Beyond Accuracy: A Cognitive Load Framework for Mapping the Capability Boundaries of Tool-use Agents

Qihao Wang[1,2], **Yue Hu**[1,2*], **Mingzhe Lu**[1,2], **Jiayue Wu**[1,2], **Yanbing Liu**[1,2*], **Yuanmin Tang**[1,2*]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences

Abstract

The ability of Large Language Models (LLMs) to use external tools unlocks powerful real-world interactions, making rigorous evaluation essential. However, current benchmarks primarily report final accuracy, revealing what models can do but obscuring the cognitive bottlenecks that define their true capability boundaries. To move from simple performance scoring to a diagnostic tool, we introduce a framework grounded in Cognitive Load Theory. Our framework deconstructs task complexity into two quantifiable components: Intrinsic Load, the inherent structural complexity of the solution path, formalized with a novel Tool Interaction Graph; and Extraneous Load, the difficulty arising from ambiguous task presentation. To enable controlled experiments, we construct ToolLoad-Bench, the first benchmark with parametrically adjustable cognitive load. Our evaluation reveals distinct performance cliffs as cognitive load increases, allowing us to precisely map each model's capability boundary. We validate that our framework's predictions are highly calibrated with empirical results, establishing a principled methodology for understanding an agent's limits and a practical foundation for building more efficient systems.

Introduction

The paradigm of Large Language Models (LLMs) is rapidly evolving from passive text generators into capable autonomous agents that can interact with the world and solve complex problems (Yao et al. 2023; Shen et al. 2023; Wang et al. 2024; Mialon et al. 2023). A cornerstone of this transformation is the ability to use external tools, APIs, databases, and other software to overcome the limitations of their parametric knowledge (Schick et al. 2023; Mialon et al. 2023; Qin et al. 2024; Su et al. 2025). This tool-use capability is the engine driving progress in agentic AI, enabling models to tackle multi-step, real-world tasks that were previously intractable (Chen et al. 2025c; Shi et al. 2025b).

This rapid progress has spurred the development of sophisticated benchmarks designed to measure and rank the tool-use proficiency of different models. Influential testbeds like Li et al. (2023) and Qin et al. (2023) provide comprehensive evaluations with thousands of APIs and complex tasks. The Berkeley Function Calling Leaderboard (Patil et al. 2025) further standardizes evaluation, while newer benchmarks explore more realistic conversational (Farn and Shin 2023; Du, Wei, and Zhang 2024) and stateful (Shi et al. 2024) scenarios. While these evaluations are invaluable for tracking overall progress, they typically culminate in a single, final accuracy score. This black-box evaluation paradigm reveals what a model can achieve but obscures when it fails. They treat task difficulty as a singular, unanalyzed variable, lacking a granular framework to diagnose specific failure modes related to task complexity. Consequently, there is a lack of clear understanding of the specific operational limits the capability boundaries of these agents (Qu et al. 2025; Chen et al. 2025b).

To move beyond simple performance scoring towards a more diagnostic form of evaluation, we propose a new lens inspired by Cognitive Load Theory (CLT) from psychology (Sweller 1988; Plass, Moreno, and Brünken 2010). Our approach deconstructs task complexity into quantifiable components, allowing us to systematically probe an agent's capabilities under controlled conditions. This enables us to pinpoint the specific bottlenecks that limit an agent's performance, a departure from prior work that has used CLT primarily as a metaphor for computational cost or to reduce the cognitive burden on human users (Yang et al. 2025b; Guidroz et al. 2025).

Our framework operationalizes this by modeling an agent's performance as a function of cognitive load. This allows us to characterize each agent's capability not as a single point of accuracy, but through a more descriptive cognitive profile defined by two key parameters: its Baseline Capability, reflecting its intrinsic proficiency on low-complexity tasks, and its Load Sensitivity, which measures how gracefully its performance degrades as task complexity increases. Crucially, we validate the soundness of our theoretical model through rigorous statistical goodness-of-fit tests, confirming that its predictions are highly calibrated with empirical observations.

Our primary contributions are as follows:

1. **A Formal Cognitive Load Framework:** We propose and formalize a novel evaluation framework, grounded in Cognitive Load Theory, that deconstructs task difficulty into quantifiable components: Intrinsic Load, derived from the inherent structural complexity of the task, and Extraneous Load, arising from the ambiguity of its

presentation.

2. **A Parametrically-Controlled Benchmark:** We construct and release **ToolLoad-Bench**, the first benchmark designed for controlled experimentation, with instances that allow for the parametric adjustment of cognitive load to systematically probe model limits.
3. **Empirical Mapping of Capability Boundaries**: We conduct an extensive evaluation of leading models, using our framework to map their distinct capability boundaries. This analysis reveals performance cliffs and characterizes each agent with a unique cognitive profile based on its baseline capability and load sensitivity.
4. **A Validated Evaluation Methodology:** We validate our framework's predictive power, showing through statistical testing that it is well-calibrated with empirical results. This establishes a principled and diagnostic methodology for the future assessment of tool-use agents.

