How Do Prompt Variations Affect Energy Consumption in On-Device LLMs?

Wei Hu[1]*, **Xiaolong Tu**[1]*, **Dawei Chen**[2], **Yitao Chen**[2], **Kyungtae Han**[2], **Haoxin Wang**[1]†
[1]Georgia State University [2]Toyota Motor North America
{whu6, xtu1}@student.gsu.edu, haoxinwang@gsu.edu
{dawei.chen1, yitao.chen, kt.han}@toyota.com

Abstract

Large language models (LLMs) are increasingly deployed on mobile devices, making energy efficiency a key deployment constraint, yet the energy impact of prompt design remains underexplored. This paper aims to understand how two prompt properties, cognitive load and phrasing pattern, shape the energy behavior of on-device LLM inference. We conduct a broad empirical study covering prompt properties, datasets, models, and devices, with phase-level profiling that separates prefill and decode energy. We find that cognitive load primarily affects the energy cost per token, while phrasing pattern affects energy largely through token usage. Our energy-quality analysis further shows that prompt design reshapes the attainable frontier differently across models, highlighting the need for model-aware prompt design in energy-efficient on-device LLM inference. Code, datasets, and scripts are available at https://amai-gsu.github.io/PromptProperty/.

1 Introduction

The deployment of Large Language Models (LLMs) is increasingly expanding beyond cloud serving toward on-device execution on mobile and edge platforms (Chen et al., 2020; Zhou et al., 2019). This trend is driven by the growing demand for stronger user privacy guarantees, lower inference latency, and reliable offline access (Das et al., 2025). However, unlike cloud-based inference, where energy consumption is amortized across large-scale infrastructure and has little immediate impact on users, on-device inference is powered by finite batteries, making energy efficiency a strict and user-visible constraint (Liu et al., 2024). As a result, energy efficiency has emerged as a key bottleneck limiting widespread adoption of on-device LLMs (Schwartz et al., 2020).

Prior research on efficient LLMs has overwhelmingly focused on model-centric optimization techniques, including quantization, pruning, architectural compression, and distillation (Xiao et al., 2023; Dettmers et al., 2022; Frankle and Carbin, 2019; Gu et al., 2024; Hinton et al., 2015). These approaches improve inference efficiency primarily by reducing model size and computational cost through parameter or architectural modification, and have become the dominant strategy for on-device LLM deployment. In addition, several recent studies have examined prompt sensitivity by analyzing how variations in prompt phrasing, structure, and formatting affect the quality and stability of model outputs (Long et al., 2025; Ismithdeen et al., 2025; Zhu et al., 2024; Lu et al., 2022). However, a fundamental question remains largely unexplored: *how does prompt design itself affect the energy consumption of on-device LLM inference?* If different prompt variations induce substantially different energy costs, even when producing comparable outputs, prompting could serve as a new, model-agnostic approach to improving energy efficiency beyond model-centric optimization.

Studying how prompt variations affect the energy consumption of on-device LLMs is non-trivial. First, prompts are linguistic artifacts, whereas energy consumption depends on the computation they trigger inside the model, and there is no direct mapping between the two. Understanding why a prompt is energy-consuming therefore requires carefully designed empirical analysis that systematically links prompt variations to differences in computational behavior and measured power consumption. Second, it is challenging to construct prompt variants in a principled way. Prompt variants can differ along many dimensions, such as phrasing, cognitive load, or length, and naïve changes may unintentionally alter task semantics. Designing prompt variants that isolate specific prompt properties while preserving comparable task intent is es-

* Equal contribution.
† Corresponding author.

sential for attributing observed energy differences to prompt design. Finally, deriving generalizable insights requires moving beyond isolated examples to identify consistent patterns that hold across prompt variants, tasks, models, and hardware devices.

To address these challenges, we design and conduct a comprehensive empirical study that systematically examines the impact of prompt variants on the energy consumption of on-device LLM inference. Our contributions are summarized as follows:

- **New dataset construction for cognitive load.** We construct a new prompt dataset that varies cognitive demand while preserving task intent. We then develop LLM-based scoring and embedding similarity to ensure semantic consistency and limit semantic variation. This dataset enables a controlled examination of energy variation associated with cognitive load demand rather than semantic differences.

