither task is tiring for her, even if
done twice in a row. She is planning to alternate: a row of brick, then
a puzzle, then the other row, then the other puzzle.

If instead she lays both rows first and then solves both puzzles, will
the afternoon's work take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 12: T5, overlap, fill wins: the fill fits inside the wait and the re-entry toll is stipulated folk-small. Item `T5-001`, gold (b).

```
Question:
```

```
This afternoon, Katrina is laying a brick garden wall and also needs to
solve a logic puzzle for a class she is taking. The wall's lower rows
are done, and their mortar needs an hour or so to firm up before it can
take the weight of the final row. She isn't tired at all. The puzzle
would take her about twenty minutes, and it only takes her a moment to
get back into the rhythm of laying. She is planning to sit and wait out
the hour, lay the final row, and then solve the puzzle.

If instead she solves the puzzle during the hour's wait, will finishing
everything take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 13: T5, overlap, fill loses: the re-entry toll is pinned comparatively above the wait it would save. Item `T5-002`, gold (a).

```
Question:
This afternoon, Keith is laying a brick garden wall and also needs to
reply to a message from his class group, which would take him a few
minutes. He has just finished laying a row and has three rows left to
lay, and the mortar he mixed for them needs to sit for ten minutes
before it's ready to use. He isn't tired at all. But once he turns to
something else, it takes him a while to find the rhythm of laying again,
more than the ten minutes the mortar needs. He is planning to wait out
the ten minutes, lay the three rows, and reply afterward.

If instead he replies to the message during the ten-minute wait, will
finishing everything take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

Listing 14: T5, overlap, open sign: the re-entry toll is declared variable over a range that straddles the wait. Item `T5-003`, gold (d).

```
Question:
This afternoon, Katrina is laying a brick garden wall and also needs to
reply to a message from her class group, which would take her about five
minutes. She has just finished laying a row and has three rows left to
lay, and the mortar she mixed for them needs to sit for ten minutes
before it's ready to use. She isn't tired at all. But once she turns to
something else, it takes her some time to find the rhythm of laying
again, a minute or two on some days and ten minutes or more on others.
She is planning to wait out the ten minutes, lay the three rows, and
reply afterward.
```

```
If instead she replies to the message during the ten-minute wait, will
finishing everything take longer, shorter, or about the same?

(a) Longer
(b) Shorter
(c) About the same
(d) Could go either way

Answer guidelines:
Count any difference, however small, as longer or shorter. Answer 'about
the same' only if it would make no difference at all. Answer in the
following format, do not include anything else:
Answer: a single letter
Explanation: 1 to 2 sentences explaining why you chose that answer.
```

A.5 Option-order sensitivity in Experiment 1

In Experiment 1, our evaluation presents the four response options in a fixed order: (a) Longer, (b) Shorter, (c) About the same, (d) Could go either way. This positioning makes the meaning of option (d) more salient, as "either way" now refers to options (a) and (b). However, a potential confounder is models' position bias, which has previously been documented in multiple choice tasks (Wang et al., 2024), where a model favors a particular slot simply because of the label.

To address this possible confounder, we ran a pilot that permutes the options in a subset of the items, and compare results against the fixed order options under otherwise identical conditions.

A.5.1 Design

Items. We drew a 64-item subset covering all nine axiom and theorem families. The subset deliberately over-samples the two families with the highest share of "(d) could go either way" ground truths, because those are the items that, if shuffled, would likely be most sensitive to answer ordering. The remaining seven families contribute two to four items each, yielding a ground truth balance of 16 (a), 16 (b), and 32 (d). As items appear in both orientations of a pairwise counterfactual structure, for the sake of balance a base item and its flip enter the subset together.

Each item was presented under two option orders:

- **Fixed.** The canonical order used in the main experiment.
- **Shuffled.** The four options permuted uniformly at random, with the gold remapped to whichever display letter carries the gold's label.

Permutations are drawn from a seed keyed on the item's base identifier and run index. Keying on the base identifier makes an item and its flip share a permutation, so that the base/flip contrast is not itself confounded with a change of layout. Keying on the run index makes successive presentations of the same item differ. Across six repeated runs, each item drew a mean of 5.4 distinct permutations (minimum 3, maximum 6), and 14 of 384 draws happened to return the identity permutation.

Sampling and decoding. Every item was presented six times per model in both conditions, resulting in giving $64 \times 6 \times 3 \times 2 = 2{,}304$ requests in total. Each request is a single turn with no system prompt, at each model's default reasoning level, matching the main experiment.

Scoring. For each answer, we invert the permutation and recover the canonical label. We report two quantities:

- **Pooled accuracy**, over all repeated runs.
- **Unanimity**, the share of items whose six replicates all returned the same canonical label.

Claude Opus 5 declined 30 of 384 rows in each arm with a content-classifier refusal. The refusals fall on the same five items in both arms (`C5-003`, `C5-003-R`, `INT-007`, `OVL-009`, `OVL-009-R`). Those rows are dropped and Claude's results are on the remaining subset of 59 items.

Table 3: Accuracy and consistency (%) under fixed and shuffled option order. *Unanimous* is the share of items answered with the same canonical label on all six shuffled reruns. Δ is shuffled minus fixed accuracy, reported with a 95% paired cluster bootstrap interval.

Model	Order	Scored	Accuracy	Unanimous	Δ accuracy (95% CI)
Claude Opus 5	Fixed	351	65.5	72.9	+2.2 (−1.9, +6.5)
	Shuffled	353	67.7	74.6	
GPT-5.6 Sol	Fixed	384	93.5	98.4	+0.3 (+0.0, +1.3)
	Shuffled	384	93.8	100.0	
Gemini 3.7 Flash	Fixed	384	94.5	96.9	−0.8 (−2.3, +0.0)
	Shuffled	384	93.8	100.0	

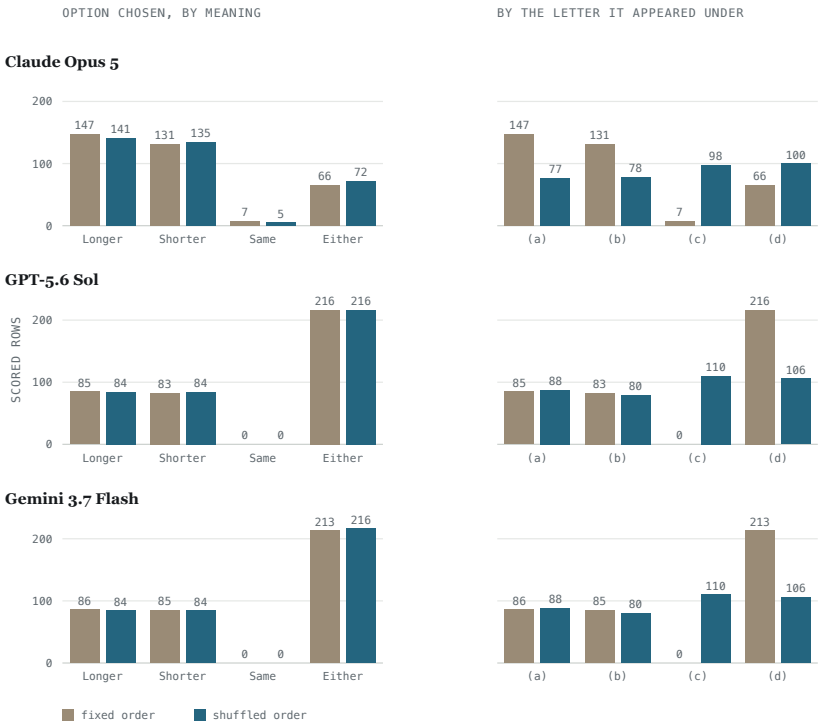

Figure 5: Scored rows by the option a model chose (left) and by the letter that option appeared under (right). Under shuffling the label counts barely move, but the slot counts flatten, indicative of a model responding to option content rather than option position.

A.5.2 Results

Table 3 gives accuracy and consistency for both conditions across three models tested. Permuting the options moves pooled accuracy by less than a percentage point for GPT and Gemini and by +2.2 points for Claude, and every 95% confidence interval on the paired difference is consistent with no effect. Each interval is the 2.5th–97.5th percentile range of 10,000 two-stage cluster-bootstrap replicates. Each replicate resamples items with replacement, and within each item, resamples that item's response rows with replacement (independently for the two conditions); the difference in pooled accuracy is recomputed on each sample.

Figure 5 compares between what the models chose and which position they chose it. Under the shuffled order, the distribution over options flattens towards uniform, with χ^2 values of 5.3, 6.4 and 6.4 on three degrees of freedom for Claude, GPT and Gemini respectively. This suggests that models are not answering based on the display option.

Table 4: Shuffled-arm pooled accuracy (percent) by the display slot the gold was permuted into, with 95% cluster-bootstrap intervals. Row counts per cell range from 73 to 116.

Model	Gold in (a)	Gold in (b)	Gold in (c)	Gold in (d)
Claude Opus 5	62.0 (47.6, 75.9)	69.9 (55.7, 82.7)	67.6 (54.0, 79.6)	71.0 (58.0, 82.4)
GPT-5.6 Sol	94.1 (86.4, 100)	91.1 (81.0, 98.8)	93.1 (85.3, 99.1)	96.2 (90.7, 100)
Gemini 3.7 Flash	94.1 (86.4, 100)	91.1 (81.0, 98.8)	93.1 (85.3, 99.1)	96.2 (90.8, 100)

The same conclusion follows from splitting the shuffled arm by the slot the gold happened to land in (Table 4). A model with a slot preference would score well when the gold coincided with its preferred letter and poorly otherwise; instead accuracy is flat across the four slots for all three models, suggesting no positional biases.