Related Work

Enhancing Tool-Use Capabilities

Research in tool-augmented LLMs has rapidly advanced, focusing heavily on improving model proficiency. Foundational work demonstrated that models could learn to use tools through self-supervised learning (Schick et al. 2023). This was followed by a wave of instruction tuning, using curated datasets to teach models specific tool-use formats and behaviors (Tang et al. 2023; Patil et al. 2023; Lin et al. 2024; Shen et al. 2024). To overcome the bottleneck of manual data creation, recent efforts have focused on automated data generation pipelines that simulate agentic interactions to produce complex, multi-turn training data (Prabhakar et al. 2025; Shi et al. 2025a; Yin et al. 2025; Zhang et al. 2025). Concurrently, reinforcement learning (RL) has emerged as a powerful technique to refine tool-use policies, optimizing for reward signals related to successful task completion and reasoning (Qian et al. 2025; Dong et al. 2025a,b; Wang et al. 2025; Feng et al. 2025). Our work complements these advancements by providing a more nuanced evaluation framework to measure the true capabilities of the agents they produce.

Cognitive Load Theory in LLM Research

Originating in educational psychology, Cognitive Load Theory (CLT) posits that learning is impeded when a task's demands exceed the finite capacity of working memory (Sweller 1988). This theory has recently been adapted to LLM research in two main ways. First, it serves as a metaphor for computational cost, inspiring systems that dynamically manage model resources to improve efficiency, akin to reducing "cognitive" strain (Yang et al. 2025b; Xiao and Yang 2025). Second, in its traditional sense, CLT guides the design of LLM applications that reduce the cognitive burden on human users, for example, by simplifying text or generating helpful explanations (Guidroz et al. 2025; Sîrbu, Schelhorn, and Gnewuch 2025). Our work charts a new course by being the first to apply CLT not to system efficiency or human-computer interaction, but as a formal, quantitative framework to deconstruct task complexity and evaluate the cognitive limits of the AI agents themselves.

Methodology

To quantify task difficulty, we introduce a framework grounded in a probabilistic view of task success. We begin from formally defining the problem setting and our notation.

Preliminary

Our research focus on the problem of multi-turn tool use. A task instance is defined by a tuple (Q, T), where:

- $Q = (q_1, q_2, \ldots, q_m)$ is an ordered sequence of user queries.
- $T = \{tool_1, tool_2, \ldots, tool_k\}$ is the set of available API tools that an agent can use to finish the given task.

The tool-use agents' objective is to generate the correct sequence of tool calls from T with correct parameters.

The Tool Interaction Graph

To formally represent the internal complexity of a task's solution, we introduce the **Tool Interaction Graph (TIG)**. For a given instance (Q, T), its TIG, denoted as $G = (V, E)$, is a directed acyclic graph (DAG) that models the ground-truth solution.

- **Nodes (V):** The set of nodes consists of user query nodes $\{v_{q_1}, \ldots, v_{q_m}\}$ and a set of function call nodes $\{v_{f_1}, \ldots, v_{f_n}\}$. Each v_{f_i} corresponds to a specific tool invocation required to solve the task.
- **Edges (E):** A directed edge $(v_i, v_j) \in E$ represents a dependency, which can be one of two types:
  - **Data Dependency:** The invocation of function v_j requires specific data produced by the operation at node v_i.
  - **Execution Dependency:** A procedural constraint where task logic requires v_i to be executed before v_j, even if no data is passed.

The TIG provides a complete, formal scaffold of the task's inherent structural complexity, upon which we build our theory of cognitive load.

A Primer on Cognitive Load Theory

Cognitive Load Theory (CLT), originating from educational psychology, posits that human working memory has a limited capacity (Sweller 1988). Effective learning and problem-solving are hindered when the total cognitive demand of a task exceeds this capacity. CLT traditionally deconstructs this total load into two primary components relevant to our work:

- **Intrinsic Cognitive Load:** The inherent, irreducible complexity of the subject matter itself, determined by the number of interacting elements that must be processed simultaneously. In our context, this corresponds to the structural complexity of the task's solution path, which we formalize using the Tool Interaction Graph (TIG).

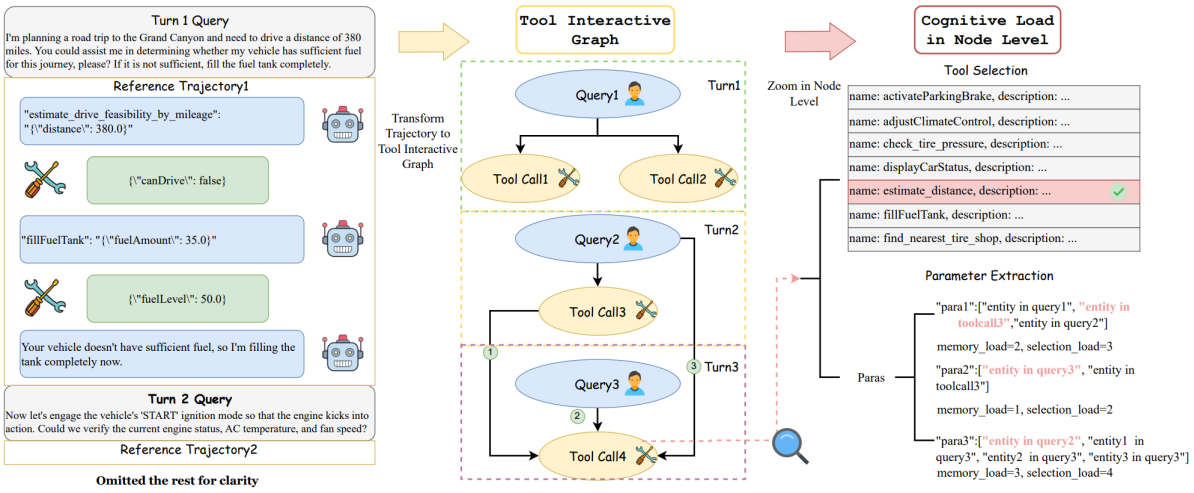

Figure 1: An illustration of our Cognitive Load Framework. A multi-turn tool-use task, defined by a sequence of user queries and a set of tools , is mapped to a TIG. The TIG represents the ground-truth solution. Zooming in node level, It shows selection load in parameter extraction and extraneous cognitive Load in tool selection.

