The Claude, The Sonnet, and the Wholly Spearmint

Michael Simon*
UCLA
Los Angeles, California, USA
mlsimon@cs.ucla.edu

Matthew Workman*
UCLA
Los Angeles, California, USA
mworkman@cs.ucla.edu

Ken Zhou*
UCLA
Los Angeles, California, USA
kbzhou01@cs.ucla.edu

Abstract

Though users prioritize efficiency in their Large Language Model (LLM)-powered workflows, perceptions of model efficiency vary between user personas and use cases. Investigations into user interactions with reasoning models and non-reasoning models exposed user perception of LLM efficiency as a sliding scale of response depth/quality v.s. low latency. In this paper, we introduce **Claude, Sonnet, and Wholly Spearmint (CSWS)**, a novel approach that uses a real-time NLP heuristic aggregation to dynamically assess a user's query and determine an optimal thinking budget for reasoning models. This budget enforces an upper bound for response length (and thus latency) while ensuring that response depth aligns with the query's requirements. Our evaluation centers around user A/B testing between CSWS and one of DeepSeek or ChatGPT. Preliminary survey and interview results illustrate that CSWS produces higher user satisfaction with response quality compared to ChatGPT, while its minimization of latency when compared to DeepSeek enabled higher user ratings and better user perceptions overall.

CCS Concepts: • **Human-centered computing** → **Natural language interfaces**.

Keywords: Human-LLM Interaction, Large Language Models

1 Introduction

The introduction of ChatGPT for public use in late 2022 took the world by storm [12]. Large Language Models (LLMs), powerful agents capable of addressing diverse use cases, are now integral to hobbyist, academic, and professional workflows. With ChatGPT being the most popular platform used in the country [13], as well as most users' first introduction to an LLM platform, user interaction techniques with LLMs have adapted to the qualities of OpenAI's platform. This adaption manifests in two significant ways: the assumption of a low-latency response, as well as the expectation of potential hallucinations and model inadequacies.

ChatGPT's low-latency responses trained users to expect quick interactions, while its known limitations in reasoning fostered a pragmatic approach, often involving iterative prompting and fact-checking. However, with the advent of reasoning models like DeepSeek-R1, users were introduced to the benefits of deeper, more considered responses, albeit at the cost of increased latency [4]. This presents a dilemma: users value both speed and quality, but existing LLM interfaces typically force a trade-off. A clear gap exists in current LLM interfaces: they fail to dynamically adapt to varying user needs in terms of speed and depth. This rigidity leads to inefficiencies: users with simple queries may be frustrated by the latency of reasoning models, while users with complex tasks may find the rapid responses of non-reasoning models lacking in depth and requiring excessive iteration.

To bridge this gap and unify these seemingly disparate approaches, we introduce **Claude, Sonnet, and Wholly Spearmint (CSWS)**[1]. CSWS is a novel system designed to dynamically adjust the "thinking effort" of a reasoning model based on the perceived complexity of the user's query. Leveraging Anthropic Claude 3.7 Sonnet's "thinking budget" parameter, CSWS employs a real-time heuristic to estimate query complexity and automatically allocate an appropriate thinking budget [1]. This allows CSWS to offer both rapid responses for simple queries and in-depth, reasoned responses for complex tasks, all within a single interface. In this paper, we will detail our user research that motivated this design, discuss related work in adaptive LLM interfaces, elaborate on the design and implementation of CSWS, present our evaluation methodology and findings, and finally, conclude with a discussion of limitations and future directions for dynamically adaptable LLM interfaces.

2 Related Work

Though LLMs have entered the HCI research space with a fury in recent years, studies specifically over optimizing LLM latencies have largely been relegated to the Natural Language domain. Natural Language Processing research prioritizes maximizing tokens per second (TPS) and minimizing memory needs, assuming faster token generation equates to faster response times. However, this paradigm is now being challenged, as concision emerges as another valuable metric to consider; a model which is fast but always provides verbose answers will have a longer end-to-end latency than a model which is slightly slower but is concise. A recent study of open weight 7-8B parameter LLMs revealed that TPS and end-to-end latency divergence is significant enough for noticeable differences between LLMs latencies for a set of common tasks to appear [3].

