CAN LLMS PERCEIVE TIME? AN EMPIRICAL INVESTIGATION

Aniketh Garikaparthi
TCS Research
aniketh.g@tcs.com

ABSTRACT

Large language models cannot estimate how long their own tasks take. We investigate this limitation through four experiments across 68 tasks and four model families. Pre-task estimates overshoot actual duration by 4–7× ($p < 0.001$), with models predicting human-scale minutes for tasks completing in seconds. Relative ordering fares no better: on task pairs designed to expose heuristic reliance, models score at or below chance (GPT-5: 18% on counter-intuitive pairs, $p = 0.033$), systematically failing when complexity labels mislead. Post-hoc recall is disconnected from reality—estimates diverge from actuals by an order of magnitude in either direction. These failures persist in multi-step agentic settings, with errors of 5–10×. The models possess propositional knowledge about duration from training but lack experiential grounding in their own inference time, with practical implications for agent scheduling, planning and time-critical scenarios.

1 INTRODUCTION

As LLM agents increasingly tackle longer-horizon tasks (Kwa et al., 2025; Wijk et al., 2025; Garikaparthi et al., 2026; Liu et al., 2026), they are also broadly deployed in settings that require planning, delegation, tool use, and coordination across multiple workers or sub-agents (Xie et al., 2024; Tran et al., 2025; Piskala, 2026). In such systems, accurate temporal self-estimation becomes part of the control problem: a manager agent must decide whether to call a fast heuristic or a slower specialist, whether to parallelize or serialize branches, when to stop exploration and commit, and how to allocate limited test-time compute across subtasks (Gray et al., 2023; Xu et al., 2025; Erdogan et al., 2025; Paglieri et al., 2025; Simhi et al., 2025). These decisions become harder in nested settings, where one scheduler calls other schedulers and timing errors compound across levels. A system that cannot estimate how long its own actions take cannot schedule itself reliably.

This matters anywhere latency, deadlines, or scarce resources constrain behavior. In interactive computer-use settings, agents already operate under tool, budget, and execution constraints (Xie et al., 2024; Yao et al., 2025; Barres et al., 2025). In policy and control settings, punctuality is itself part of task success rather than a secondary concern (Jia & Chen, 2025; Xu et al., 2024). In time-critical domains such as medical triage, emergency response, and real-world resource allocation, misjudging duration can have direct operational consequences (Shi et al., 2025; Sehgal et al., 2026; Keding & Meissner, 2021; Sravanthi et al., 2023). If LLMs are to act as planners, orchestrators, or managers, they need at least coarse temporal calibration about their own behavior.

This question is related to, but distinct from, prior work on time in language models. Existing work has studied temporal fact recall, temporal reasoning over events, external time-series forecasting, temporal point processes, and spatio-temporal understanding in video (Ding & Wang, 2025; Herel et al., 2024; Jin et al., 2024; Chen et al., 2026; Cheng et al., 2025; Song et al., 2025; Zhao et al., 2025). Those settings ask whether models can reason about time in the world. Our question is different: can a model estimate the duration of its own computation and actions? This is a problem of temporal self-estimation. It requires mapping internal processing and task demands to elapsed time, not merely recalling that one event happened before another or that a class of activities usually lasts a certain duration.

Humans develop such judgments through repeated interaction with the world. They act, wait, observe progress, and update expectations from feedback. That kind of calibration depends on continuous temporal grounding. A broad line of work in world models, predictive representation learning, embodied intelligence, and action-grounded systems argues that prediction and planning improve when a system learns from temporally extended interaction rather than text alone (Ha & Schmidhuber, 2018; Assran et al., 2023; 2025; Bruce et al., 2024; Feng et al., 2025; Li et al., 2026; Lee et al., 2025; Guo et al., 2026). Standard LLM inference lacks this grounding. The model observes tokens, not elapsed time. It does not directly perceive wall-clock duration while generating, and it does not accumulate sensorimotor memories of how long similar acts took. The eventual duration of a response also depends on factors such as model size, hardware, batching, tool latency, and network effects, none of which are represented in the prompt.

The picture becomes less stable as deployment increasingly decouples model identity from latency. Some providers now expose a single model under different speed modes.[1][2] Specialized inference hardware and newer sequence-model architectures make the mapping from semantic task difficulty to wall-clock duration even less direct. State-space models and diffusion language models change the underlying compute pattern of generation itself (Gu & Dao, 2023; Dao & Gu, 2024; Nie et al., 2025). Runtime, therefore, is increasingly a property of the full deployment stack and architecture.

We investigate this limitation through four experiments across 68 tasks and four model families. We first study absolute calibration: before acting, can models estimate how long their own task execution will take? We then test relative judgment: even if absolute estimates are poor, can models at least identify which of two tasks will take longer? Next, we examine post-hoc recall: after completing a task, can a model report how long it actually took? Finally, we test whether the same limitation persists in multi-step agentic settings, where timing errors can propagate through planning, delegation, and resource allocation.

Our findings reveal a consistent pattern. Pre-task estimates overshoot actual duration by 4–7$\times$ ($p < 0.001$), often assigning human-scale minutes to tasks that finish in seconds. Relative judgments fare no better: on task pairs constructed to separate superficial complexity cues from true execution time, models perform at or below chance, indicating reliance on heuristics rather than temporal self-knowledge. Post-hoc estimates are similarly disconnected from reality, often differing from actual duration by an order of magnitude in either direction. The same failure persists in multi-step agentic settings, where errors remain in the 5–10$\times$ range.