- **Extraneous Cognitive Load:** The cognitive burden imposed by the way information is presented. This load is not essential to the task itself but arises from suboptimal design, such as confusing instructions or distracting information. We map this directly to challenges like ambiguous user queries and the presence of irrelevant distractor tools.

Our framework adapts this established psychological model to the domain of tool-augmented LLMs. We treat the model's computational context and reasoning capacity as analogous to human working memory. This allows us to move beyond a monolithic view of task difficulty and instead provide a principled, quantitative decomposition. By measuring intrinsic (CL_I) and extraneous (CL_E) load, we can precisely diagnose the specific bottlenecks that limit an agent's performance.

Theoretical Postulates and Derivation of Additive Load

Our framework relies on two fundamental postulates that connect cognitive load to the observable accuracy.

Postulate 1: The Load-Success Relationship. We posit that the probability of successfully executing any single cognitive operation is an exponential function of its associated cognitive load, CL.

$$P_{succ}(\text{op}) = \exp(-(k \cdot CL + b)) \tag{1}$$

Here, $k > 0$ and $b \geq 0$ are model-specific sensitivity parameter, representing model's capability to solve these tasks.

Postulate 2: Probabilistic Independence in the TIG. The probability of successfully executing the entire plan is the product of the probabilities of successfully executing each constituent function call node v_f.

$$P_{succ}(G) = \prod_{v_f \in V \setminus V_Q} P_{succ}(v_f) \tag{2}$$

where V_Q is the set of all query nodes.

Derivation. From these postulates, we can derive the additive nature of cognitive load. By substituting the load-success relationship Equation (1) into the probabilistic independence model of Equation (2), the product of exponential terms becomes the sum of their exponents. This leads directly to our central proposition:

$$CL_{Total} = \sum_{v_f \in V \setminus V_Q} CL(v_f) \tag{3}$$

Proposition 1. *Given our postulates, the total cognitive load of a task is the sum of the cognitive loads of its constituent function call nodes.*

Decomposing and Quantifying Cognitive Load

This additive principle allows us to decompose the total load into its constituent sources: intrinsic load from the task's structure and extraneous load from its presentation.

$$CL_{Total} = CL_I + CL_E \tag{4}$$

Intrinsic Cognitive Load (CL_I) The intrinsic load is inherent to the TIG structure. Following Proposition , we define CL_I as the sum of loads from all function nodes, which in turn is the sum of the loads of their dependency edges, $CL(e)$:

$$CL_I(G) = \sum_{v_f \in V \setminus V_Q} \sum_{e=(v_i, v_f) \in E} CL(e) \tag{5}$$

The load of a single dependency edge, $CL(e)$, is determined by its difficulty, which we model as a weight, $w(e) = CL(e)$, composed of two factors:

- **Memory Load (Attentional Distance):** The effort to recall information. We model this with $\delta(v_i, v_j)$, the number of conversational turns (user queries and tool calls) between the operation at v_i and its use at v_j.

- **Selection Load (Interference):** The effort to select correct information. We model this with $I(v_i, v_j)$, the number of other available but incorrect entities of the same semantic type (e.g., other user IDs) in the context. For pure execution dependencies, this is zero.

These combine into a single edge weight:

$$w(e) = \delta(v_i, v_j) \cdot (1 + \lambda \cdot I(v_i, v_j)) \tag{6}$$

where λ is a balancing hyperparameter. Our final formulation for intrinsic load is:

$$CL_I(G) = \sum_{v_f \in V \setminus V_Q} \sum_{e=(v_i, v_f) \in E} w(e) \tag{7}$$

Extraneous Cognitive Load (CL_E) Extraneous load arises from how the task is presented. It is independent of the TIG's structure and is primarily incurred when parsing user queries. We define the total extraneous load as the sum of the loads from each individual query in the task sequence Q:

$$CL_E(Q, T) = \sum_{q_i \in Q} CL_E(q_i, T) \tag{8}$$

For each query q_i, its extraneous load $CL_E(q_i, T)$ is the sum of two normalized scores (each in $[0, 1]$) determined by Gemini-2.5-pro. These scores separately evaluate: 1) the query's ambiguity, and 2) the potential for distraction from irrelevant but plausible tools in the set T. A higher score in either component reflects greater cognitive load.

Final Model and Accuracy Prediction

By combining the intrinsic and extraneous components (Equation (4)), we arrive at the total cognitive load for a task instance (Q, T, G). This unified metric allows us to predict model performance directly from our first postulate:

$$\text{Accuracy}(Q, T, G) \approx \exp(-(k \cdot CL_{Total} + b)) \tag{9}$$

This model provides a comprehensive, theoretically-grounded framework for quantifying task complexity and model capability in multi-turn tool agent systems.