*All authors contributed equally to this research.

[1]The demo for our system is live at https://csdubz.streamlit.app/ with a BYO Anthropic API key policy.

Additionally, there are some pure ML works which explore dynamic test-time latency adjustments through unique model architectures:

2.1 NLP Methods

Adaptive Computation Time (ACT) [8] was an algorithm originally proposed for RNNs that dynamically allocates computational resources based on the complexity of the input. Each RNN step predicts if further computation is necessary. There is a "ponder cost" that penalizes prolonged computation, providing a trade-off between accuracy and efficiency. ACT aims to encourage the model to focus on what is truly needed. On language modeling tasks, ACT did not yield large performance gains, but did appropriately allocate resources to harder-to-predict transitions. Although RNNs have fallen out of favor since transformers took over, this was an early work which emphasized the importance of test-time latency being as important as overall model efficiency.

DACT-BERT [5] applies the Adaptive Computation Time paradigm to the BERT model. This variant of the model has a dynamic halting mechanism which determine the optimal number of transformer layers to engage based on the input. At each layer, there is some calculated halting probability and when the probabilities accumulate into a value exceeding a threshold, the model halts and combines the existing outputs. Similarly, DACT-BERT uses a "ponder cost" to penalize excessive computation. DACT-BERT excelled in low resource scenarios and performed comparably to non-halting architecture in regard to accuracy. It provides another compelling example to further research inference-time latency controllability.

2.2 LLMs in HCI

It's worth noting the context in which HCI research has explored LLM chat interfaces. Chat interfaces have become the dominant interface by which people interact with AI systems, with LLMs being used significantly more by end-users than systems like classifiers which work behind the scenes. However, people don't always use the interface purely conversationally, with a variety of different workflows beyond the standard call-and-response. Even LLMs with reasoning in a standard chat interface can be considered a different mode in the greater taxonomy of user-LLM interactions [7]. An emerging research direction in HCI is the study of interfaces which better supplement non conversational use cases, where latencies on the order of minutes or hours for multi-agent non-deterministic LLM workflows.

Other HCI works have explored additional methods to optimize system latency without modifying underlying models as in [10] where the authors created an efficient UI agent by relying on application APIs to interact with the UI instead of a slower multimodal LLM to interact with the visual interface directly.

A fully non-chat-UI based form of human-LLM interaction is the voice interface, where a text-to-speech model is directly integrated with the underlying LLM, allowing the user to interact primarily through sound. In this domain, there is a range of acceptable latencies on the TPS scale, since the text-to-speech model is only expected to "say" words at a understandable rate. Although latency still remains a challenge [2], the vocal response paradigm provides a lot more leeway for latency than a purely GUI interface.

3 User Research

3.1 Method

User Research was conducted to investigate the impact of higher latency between question and response on user attentiveness during LLM use. To this end, two models were selected:

- **ChatGPT**: an LLM platform with low latency between query and response. As mentioned earlier, its ubiquity makes it a tool that most users have used before.
- **DeepSeek**: a reasoning LLM with relatively high latency between query and response, as well as a displayed 'thinking process' that serves as indicator, reasoning, and transparency into the model's thinking.

Participants were drawn from a pool of college students, both undergraduates and graduates. We reasoned that this was appropriate for the purposes of a think-aloud due to our age-group's relative technological literacy and familiarity with LLMs, the prevalence of non-trivial workflows in the participants' majors, as well as the relative level of interpersonal familiarity already-established that would enable greater openness during the study. Additionally, students' general affinity for tools which make accomplishing advanced homework or assignments easier means the group as a whole is more likely to use LLM interfaces for non-trivial, complex tasks which could test the effectiveness of our system on it's robustness for harder tasks. Studies were conducted in a virtual format, where participants would share both their camera and their screen while executing the steps of the study.

Participants were told to (1) identify a representative, ideally non-trivial workflow to use the large language model for, (2) execute the workflow on ChatGPT until completion, (3) execute the identical workflow on DeepSeek, using the same first query as ChatGPT, until completion or server timeout. Throughout this process, users were encouraged to vocalize their thoughts and commentate their experience. Afterwards, a brief, unstructured interview concerning their perception of both models' latency was conducted.