A.6 Token limits and refusals

Token limits. All prompts included an explicit cap on output and thinking tokens. GPT-5.6 Sol and Claude Opus 5 were run with a limit of 2,000 tokens and Gemini 3.7 Flash with a cap of 5,000. Claude occasionally exceeded the 2,000-token limit and returned an empty response. For those requests, we re-ran with a limit of 5,000 tokens and no truncations re-occurred.

Refusals. GPT-5.6 Sol and Gemini 3.7 Flash answered every request. Claude Opus 5 refused to answer a subset of requests due to a content-classifier refusal with category `cyber`, corresponding to a total of 13.2% of the main evaluation. As our vignettes describe ordinary work tasks and are selected with the criteria to avoid topics that trigger safety flags, we consider the refusals as an API-side classifier failure. Refused rows are treated as missing data and are dropped from all analyses. Table 5 provides counts broken down by axiom/theorem and by domain.

Table 5: Claude Opus 5 content-classifier refusals in the main run, by axiom or theorem and domain. Queries per cell is the total number of queries made in each domain per axiom/theorem row. Cells are refused responses out of runs in that row and domain. Domains: BS = brick steps, BW = brick wall, LQ = logic questions, MP = math problems, CR = customer replies, FC = fact-check claims, SP = social posts, SS = summarize sources.

		Domain								
Axiom/theorem	Queries per cell	BS	BW	LQ	MP	CR	FC	SP	SS	Total
learning	60	0	0	0	0	0	0	0	0	0/480
decay	60	0	0	0	0	0	0	0	0	0/480
screening	120	0	0	0	0	107	120	0	108	335/960
loading	60	0	0	0	0	0	60	0	0	60/480
recovery	60	0	0	0	0	0	0	0	0	0/480
interruption count	60	0	0	0	0	60	60	30	58	208/480
rest-vs-rust	180	0	0	0	0	0	0	0	0	0/1440
batching	60	0	0	0	0	0	0	0	0	0/480
overlap	180	0	0	0	0	60	104	59	60	283/1440
Total	840	0	0	0	0	227	344	89	226	886/6720

B Details for experiment 2: estimating completion time

B.1 Human task-duration estimation prompt

The human task-duration experiment uses the following prompt. The placeholder `[TASK INSTRUCTIONS]` is replaced with the complete instructions for the corresponding HCAST task. Each request is independent and contains no system prompt.

Listing 15: Prompt used to elicit human task-duration estimates.

```
Estimate how long a person from the following population would take to
complete the task below. Estimate active wall-clock working time,
including reading the instructions, investigation, implementation,
testing, and debugging if applicable. Assume that the person does not
use AI assistance and has not seen any previous attempts at the task,
and that they work independently in the supplied task environment and
can use its normal tools, documentation, terminal, and tests. Estimate
time conditional on successful task completion.

Your primary estimate is the person's completion time of the task. Also
provide 50%, 80%, and 95% uncertainty interval estimates. Do not search
for or rely on remembered benchmark results.

POPULATION
Skilled professionals in software engineering, machine learning, and
cybersecurity, with the majority having attended world top-100
universities. They have an average of about 5 years of relevant
experience. These individuals are unfamiliar with this particular task
but have the stated general background.

TASK INSTRUCTIONS
[TASK INSTRUCTIONS]

ESTIMATION INSTRUCTIONS
Return exactly these values in this format:
estimated completion time (minutes): number
50% interval: lower value, upper value
80% interval: lower value, upper value
95% interval: lower value, upper value
rationale: 1-2 sentences explaining the reasoning behind the estimates.
```

B.2 Additional results for human task-duration estimation

Table 6 reports the complete reference-containment results for the human task-duration experiment. For each model and interval level, we calculate the proportion of tasks for which the HCAST geometric-mean human completion time falls within the model's median interval across 30 repetitions. Because models predict the completion time of an individual successful attempt, while the HCAST reference is an aggregate computed from a limited number of successful attempts, these values should be interpreted as a permissive reference-containment diagnostic rather than formal predictive coverage.

Table 6: **Reference containment for human completion-time estimates.** Each cell reports the proportion and number of tasks for which the HCAST human geometric-mean reference falls within the model's median uncertainty interval across 30 repetitions.

Model	50% interval	80% interval	95% interval
Claude Opus 5	43.75% (7/16)	56.25% (9/16)	93.75% (15/16)
GPT-5.6 Sol	43.75% (7/16)	75.00% (12/16)	87.50% (14/16)
Gemini 3.7 Flash	18.75% (3/16)	31.25% (5/16)	50.00% (8/16)
All model–task pairs	35.42% (17/48)	54.17% (26/48)	77.08% (37/48)

Containment generally increases with the nominal interval width, but remains below the corresponding nominal level in nearly every condition. Gemini 3.7 Flash exhibits the largest shortfall, containing the human reference on only 50.00% of tasks at the nominal 95% level. GPT-5.6 Sol and Claude Opus 5 perform better at the widest interval, but neither model is consistently calibrated across all three levels. Aggregated across models, only 77.08% of model–task pairs contain the human reference at the nominal 95% level.

B.3 SELF-ESTIMATION PROMPTS

We use the same first-person prompt template for all three self-estimation conditions. Each API call contains no system prompt and is independent of all other calls. The placeholders `[AGENT CONFIGURATION]` and `[TASK INSTRUCTIONS]` are replaced with the corresponding agent profile and complete HCAST task instructions. The three rendered agent-configuration blocks are reported below.

Listing 16: Shared prompt template for the self-estimation experiment.

```
Estimate how long you would take to complete the task below, operating
with the agent scaffold specified in YOUR AGENT CONFIGURATION. Estimate
elapsed wall-clock time from the beginning of your agent run until
successful task completion, including model generation, scaffolding
runtime, tool use, investigation, implementation, testing, and debugging
if applicable. Assume that you do not have human assistance and do not
have access to any previous attempts at the task, and that you run
independently in the supplied task environment and can use its provided
tools, documentation, terminal, and tests. Estimate time conditional on
your successful task completion.

Your primary estimate is your completion time of the task. Also provide
50%, 80%, and 95% uncertainty interval estimates. Do not search for or
rely on remembered benchmark results.

YOUR AGENT CONFIGURATION
[AGENT CONFIGURATION]

TASK INSTRUCTIONS
[TASK INSTRUCTIONS]

ESTIMATION INSTRUCTIONS
Return exactly these values in this format:
 estimated completion time (minutes): number
50% interval: lower value, upper value
80% interval: lower value, upper value
95% interval: lower value, upper value
rationale: 1-2 sentences explaining the reasoning behind the estimates.
```

Claude Opus 4.6 configuration.

```
Model: Claude Opus 4.6
Reasoning configuration: not specified.
Agent scaffold: ReAct. The agent repeatedly selects an action, observes
    the result, and then selects its next action.
```

GPT-5.2 configuration.

```
Model: GPT-5.2
Reasoning configuration: not specified.
Agent scaffold: Triframe. Before executing an action, the scaffold uses
separate stages to provide strategy, generate candidate actions,
evaluate the candidates, and select an action to execute.
```

Gemini proxy self-estimation configuration. Gemini 3 Pro was not callable when we conducted the experiment. We therefore send the shared first-person prompt to Gemini 3.1 Pro Preview while identifying the agent configuration as Gemini 3 Pro. The resulting estimates concern the released Gemini 3 Pro configuration and are not literal self-estimates by the queried model.

```
Model: Gemini 3 Pro
Reasoning configuration: not specified.
```

```
Agent scaffold: ReAct. The agent repeatedly selects an action, observes
the result, and then selects its next action.
```

B.4 Additional self-estimation results

Table 7 reports reference containment for the self-estimation experiment. For each model and interval level, we calculate the proportion of evaluable tasks for which the geometric mean duration across successful METR runs falls within the model's median interval. Claude Opus 4.6 has ground-truth runtimes for 15 tasks because no successful released run is available for one task. GPT-5.2 and Gemini 3 Pro have ground-truth runtimes for all 16 tasks.

Table 7: **Reference containment for agent-time estimates.** Each cell reports the proportion and number of evaluable tasks for which the geometric mean duration across successful METR runs falls within the model's median uncertainty interval across 30 repetitions. The Gemini condition is proxy self-estimation: Gemini 3.1 Pro Preview estimates Gemini 3 Pro.

Estimator and target	**50% interval**	**80% interval**	**95% interval**
Claude Opus 4.6 → self	40.00% (6/15)	66.67% (10/15)	66.67% (10/15)
GPT-5.2 → self	0.00% (0/16)	0.00% (0/16)	0.00% (0/16)
Gemini 3.1 Pro Preview → Gemini 3 Pro	0.00% (0/16)	0.00% (0/16)	0.00% (0/16)
All evaluable model–task pairs	12.77% (6/47)	21.28% (10/47)	21.28% (10/47)

Reference containment is substantially worse than in the human-duration experiment. Across all evaluable model–task pairs, only 21.28% of the median 95% intervals contain the observed agent duration. Moreover, increasing Claude Opus 4.6's interval width from 80% to 95% does not contain any additional references, while the GPT-5.2 and Gemini conditions remain at zero containment at every interval level.

B.5 Robustness to additional task information

The main self-estimation experiment provides each estimator with the complete task instructions and its target agent configuration. However, the agents whose runtimes we predict can also inspect files, services, and other details of their execution environments. We therefore repeated the experiment with as much task and environment information as we could reconstruct from METR's public task repository. We used the same 16 tasks, estimator and target configurations, elicitation format, and 30 independent repetitions per estimator–task pair, yielding 1,440 parsed estimates.