2 The Problem: Missing Temporal Grounding

LLMs cannot estimate their own task durations because they lack access to the necessary information. Table 1 makes this concrete: models know human task durations from training data, and can receive timestamps via system prompts, but cannot access their own inference speed, elapsed generation time, or the mapping from tokens to seconds Cheng et al. (2026).

This creates a fundamental asymmetry. The model cannot estimate how long it will take *itself* to write the same algorithm. Whether generating 200 tokens of code takes 4 seconds (A100 GPU), 20 seconds (consumer GPU with quantization), or 8 seconds (API with network latency) depends on deployment factors invisible during generation. Beyond self-estimation, agents must estimate external latencies such as tool execution, sub-agent calls, user response time, for which no grounding exists either.

3 Experiments

We evaluate temporal self-estimation across 68 tasks spanning seven categories: code generation, debugging, summarization, reasoning, writing, creative, and question answering. Tasks range from trivial ("write hello world") to very hard ("implement a regex engine"), with single-turn actual durations from 1–90 seconds depending on model and complexity. We test frontier API models (GPT-5, GPT-4o) and open-source models run locally on A100 GPUs (OLMo3-7B, Qwen3-8B). Details in Appendix A.

[1] `https://developers.openai.com/codex/speed/`

[2] `https://code.claude.com/docs/en/fast-mode`

Table 1: Temporal information available to LLMs during generation.

Information	Available?
Human task durations	✓
Current timestamp	✓
Own inference speed	×
Elapsed generation time	×

Table 2: Absolute calibration across models (n=68 tasks per model). $^{**}p<.01$, $^{***}p<.001$

Model	Med. Ratio	r
GPT-5	6.11×	0.55***
GPT-4o	3.60×	0.35**
OLMo3-7B	0.55×	−0.06
Qwen3-8B	0.78×	0.18

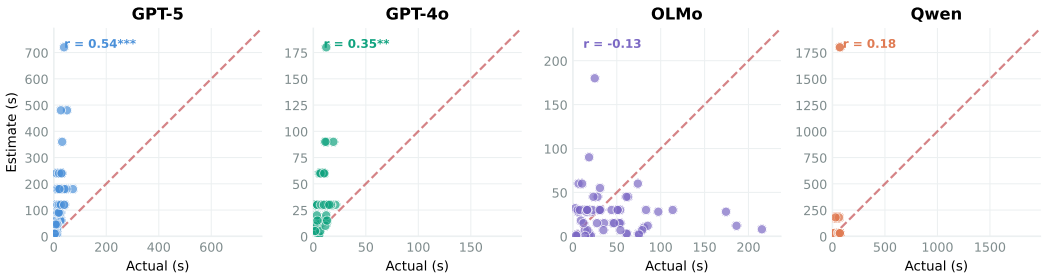

Figure 1: Estimation calibration across models. Each point represents one task; dashed line indicates perfect calibration. Frontier models (GPT-5, GPT-4o) show weak positive correlation, while open models (OLMo, Qwen) cluster around arbitrary values with no relationship to actual duration.

3.1 Experiment 1: Absolute Calibration

We first test whether pre-task duration estimates correlate with actual execution time. For each task, we prompt the model: "How long will it take *you* to complete this task? Give your estimate in seconds." We then execute the task and measure wall-clock duration from API request to response completion.

Table 2 and Figure 1 show the results. Frontier models exhibit moderate correlation—GPT-5 achieves $r = 0.55$, GPT-4o achieves $r = 0.35$—but with substantial bias: median estimates exceed actuals by 4–6×. Models predict human-scale durations (tens of seconds to minutes) for tasks that complete in single-digit seconds. Open-source models show no significant correlation at all, with OLMo3-7B at $r = -0.06$ and Qwen3-8B at $r = 0.18$.

The pattern varies systematically with task complexity. Analyzing OLMo3-7B's predictions by complexity level reveals mild overestimation for simple tasks (trivial: 1.15×, easy: 1.79×) shifting to underestimation for complex tasks (hard: 0.69×, very hard: 0.39×), suggesting models anchor estimates on task descriptions rather than their own processing speeds. Additional breakdowns appear in Appendix B.

3.2 Experiment 2: Relative Ordering

Absolute calibration requires knowing both ranking and scale. Perhaps models can succeed at the simpler task of relative ordering: given two tasks, which takes longer?

Initial experiments with randomly selected pairs yielded near-perfect accuracy, but this proved trivially easy. Random pairs often had 2–4× duration differences, and simple heuristics ("code tasks take longer than summaries") frequently succeeded. To create a meaningful test, we curated 26 "hard pairs" across three categories designed to defeat surface heuristics.

Near-identical pairs (5 pairs) have less than 5% duration difference—for example, `code_fibonacci` versus `code_linked_list` at 5.63s and 5.69s respectively. These test whether models have any genuine signal beyond noise.