Dataset Construction

To create a benchmark with fine-grained control over cognitive load, we constructed **ToolLoad-Bench**. Our methodology began with the 200 high-quality instances from the multi-turn base split of the Berkeley Function Calling Leaderboard (BFCL) v3 (Patil et al. 2025). From this foundation, we first extracted the tool dependency relationships to form initial Tool Interaction Graphs.

Benchmark	Num	Domains	Tools	Avg. Calls
BFCL	200	8	84	4.1
ToolLoad	500	10	106	4.9

Table 1: Statistical comparison of ToolLoad-Bench and the original BFCL-v3 (multi-turn base)(Patil et al. 2025) dataset.

We then employed a novel graph-to-task generation pipeline to build a larger and more diverse dataset. This involved two key strategies: 1) **Graph Generation**, where we synthesized entirely new, complex task graphs, and 2) **Edge Insertion**, where we systematically added new dependency edges to existing graphs to increase their structural complexity. Furthermore, to address the lack of scenarios with highly complex dependencies, we designed two new tool categories and meticulously annotated new instances following the official BFCL-v3 protocol.

This pipeline resulted in a final dataset of 500 instances designed to push the limits of current models. The statistical profile of ToolLoad-Bench is summarized in Table 1. The algorithms for data generation and cognitive load computation are detailed in the Appendix.

Experiments and Analysis

We conducted a series of experiments on our ToolLoad-Bench to evaluate leading language models and validate our cognitive load framework.

Experimental Setup

Models. We evaluated a comprehensive suite of models to cover different capability tiers (Patil et al. 2025; Chen et al. 2025a, 2023).

- **Closed-Source models:** GPT-4o, GPT-4o-mini (Achiam et al. 2023), Gemini 2.5 Pro (Team et al. 2024), and Claude 3.7 Sonnet.
- **Open-Source models:** A range of models from the Qwen3 (Yang et al. 2025a) and Llama3.3 families (Dubey et al. 2024), including Qwen3-8B, Qwen3-32B, Qwen3-235B, and Llama3.3-70B.
- **Fine-tuned model** xLAM2-32B (Prabhakar et al. 2025; Zheng et al. 2024), a model specifically fine-tuned for advanced multi-turn tool use, to compare against general-purpose models.

Metrics. Our primary evaluation metric is **Accuracy**, which measures the rate of successful task completion. The detailed methodology for calculating accuracy is provided in the Appendix.

Overall Performance

Table 2 presents the overall accuracy for each model across the entire ToolLoad-Bench dataset. The results immediately underscore the challenging nature of our benchmark. A clear performance hierarchy emerges: the leading closed-source models form a top tier, with GPT-4o at 68.0%. Among the open-source models, performance generally scales with size, with Qwen3-235B outperforming its smaller variants.

However, the most striking result comes from the specialized fine-tuned model, xLAM2-32B, which achieves the highest accuracy at 78.8%. Despite its smaller size, its focused training allows it to significantly outperform larger general-purpose models. This finding strongly suggests that targeted fine-tuning is a highly effective strategy for boosting tool-use capabilities. The wide performance delta across all models demonstrates that ToolLoad-Bench effectively differentiates model capabilities.

Model	Overall Accuracy (%)
Closed-Source models	
GPT-4o	68
Claude 3.7 Sonnet	64.8
Gemini 2.5 Pro	60
GPT-4o-mini	62.2
Open-Source models	
Qwen3-235B	58
Llama3.3-70B	17
Qwen3-32B	55.2
Qwen3-8B	38.6
Fine-tuned model	
xLAM2-32B	**78.8**

Table 2: Overall Accuracy (%) on ToolLoad-Bench.

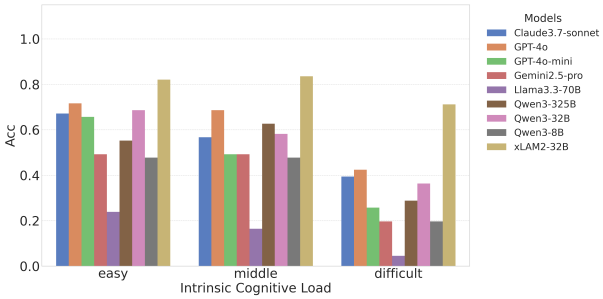

Figure 2: Accuracy vs. Intrinsic Cognitive Load (CL_I).

Impact of Cognitive Load

To understand how different facets of complexity affect performance, we analyzed model accuracy as a function of both intrinsic and extraneous cognitive load. We partitioned the dataset into low, medium, and high load buckets for each load type, with each bucket containing one-third of the instances.

Intrinsic Load Analysis. Figure 2 shows that accuracy consistently drops as the task's structural complexity (CL_I) increases. In the low-load regime, most models perform well, with the notable exception of Llama3.3-70B (23% accuracy), indicating a fundamental weakness. At high loads, performance collapses for general-purpose models. Only the specialized xLAM2-32B maintains over 60% accuracy, demonstrating its superior capability in handling complex reasoning structures.

Extraneous Load Analysis. A similar performance degradation is observed with rising Extraneous Cognitive Load (CL_E), as shown in Figure 3. Higher query ambiguity and the presence of distractor tools consistently reduce accuracy across all models. The performance patterns mirror the CL_I results: Llama3.3-70B again struggles, while only the fine-tuned xLAM2-32B sustains high accuracy under high extraneous load. This confirms that confusing task presentation is as significant a hurdle as inherent task complexity.