For each task, we added a Task Environment and Available Materials block describing the files and services available at the beginning of the run, corresponding public-repository materials, and relevant environment and variant details. Where feasible, we also included the complete contents of relevant text files. The prompt stated that the target agent did not begin with these materials already read or understood and instructed the estimator to include the time required to inspect and comprehend them.

The additional information varied by task family. For the three debugging tasks, we included the supplied Python source files and readable tests. For the two esoteric-language tasks, we included the variant-specific sample programs and described the interpreter interface and environment. For the six hypothesis-testing tasks, we specified the black-box service protocol, input arity, and environment constraints but did not expose the protected implementation. For the two local-research tasks, we included extracted text from the two supplied papers and two supplied web pages. For the three `local_research_tex` tasks, we described the complete resource collection and included the relevant public LaTeX sources. We excluded protected graders, hidden tests, black-box implementations, answer keys, and official solutions.

The additional-information condition supplies substantially more information than the original prompt but does not recreate interactive services, executable files, or every detail of the original execution environment.

Table 8: **Effect of additional task information on self-estimates.** Forecasts are geometric means, in minutes, of task-level median predictions. Factor is the additional-information forecast divided by the original forecast. Confidence intervals are unadjusted 95% intervals from paired t-tests on log forecasts; p-values use Holm correction across the three estimators.

Estimator	Original	Additional	Factor (95% CI)	Adj. p
Claude Opus 4.6	8.93	7.01	0.785 [0.577, 1.067]	.113
GPT-5.2	18.19	11.22	0.617 [0.507, 0.751]	$< .001$
Gemini proxy	8.55	6.31	0.738 [0.567, 0.960]	.053

Effect on forecasts. We test whether the additional information changes the models' forecasts. For each estimator–task pair, we aggregate the 30 repetitions using the median point estimate. We then log-transform the task-level estimates and compare the original and additional-information conditions using paired t-tests across the same 16 tasks. The log transformation reflects the multiplicative scale of completion-time estimates. We adjust the three model-specific p-values using Holm's procedure.

Table 8 reports the results. Additional information generally shifts forecasts downward. Claude Opus 4.6's geometric-mean forecast across tasks decreases from 8.93 to 7.01 minutes, but the change is not statistically significant ($p = .113$). GPT-5.2's forecast decreases significantly, from 18.19 to 11.22 minutes (Holm-adjusted $p < .001$). The Gemini proxy's forecast decreases from 8.55 to 6.31 minutes; the difference is significant before correction ($p = .026$) but not after Holm correction ($p = .053$). Thus, after correction, only GPT-5.2 shows a statistically significant change in forecasts.

The downward shifts do not correct the original underestimation. Descriptively, GPT-5.2's median estimate-to-reference ratio decreases from 0.0052 to 0.0041, while the Gemini proxy's changes from 0.0239 to 0.0226 and Claude Opus 4.6's remains 0.627. Rank correlations remain non-positive in both conditions. Claude's 95% reference containment increases from 10/15 to 12/15 tasks, whereas GPT and Gemini contain no references at any interval level in either condition. Overall, the additional information does not consistently improve the estimates or explain the underestimation observed in the main experiment.

B.6 Cross-model task-duration estimation prompt

The cross-model experiment uses the following prompt. The placeholders `[TARGET AGENT CONFIGURATION]` and `[TASK INSTRUCTIONS]` are replaced with the target agent's profile and the complete HCAST task instructions, respectively. Each request is independent and contains no system prompt.

Listing 17: Prompt used to elicit completion-time estimates for other LLM agents.

```
Estimate how long the specified LLM agent would take to complete the
task below. Estimate elapsed wall-clock time from the beginning of the
agent run until successful task completion, including model generation,
scaffolding runtime, tool use, investigation, implementation, testing,
and debugging if applicable. Assume that the agent does not have human
assistance and does not have access to any previous attempts at the
task, and that they run independently in the supplied task environment
and can use its provided tools, documentation, terminal, and tests.
Estimate time conditional on successful task completion.

Your primary estimate is the agent's completion time of the task. Also
provide 50%, 80%, and 95% uncertainty interval estimates. Do not search
for or rely on remembered benchmark results.

TARGET AGENT CONFIGURATION
[TARGET AGENT CONFIGURATION]

TASK INSTRUCTIONS
[TASK INSTRUCTIONS]

ESTIMATION INSTRUCTIONS
```

```
Return exactly these values in this format:
estimated completion time (minutes): number
50% interval: lower value, upper value
80% interval: lower value, upper value
95% interval: lower value, upper value
rationale: 1-2 sentences explaining the reasoning behind the estimates.
```

For each target agent, the configuration block is rendered in the following format:

```
Model: [MODEL DESCRIPTION]
Reasoning configuration: [REASONING CONFIGURATION]
Agent scaffold: [SCAFFOLD NAME]. [SCAFFOLD DESCRIPTION]
```

The target profiles contain only information about the model and its execution configuration. We do not expose observed completion times, success rates, token consumption, previous trajectories, or other outcome information.

B.7 Additional results for estimating other LLM agents

Available METR data constrain our target set. We include a target-agent/task pair only when the released data contain at least one successful run, so the number of evaluable tasks varies by target. Among targets with configurable reasoning effort, the released metadata record the reasoning configuration only for Claude 3.7 Sonnet and o1; we describe it as unrecorded for the others.

Table 9 reports point-estimate bias and reference containment separately for each target agent, together with the aggregate results summarized in Section 4. The aggregate estimate-to-reference ratios correspond to observed completion times approximately $9\times$, $13\times$, and $49\times$ longer than the estimates from Claude Opus 5, GPT-5.6 Sol, and Gemini 3.7 Flash, respectively.

Underestimation predominates across target agents. The median estimate-to-reference ratio is below one for nearly every estimator–target combination; the principal exception is GPT-4 0314, which all three estimators overestimate at the median. Claude Opus 5 also slightly overestimates Claude 3.5 Sonnet 0620 and Claude Opus 4.5. The most severe underestimation occurs for o1, o3, GPT-5, GPT-5.1-Codex-Max, GPT-5.2, GPT-5.3-Codex, and Gemini 3 Pro: for every estimator, the median ratio is below 0.03 and the median 95% intervals contain none of the references for these targets.

These target-level comparisons are descriptive. The available agents differ jointly in model generation and scaffold, while METR's released metadata identify the reasoning configuration for only Claude 3.7 Sonnet and o1 among the reasoning-capable targets. Differences between targets therefore cannot be attributed solely to model identity or generation.

C Details for experiment 3: duration judgments

C.1 Prompts

Each prompt reproduces the instructions given to the human participants. The API models returned structured JSON through each provider's schema-constrained output mode. Qwen2.5-32B, served through vLLM, received the same prompt followed by an explicit instruction to answer in JSON with the corresponding fields; answers were parsed with one retry on a malformed response.

Listing 18: UDS-T, base prompt.

```
You will read a sentence from English web text (for example a blog,
review, email, or forum post). One word in the sentence is marked with
[[double brackets]]. It refers to an event or state.

What is the likely duration of the event or state referred to by the
marked word?

Choose exactly one of: instant, seconds, minutes, hours, days, weeks,
months, years, decades, centuries, forever.
```

Table 9: **Point-estimate bias and reference containment for other LLM agents.** For each target, n is the number of tasks with at least one successful released METR run. Ratio is the median task-level estimate-to-reference ratio; values below one indicate underestimation. Containment entries report the percentage and count of tasks for which the geometric mean of successful METR run times falls within the median forecast interval. "Unknown" indicates that the reasoning configuration was not recorded in the available METR metadata or specified to the estimators; "None" denotes a model without a configurable reasoning-effort setting.