3.2 Findings

When interviewed on their experience with higher latency and their preferences between DeepSeek and ChatGPT, users were split roughly evenly on their preferences. Users that

preferred ChatGPT emphasized the relative simplicity of their use case, asserting the redundancy of DeepSeek's reasoning process as well as its prohibitive latency. Users that preferred DeepSeek emphasized the higher level of insight in its responses, as well as the transparency in its reasoning that enabled user trust in its output, especially in more complex use cases. From observing user interactions with the models as well as synthesizing their perceptions and feedback, three approximate personas were determined:

1. **Persona A:** The Oracle Seeker ("Adam"): a user with a generally high level of trust in the LLM, treating the model as a resource with which to gain powerful insights or solutions. To this end, this persona was generally more resilient to higher wait times under the expectation of a satisfactory response. This persona generally appreciated the thinking response as an exciting insight to the model's processes.
   a. Workflow example: explaining a complex topic, e.g. "Explain to me the benefit of quantum computing, specifically how multiple states can enable greater power than binary states."
2. **Persona B:** The Google Searcher ("Barb"): a user with a blanket assumption of trust in the LLM, treating the model as a resource to answer queries with very limited amounts of overhead or complexity. To this end, this persona was extremely sensitive to latency, prioritizing near-instantaneous response times over answer depth. This persona heavily disliked the thinking response as a result of its high latency.
   a. Workflow example: a trivial example generation, e.g. "Generate for me a list of 10 molecular chemistry problems"
3. **Persona C:** The Intern Manager ("Chet"): a user with a low level of trust in the LLM, treating the model as an agent with the potential to solve a workflow, but without the individual ability to arrive to the solution without human guidance and prompt iteration. To this end, this persona highly valued transparency, and naturally emphasized answer depth and quality as an important value due to the complexity of their use cases. This persona appreciated the thinking response due to its enablement of higher quality, higher correctness answers: as well as the transparency it provided in the event these answers came under question.
   a. Workflow example: decoding an encrypted file, e.g. "I have this encrypted file with the following contents. First identify its encryption format. Then, describe to me how to deencrypt it. Finally, provide a script that executes that deencryption."

3.2.1 User Perceptions of Response Quality and Depth.

Qualitatively, users frequently commented on the perceived depth and quality of responses from DeepSeek in contrast to ChatGPT. For example, a user fitting Persona A noted how "DeepSeek was able to create a more convincing argument for its responses and scenarios," making the longer latency tolerable, but only when "[end] quality is valued." In contrast, users sometimes found ChatGPT's responses to be sufficient for simple tasks but lacking in depth for more complex queries. A user fitting Persona C, mentioned that while the "thinking" visibility of DeepSeek "[ensured] that the AI is proving its understanding", their general skepticism in the ability for any AI to one-shot a tough task still meant quick iterations were important.

The tension between the desired model qualities of Persona A and Persona C guided our synthesis of the problem statement.

3.3 Problem Statement

We define the high-level concept of "net time spent" as a metric of efficiency, as this concept reappeared frequently in our user research. Large language models accumulate 'time spent' over three intervals. The first: the time spent to engineer a prompt, waiting for the model to finish its response (including reasoning time), and iterating and correcting. If the perception of net time spent using an LLM exceeds that of the time that would otherwise be spent finishing the task oneself, then the user is discouraged from using an LLM. Therefore, to synthesize our qualitative findings from our research, we list the following conclusions:

1. All users value answer correctness, e.g. "will the model provide the correct answer?"
2. All users value efficiency, e.g. "how much net time will be spent to receive the final answer?"
3. The values or qualities that determine efficiency depend on the assumption of correctness, or the trust the user has in the system:
   a. Workflows and use-cases where answer correctness can be assumed (e.g. persona B's simplistic tasks) make low-latency systems paramount.
   b. Workflows and use-cases where answer correctness cannot be assumed (e.g. persona C's complex, novel workflows) make answer 'quality' and transparency important.
   c. If a user engages with these workflows in an iterative manner, low latency per iteration becomes heavily important; however, in the case of persona C's workflows, these iterations generally represented workarounds to intrinsic weaknesses in the model.
4. If a user's LLM interaction is not efficient, this discourages the user from using an LLM to execute their workflow.