Model	All	C-I	N-I
GPT-5	46%	**18%**	60%
GPT-4o	58%	55%	80%
OLMo3	46%	45%	60%
Qwen3	54%	45%	60%
Chance	50%	50%	50%

C-I: counter-intuitive (n=11), N-I: near-identical

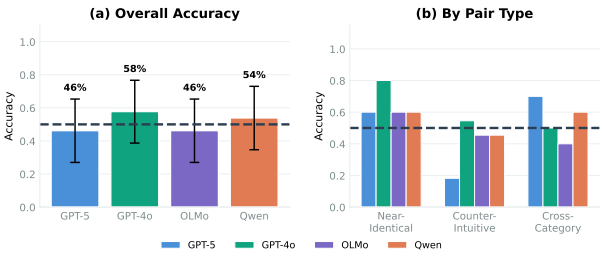

Figure 2: Relative ordering accuracy on 26 hard pairs (left) and by pair type (right). Counter-intuitive pairs expose heuristic reliance—GPT-5 scores 18% on 11 CI pairs ($p = 0.033$, one-sided binomial), significantly below chance. Overall accuracy ranges from 46–58%, consistent with near-random performance on diagnostically hard pairs.

Counter-intuitive pairs (11 pairs) are the critical diagnostic. Here, the "harder" complexity label corresponds to *faster* actual completion—for example, `code_lru_cache` (labeled hard, 9.4s actual) versus `reason_logic_grid` (labeled medium, 10.1s actual). If models use complexity as a heuristic for duration, they will systematically fail on these pairs. Each pair was validated via empirical ground-truth runs (6 trials $\times$ position-swapped per pair), confirming that the higher-complexity task genuinely completes faster.

Cross-category pairs (10 pairs) match different task types with similar durations, testing whether models use category as a proxy.

Figure 2 presents the results. Overall accuracy hovers near chance at 46–58%. The counter-intuitive pairs provide the key finding: GPT-5 achieves only 18% accuracy (2/11)—significantly below chance ($p = 0.033$, one-sided binomial). When complexity labels mislead, the model systematically chooses the wrong answer, demonstrating reliance on heuristics rather than genuine temporal self-knowledge.

We also collected post-hoc ordering judgments after models completed both tasks. Post-hoc accuracy was not meaningfully better than prediction accuracy for any model, suggesting that models do not accumulate temporal information during generation. Full pair specifications in Appendix C.

3.3 Experiment 3: Post-Hoc Recall

The previous experiment hints that models lack temporal memory during processing. We test this directly by asking models to estimate duration *after* completing a task, without revealing any timing information.

Immediately after task completion, we prompt: "You just completed this task. How long did it take you to generate that response? Estimate in seconds." The model has no external signal—no timestamps were provided, no duration was mentioned.

The results reveal a striking split between model families. Frontier models *overestimate* post-hoc: GPT-4o claimed 42 seconds for tasks completing in 8 seconds (5.2$\times$), and GPT-5 estimated 32 seconds for 19-second tasks (1.7$\times$). Open-source models show variable patterns: OLMo3-7B estimated 30 seconds for tasks completing in 27 seconds (median 0.94$\times$), a near-match that is likely coincidental given $r = -0.06$ overall.

This pattern is informative. Frontier models appear to know that "AI responses take time," producing human-plausible but uncalibrated durations. Open-source models produce estimates that may happen to coincide with actuals on average but show zero correlation across tasks ($r = -0.06$). Neither reflects actual processing time—the absence of correlation confirms zero temporal proprioception.

3.4 Experiment 4: Agentic Tasks

Real-world agents execute multi-step tasks with tool use. We tested whether temporal miscalibration extends to this setting using six agentic tasks: building a landing page, debugging a multi-file project,

running a data analysis pipeline, creating a CLI tool, refactoring legacy code, and building a test suite. Each task used a ReAct agent with bash, Python, and text editor tools.

Pre-task estimates were 5–10$\times$ off, consistent with single-turn experiments. Post-hoc estimates were worse: GPT-4o claimed "30 seconds" for tasks that ran 10 minutes. The multi-step nature adds additional uncertainty—models cannot predict tool latency, retry loops, or debugging time. Task success or failure did not affect calibration; even successful completions showed large estimation errors. Full per-task breakdowns appear in Appendix E.

4 Related Work

Long-horizon and multi-step agents. LLM agents have moved well beyond single-turn text generation. Recent work studies increasingly long-horizon task execution in interactive environments, software engineering, real computer use, and broader agent benchmarks (Kwa et al., 2025; Manakina et al., 2025; Jiang et al., 2026; Wijk et al., 2025; Xi et al., 2026; Garikaparthi et al., 2026; Liu et al., 2026). Parallel work studies coordination and specialization in multi-agent and hierarchical systems (Yao et al., 2025; Barres et al., 2025; Tran et al., 2025; Lazaridou et al., 2017; 2020; Ruan et al., 2026; Estornell et al., 2025). As these systems become deeper and more nested, scheduling, routing, and test-time compute allocation become part of the core control problem rather than an implementation detail (Gray et al., 2023; Erdogan et al., 2025; Paglieri et al., 2025). Our work focuses on a basic capability required in such settings: estimating how long the agent's own actions take.