				Claude Opus 5				GPT-5.6 Sol				Gemini 3.7 Flash			
Target agent	Reasoning	Scaffold	n	Ratio	50%	80%	95%	Ratio	50%	80%	95%	Ratio	50%	80%	95%
Claude 3 Opus	None	ReAct	7	0.646	28.6 (2/7)	57.1 (4/7)	85.7 (6/7)	0.287	0.0 (0/7)	28.6 (2/7)	57.1 (4/7)	0.112	28.6 (2/7)	28.6 (2/7)	28.6 (2/7)
Claude 3.5 Sonnet 0620	None	ReAct	12	1.020	41.7 (5/12)	50.0 (6/12)	66.7 (8/12)	0.462	8.3 (1/12)	25.0 (3/12)	75.0 (9/12)	0.185	8.3 (1/12)	25.0 (3/12)	33.3 (4/12)
Claude 3.5 Sonnet 1022	None	ReAct	13	0.542	7.7 (1/13)	46.2 (6/13)	69.2 (9/13)	0.241	23.1 (3/13)	38.5 (5/13)	46.2 (6/13)	0.094	0.0 (0/13)	0.0 (0/13)	23.1 (3/13)
Claude 3.7 Sonnet	Extended (32k)	ReAct	16	0.415	6.2 (1/16)	31.2 (5/16)	75.0 (12/16)	0.194	6.2 (1/16)	25.0 (4/16)	31.2 (5/16)	0.086	0.0 (0/16)	12.5 (2/16)	25.0 (4/16)
Claude 4 Opus	Unknown	ReAct	16	0.679	43.8 (7/16)	68.8 (11/16)	81.2 (13/16)	0.344	18.8 (3/16)	31.2 (5/16)	68.8 (11/16)	0.179	6.2 (1/16)	12.5 (2/16)	25.0 (4/16)
Claude 4.1 Opus	Unknown	ReAct	16	0.053	12.5 (2/16)	12.5 (2/16)	12.5 (2/16)	0.024	6.2 (1/16)	6.2 (1/16)	25.0 (4/16)	0.010	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
Claude Opus 4.5	Unknown	ReAct	15	1.114	40.0 (6/15)	46.7 (7/15)	60.0 (9/15)	0.771	46.7 (7/15)	53.3 (8/15)	53.3 (8/15)	0.288	13.3 (2/15)	26.7 (4/15)	46.7 (7/15)
Claude Opus 4.6	Unknown	ReAct	15	0.503	20.0 (3/15)	66.7 (10/15)	100.0 (15/15)	0.335	0.0 (0/15)	40.0 (6/15)	86.7 (13/15)	0.132	13.3 (2/15)	13.3 (2/15)	26.7 (4/15)
GPT-4 0314	None	Modular	8	9.220	12.5 (1/8)	12.5 (1/8)	12.5 (1/8)	4.432	12.5 (1/8)	12.5 (1/8)	25.0 (2/8)	1.740	25.0 (2/8)	50.0 (4/8)	50.0 (4/8)
GPT-4 1106	None	ReAct	9	0.823	55.6 (5/9)	55.6 (5/9)	66.7 (6/9)	0.398	11.1 (1/9)	22.2 (2/9)	66.7 (6/9)	0.158	0.0 (0/9)	11.1 (1/9)	22.2 (2/9)
GPT-4 Turbo	None	ReAct	7	0.186	14.3 (1/7)	14.3 (1/7)	28.6 (2/7)	0.077	14.3 (1/7)	14.3 (1/7)	14.3 (1/7)	0.031	0.0 (0/7)	0.0 (0/7)	0.0 (0/7)
GPT-4o	None	ReAct	10	0.737	20.0 (2/10)	60.0 (6/10)	80.0 (8/10)	0.333	0.0 (0/10)	20.0 (2/10)	70.0 (7/10)	0.136	0.0 (0/10)	20.0 (2/10)	40.0 (4/10)
o1-preview	Unknown	Duet	14	0.823	78.6 (11/14)	92.9 (13/14)	100.0 (14/14)	0.521	14.3 (2/14)	78.6 (11/14)	100.0 (14/14)	0.185	0.0 (0/14)	0.0 (0/14)	0.0 (0/14)
o1	Medium (default)	Triframe	15	0.013	0.0 (0/15)	0.0 (0/15)	0.0 (0/15)	0.006	0.0 (0/15)	0.0 (0/15)	0.0 (0/15)	0.003	0.0 (0/15)	0.0 (0/15)	0.0 (0/15)
o3	Unknown	Triframe	16	0.010	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.005	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.002	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
GPT-5	Unknown	Triframe	16	0.010	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.006	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.002	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
GPT-5.1-Codex-Max	Unknown	Triframe	16	0.013	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.008	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.003	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
GPT-5.2	Unknown	Triframe	16	0.004	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.002	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.001	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
GPT-5.3-Codex	Unknown	Triframe	16	0.014	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.008	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.003	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
Gemini 3 Pro	Unknown	ReAct	16	0.026	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.022	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)	0.006	0.0 (0/16)	0.0 (0/16)	0.0 (0/16)
All eligible pairs	—	—	269	**0.111**	**17.5**	**28.6**	**39.0**	**0.075**	**7.8**	**19.0**	**33.5**	**0.020**	**3.7**	**8.2**	**14.1**

Table 10: Duration judgments on 5000 UDS-T events. ρ: Spearman correlation with the mean human label. Bias: mean model label minus mean human label, in scale steps. Gain: sensitivity to sentence features relative to people (g; defined in Section 5), over all features and over the predicate and context groups. Brackets give 95% bootstrap intervals. †Qwen2.5-32B was run on the 1000-event subset only.

			Sensitivity gain		
Judge	ρ	Bias	All features	Predicate	Context
One annotator (reference)	0.49	–	1	1	1
Claude Opus 5	0.688 [0.67, 0.71]	+0.49	1.35 [1.22, 1.47]	1.62 [1.50, 1.76]	0.89 [0.69, 1.11]
GPT-5.6 Sol	0.619 [0.60, 0.64]	+0.01	1.27 [1.13, 1.39]	1.57 [1.40, 1.72]	0.76 [0.52, 0.99]
Gemini 3.7 Flash	0.632 [0.61, 0.65]	+0.50	1.30 [1.15, 1.42]	1.66 [1.49, 1.83]	0.69 [0.47, 0.92]
Qwen2.5-32B†	0.430	-0.91	0.67 [0.30, 1.03]	1.13 [0.75, 1.51]	0.37 [-0.17, 0.90]

```
Then say how confident you are in your duration answer:
0 = not at all confident
1 = not very confident
2 = somewhat confident
3 = very confident
4 = totally confident

Sentence: {sentence}
```

Prompt variants. *Reworded* rephrases every instruction while keeping the scale. *Reversed* lists the categories from *forever* to *instant*. *Numeric* presents the scale as numbered points and asks for a number from 0 to 10, which is mapped back to its category. The reworded and numeric prompts are:

```
Below is a sentence taken from text people wrote online, such as
reviews, emails, blog
posts, or forum threads. The word inside [[double brackets]] names an
event or a state.

How long does that event or state most likely last?

Pick one answer from this list: instant, seconds, minutes, hours, days,
weeks, months,
years, decades, centuries, forever.

Also rate your confidence in that answer, from 0 to 4:
0 = not at all confident
1 = not very confident
2 = somewhat confident
3 = very confident
4 = totally confident

Sentence: {sentence}
```

```
You will read a sentence from English web text (for example a blog,
review, email, or forum
post). One word in the sentence is marked with [[double brackets]]. It
refers to an event or
state.

What is the likely duration of the event or state referred to by the
marked word? Answer
with a number on this scale:
0 = instant
1 = seconds
2 = minutes
```

```
3 = hours
4 = days
5 = weeks
6 = months
7 = years
8 = decades
9 = centuries
10 = forever

Then say how confident you are in your duration answer:
0 = not at all confident
1 = not very confident
2 = somewhat confident
3 = very confident
4 = totally confident

Sentence: {sentence}
```

Wittenberg & Levy, Experiment 1a. The instructions are those given to participants in the original study.

```
In this survey, we would like your opinion on how long you think events
take. For example,
the event "Frida is eating an apple" might take about seven minutes;
"Frida is driving 150
miles" might take about 2 hours 15 minutes; or "Frida sneezed" might
take about one second.

Please read the following sentence carefully and estimate how long the
described event took
as best and accurately as you can. Give your estimate as hours, minutes,
and seconds.

Sentence: {sentence}
```

C.2 Models, settings and cost

We use default settings for all models. Neither Claude Opus 5 nor GPT-5.6 Sol accepts a temperature parameter, and we use temperature of 0 for Gemini. Qwen2.5-32B-Instruct (4-bit AWQ) was served with vLLM in fp16 on two NVIDIA TITAN RTX GPUs using the server's default sampling. Refusal fallbacks were disabled so that every answer comes from the named model. Table 11 lists calls, tokens and cost for all runs.

Table 11: API usage for all duration runs: the full sample, four additional runs and three prompt variants on the 1000-event subset, and pilot calls.

Model	Calls	Tokens (in / out)	Cost
Claude Opus 5	14315	7.13M / 0.21M	$40.99
GPT-5.6 Sol	12202	2.87M / 0.69M	$25.17
Gemini 3.7 Flash	12211	2.01M / 1.72M	$7.95
Qwen2.5-32B	1042	–	local
Total			$74.11

C.3 Item sampling and feature definitions

The 5000 events are a uniform random sample without replacement from the 25192 UDS-T predicates with at least three duration labels, pooling the train, development and test splits. They were drawn in two stages: a first sample of 1000 events (seed 0), which forms the subset used for repeated runs

and prompt variants, and 4000 further events from the remaining pool (seed 1). Each event is shown as its full sentence with the predicate word marked by double brackets. Features were read off the Universal Dependencies parse of the UD English Web Treebank (v1.2) on which UDS-T is annotated:

- *Predicate form*: a non-verbal predicate (adjective, noun or copular construction); otherwise the verb form, determined by its tag and auxiliaries (simple past as the reference, simple present, perfect *have*+participle, progressive *be*+gerund, imperative mood). Modal, future, infinitival and participial forms are included as further levels of the same factor but are not reported.
- *Context*: the subject type (first- or second-person pronoun as the reference; third-person pronoun, singular noun, plural noun, proper noun, or no subject); a habitual adverb modifying the predicate (*always*, *usually*, *often*, *never*, ...); an *if* marker on the predicate's clause; passive voice. Object type, negation, clause type (main, subordinate, conjunct) and question status are included as controls.

Duration phrases (*for years*), *every*-phrases and point-in-time adverbials were too rare to estimate reliably and were dropped. All regressions use ordinary least squares with heteroskedasticity-robust (HC1) standard errors; the human regression uses the mean label per event.

C.4 Per-feature effects

The two largest divergences in Figure 3a are the predicate features. Describing an event as a state rather than an action raises the human estimate by 1.38 steps and the models' by 2.69–3.00; conversely a habitual adverb (*always*, *usually*) raises the human estimate by 0.90 steps but the models' by at most 0.36. The predicate gain exceeds 1 for all three API models, and the context gain falls below 1 for GPT-5.6 Sol and Gemini 3.7 Flash (Claude Opus 5: 0.89 [0.69, 1.11]). Because the predicate and context groups were defined after inspecting the per-feature effects, we treat that split as exploratory.