Following these conclusions, we identify the following problem statement: Users highly prioritize efficiency in their AI-powered workflows; in higher complexity workflows, this is a balance between maximizing answer quality, minimizing

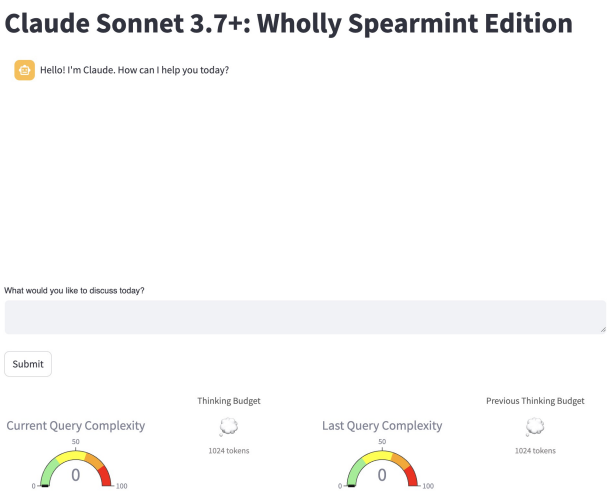

Figure 1. Web-App User Interface

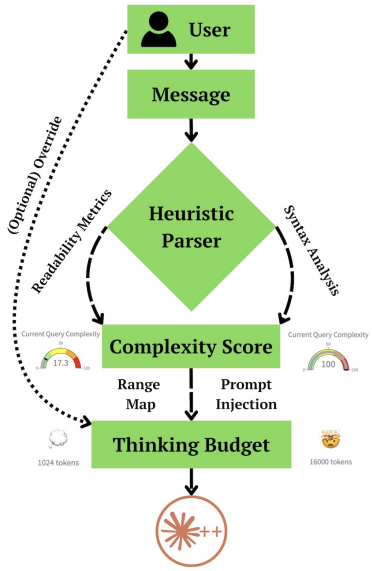

Figure 2. Backend process executed on a per-query basis

net time spent, and preserving transparency. While the qualities desirable for low-complexity workflows such as low latency are largely served by popular market options: existing market options fail to deterministically distinguish between high and low complexity queries, introducing inefficiencies that discourage user engagement with these systems.

4 Design Goals

To address the problem illustrated in the problem statement, two design goals were created that guided the creation of the CSWS system.

The first design goal: "Reduce the net time spent to complete higher complexity workflows without significantly compromising the efficiency of lower complexity workflows." User research revealed that issues with transparency and answer quality necessitated frequent iterative queries to either correct or guide the model under high complexity scenarios, both of which heavily impact the net time spent.

Addressing this goal takes on two implementation details. First, to promote higher answer quality, CSWS wraps around Anthropic's Claude-3.7 Sonnet as its core model, leveraging Claude's reasoning model to reduce iteration counts with the power of reasoning-enabled responses. In so doing, we hoped to reduce the net time spent in iteration or correction. Second, we leverage Claude's thinking budget API to provide a deterministic upper bound to thinking latency, thereby deterministically capping the overhead associated with the model's reasoning.

The second design goal: "Minimize the effort required for users to adapt a large language model to their workflow."

While user research revealed that both DeepSeek and ChatGPT could be well suited to their respective extremes of use cases, neither model provided a one-size fits all approach to the vast range of query complexities that users could arrive with. Furthermore, there is a non-trivial amount of overhead for trying multiple models for a single query.

To address this, we implemented a heuristic analyzer of user prompts that maps user workflows to a *thinking budget*: a token limit that serves as an upper bound for the model's reasoning process. Such a heuristic enables low-complexity queries a relatively faster response, while enabling higher-complexity queries the requisite thinking budget generate a response with high answer quality.

5 System Design & Implementation

5.1 Overview

A user inputs their query in a text area. This query, upon key-up, will have its complexity score determined live and shown to the user. The user may also choose to toggle an override, which lets them set a complexity of their choice. A thinking budget corresponding to the complexity score is simultaneously determined and displayed. Upon query submission, the Claude Sonnet 3.7 model responds, with its thinking tokens limited by our budget.