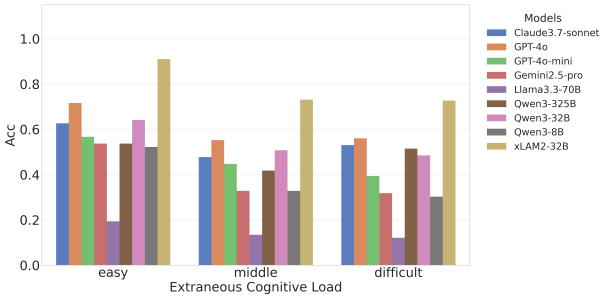

Figure 3: Accuracy vs. Extraneous Cognitive Load (CL_E).

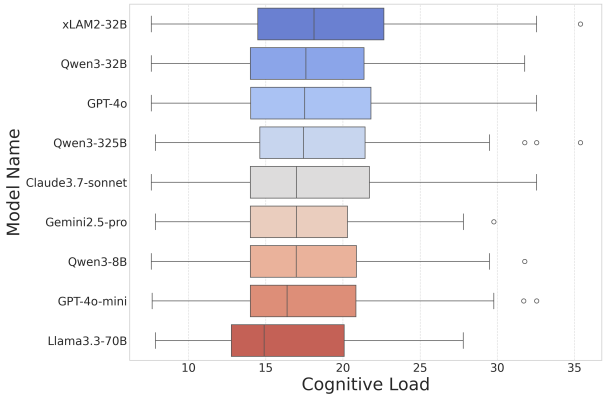

Figure 4: Boxplots showing the distribution of Total Cognitive Load (CL_{Total}) for successfully completed tasks by different models.

Synthesizing Total Cognitive Load

While analyzing intrinsic and extraneous loads separately provides valuable insights, a unified **Total Cognitive Load** (CL_{Total}) is required to holistically measure task difficulty and map a model's operational limits. The synthesis of these two components is not arbitrary but is directly guided by our Load-Success Relationship postulate (Equation 1).

The postulate implies that any two distinct cognitive challenges that cause an equivalent drop in success probability must correspond to an equivalent amount of cognitive load. This principle provides a direct method for calibrating the relative contributions of CL_I and CL_E onto a single, unified scale. We can empirically determine a scaling factor, ω_E, that quantifies how much a unit of our measured extraneous load impacts accuracy relative to a unit of intrinsic load. This factor is calculated as the ratio of their observed effects on model performance:

$$\omega_E = \frac{\Delta_{\text{Acc}}(CL_E)}{\Delta_{\text{Acc}}(CL_I)} \tag{10}$$

where $\Delta_{\text{Acc}}(CL)$ represents the empirically measured drop in accuracy associated with an increase in that type of load. By setting the weight of intrinsic load to 1 as our baseline, this ratio places extraneous load on the same effective scale. The final, model-specific Total Cognitive Load is then

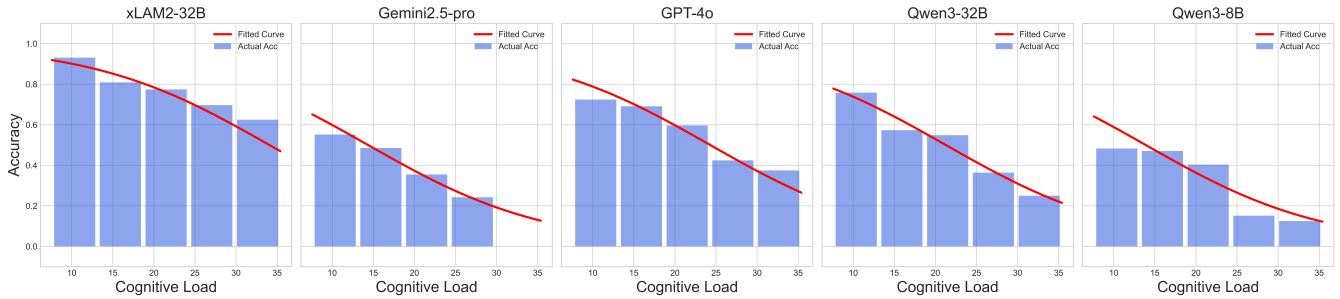

Figure 5: Empirical accuracy vs. Total Cognitive Load. The blue bars show the actual accuracy within binned load intervals, while the red line represents the fitted exponential decay curve from our theoretical model.

a principled weighted sum:

$$CL_{Total} = CL_I + \omega_E \cdot CL_E \tag{11}$$

This unified score reflects how a specific model weighs the challenges of inherent task complexity versus ambiguous presentation, providing a single, powerful metric for defining its capability boundary.

Defining Model Capability Limits with Total Cognitive Load

Having established a unified CL_{Total} score, we can move beyond overall accuracy to precisely map the operational boundaries of each model. This analysis proceeds in three steps: first, visualizing the range of solvable tasks; second, modeling the performance decay as a function of load; and third, extracting key parameters that define each model's cognitive profile.

Visualizing the Operational Range. We begin by visualizing the distribution of CL_{Total} for only those tasks that each model *successfully* completed. The boxplots in Figure 4 powerfully illustrate the effective cognitive capacity of each model. The distributions reveal stark differences: less capable models like Llama3.3-70B are confined to a narrow band of low-load tasks. In contrast, top-performing agents like the specialized xLAM2-32B and GPT-4o exhibit distributions with a higher median and a significantly wider range. The upper quartile and whisker of each distribution serve as a clear visual signature of a model's capability.