Table 12 gives every feature effect underlying Figure 3a. Standard errors for the models are bootstrap standard deviations. Qwen2.5-32B† was estimated on the 1000-event subset, so the last column gives the human effects on that subset for comparison.

Table 12: Effect of each feature on the duration label, in scale steps (standard error in parentheses). Upper block: predicate features; lower block: context features.

Feature	Humans (5000)	Claude Opus 5	GPT-5.6 Sol	Gemini 3.7 Flash	Qwen2.5-32B†	Humans (1000)
Non-verbal predicate	+1.38 (0.09)	+2.76 (0.12)	+2.69 (0.14)	+3.00 (0.13)	+1.77 (0.35)	+1.16 (0.18)
Simple present	+1.14 (0.08)	+2.03 (0.11)	+2.02 (0.13)	+2.44 (0.12)	+0.99 (0.35)	+1.36 (0.18)
Perfect	+1.27 (0.14)	+1.56 (0.19)	+1.28 (0.23)	+1.03 (0.22)	+1.23 (0.50)	+0.72 (0.28)
Progressive	+0.92 (0.12)	+1.19 (0.16)	+1.33 (0.18)	+1.29 (0.16)	+0.54 (0.35)	+0.62 (0.25)
Imperative	-0.20 (0.14)	-0.37 (0.16)	-0.35 (0.18)	-0.28 (0.18)	-0.02 (0.53)	-0.03 (0.28)
Plural-noun subject	+1.15 (0.11)	+1.25 (0.14)	+1.00 (0.18)	+1.06 (0.17)	+0.20 (0.42)	+0.89 (0.24)
Named subject	+0.88 (0.11)	+0.88 (0.15)	+0.69 (0.17)	+0.85 (0.17)	+0.74 (0.43)	+0.78 (0.25)
Habitual adverb	+0.90 (0.22)	+0.36 (0.37)	+0.00 (0.44)	-0.15 (0.37)	+0.22 (1.02)	+1.63 (0.53)
If-clause	-0.67 (0.13)	-0.52 (0.19)	-0.93 (0.23)	-0.49 (0.22)	-0.90 (0.63)	-1.36 (0.29)
Passive	-0.26 (0.16)	-0.63 (0.23)	-0.86 (0.25)	-0.74 (0.25)	+0.29 (0.50)	-0.42 (0.33)

C.5 Robustness to the prompt

Across the five runs on the 1000-event subset, the pooled sensitivity gain varies by at most 0.08 (SD). Under the three prompt variants, predicate gains remain within 1.61–1.99 and context gains within 0.45–0.79 (Table 13).

C.6 Run stability and self-consistency

We report the run stability in Table 14. The left column states the model or subject, and we report same category agreement, mean difference, and standard deviations of ρ and the pooled sensitivity gain.

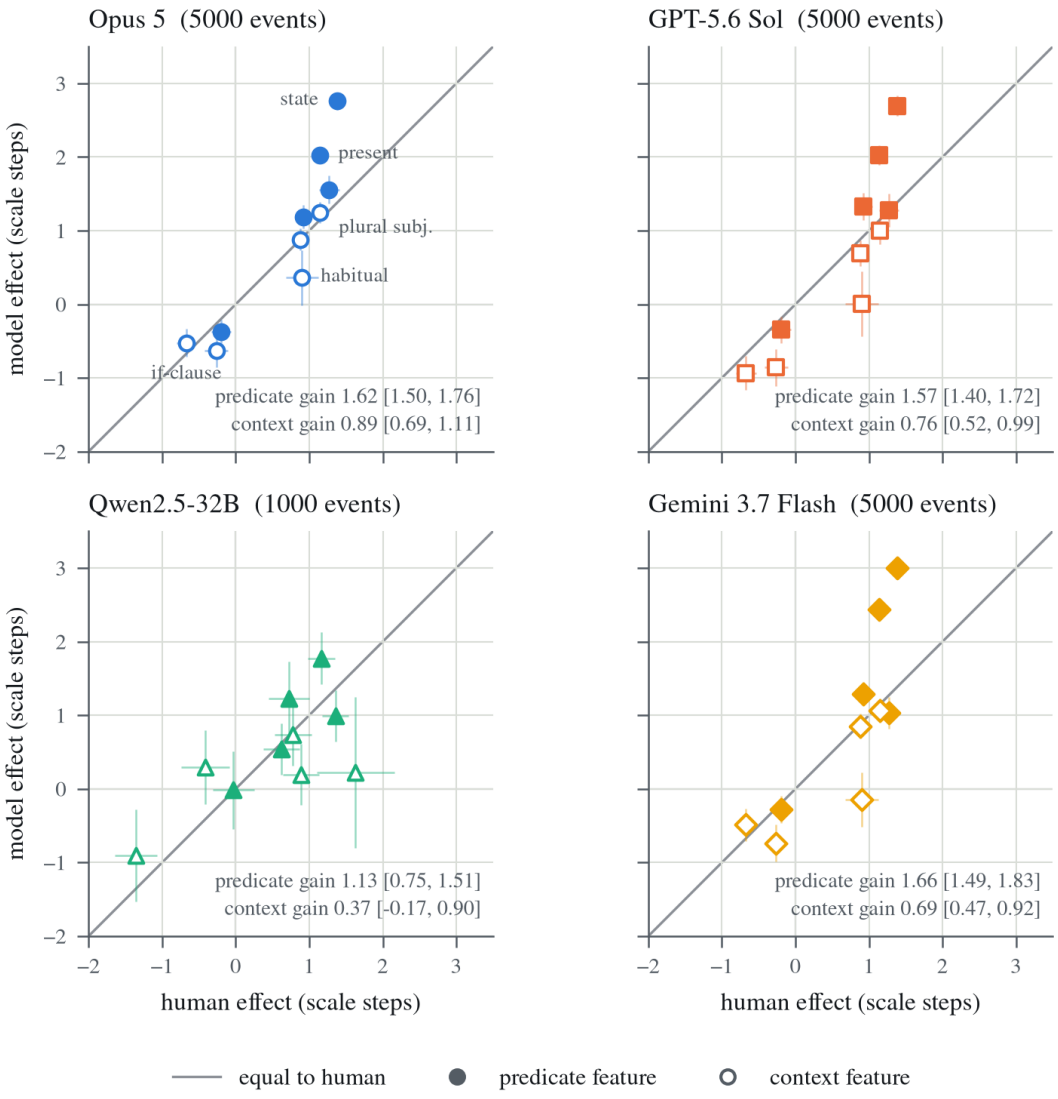

Figure 6: Per-model version of Figure 3a: human effect (x) against model effect (y) for each of the ten features, one panel per model, with ± 1 SE. Filled markers are predicate features, open markers context features. Qwen2.5-32B is estimated on the 1000-event subset.

C.7 Use of the duration scale

Models also differ from people in which parts of the scale they use (Table 15). Claude Opus 5 assigns *years* to 26% of events against 13% for people, and Qwen2.5-32B assigns *instant* to 27% against 6%.

C.8 Human judgments on the full corpus

To characterize the human judgments independently of the model comparison, we fit the same kind of regression to all 163338 UDS-T labels with annotator fixed effects and standard errors clustered by sentence. Annotator identity alone explains $R^2 = 0.107$ of the variance; adding the grammatical features raises it to 0.184, adding verb identity (lemmas with at least 30 events) instead raises it to 0.181, and adding both to 0.229. Holding grammar fixed, the verbs judged shortest denote brief acts (*thank* -1.6, *sign* -1.5, *attach* -1.5, *guess* -1.5), and those judged longest denote states and gradual changes (*live* $+2.9$, *grow* $+2.5$, *remain* $+2.2$, *support* $+2.1$).

Table 13: Agreement and sensitivity gain under each prompt, on the 1000-event subset. Base values average five runs; each variant is a single run.

Model	Prompt	ρ	Gain (all)	Gain (predicate)	Gain (context)
Claude Opus 5	base (mean of 5)	0.676	1.16	1.82	0.72
	reworded	0.651	1.09	1.85	0.57
	reversed	0.679	1.14	1.87	0.64
	numeric	0.653	1.12	1.85	0.62
GPT-5.6 Sol	base (mean of 5)	0.591	1.05	1.73	0.58
	reworded	0.608	0.97	1.64	0.52
	reversed	0.600	1.12	1.61	0.79
	numeric	0.613	0.98	1.76	0.45
Gemini 3.7 Flash	base (mean of 5)	0.634	1.15	1.94	0.62
	reworded	0.625	1.13	1.95	0.57
	reversed	0.636	1.19	1.99	0.65
	numeric	0.627	1.04	1.91	0.45

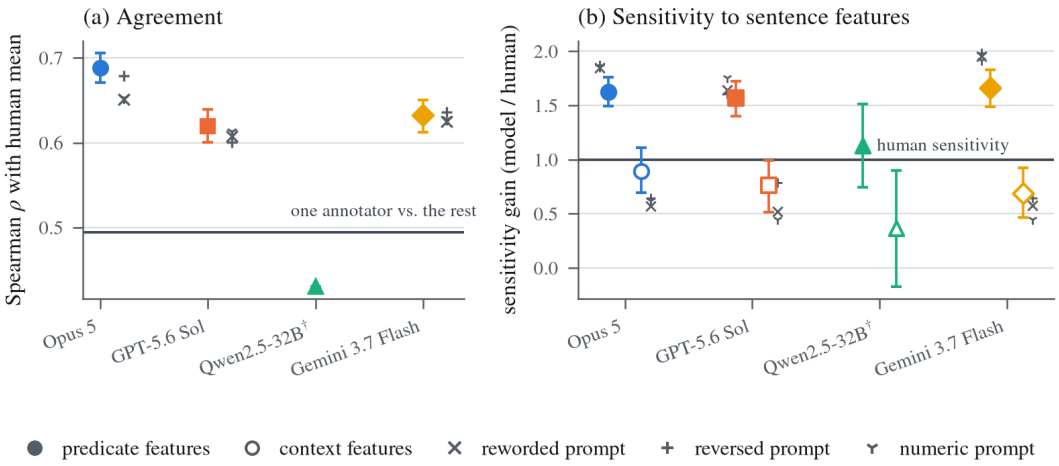

Figure 7: (a) Correlation between each model's labels and the mean human label, with the single-annotator reference. (b) Sensitivity gain by feature group with 95% intervals; small gray markers give the three prompt variants. Both panels use the 1000-event subset, where the repeated runs and prompt variants were collected.