5.2 Heuristic

To ensure fast evaluation, the complexity score heuristic is comprised of syntactic analysis, readability metrics, and punctuation density.

5.2.1 Syntactic Analysis. This component is of the heuristic is calculated from type-token ratio (TTR), average word length, and punctuation density. Type-token ratio measures lexical diversity by comparing the number of unique words, types, to total words. A high TTR is indicative of a varied vocabulary, which could indicate a more complex input. Both

of these metrics are normalized, assuming ranges of 0.3 - 1 and 3 to 10 letters respectively, and then averaged to produce an overall lexical score.

5.2.2 Readability Metrics. The following are the bulk of the overall complexity score:

Flesch-Kincaid Grade Level[6]: This metric aims to approximate the United States grade level that corresponds to a piece of text and is calculated using total words, syllables, and sentences.

$$0.39 \times \left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8 \times \left(\frac{\text{total syllables}}{\text{total words}}\right)$$

This metric is limited by its lack of an upper bound and can be "gamed" by producing a high syllable count string.

Gunning FOG Index[9]: This metric also aims to approximate the number of years of formal education required to read a text. It's based on the assumption that high syllable words are more complex.

$$0.4 \times \left(\frac{\text{words}}{\text{sentences}} + 100 \times \left(\frac{\text{complex words}}{\text{words}}\right)\right)$$

SMOG Index[11]: This metric attempts to estimate the number of years of education required to comprehend a text.

$$1.043 \times \sqrt{\text{Polysyllabic Word Count} \times \frac{30}{\text{Sentence Count}}} + 3.1291$$

These scores are normalized assuming a range of 0 - 20.

5.2.3 Punctuation Density. A higher density of punctuation implies a more involved sentence structure with clauses, lists, and/or embedded phrases. This score is normalized, assuming a range of 0 - 0.2.

5.2.4 Overall Formula. The overall complexity is determined by taking the minimum of 1 and the following weighting of above scores:

- Lexical Score: **0.15**
- Flesch-Kincaid: **0.2**
- Gunning FOG Index: **0.3**
- SMOG Index: **0.3**
- Punctuation Density: **0.05**

The speed of this heuristic comes at a cost, as its determination of complexity is only a rough scoring and this computation overly weights high syllable count, so a long single word string will net a high complexity score.

To compensate for this, we provide the user the ability to override this score by manually setting a slider.

5.3 Prompting

Chat oriented LLMs have a system prompt that determines the persona, priorities, and initial understanding the model has about its role in its interactions with the user. As important as the heuristic is for deciding an upper bound for the model's thinking, a model prompt is integral to defining how the model reasons using the heuristic's provided budget.

Preliminary user pilots revealed a key concern. While a higher complexity score, implies a greater use of the thinking budget: users did not necessarily see a difference in the level of thinking when measured in tokens used. For example, the same prompt at two different complexity levels may result in the exact same reasoning and answer, whereas the user expectation is for the higher complexity version of the prompt to be answered with more insight.

To address this feedback, our prompt was augmented to specifically detail the following characteristics:

- Scenario setup with the computed thinking budget and complexity score of the user's prompt.
- A general set of defined thinking budget 'intervals' for low-medium, medium-high, and high complexity prompt (e.g. >7000 tokens defines a high-complexity prompt).
- Level of 'insight' expected at each complexity level. For example, low complexity scored prompts shouldn't seek to provide any more insight than what is explicitly relevant to the user's query, while the high complexity scored prompts should seek to provide this insight.

6 Evaluation

6.1 Guiding Question

The aim of this system is to improve user interactions with LLM chat interfaces by increasing the sense of perceived efficiency, be it through balancing latency and quality and/or reducing iteration count.

The evaluation conducted was guided by the following question: How well does heuristically defining a token limit for thinking models correlate with improving user satisfaction with their LLM-usage experience?

6.2 Methods

We performed A/B testing between the models initially used during user research (ChatGPT and DeepSeek) and CSWS. In the comparison with ChatGPT, we sought to discover deltas in answer quality, transparency, and iteration count with respect to the user's perceptions. In the comparison with DeepSeek, we wished to ascertain the effectiveness of defining thinking budget caps w.r.t. reducing the latency a user experiences when waiting for query responses. Both of these comparisons allow us to measure the validity of tuning the thinking budget specifically to a user's query complexity.