Modeling the Performance Decay Curve. While the boxplots show the range of solvable problems, they don't capture the probability of accuracy at different cognitive loads. We fit our theoretical Load-Success Relationship (Equation 9) to the empirical data for each model. Figure 5 plots the actual accuracy (blue bars) across binned cognitive load levels against the fitted exponential decay curve (red line). The close alignment between the empirical data and the theoretical curve provides strong visual validation for our framework.

Quantifying Capability with Model-Specific Parameters. The fitted curves can be characterized by the parameters k and b from our core equation, Accuracy $\approx \exp(-(k \cdot CL_{Total} + b))$. These parameters provide a concise, quantitative summary of a model's cognitive profile:

- **Baseline Capability (b):** This parameter reflects the model's intrinsic capability at near-zero cognitive load. A lower b corresponds to a higher starting accuracy (Acc $\approx e^{-b}$ at $CL_{Total} = 0$), indicating a stronger foundational ability.

- **Load Sensitivity (k):** This parameter measures how resilient the model is to increasing cognitive load. A smaller k signifies a flatter decay curve, meaning the model's performance degrades more gracefully under pressure.

Table 3 presents the fitted k and b values for key models. The specialized xLAM2-32B exhibits the lowest k and a very low b, quantifying its dual strength: the highest baseline proficiency and exceptional robustness to complexity. In contrast, GPT-4o shows a similarly low sensitivity but a slightly higher baseline load, suggesting it is highly capable on simpler tasks but its performance degrades more quickly than the specialized fine-tuned model. The open source Qwen3 models show a clear progression, with the 235B model approaching the capability of closed-source giants, while the smaller 8B model has both a weaker baseline and higher sensitivity. This parametric analysis transforms the abstract notion of "capability" into a concrete, two-dimensional profile, precisely defining each agent's strengths and breaking points.

Model	Load Sensitivity(k)	Baseline Load(b)
xLAM2-32B	**0.034**	1.22
GPT-4o	0.067	1.71
Claude 3.7	0.073	1.57
Gemini2.5-pro	0.088	1.22
Qwen3-32B	0.075	1.60
Qwen3-8B	0.085	**1.12**

Table 3: Fitted cognitive load parameters for different models. Lower k indicates better resilience to load, and lower b indicates higher baseline accuracy.

Validating the Cognitive Load Distributional Model

Our framework's central hypothesis is that cognitive load shapes the *probability distribution* of accuracy, not that it deterministically predicts an outcome. To validate this, we assess the **goodness-of-fit** between our model's predicted probabilities and the empirically observed accuracy. We use two complementary methods for this validation.

First, we employ the formal Hosmer-Lemeshow (H-L) statistical test (Paul, Pennell, and Lemeshow 2013). For this test, the null hypothesis is that the model is well-calibrated, meaning a high p-value is the desired outcome. As shown in Table 4, the p-values for all evaluated models are well above the conventional 0.05 significance level. This provides strong statistical evidence that our framework generates a probability distribution of accuracy that is statistically indistinguishable from the observed reality.

Model	H-L χ^2 Statistic	p-value
Closed-Source models		
GPT-4o	4.87	0.77
Claude 3.7 Sonnet	10.47	0.23
Gemini 2.5 Pro	13.15	0.11
GPT-4o-mini	8.91	0.35
Open-Source models		
Qwen3-235B	5.19	0.74
Llama3.3-70B	13.21	0.10
Qwen3-32B	7.50	0.48
Qwen3-8B	7.90	0.44
Finetuned model		
xLAM2-32B	**3.59**	**0.89**

Table 4: Hosmer-Lemeshow Goodness-of-Fit Test Results.

Second, we visually corroborate this statistical finding with calibration plots, as shown in Figure 6. The plots reveal a close alignment between the predicted probabilities and the observed accuracy, with points lying near the diagonal line of perfect calibration. Together, these results validate our framework's foundational assumption, demonstrating that it accurately models the nuanced, probabilistic nature of tool-use accuracy.

Discussion

Theoretical Contribution: A Scientific Framework for Evaluation. From a theoretical standpoint, our primary contribution is the introduction and validation of cognitive load as a formal construct for measuring task complexity in tool-use scenarios. We moved beyond abstract notions of difficulty by operationalizing it into measurable components: Intrinsic Load (CL_I) from the task's inherent structure and Extraneous Load (CL_E) from its presentation. To test this theory, we constructed **ToolLoad-Bench**, a benchmark specifically designed for controlled experimentation. Our experiments (Figures 2 and 3) provide strong empirical evidence for our central hypothesis: a direct and predictable relationship exists between a task's cognitive load and a model's accuracy. This provides the research community with a more scientific and comprehensive methodology for evaluation, enabling a shift from simple leaderboards to diagnostic analysis.

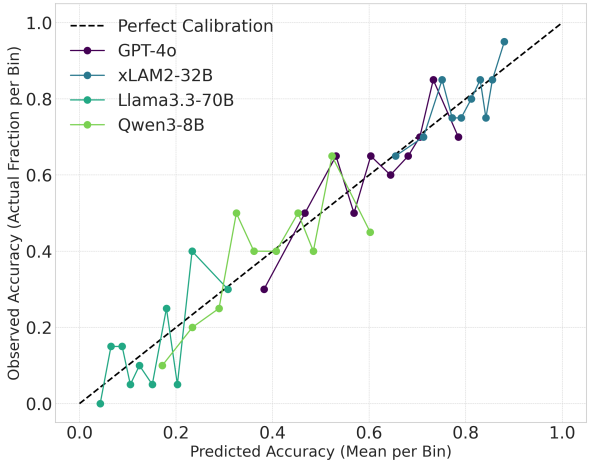

Figure 6: Calibration plot for representative models, showing predicted probabilities vs. observed accuracy.