C.9 Wittenberg & Levy items

On the minimal pairs of Wittenberg & Levy (2017), people judge the light-verb version as shorter for all three event classes: 0.46× for punctive events, 0.75× in mass syntax and 0.85× in count syntax. The API models shorten punctive (0.50–0.61×) and mass-syntax events (0.25–0.64×), but lengthen count-syntax events (1.26–1.46×), the class for which human estimates barely moved. Because these items hold the event fixed, the divergence cannot be attributed to different beliefs about the events themselves.

Light-verb sentences were built from each item's simple-verb sentence (Appendix A of Wittenberg & Levy (2017)) using the noun forms in their Table 1, following their example (*After their first date, Douglas kissed Mary* → *… Douglas gave a kiss to Mary*). Four of their 24 events (*condoling*, *punching*, *respecting*, *sermoning*) have no sentence in the published appendix and were excluded. Human values are medians of the released per-participant estimates; model values are medians over five runs (Qwen2.5-32B: one run). Ratios below 1 mean the light-verb version was judged shorter.

Table 14: Agreement between two annotators on the same event, and between two runs of the same model, on the 1000-event subset. The last two columns give the standard deviation of ρ and of the pooled sensitivity gain across five runs.

	Same category	Mean difference (steps)	SD of ρ	SD of gain
Two annotators	29.2%	1.82	–	–
Claude Opus 5, two runs	92.2%	0.15	0.004	0.058
GPT-5.6 Sol, two runs	80.4%	0.45	0.007	0.080
Gemini 3.7 Flash, two runs	89.4%	0.20	0.008	0.035

Table 15: Share of labels (%) in each duration category. Humans and the API models: 5000 events. Qwen2.5-32B†: 1000-event subset.

	inst.	sec.	min.	hours	days	weeks	months	years	dec.	cent.	forever
Humans	6.0	8.6	27.7	11.5	11.1	6.5	10.2	12.5	2.6	0.7	2.6
Claude Opus 5	2.4	6.6	27.2	11.5	8.8	4.8	8.1	26.3	2.0	0.4	2.0
GPT-5.6 Sol	9.8	13.1	21.0	10.3	7.7	2.9	8.5	22.7	1.0	0.4	2.6
Gemini 3.7 Flash	5.1	10.5	19.1	11.3	10.0	4.2	11.7	20.6	2.7	0.9	4.0
Qwen2.5-32B†	27.2	5.1	34.5	4.6	7.7	1.0	3.2	9.7	0.4	0.5	6.1

C.10 Refusals and failures

Claude Opus 5 declined one UDS-T sentence, which mentions the Ames strain of anthrax, in every run (`stop_reason=refusal`); it is excluded from that model's statistics.

D Details for Experiment 4: decisions involving time

D.1 Prompts and tasks

Placeholders in braces are substituted at run time. `{question}` is replaced with the task text; `{tool_description}` is replaced with the tool block below; `{tool_delay_seconds}` is replaced with the stated time cost Δt; `{reasoning_level}` is replaced with the model's default reasoning level; `{calculator_query}` and `{calculator_results}` are replaced with the model's query and the tool output. Each request is independent and contains no system prompt. We only show the reasoning-condition prompts for the mathematics task. The no-reasoning templates are identical except that the `Reason:` field is removed, "at the {reasoning_level} reasoning level" is replaced with "without any reasoning", and "You will be reasoning at the {reasoning_level} reasoning level." is replaced with "You will not be allowed to reason." The factual question-answering templates are identical except that the decision stage asks for a `Query:` field containing a free-text search query in place of `Expressions:`, answers are requested as "your answer" rather than "all roots, comma separated", and Stage 4 refers to the search tool, its query, and its results. In the no-delay baseline, the sentence "The tool takes {tool_delay_seconds} seconds to finish." is removed from Stages 2 and 3; the prompts are otherwise identical.

Stage 1: confidence without the tool.

```
Question:
{question}

No tool is available to you yet.
Do not solve or answer the question yet.

Estimate the probability from 0 to 1 that you would answer this question
    correctly WITHOUT the help of any tools, at the {reasoning_level}
    reasoning level. Do not solve the question in order to provide the
    estimate.

Respond in this exact format, do not output anything else:
Estimate: your estimate as a single number from 0 to 1
```

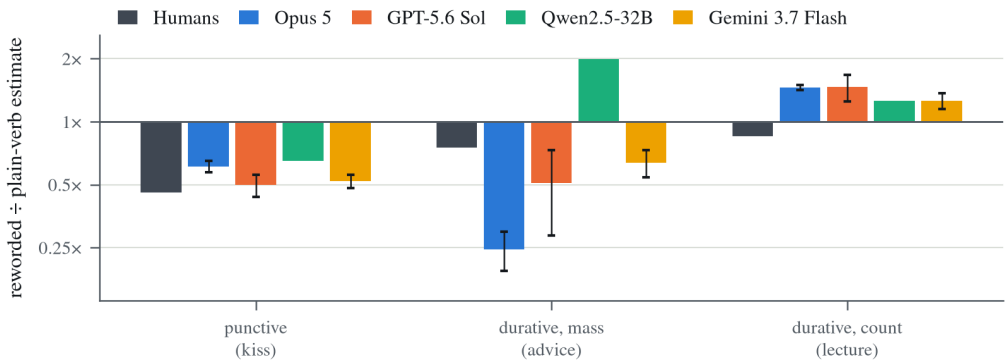

Figure 8: Per-model version of Figure 3c, including Qwen2.5-32B: light-verb estimate divided by simple-verb estimate, as a geometric mean over the events of each class (log scale). Error bars: SD across five runs.

Table 16: Light-verb / simple-verb duration ratio for each Wittenberg & Levy event.

Event	Class	Humans	Claude Opus 5	GPT-5.6 Sol	Gemini 3.7 Flash	Qwen2.5-32B
embracing	punctive	0.50	0.80	1.00	0.83	1.67
hugging	punctive	1.21	1.00	1.00	1.00	1.00
cuddling	punctive	0.10	0.07	0.03	0.03	0.30
kicking	punctive	0.67	1.00	0.67	1.00	1.00
shaking	punctive	0.41	0.67	0.50	1.00	0.17
poking	punctive	0.40	1.00	1.00	1.00	1.00
kissing	punctive	0.67	1.00	0.40	0.60	0.60
advising	durative, mass	0.57	0.67	0.50	1.00	0.50
recognizing	durative, mass	0.67	0.17	1.00	0.67	1.00
encouraging	durative, mass	0.67	1.00	0.50	1.00	1.00
assuring	durative, mass	0.77	1.00	3.00	2.00	1.00
assisting	durative, mass	0.69	0.33	0.12	0.06	0.50
thanking	durative, mass	1.00	1.67	4.00	1.00	1.67
supporting	durative, mass	1.00	0.00	0.33	0.49	300.00
addressing	durative, count	1.20	1.00	1.00	1.00	2.00
lecturing	durative, count	1.02	1.25	1.00	1.00	0.50
scolding	durative, count	1.00	1.00	1.00	1.00	2.00
presenting	durative, count	0.30	1.67	3.00	3.00	1.00
speaking	durative, count	1.42	1.20	1.00	1.17	2.00
talking	durative, count	0.75	3.00	1.50	1.00	1.00

```
Reason: briefly provide your reasoning
```

Stage 2: confidence with the tool.

```
You now have access to one query for the following tool.

{tool_description}

The tool takes {tool_delay_seconds} seconds to finish.
Do not solve or answer the question yet.

Estimate the probability from 0 to 1 that you would answer this question
    correctly WITH the help of this tool, at the {reasoning_level}
    reasoning level. Do not solve the question in order to provide the
    estimate.

Respond in this exact format, do not output anything else:
Estimate: your estimate as a single number from 0 to 1
Reason: briefly provide your reasoning
```

Stage 3: decision.

```
You need to answer the following question.

Question:
{question}

You have access to one query for the following tool:

{tool_description}

The tool takes {tool_delay_seconds} seconds to finish.

You must choose exactly one of two mutually exclusive actions:
1. Query the tool.
2. Answer directly without querying the tool.

You will only be allowed to query the tool once.

You will be reasoning at the {reasoning_level} reasoning level. Once you
    begin your response, do not revise, replace, or append another action.


If you choose to query the tool, output exactly one 'Expressions:' field.
     Do not output an 'Answer:' field. Respond in this exact format, do
    not output anything else:
Expressions: [expression 1, expression 2, expression 3, ...]
Reason: briefly provide your reasoning for why you chose to query the
    tool

If you choose to answer directly, output exactly one 'Answer:' field. Do
    not output an 'Expressions:' field. Respond in this exact format, do
    not output anything else:
Answer: all roots, comma separated
Reason: briefly provide your reasoning for why you chose not to query the
     tool
```