6.2.1 Quantitative Metrics.

- **Response Latency per Question** (sec): Allows us to see if response latency is improved with a thinking budget cap between the same queries on different

models, albeit this value is also dependent on server difficulties or rate limits for other models.
- **Number of Corrections/Clarifications**: Allows us to measure response quality (i.e. how much smarter a model is over another); however, the generalizability of this metric is not great because of differing use cases/user behaviors
- **User Rating on Quality/Latency**: Allows us to approximate a user's experience with the models, but, like all rating systems, is affected by user's hidden perceptions of what a rating corresponds to
- **Iteration Count**: Allows us to observe trustworthiness and extrapolative ability of models; however, this metric can be conflated with user assumptions or prompting behaviors.

Due to the subjective nature of the problem we tackle, these approximative metrics are subject to biases. However, by representing them in the aforementioned manners, we can aggregate data and observe trends/correlations.

6.2.2 Qualitative Metrics. In regards to qualitative data, we want to perform a structured interview in order do obtain responses to the following questions:

1. What was their attitude towards the detail, depth, and response length of the responses generated by our model vs. competitor models?
2. Did the amount of thinking they witnessed in our model accurately correlate to the level of transparency and depth required for their use case?
3. What did they feel was still missing from their experience?
4. Does this make them more willing to use LLMs for their applications?

Because the evaluation conducted is an A/B test between CSWS and older models, we are able to accurately glean the differences in user perception and user experience via the quantitative data we obtain as well as the user's 'think aloud' experience, which informs us of the effectiveness of heuristically guided thinking limits in both limiting the latency of thinking models and strengthening responses relative to 'over-thinking' or 'no-reasoning' models.

6.3 Approach

In an attempt to mitigate some biases, we varied the ordering of models shown and which models were selected as the Other Model. We covered all permutations of (ChatGPT/DeepSeek) vs. our system and vice versa, aiming to have roughly even representation. Because our evaluation pool consisted of 10 users, A perfectly even representation was impossible. The non-CSWS model assigned to each user was based on the user's familiarity with reasoning LLMs (those more familiar with the latest LLMs were assigned DeepSeek).

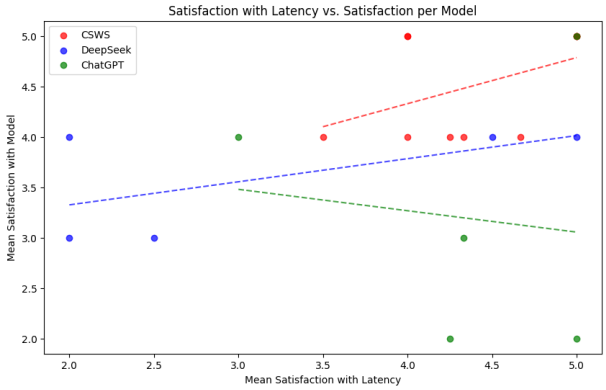

Figure 3. Linear Correlations between model satisfaction and latency satisfaction

The order of models (CSWS before/after the incumbent system) was randomly assigned to maximize permutation coverage.

Because all of the evaluation participants were people we knew, there was bias when it came to delivering feedback of our system as they could, consciously or not, have toned down their negative feelings. Verbal assurances were given that all thoughts were welcome, but this measure is not enough to fully counteract this bias.

7 Findings

Our evaluation, conducted with a limited sample size of 10 participants, was primarily exploratory, aiming to identify trends and gather qualitative insights into user perceptions of CSWS in comparison to ChatGPT and DeepSeek. While quantitative metrics provide a valuable overview of response latency and user ratings, the richness of our findings lies in the qualitative data gathered from user interviews and think-aloud protocols, which provide crucial context for the observed quantitative trends. The following sections will present both quantitative summaries and illustrative qualitative findings to paint a more complete picture of user experiences with CSWS.