Application Value: Enabling Intelligent Task Routing. From a practical standpoint, the ability to quantify cognitive load offers immediate application value. Our framework quantitatively reveals tool-use agents' distinct capabilities for handling tasks with varying cognitive loads. This provides a principled basis for a critical real-world application: intelligent task routing (Yue et al. 2025; Hu et al. 2024). In a production system, it could dynamically route the task to the most appropriate LLM based on cognitive scores. This approach enables the design of highly efficient, scalable, and economically viable tool-use agent systems.

Limitations and Future Work. Finally, we acknowledge the limitations of our current framework. While ToolLoad-Bench is highly controlled, its domain coverage could be expanded to ensure broader generalizability. Furthermore, our measurement of extraneous load (CL_E) currently relies on LLM evaluation, and developing more objective, feature-based metrics would strengthen the framework.

Conclusion

This work challenges the prevailing evaluation paradigm for tool-augmented LLMs, which reduces agent capability to a single, opaque score. We introduce a diagnostic framework grounded in Cognitive Load Theory that moves beyond simple accuracy to map the capability boundaries of tool-use agents. Our central finding is that models possess distinct cognitive frontiers, exhibiting sharp, predictable performance cliffs as task complexity increases. Compared to traditional tool-use agent evaluations, our research provides a principled methodology for understanding their true limits.

Acknowledgements

This work was supported by the National Natural Science Foundation of China(No.U21B2009).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Chen, C.; Hao, X.; Liu, W.; Huang, X.; Zeng, X.; Yu, S.; Li, D.; Wang, S.; Gan, W.; Huang, Y.; et al. 2025a. ACEBench: Who Wins the Match Point in Tool Learning? *arXiv e-prints*, arXiv–2501.

Chen, H.; Song, Z.; Niu, B.; Zhang, K.; Ou, L.; Lu, Y.; Zhang, Z.; Cong, X.; Lin, Y.; Liu, Z.; et al. 2025b. ToLeaP: Rethinking Development of Tool Learning with Large Language Models. *arXiv preprint arXiv:2505.11833*.

Chen, H.; Zhu, C.; Li, Y.; and Driggs-Campbell, K. 2025c. Tool-as-interface: Learning robot policies from human tool usage through imitation learning. *arXiv preprint arXiv:2504.04612*.

Chen, Z.; Du, W.; Zhang, W.; Liu, K.; Liu, J.; Zheng, M.; Zhuo, J.; Zhang, S.; Lin, D.; Chen, K.; et al. 2023. T-eval: Evaluating the tool utilization capability of large language models step by step. *arXiv preprint arXiv:2312.14033*.

Dong, G.; Chen, Y.; Li, X.; Jin, J.; Qian, H.; Zhu, Y.; Mao, H.; Zhou, G.; Dou, Z.; and Wen, J.-R. 2025a. Tool-Star: Empowering LLM-Brained Multi-Tool Reasoner via Reinforcement Learning. *arXiv preprint arXiv:2505.16410*.

Dong, G.; Mao, H.; Ma, K.; Bao, L.; Chen, Y.; Wang, Z.; Chen, Z.; Du, J.; Wang, H.; Zhang, F.; et al. 2025b. Agentic Reinforced Policy Optimization. *arXiv preprint arXiv:2507.19849*.

Du, Y.; Wei, F.; and Zhang, H. 2024. Anytool: Self-reflective, hierarchical agents for large-scale api calls. *arXiv preprint arXiv:2402.04253*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Farn, N.; and Shin, R. 2023. ToolTalk: Evaluating Tool-Usage in a Conversational Setting. arXiv:2311.10775.

Feng, J.; Huang, S.; Qu, X.; Zhang, G.; Qin, Y.; Zhong, B.; Jiang, C.; Chi, J.; and Zhong, W. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.

Guidroz, T.; Ardila, D.; Li, J.; Mansour, A.; Jhun, P.; Gonzalez, N.; Ji, X.; Sanchez, M.; Kakarmath, S.; Bellaiche, M. M.; et al. 2025. LLM-based Text Simplification and its Effect on User Comprehension and Cognitive Load. *arXiv preprint arXiv:2505.01980*.

Hu, Q. J.; Bieker, J.; Li, X.; Jiang, N.; Keigwin, B.; Ranganath, G.; Keutzer, K.; and Upadhyay, S. K. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

Li, M.; Zhao, Y.; Yu, B.; Song, F.; Li, H.; Yu, H.; Li, Z.; Huang, F.; and Li, Y. 2023. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. arXiv:2304.08244.

Lin, Q.; Wen, M.; Peng, Q.; Nie, G.; Liao, J.; Wang, J.; Mo, X.; Zhou, J.; Cheng, C.; Zhao, Y.; et al. 2024. Hammer: Robust function-calling for on-device language models via function masking. *ArXiv preprint*, abs/2410.04587.

Mialon, G.; Dessì, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Rozière, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Patil, S. G.; Mao, H.; Cheng-Jie Ji, C.; Yan, F.; Suresh, V.; Stoica, I.; and E. Gonzalez, J. 2025. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Forty-second International Conference on Machine Learning*.