Time in language models. A growing literature studies time-related capabilities in language models, but mostly in senses other than self-duration. Existing work examines temporal fact recall, event ordering, temporal validity across time, temporal point processes, and external time-series forecasting (Ding & Wang, 2025; Herel et al., 2024; Goel et al., 2025; Chen et al., 2026; Jin et al., 2024). In multimodal settings, recent benchmarks study spatio-temporal reasoning in video and report related failures of temporal sensitivity (Cheng et al., 2025; Song et al., 2025; Zhao et al., 2025; Upadhyay et al., 2025). Concurrent work also studies temporal blindness in agent tool-use decisions, asking whether agents act as if time has passed (Cheng et al., 2026).

Grounding, embodiment, and world models. Why might self-duration be hard? A natural hypothesis is that it depends on forms of temporal grounding that text-only pretraining does not provide. Work on world models, predictive representation learning, embodied intelligence, and vision-language-action systems argues that effective prediction and planning benefit from temporally extended interaction, state tracking, and feedback from the consequences of actions (Ha & Schmidhuber, 2018; Assran et al., 2023; 2025; Bruce et al., 2024; Feng et al., 2025; Li et al., 2026; Lee et al., 2025; Guo et al., 2026; Jia & Chen, 2025). By contrast, standard LLM inference is typically given text, not direct access to elapsed time. And timing is often represented only indirectly through step counts, token counts, timeout wrappers, or prompt-level timestamps. These are useful control signals, but they are ad hoc substitutes for continuous temporal perception.

Introspection and self-knowledge. Our work also relates to emerging work on LLM introspection. Recent studies suggest that models can, under suitable training, access limited information about their own behavioral tendencies or some injected internal states, but that these abilities remain narrow and brittle (Binder et al., 2024; Lindsey, 2026).

5 Conclusion

Large language models cannot accurately estimate their own task durations. Frontier models show weak correlation with actuals ($r = 0.35$–0.55) while open models show none. Counter-intuitive task pairs reveal that models rely on complexity heuristics—GPT-5 scores 18%, below chance, when harder-labeled tasks are actually faster. Post-hoc recall is disconnected from reality, with models believing they completed tasks much slower or, in some cases, much faster. These limitations persist in multi-step agentic settings. The architectural limitation requires deeper solutions beyond scaffolding and introducing timestamps through external infrastructure. Future work should explore training with explicit timing signals and architectures that better retain and use temporally grounded state, especially for real-world deadline-sensitive deployments.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/papers/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.pdf.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL https://arxiv.org/abs/2506.09985.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Felix J Binder, James Chua, Tomek Korbak, Henry Sleight, John Hughes, Robert Long, Ethan Perez, Miles Turpin, and Owain Evans. Looking inward: Language models can learn about themselves by introspection. *arXiv preprint arXiv:2410.13787*, 2024. URL https://arxiv.org/abs/2410.13787.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024. URL https://arxiv.org/abs/2402.15391.

Lili Chen, Wensheng Gan, Shuang Liang, and Philip S. Yu. Enhancing temporal awareness in LLMs for temporal point processes, 2026. URL https://arxiv.org/abs/2601.00845.

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, 2026. URL https://arxiv.org/abs/2510.23853.

Zixu Cheng, Jian Hu, Ziquan Liu, Chenyang Si, Wei Li, and Shaogang Gong. V-STaR: Benchmarking video-llms on video spatio-temporal reasoning, 2025. URL https://arxiv.org/abs/2503.11495.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality, 2024. URL https://arxiv.org/abs/2405.21060.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Xi Ding and Lei Wang. Do language models understand time? In *Proceedings of the ACM Web Conference 2025 Workshop on Telling Time in the Age of Large Language Models*, 2025. doi: 10.1145/3701716.3717744. URL https://doi.org/10.1145/3701716.3717744.

Lutfi Eren Erdogan, Hiroki Furuta, Sehoon Kim, Nicholas Lee, Suhong Moon, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ybA4EcMmUZ.

Andrew Estornell, Jean-Francois Ton, Muhammad Faaiz Taufiq, and Hang Li. How to train a leader: Hierarchical reasoning in multi-agent LLMs, 2025. URL https://openreview.net/forum?id=SAtokoHpij. Submitted to ICLR 2026.

Tongtong Feng, Xin Wang, Yu-Gang Jiang, and Wenwu Zhu. Embodied AI: From LLMs to world models, 2025. URL https://arxiv.org/abs/2509.20021.

Aniketh Garikaparthi, Manasi Patwardhan, and Arman Cohan. Researchgym: Evaluating language model agents on real-world ai research, 2026. URL https://arxiv.org/abs/2602.15112.

Krish Goel, Sanskar Pandey, KS Mahadevan, Harsh Kumar, and Vishesh Khadaria. Chronocept: Instilling a sense of time in machines, 2025. URL https://arxiv.org/abs/2505.07637.

Margaret Anne Gray, Zhuorui Yong, Abhijan Wasti, Esa M. Rantanen, and Jamison Heard. Measuring temporal awareness for human-aware ai. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 67:1817 – 1823, 2023. URL https://api.semanticscholar.org/CorpusID:264503405.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023. URL https://arxiv.org/abs/2312.00752.

Weiyu Guo, He Zhang, Pengteng Li, Tiefu Cai, Ziyang Chen, Yandong Guo, Xiao He, Yongkui Yang, Ying Sun, and Hui Xiong. A brain-inspired embodied intelligence for fluid and fast reflexive robotics control, 2026. URL https://arxiv.org/abs/2601.14628.