7.1 Quantitative Results

The most telling result from our quantitative analysis of user feedback stems from Figure 3, where we at least loosely confirm some of our original assumptions about reasoning models vs standard LLMs. As expected, ChatGPT has the highest satisfaction scores w.r.t. latency, yet lags behind the thinking models in terms of overall model satisfaction. So though snappy, ChatGPT's overall quality lags the more sophisiticated models. There was a wide spread of satisfaction with DeepSeek's latency, likely due to the extreme variance in model response time (anywhere from 10 sec to 7 min). Lastly, CSWS has the highest general satisfaction, both with answer

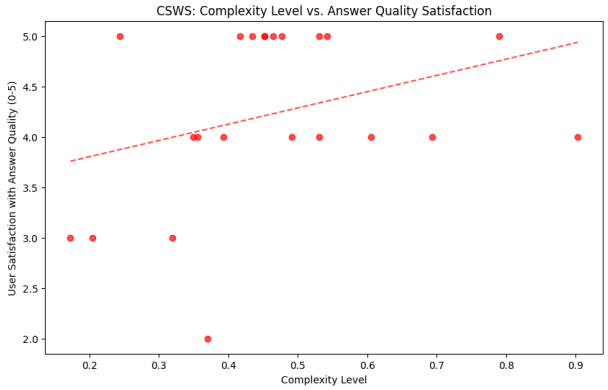

Figure 4. Linear Correlation between complexity and answer satisfaction

quality and latency, surpassing ChatGPT and CSWS in this regard.

Though the trends are not statistically significant given the small sample size, they provide some initial validation of CSWS's benefits over plain chat interfaces for either quick or slow reasoning models.

Another important quantitative evaluation is determining the effectiveness of our complexity metric, in Figure 4. Here we can observe a general positive correlation between the complexity score (and by extension, thinking budget) and answer quality rated by users. Importantly, this suggests that our complexity metric is at least a decent proxy for answer depth, since users were executing non-trivial workflows where answer depth was generally valued over concision. However, given the wide spread of answer satisfactions overall, this is far from conclusive evidence. Our qualitative analysis is a bit more enlightening in this regard.

7.2 Qualitative Results: User Experiences and Satisfaction

We observed a variety of qualitative reactions to our tool, which help contextualize our quantitative results. With the obvious caveat being that some reactions can be attributed to the underlying model's effectiveness rather than the user interface, we saw some cases where CSWS addressed our problem statement effective, and other cases where it fell short.

The effectiveness of CSWS was most apparent in Individual 6's evaluation. Individual 6's task was decoding a string into an executable and determining what the executable would do/determining if it was malicious. Individual 6 started with DeepSeek, which spent 7 minutes thinking and only to get the answer incorrect. Individual 6 decided to give up on the DeepSeek process afterwards due to the exceptional delay. Conversely, CSWS responded to the first query in 20 seconds with a comparable quality answer and, with slight guidance, was able to achieve the correct answer after additional prompting. DeepSeek doubled back and doubted itself multiple times in the 7 minute thinking period. By curtailing the thinking process, this unnecessary rambling was removed and the user was able to get to the desired answer in a timely, non-frustrating manner. This is a case where neither the latency nor answer quality was sufficient for DeepSeek, something that with it's current model, R1, is not addressable given the lack of a "thinking budget" parameter during inference.

We also got positive validation for our user interface, which provided easy to understand gauges and emojis to represent the complexity score and thinking depth "tier" without bombarding users with less intuitive pure numbers like the raw thinking token budget. As Individual 9 noted:

> "I think the visual gauges and emojis were a nice touch since it made it clear how long to expect the response to be and to take. With DeepSeek, there's no telling how long it's going to take to get to the final answer"

A more middling response to our tool's overall quality came from Individual 10, who was more familiar with LLM interfaces:

> "I noticed as I manually maxed out the complexity, the thinking time increased, but I still got the same response. So, it looks like the model used a more complicated approach to get to the same response."

In other words, when specifically A/B testing the complexity score with a manual override for the same prompt, Individual 10 found that increasing the thinking budget did not affect the final answer. They found this a shortcoming of our approach, even with our prompt injection working behind the scenes to try and extract maximum depth with increasing "depth" tiers. This is constructive criticism for future work, since this is mainly a shortcoming of the underlying model rather than an oversight on our interface's design.