Patil, S. G.; Zhang, T.; Wang, X.; and Gonzalez, J. E. 2023. Gorilla: Large Language Model Connected with Massive APIs. arXiv:2305.15334.

Paul, P.; Pennell, M. L.; and Lemeshow, S. 2013. Standardizing the power of the Hosmer–Lemeshow goodness of fit test in large data sets. *Statistics in medicine*, 32(1): 67–80.

Plass, J. L.; Moreno, R.; and Brünken, R. 2010. Cognitive load theory.

Prabhakar, A.; Liu, Z.; Zhu, M.; Zhang, J.; Awalgaonkar, T.; Wang, S.; Liu, Z.; Chen, H.; Hoang, T.; Niebles, J. C.; et al. 2025. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*.

Qian, C.; Acikgoz, E. C.; He, Q.; Wang, H.; Chen, X.; Hakkani-Tür, D.; Tur, G.; and Ji, H. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*.

Qin, Y.; Hu, S.; Lin, Y.; Chen, W.; Ding, N.; Cui, G.; Zeng, Z.; Zhou, X.; Huang, Y.; Xiao, C.; et al. 2024. Tool learning with foundation models. *ACM Computing Surveys*, 57(4): 1–40.

Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; Zhao, S.; Hong, L.; Tian, R.; Xie, R.; Zhou, J.; Gerstein, M.; Li, D.; Liu, Z.; and Sun, M. 2023. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. arXiv:2307.16789.

Qu, C.; Dai, S.; Wei, X.; Cai, H.; Wang, S.; Yin, D.; Xu, J.; and Wen, J.-R. 2025. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8): 198343.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Shen, W.; Li, C.; Chen, H.; Yan, M.; Quan, X.; Chen, H.; Zhang, J.; and Huang, F. 2024. Small llms are

weak tool learners: A multi-llm agent. *arXiv preprint arXiv:2401.07324*.

Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; and Zhuang, Y. 2023. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in HuggingFace. In *Advances in Neural Information Processing Systems*.

Shi, D.; Cao, J.; Chen, Q.; Sun, W.; Li, W.; Lu, H.; Dong, F.; Qin, T.; Zhu, K.; Liu, M.; et al. 2025a. Taskcraft: Automated generation of agentic tasks. *arXiv preprint arXiv:2506.10055*.

Shi, Z.; Gao, S.; Chen, X.; Feng, Y.; Yan, L.; Shi, H.; Yin, D.; Ren, P.; Verberne, S.; and Ren, Z. 2024. Learning to use tools via cooperative and interactive agents. *arXiv preprint arXiv:2403.03031*.

Shi, Z.; Gao, S.; Yan, L.; Feng, Y.; Chen, X.; Chen, Z.; Yin, D.; Verberne, S.; and Ren, Z. 2025b. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM on Web Conference 2025*, 2222–2237.

Sîrbu, A.-M.; Schelhorn, T. C.; and Gnewuch, U. 2025. Explanation Provision Strategies in LLM-based Data Assistants: Impact on Extraneous Cognitive Load, Trust, and Task Performance.

Su, Z.; Li, L.; Song, M.; Hao, Y.; Yang, Z.; Zhang, J.; Chen, G.; Gu, J.; Li, J.; Qu, X.; et al. 2025. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*.

Sweller, J. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2): 257–285.

Tang, Q.; Deng, Z.; Lin, H.; Han, X.; Liang, Q.; Cao, B.; and Sun, L. 2023. ToolAlpaca: Generalized Tool Learning for Language Models with 3000 Simulated Cases. arXiv:2306.05301.

Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv preprint*, abs/2403.05530.

Wang, H.; Qian, C.; Zhong, W.; Chen, X.; Qiu, J.; Huang, S.; Jin, B.; Wang, M.; Wong, K.-F.; and Ji, H. 2025. Otc: Optimal tool calls via reinforcement learning. *arXiv e-prints*, arXiv–2504.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Xiao, C.; and Yang, B. 2025. Streaming, Fast and Slow: Cognitive Load-Aware Streaming for Efficient LLM Serving. *arXiv preprint arXiv:2504.17999*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, Y.; Wang, Y.; Ma, C.; Yu, L.; Chersoni, E.; and Huang, C.-R. 2025b. Sparse Brains are Also Adaptive Brains: Cognitive-Load-Aware Dynamic Activation for LLMs. *arXiv preprint arXiv:2502.19078*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yin, F.; Wang, Z.; Hsu, I.; Yan, J.; Jiang, K.; Chen, Y.; Gu, J.; Le, L. T.; Chang, K.-W.; Lee, C.-Y.; et al. 2025. Magnet: Multi-turn tool-use data synthesis and distillation via graph translation. *arXiv preprint arXiv:2503.07826*.

Yue, Y.; Zhang, G.; Liu, B.; Wan, G.; Wang, K.; Cheng, D.; and Qi, Y. 2025. Masrouter: Learning to route llms for multi-agent systems. *arXiv preprint arXiv:2502.11133*.

Zhang, S.; Dong, Y.; Zhang, J.; Kautz, J.; Catanzaro, B.; Tao, A.; Wu, Q.; Yu, Z.; and Liu, G. 2025. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*.

Zheng, L.; Huang, Z.; Xue, Z.; Wang, X.; An, B.; and Yan, S. 2024. Agentstudio: A toolkit for building general virtual agents. *arXiv preprint arXiv:2403.17918*.