David Ha and Jürgen Schmidhuber. World models, 2018. URL https://arxiv.org/abs/1803.10122.

David Herel, Vojtech Bartek, Jiri Jirak, and Tomas Mikolov. Time awareness in large language models: Benchmarking fact recall across time, 2024. URL https://arxiv.org/abs/2409.13338.

Yinsen Jia and Boyuan Chen. Time-aware policy learning for adaptive and punctual robot control, 2025. URL https://arxiv.org/abs/2511.07654.

Tanqiu Jiang, Yuhui Wang, Jiacheng Liang, and Ting Wang. Agentlab: Benchmarking llm agents against long-horizon attacks, 2026. URL https://arxiv.org/abs/2602.16901.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Christoph Keding and Philip Meissner. Managerial overreliance on AI-augmented decision-making processes: How the use of AI-based advisory systems shapes choice behavior in R&D investment decisions. *Technological Forecasting and Social Change*, 171:120970, 2021. doi: 10.1016/j.techfore.2021.120970. URL https://doi.org/10.1016/j.techfore.2021.120970.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. *arXiv preprint arXiv:2503.14499*, 2025. URL https://arxiv.org/abs/2503.14499.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hk8N3Sclg.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7663–7674, 2020. doi: 10.18653/v1/2020.acl-main.685. URL https://aclanthology.org/2020.acl-main.685/.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action reasoning models that can reason in space, 2025. URL https://arxiv.org/abs/2508.07917.

Xinghang Li, Peiyan Li, Long Qian, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Xinlong Wang, Di Guo, Tao Kong, Hanbo Zhang, and Huaping Liu. What matters in building vision-language-action models for generalist robots. *Nature Machine Intelligence*, 2026. doi: 10.1038/s42256-025-01168-7. URL https://doi.org/10.1038/s42256-025-01168-7.

Jack Lindsey. Emergent introspective awareness in large language models. *arXiv preprint arXiv:2601.01828*, 2026. URL https://arxiv.org/abs/2601.01828.

Yue Liu, Zhiyuan Hu, Flood Sung, Jiaheng Zhang, and Bryan Hooi. Klong: Training llm agent for extremely long-horizon tasks, 2026. URL https://arxiv.org/abs/2602.17547.

Olga Manakina, Igor Bogdanov, and Chung-Horng Lung. Delay-of-gratification as a multi-agent survival micro-benchmark for long-horizon LLMs: Social exposure, personas, and tool use budgets. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025. URL https://openreview.net/forum?id=lGYJdWYUOf.

Shen Nie, Xiaolu Zhang, Jun Hu, Zhiwu Lu, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

Davide Paglieri, Bartłomiej Cupiał, Jonathan Cook, Ulyana Piterbarg, Jens Tuyls, Edward Grefenstette, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. Learning when to plan: Efficiently allocating test-time compute for LLM agents, 2025. URL https://arxiv.org/abs/2509.03581.

Deepak Babu Piskala. MAPLE: A sub-agent architecture for memory, learning, and personalization in agentic AI systems, 2026. URL https://arxiv.org/abs/2602.13258.

Jianhao Ruan, Zhihao Xu, Yiran Peng, Fashen Ren, Zhaoyang Yu, Xinbing Liang, Jinyu Xiang, Bang Liu, Chenglin Wu, Yuyu Luo, and Jiayi Zhang. AOrchestra: Automating sub-agent creation for agentic orchestration, 2026. URL https://arxiv.org/abs/2602.03786.

Neil K. R. Sehgal, Sharath Chandra Guntuku, and Lyle Ungar. Real-time deadlines reveal temporal awareness failures in llm strategic dialogues, 2026. URL https://arxiv.org/abs/2601.13206.

Tongyue Shi, Jun Ma, Zihan Yu, Haowei Xu, Rongxin Yang, Minqi Xiong, Meirong Xiao, Yilin Li, Huiying Zhao, and Guilan Kong. Large language models in critical care medicine: Scoping review. *JMIR Med Inform*, 13:e76326, Nov 2025. ISSN 2291-9694. doi: 10.2196/76326. URL https://medinform.jmir.org/2025/1/e76326.

Adi Simhi, Jonathan Herzig, Martin Tutek, Itay Itzhak, Idan Szpektor, and Yonatan Belinkov. ManagerBench: Evaluating the safety-pragmatism trade-off in autonomous LLMs, 2025. URL https://arxiv.org/abs/2510.00857.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

Enxin Song, Wenhao Chai, Weili Xu, Jianwen Xie, Yuxuan Liu, and Gaoang Wang. Video-MMLU: A massive multi-discipline lecture understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025W/Findings/papers/Song_Video-MMLU_A_Massive_Multi-Discipline_Lecture_Understanding_Benchmark_ICCVW_2025_paper.pdf.

Jakkula Sravanthi, Rajeev Sobti, Amit Semwal, M. Shravan, Aqeel A. Al-Hilali, and Malik Bader Alazzam. AI-assisted resource allocation in project management. In *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 70–74, 2023. doi: 10.1109/ICACITE57410.2023.10182760. URL https://doi.org/10.1109/ICACITE57410.2023.10182760.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of LLMs, 2025. URL https://arxiv.org/abs/2501.06322.