8 Conclusion

In this paper, we introduced Claude, Sonnet, and Wholly Spearmint (CSWS), a novel system designed to dynamically adjust the thinking budget of a reasoning LLM based on heuristically assessed query complexity. Our evaluation, involving A/B testing with ChatGPT and DeepSeek, suggests that CSWS effectively balances response latency and quality, leading to improved user satisfaction, particularly for complex tasks. Our evalution is a step forwards for our hypothesis, that dynamically adapting LLM "thinking effort" based on query complexity can enhance user-perceived efficiency and bridge the gap between fast, shallow responses and slow, deep reasoning.

However, our study has several limitations. Our evaluation involved a small sample of college students, potentially limiting the generalizability of our findings to other user groups and tasks. The heuristic, while fast, is a simplification of true query complexity and has limitations, such as over-weighting long single-word strings. Furthermore, our A/B testing focused on comparisons with ChatGPT and DeepSeek; future work should include a direct comparison with Claude 3.7 Sonnet Thinking to isolate the impact of the thinking budget heuristic itself versus Anthropic's internal hyperparameter settings.

Future work should focus on refining the heuristic, exploring more sophisticated NLP techniques for complexity assessment, and conducting evaluations with larger and more diverse user populations and task domains. Finally, further research is needed to understand the psychological impact of presenting the LLM's "thinking process" to users, and how to optimize this presentation to maximize transparency without causing information overload.

Despite these limitations, our work provides initial evidence that dynamically adaptable LLM interfaces, like CSWS, offer a promising approach to addressing the latency vs. quality trade-off and enhancing user efficiency and satisfaction in LLM-powered workflows.

References

[1] Anthropic. 2025. Claude 3.7 Sonnet. https://www.anthropic.com/news/claude-3-7-sonnet

[2] Szeyi Chan, Shihan Fu, Jiachen Li, Bingsheng Yao, Smit Desai, Mirjana Prpa, and Dakuo Wang. 2024. Human and LLM-Based Voice Assistant Interaction: An Analytical Framework for User Verbal and Nonverbal Behaviors. arXiv:2408.16465 [cs.HC] https://arxiv.org/abs/2408.16465

[3] Javier Conde, Miguel González, Pedro Reviriego, Zhen Gao, Shanshan Liu, and Fabrizio Lombardi. 2024. Speed and Conversational Large Language Models: Not All Is About Tokens per Second. *Computer* 57, 8 (2024), 74–80. doi:10.1109/MC.2024.3399384

[4] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[5] Cristóbal Eyzaguirre, Felipe del Río, Vladimir Araujo, and Álvaro Soto. 2021. DACT-BERT: Differentiable Adaptive Computation Time for an Efficient BERT Inference. arXiv:2109.11745 [cs.CL] https://arxiv.org/abs/2109.11745

[6] Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology* 32, 3 (1948), 221–233. doi:10.1037/h0057532

[7] Jie Gao, Simret Araya Gebreegziabher, Kenny Tsu Wei Choo, Toby Jia-Jun Li, Simon Tangi Perrault, and Thomas W Malone. 2024. A Taxonomy for Human-LLM Interaction Modes: An Initial Exploration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 1–11. doi:10.1145/3613905.3650786

[8] Alex Graves. 2017. Adaptive Computation Time for Recurrent Neural Networks. arXiv:1603.08983 [cs.NE] https://arxiv.org/abs/1603.08983

[9] Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill.

[10] Junting Lu, Zhiyang Zhang, Fangkai Yang, Jue Zhang, Lu Wang, Chao Du, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. Turn Every Application into an Agent: Towards Efficient Human-Agent-Computer Interaction with API-First LLM-Based Agents. arXiv:2409.17140 [cs.AI] https://arxiv.org/abs/2409.17140

[11] G. Harry McLaughlin. 1969. SMOG grading: A new readability formula. *Journal of Reading* 12, 8 (1969), 639–646.

[12] OpenAI. [n. d.]. ChatGPT. https://openai.com/index/chatgpt/. Accessed: 2025-03-20.

[13] Zapier. 2025. The Best Large Language Models (LLMs) in 2025. https://zapier.com/blog/best-llm/ Accessed: 2025-03-20.