Stage 4: answer after a tool query.

```
You chose to query the calculator tool.

Your query was:
{calculator_query}

Tool results:
{calculator_results}

You have used up your one query for the calculator tool. Next, provide
    your final answer.

Respond in this exact format, do not output anything else:
Final Answer: all roots, comma separated
Reason: briefly provide your reasoning for your final answer
```

Tool descriptions.

```
CALCULATOR
Evaluates fully numerical mathematical expressions.
It cannot solve equations or perform symbolic algebra.
Real and complex arithmetic are supported.
Each expression is evaluated independently and returns a numerical value.
Returns an array of decimal values in the same order as the given
    expressions.

A list of self-contained mathematical expressions to evaluate.
Each expression is evaluated independently and returns a single number.
```

```
Supported operations include:
   - +, -, *, /, and ^ or ** for powers
   - parentheses
   - sqrt(), cbrt(), abs(), log(), exp()
   - sin(), cos(), tan(), asin(), acos(), atan()
   - scientific notation such as 1e10 or 2.5e-3
   - the constants pi and E, where E is Euler's number
   - complex arithmetic using I as the imaginary unit

Not supported; these return an error rather than a value:
   - equation solving. There is no solve(). Send an expression to
      evaluate, not an equation to solve.
   - "=" anywhere, including assignments such as "p = b^2 - 3*c".
   - variables. Expressions cannot contain free variables.

Complex results are returned, written like "1.0 + 1.73205*I".

Undefined results such as 1/0 are reported as errors.

Example:
   ["125 / 8", "sqrt(72)", "3^4 - 10", "2*sqrt(19/3)*cos(acos(-0.5)/3)"]

WEB SEARCH
Searches the web for up-to-date information relevant to the query.
Returns a set of search results with titles, URLs, snippets, and source
    metadata.
```

D.2 Example Tasks

Example tasks.

```
EASY: Suppose 2*q**3 - 18*q**2 - 20*q + 336 = 0. What is q?
MEDIUM: Let -5*b**3 - 125*b**2 + 440*b + 560 = 0. What is b?
HARD: What is v in -4*v**3 + 1099208*v**2 - 75512841472*v - 453116621184
    = 0?

FRESHQA: What's the current Tomatometer score of the most recent season
    of The Lord of the Rings: The Rings of Power on Rotten Tomatoes?
FRESHQA: Who is the latest winner of The Voice US?
FRESHQA: Who won the most recent Super Bowl?
```

D.3 Task Generation

The 30 cubic equations were generated with a wrapper around DeepMind's Mathematics Dataset (Saxton et al., 2019) that restricts its `algebra__polynomial_roots` module to degree-3, solve-for-the-variable problems; upstream samples degrees 2–5 and asks for a factorization about half the time. The three difficulty levels are the upstream entropy schedule, which increases the magnitude of the coefficients. Every equation is built from rational roots and factors exactly, and ten were drawn at each difficulty level.

D.4 Data Exclusion

We exclude 289 of 28,799 trials whose response visibly exposes problem-solving or reasoning (mathematical work, verification, rejection of an answer candidate, self-evaluation, or reasoning about the required format) anywhere other than the requested `Reason:` field, including inside an answer or within a tool query. Because such text can serve as extra computational space the prompt protocol did not provide other trials, these trials are treated as protocol-noncompliant. Presentation changes, malformed output, and unexplained answer replacements are retained, as is reasoning the model performs internally without exposing it. The exclusion is concentrated in Claude Opus 5 without reasoning on cubic equations, which includes 221 of the 289.

D.5 Model-Call Durations

Table 17 reports median differences in model-call duration between matched tool-use and no-tool trials. This pattern is consistent with API overhead contributing to the differences, but each call also includes model computation and response generation. These measurements therefore cannot isolate internal reasoning time or establish that reasoning durations are equivalent.

Table 17: **Model-call durations.** All results use default reasoning. For each task and stated time cost where both tool and no-tool trials occur, we subtract the mean no-tool duration from the mean tool duration. Then, the differences are averaged equally across matched delays within each task, and the median across matched tasks is reported. Duration is the sum of recorded model-call latencies in the task-answering conversation, excluding confidence estimation and tool execution. Positive values indicate longer model-call durations with tool use.

Domain	Model	Matched tasks	Median difference (s)
Cubics	GPT-5.6 Sol	16	+6.69
	Gemini 3.7 Flash	13	+2.77
	Claude Opus 5	0	—
Search	Claude Opus 5	4	+2.06
	GPT-5.6 Sol	6	+2.46
	Gemini 3.7 Flash	14	+1.29

D.6 Task Performance

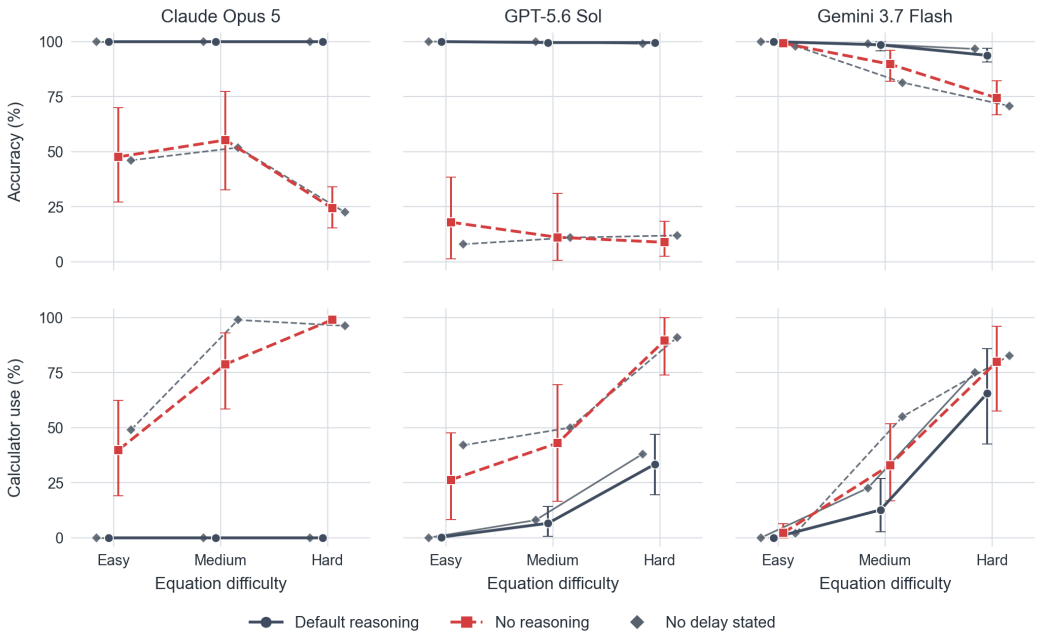

Figure 9: **Accuracy and calculator use by equation difficulty.** Each point averages the 10 tasks at that level, pooled over all seven Δt; bars are 95% bootstrap intervals over tasks. Gray diamonds beside each series mark the no-delay baseline. Accuracy under default reasoning remains high across levels. Without reasoning it is lower on hard than on easy equations, though not monotonically in every condition. Calculator use generally increases with difficulty, except for Claude Opus 5 under default reasoning, which never queries the calculator.

Figure 9 reports the accuracy and calculator use in the three cubic difficulty levels, along with the no-delay baseline. Across models, overall accuracy ranges from 97.4% to 100% with reasoning and from 12.7% to 87.8% without reasoning (these averages pool all three difficulty levels). Claude with reasoning answers all trials correctly without ever querying the calculator. In the other conditions, calculator use increases with equation difficulty. This does not by itself show an accuracy benefit, since harder equations are both solved with the calculator more often and answered incorrectly more often. The no-delay baseline shows the same pattern, where accuracy ranges from 98.6% to 100% with reasoning and from 10.3% to 83.4% without, and calculator use is higher on harder questions.

D.7 Results Summary

In Table 18, Spearman's ρ measures if models use the tool more often on tasks where they expect it to be more helpful. For each of the 30 tasks, we compare the median reported Δp with the overall tool-use rate across Δt, then compute the rank correlation across tasks. ρ is undefined when either quantity does not vary across tasks.

The slope columns (pp and OR) measure if models use the tool less as the stated time cost increases, comparing each task only with itself. Both are estimated from individual trials where each trial's binary choice (tool or no tool) is related to $\log_{10} \Delta t$, with each task given its own baseline rate of tool use and uncertainty computed over tasks rather than trials. *pp* comes from a linear model and gives the change in the probability of using the tool each time Δt grows $10\times$. *OR* comes from a logistic model and gives how much the odds of using the tool are multiplied over the same increase. A 1.0 OR means no change and values below 1.0 mean less tool use. *Var.* counts the tasks on which the model both used and declined the tool at least once. Because each task is compared only with itself, tasks on which the model always or never used the tool carry no information about the slope.

N is the number of trials kept after exclusions. *Tool* and *Acc.* are first computed for each task and then averaged across the 30 tasks. Δp is the median of the model's reported benefit from the tool, in percentage points. ρ is Spearman's rank correlation across the 30 tasks between each task's median Δp and its tool usage.

Table 19 reports the same quantities but for the baseline, in which no time cost is stated.

Table 18: **Results Summary.** Tool use and accuracy are unweighted means over the 30 task-level rates; accuracy is under the model's own chosen action. Slopes are per tenfold increase in Δt. *Var.* counts tasks that both used and declined the tool.