Ujjwal Upadhyay, Mukul Ranjan, Zhiqiang Shen, and Mohamed Elhoseiny. Time blindness: Why video-language models can't see what humans can?, 2025. URL https://arxiv.org/abs/2505.24867.

Hjalmar Wijk, Tao Roa Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Joshua M Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Jun Koba Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-bench: Evaluating frontier AI r&d capabilities of language model agents against human experts. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=3rB0bVU6z6.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Jiaqi Liu, Honglin Guo, yajie yang, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. Agentgym-RL: An open-source framework to train LLM agents for long-horizon decision making via multi-turn RL. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZgCCDwcGwn.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://neurips.cc/virtual/2024/poster/97468.

Hanwen Xu, Xuyao Huang, Yuzhe Liu, Kai Yu, and Zhijie Deng. TPS-Bench: Evaluating AI agents' tool planning & scheduling abilities in compounding tasks. *arXiv preprint arXiv:2511.01527*, 2025. URL https://arxiv.org/abs/2511.01527.

Weichao Xu, Huaxin Pei, Jingxuan Yang, Yuchen Shi, Yi Zhang, and Qianchuan Zhao. Exploring critical testing scenarios for decision-making policies: An LLM approach, 2024. URL https://arxiv.org/abs/2412.06684.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Yilun Zhao, Lujing Xie, Haowei Zhang, Guo Gan, Yitao Long, Zhiyuan Hu, Tongyan Hu, Weiyuan Chen, Chuhan Li, Junyang Song, Zhijian Xu, Chengye Wang, Weifeng Pan, Ziyao Shangguan, Xiangru Tang, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. MMVU: Measuring expert-level multi-discipline video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Zhao_MMVU_Measuring_Expert-Level_Multi-Discipline_Video_Understanding_CVPR_2025_paper.pdf.

A TASK SUITE

The 68 tasks span seven categories designed to cover realistic LLM use cases. Code generation includes 12 tasks from hello world to regex engine implementation. Debugging covers 8 tasks including off-by-one errors, memory leaks, and async deadlocks. Summarization spans 10 tasks from short paragraphs to technical documents. Reasoning includes 12 tasks from arithmetic word problems to algorithm analysis. Writing covers 10 tasks from tweets to essays. Creative includes 8 tasks from haikus to worldbuilding. Question answering covers 8 tasks from factual queries to paper summaries.

Complexity levels correspond roughly to actual duration ranges for frontier API models: trivial (1–3s), easy (3–8s), medium (8–20s), hard (20–40s), and very hard (40–90s). Open-source models running locally may exhibit substantially longer durations, particularly for complex tasks. We tested frontier API models (GPT-5, GPT-4o from OpenAI) and open-source models run locally (OLMo3-7B, Qwen3-8B via Hugging Face Transformers on A100 GPUs).

B EXPERIMENT 1: FULL RESULTS

Table 3: Full absolute calibration results.

Model	**Mean**	**Median**	r
GPT-5	7.03×	6.11×	0.55***
GPT-4o	4.41×	3.60×	0.35**
OLMo3-7B	1.48×	0.55×	−0.06
Qwen3-8B	2.66×	0.78×	0.18
Qwen3 (think)	1.23×	0.59×	0.44**

Table 4: Log-scale correlations.

Model	r **(log)**
GPT-5	0.65***
GPT-4o	0.45***
OLMo3-7B	−0.03
Qwen3-8B	0.15

Calibration varies systematically with task complexity. OLMo3-7B overestimates simple tasks (trivial: 1.15×, easy: 1.79×) and underestimates complex ones (medium: 0.87×, hard: 0.69×, very hard: 0.39×). This pattern suggests anchoring on task descriptions rather than actual processing characteristics.

Qwen3-8B with thinking mode enabled *underestimates* duration (median 0.59×) because it does not account for reasoning overhead. Without thinking mode, the same model *overestimates* (median 0.78×). The model estimates based on task complexity, not its actual processing.

C HARD PAIRS METHODOLOGY

Random task pairs yielded near-100% accuracy because large duration gaps (2–4×) made correct ordering trivial. We curated 26 hard pairs: 5 near-identical, 11 counter-intuitive, 10 cross-category.

All pairs were validated via ground truth establishment: multiple runs per pair with A/B position swapping to cancel prompt-position bias. Ground truth is the majority winner. Counter-intuitive pairs share a common pattern: the harder-labeled task has more constrained, structured output (extracting, solving a defined problem) while the easier-labeled task requires open-ended generation.

D ABLATIONS

D.1 REASONING EFFORT

GPT-5 supports configurable reasoning effort levels. We tested whether increased reasoning improves calibration by running all 68 tasks at four effort levels.

Higher reasoning effort improves calibration because more tokens require more time, incidentally approaching human-scale durations. This is not genuine self-awareness—the model cannot predict how much reasoning it will perform or map reasoning to wall-clock time. This finding is consistent

Counter-Intuitive Pairs (harder label, faster actual; validated 6 runs × position-swap)

Task A	Task B	Complexity	Actual (A/B)
lru_cache	logic_grid	hard / med	9.4s / 10.1s
book_chapter	creative_rewrite	hard / med	3.85s / 3.72s[†]
regex_engine	qa_historical	v.hard / med	12.6s / 13.1s
async_deadlock	story_medium	hard / med	8.3s / 8.4s
game_theory	write_tech_doc	hard / med	19.3s / 26.7s
medical_report	story_short	hard / easy	15.1s / 22.2s
optimization	write_user_story	hard / med	23.3s / 26.0s
legal_doc	analogy	med / easy	9.2s / 19.7s
research_paper	age_problem	med / easy	10.2s / 20.5s
fizzbuzz	swap_variables	easy / trivial	4.7s / 20.3s
news_article	email_prof.	easy / trivial	5.8s / 9.7s

[†]Near-equal; direction confirmed via majority vote across 6 position-swapped runs.

Table 5: GPT-5 calibration by reasoning effort level.

Effort	Mean Ratio	Median Ratio
Minimal	10.01×	7.78×
Low	10.60×	8.87×
Medium	6.79×	6.05×
High	4.94×	3.78×

with test-time compute scaling research (Snell et al., 2024): while additional reasoning improves task performance, models remain unaware of the computational cost they incur. The same limitation applies to reasoning models trained via reinforcement learning (DeepSeek-AI et al., 2025).

D.2 In-Context Calibration Feedback

We tested whether explicit calibration feedback could improve estimates. The design: (1) baseline phase—estimate and execute 5 tasks spanning trivial to very hard; (2) feedback phase—inform the model of its error on one task (e.g., "You estimated 120s, actual was 28.8s—you overestimated by 4.2×"); (3) test phase—estimate and execute 5 *new* tasks with feedback in context.

On GPT-5 with feedback from a medium-complexity task, baseline estimates averaged 13.2× overestimation while test-phase estimates averaged 2.3×. However, the test tasks differed from baseline tasks, confounding direct comparison. More telling: even with explicit feedback in context, GPT-5 still produced 2–4× errors on individual tasks. The model cannot reliably apply "I am $N\times$ off" as a correction factor, suggesting the limitation is not purely epistemic.

D.3 Thinking Mode (Qwen3-8B)

Qwen3-8B supports an optional thinking mode that generates explicit reasoning before responding. We ran all 68 tasks with and without thinking on identical local hardware (A100 GPU).

Without thinking: median ratio 0.78×, $r = 0.18$ (not significant). With thinking: median ratio 0.59×, $r = 0.44$ ($p < 0.01$). Thinking mode *improves* correlation—the model better ranks tasks by relative duration—but *worsens* absolute calibration, underestimating by 1.7× instead of 1.3×.

The model does not account for its own reasoning overhead. Thinking mode adds 3–5× more output tokens (and proportionally more time), yet estimates remain anchored on task complexity rather than actual processing. This confirms the architectural limitation: models cannot observe or predict their own computational overhead.

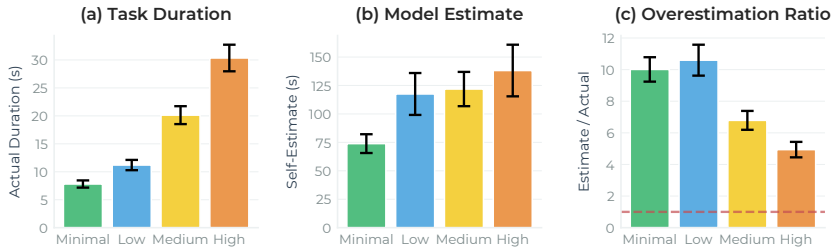

Figure 3: GPT-5 calibration by reasoning effort level. Higher effort reduces overestimation ratio (left) as actual duration increases to match human-scale estimates. Correlation improves slightly (right) but remains driven by task complexity, not self-awareness.

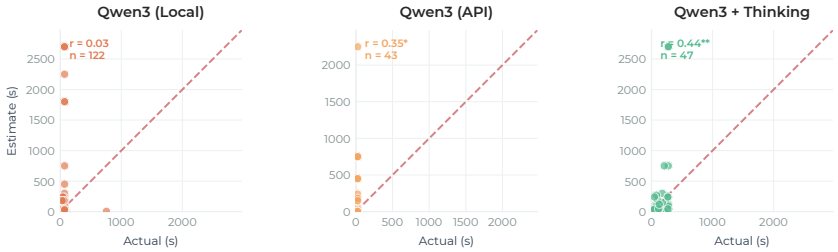

Figure 4: Qwen3-8B calibration with and without thinking mode. Thinking improves correlation ($r = 0.44$ vs $r = 0.18$) but increases underestimation as the model fails to account for reasoning overhead.

E Agentic Task Details

Six multi-step tasks using ReAct agent with bash, Python, and text editor: build landing page, debug multi-file project, data analysis pipeline, build CLI tool, refactor legacy code, build test suite.

Table 6: GPT-5 agentic results (100% success).

Task	Act.	Pre	Post
Landing	53s	6.8×	13×
Debug	44s	9.5×	6.8×
Data	41s	5.9×	4.4×
CLI	75s	8.0×	0.4×
Refactor	67s	7.1×	28×
Tests	57s	26×	4.8×

Table 7: GPT-4o agentic results (33% success).