Model	Reas.	N	Tool	Acc.	Δp	ρ	pp [95% CI]	OR [95% CI]	Var.
Cubic equations									
Claude Opus 5	Def.	2,100	0.0%	100.0%	3	–	–	–	0/30
	None	1,912	72.5%	42.4%	10	0.79	−7.1 [-10.8, -3.4]	0.26 [0.14, 0.48]	18/30
GPT-5.6 Sol	Def.	2,100	13.4%	99.7%	7	0.86	+0.1 [-1.0, +1.2]	1.02 [0.88, 1.18]	16/30
	None	2,100	53.0%	12.7%	19	0.91	+0.5 [-1.0, +2.1]	1.08 [0.85, 1.37]	13/30
Gemini 3.7 Flash	Def.	2,068	26.1%	97.4%	18	0.81	−3.1 [-5.2, -0.9]	0.56 [0.42, 0.74]	13/30
	None	2,076	38.4%	87.8%	15	0.83	−4.2 [-6.3, -2.0]	0.55 [0.43, 0.69]	15/30
FreshQA search									
Claude Opus 5	Def.	2,100	94.9%	71.0%	38	0.58	−2.4 [-5.2, +0.5]	0.05 [0.03, 0.09]	4/30
	None	2,098	98.1%	66.6%	52	0.31	−1.3 [-3.9, +1.3]	–	1/30
GPT-5.6 Sol	Def.	2,100	94.8%	62.6%	40	0.68	−0.0 [-0.6, +0.5]	0.98 [0.74, 1.30]	6/30
	None	2,100	99.5%	64.1%	44	0.50	+0.0 [-0.1, +0.1]	1.01 [0.85, 1.19]	3/30
Gemini 3.7 Flash	Def.	2,100	88.0%	56.0%	30	0.76	−10.5 [-15.3, -5.8]	0.06 [0.02, 0.13]	15/30
	None	2,099	86.2%	44.0%	29	0.71	−10.0 [-14.5, -5.5]	0.11 [0.05, 0.21]	20/30

D.8 Perceived Benefit and Tool Choice

Figure 10's curves come from a separate descriptive fit, estimated separately for each condition over all seven Δt: a binomial GLM on task–delay cell rates, weighted by retained trials,

$$\operatorname{logit} \pi_c = \beta_0 + \beta_1 z_c^p + \beta_2 z_c^t + \beta_3 z_c^p z_c^t, \tag{1}$$

Table 19: **No-delay baseline summary.** Same quantities as Table 18 for trials in which no time cost is stated. Tool use and accuracy are unweighted means over the 30 task-level rates; accuracy is under the model's own chosen action.

Model	Reas.	N	Tool	Acc.	Δp	ρ
Cubic equations						
Claude Opus 5	Def.	300	0.0%	100.0%	4	–
	None	267	81.4%	40.2%	13	0.59
GPT-5.6 Sol	Def.	300	15.3%	99.7%	7	0.61
	None	300	61.0%	10.3%	18	0.89
Gemini 3.7 Flash	Def.	295	32.5%	98.6%	18	0.80
	None	295	46.6%	83.4%	15	0.70
FreshQA search						
Claude Opus 5	Def.	300	96.7%	76.0%	36	0.29
	None	300	100.0%	66.3%	53	–
GPT-5.6 Sol	Def.	300	97.7%	58.0%	40	0.38
	None	300	100.0%	61.0%	49	–
Gemini 3.7 Flash	Def.	300	99.3%	41.3%	30	0.37
	None	300	97.0%	39.0%	30	0.43

where π_c is the probability of tool use in cell c, z_c^p is the standardized median Δp of the cell, and z_c^t is the standardized $\log_{10} \Delta t$. The fit has no task fixed effects. Each curve spans only the Δp range observed at that Δt. A rightward shift from 1s to 20 minutes is consistent with a larger perceived accuracy gain being needed to offset the longer stated Δt.

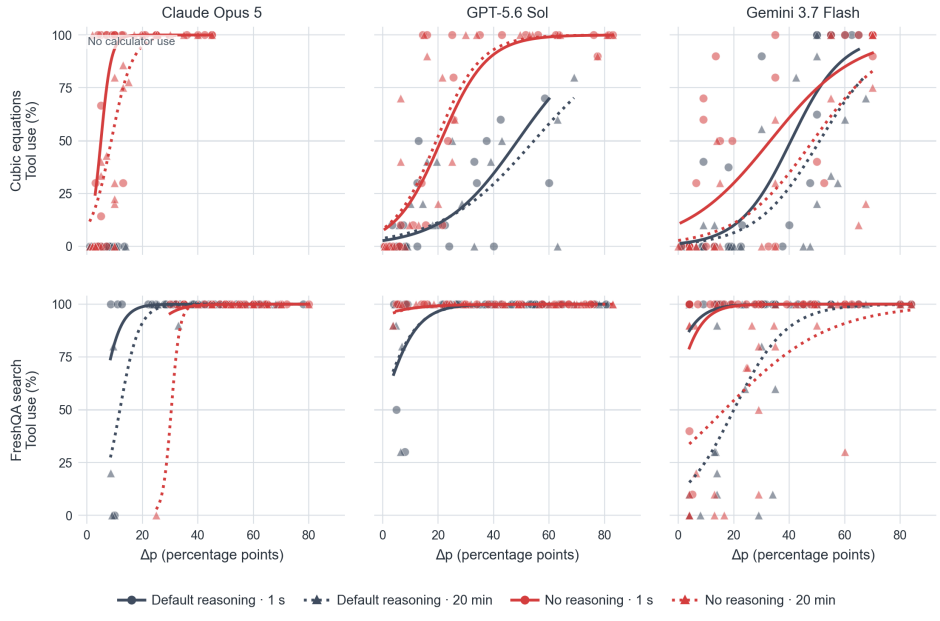

Figure 10: **Perceived accuracy benefit and tool choice at the shortest and longest Δt.** Points are task–Δt cells, with median reported benefit Δp on the horizontal axis and tool-use rate on the vertical axis. Color denotes reasoning condition; marker and line style denote the Δt. Curves are descriptive logistic fits in Δp, $\log_{10}(\Delta t)$, and their interaction, each drawn only over the Δp range observed at that Δt, so they should be compared where those ranges overlap. Claude with reasoning on cubics is labeled rather than fitted, since it never used the calculator. Several search panels sit near ceiling with few tasks varying, and their curves should be read accordingly (Table 18).

D.9 Basic Descriptive Statistics

Table 20, Table 21, Table 22 report basic descriptive statistics of the data.

Table 20: **Tool use by stated time cost.** Each entry is the unweighted mean of the 30 task-level rates at that Δt, matching Figure 4. The first column is the no-delay baseline.

Model	Reas.	No Δt	1 s	5 s	10 s	20 s	30 s	3 min	20 min
Cubic equations									
Claude Opus 5	Def.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	None	81.4	77.3	76.9	71.1	79.6	73.8	67.0	50.2
GPT-5.6 Sol	Def.	15.3	13.3	15.0	12.7	12.7	13.0	12.7	14.7
	None	61.0	51.7	55.7	51.3	50.3	53.7	54.3	53.7
Gemini 3.7 Flash	Def.	32.5	32.7	30.1	26.6	22.5	24.0	25.1	22.0
	None	46.6	46.3	39.8	38.6	36.3	40.2	35.7	31.5
FreshQA search									
Claude Opus 5	Def.	96.7	96.7	96.3	96.7	95.7	96.3	93.0	89.7
	None	100.0	100.0	100.0	99.3	97.7	96.7	96.7	96.7
GPT-5.6 Sol	Def.	97.7	96.0	94.7	94.7	94.3	94.0	93.3	96.3
	None	100.0	100.0	99.7	99.3	98.7	99.3	100.0	99.7
Gemini 3.7 Flash	Def.	99.3	100.0	95.7	90.3	91.3	90.7	81.0	66.7
	None	97.0	95.0	94.3	93.0	88.0	88.7	78.7	65.7

Table 21: **Calculator use and accuracy by equation difficulty.** Cubic tasks only, ten per level. Each entry is the unweighted mean of the ten task-level rates, pooled over all seven Δt.

		Calculator use (%)			Accuracy (%)		
Model	Reas.	Easy	Medium	Hard	Easy	Medium	Hard
Claude Opus 5	Def.	0.0	0.0	0.0	100.0	100.0	100.0
	None	39.8	78.8	99.0	47.7	55.3	24.3
GPT-5.6 Sol	Def.	0.3	6.6	33.4	100.0	99.6	99.6
	None	26.3	43.0	89.6	18.0	11.1	8.9
Gemini 3.7 Flash	Def.	0.0	12.6	65.6	100.0	98.5	93.8
	None	2.1	33.0	79.9	99.3	89.7	74.4

Table 22: **Elicited confidence estimates.** Medians over retained trials. $p_{\text{no-tool}}$ and p_{tool} are the model's stated probabilities of answering correctly without and with the tool; Δp is their difference. Cubic entries are broken out by difficulty.

		Median $p_{\text{no-tool}}$ (%)			Median Δp (pp)		
Model	Reas.	Easy	Med.	Hard	Easy	Med.	Hard
Cubic equations							
Claude Opus 5	Def.	95	91	85	2	4	8
	None	85	75	35	5	10	35
GPT-5.6 Sol	Def.	97	95	65	3	4	33
	None	93	85	35	6	13	63
Gemini 3.7 Flash	Def.	95	85	35	4	14	50
	None	95	80	20	4	15	70
FreshQA search		(no difficulty levels)					
Claude Opus 5	Def.	55			38		
	None	35			52		
GPT-5.6 Sol	Def.	55			40		
	None	55			44		
Gemini 3.7 Flash	Def.	65			30		
	None	70			29		