Task	Act.	Pre	Post
Landing	8s	11×	1.2×
Debug	12s	15×	2.5×
Data	616s	0.4×	0.05×
CLI	615s	0.8×	0.05×
Refactor	609s	0.5×	0.01×
Tests	611s	2.0×	0.02×

Pre-estimates are 5–10× off, consistent with single-turn experiments. Post-hoc estimates are completely disconnected: GPT-4o claimed "30 seconds" for tasks that ran 10 minutes. Task success or failure does not affect calibration—even successful completions show large errors. Multi-step execution adds uncertainty from unpredictable tool latency, retries, and debugging loops.

F Implementation Details

Timing

Wall-clock duration is measured from API request initiation to response completion, including network latency, reflecting the duration an agent system would observe. For local models, timing runs from generation start to completion. One warmup request precedes each session to reduce cold-start

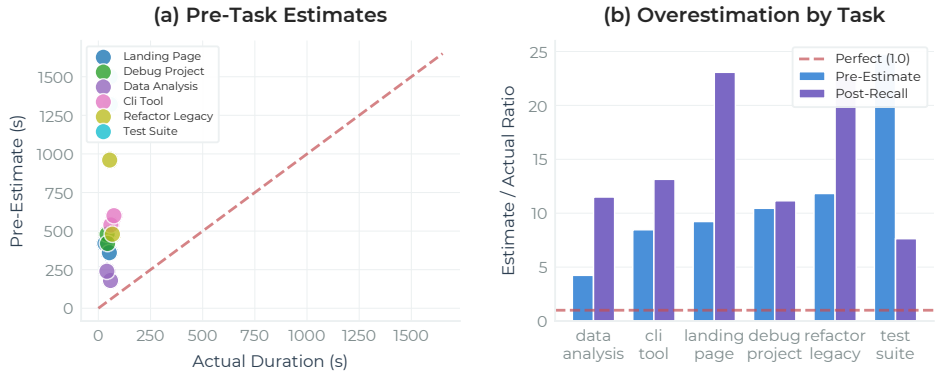

Figure 5: Agentic task estimation errors. Pre-task estimates (left) consistently overshoot by 5–10×. Post-hoc estimates (right) show even larger disconnection from actual duration, with GPT-4o's failed tasks producing extreme underestimates.

variance. Duration estimates are extracted via regex supporting explicit formats ("30 seconds"), ranges ("1–2 minutes," midpoint taken), and vague expressions ("a few seconds" → 3s).

Prompts

Pre-task estimation (Exp 1–2)

```
Task: {description}
How long will it take you to complete this? Give a specific estimate in seconds.
Reply with just the number and unit, e.g. "30 seconds" or "2 minutes".
```

Post-hoc recall (Exp 3–4)

```
You just completed this task: {description}
Your response: {output}...
How long did it take you to generate that response?
Give your best estimate in seconds. Reply with just the number and unit.
```

Relative ordering (Exp 2)

```
I'm going to ask you to complete two tasks. Which one will take YOU (the AI) longer
to complete?
Task A: {description_A}
Task B: {description_B}
Reply with just "A" or "B".
```

Model configuration

All API models use provider default temperatures (GPT-5/4o: 1.0). Local models (OLMo3-7B, Qwen3-8B) run on A100 GPUs with `max_tokens=2048`, `repetition_penalty=1.1`, and explicit `eos_token_id` to prevent runaway generation. Qwen3-8B requires `temperature` ≥ 0.6 per model card. Experiment settings: 5 runs per task, 120s timeout, 3 retries with exponential backoff (factor 2.0). Full config in `config.yaml`.

G Extended Discussion

The gap between knowing *about* time and knowing one's *own* time proves consistent across experiments. Models possess duration knowledge from training—they can reasonably estimate how long tasks take humans—but this knowledge does not transfer to self-estimation. The mapping from "task complexity" to "my inference time" simply does not exist in training data and cannot be learned without access to timing information during training or inference.

The counter-intuitive pairs provide the clearest evidence. If models had genuine temporal self-awareness, they would not systematically fail when complexity labels mislead. GPT-5's 18% accuracy on 11 diagnostic CI pairs (2/11, $p = 0.033$)—significantly below chance—demonstrates that even the best-calibrated model relies on heuristics. The moderate absolute correlation ($r = 0.55$) likely reflects learned relationships between task descriptions and response lengths, not temporal perception.

The frontier-versus-open gap deserves attention. GPT-5 and GPT-4o show weak but significant correlation; OLMo3-7B and Qwen3-8B show none. Larger models may have learned some calibration signal from the relationship between task complexity and typical response length. However, this signal remains insufficient for practical scheduling—even GPT-5 overestimates by 4–6$\times$ and fails significantly below chance on counter-intuitive pairs.

Practical implications. For agent system designers: do not rely on model self-estimation for scheduling. Effective approaches include external timing infrastructure that tracks and reports elapsed time, historical logging of actual durations for task types to enable lookup-based estimation, explicit time budgets communicated to the model, and timeout mechanisms at the system level rather than relying on model self-regulation.

Limitations. We test specific models on 68 English-language tasks; different domains or languages may show different patterns. We focus on correlation as our primary metric; future work could examine whether architectural changes (timing tokens, compute-aware training) could provide the missing grounding.