- **Empirical study of prompt property effects.** We present the first large-scale empirical study of how two prompt properties, *Phrasing Pattern* and *Cognitive Load*, affect energy consumption in on-device LLM inference, spanning an evaluation space of prompt variants $\times$ datasets $\times$ models $\times$ devices. This enables *the first study of how linguistic form and reasoning demand influence the computational behavior of on-device LLM inference.*

- **Energy behavior analysis and empirical findings.** We analyze energy behavior using per-token energy and token usage, which indicate the cost per token and the number of tokens processed or generated, respectively. We find that cognitive load mainly affects per-token energy variation, while phrasing patterns primarily affect token usage. We further show that energy-quality trade-offs are model-dependent, offering guidance for model-aware, energy-efficient prompt design.

2 Prompt Properties and Datasets

This section first describes the two prompt properties motivated by prior literature, and then introduces the datasets used to study them.

2.1 Prompt Properties

To analyze how prompt variations affect energy behavior, we focus on two representative prompt properties: *Phrasing Pattern* and *Cognitive Load*. Phrasing pattern characterizes linguistic and structural variation under semantic equivalence, while cognitive load introduces the reasoning structure and cognitive demand. The two properties capture complementary aspects of prompt variation, covering both expression form and reasoning demand, as shown in Figure 1. The detailed definitions of the phrasing pattern and cognitive load sub-properties are provided in Appendix A.

Phrasing pattern. Prior studies have shown that semantically equivalent paraphrases can induce substantial variability in model behavior, raising concerns about robustness and evaluation reliability (Ismithdeen et al., 2025; Mizrahi et al., 2024). Motivated by this sensitivity, we study phrasing pattern as a prompt property that characterizes variations in tone, linguistic style, and presentation structure while maintaining semantic intent. Drawing on the taxonomy of phrasing styles introduced by Sotic and Kamps (2025) for the CLEF 2025 ELOQUENT Lab (Karlgren et al., 2025), we consider seven sub-properties: *Aggressive Tone*, *Conversational Tone*, *Chain-of-Thought (CoT)*, *Formatting Differences*, *Persona-Based Prompts*, *Polite Tone*, and *Technical/Jargon-Heavy Prompts*.

Cognitive load. Cognitive Load Theory categorizes cognitive load into intrinsic load, extraneous load, and germane load, emphasizing that problem-solving requires meticulous management of cognitive load (Sweller and Chandler, 1991). Recent work conceptualizes cognitive load as a prompt-level property that systematically influences model reasoning behavior (Long et al., 2025). In this paper, we examine cognitive load through three sub-properties: *Intrinsic Load*, *Extraneous Load*, and *Germane Load*. Each sub-property is represented by explicit prompt cues that vary the reasoning demand while preserving the original task intent.

2.2 Prompt Datasets

For phrasing pattern, we select variants from an existing robustness evaluation dataset; for cognitive load, we construct a new dataset.

Public benchmark dataset for phrasing pattern. To study phrasing pattern, we use the dataset introduced by Sotic and Kamps (2025), which offers controlled stylistic variations of semantically equivalent prompts for robustness evaluation. We retain prompts covering the seven selected sub-properties.

New dataset construction for cognitive load. As no existing public benchmark is designed to capture cognitive load across its three sub-properties

Figure 1: Prompt examples for two prompt properties. (a) Phrasing pattern includes a base prompt and seven sub-properties that capture surface-level linguistic and structural variations under the same semantic intent. (b) Cognitive load includes a base prompt and three sub-properties that vary the information density and reasoning demand of the prompt.

(intrinsic, extraneous, and germane load), we construct a new dataset for cognitive load property using a manually designed template that guides the LLM in generating prompt variants (see Appendix B). For each sub-property, we create dedicated prompt variants while minimizing confounding cues from the other sub-properties. We sample base prompts as semantic anchors from three public datasets that represent distinct reasoning types: SVAMP (arithmetic word problems) (Patel et al., 2021), BoolQ (binary yes/no question answering) (Clark et al., 2019), and AI2-ARC (multiple-choice science questions) (Clark et al., 2018).

3 Empirical Study Pipeline

This section describes the empirical study pipeline we designed. As shown in Figure 2, the workflow consists of three stages: prompt generation, prompt validation, and on-device energy profiling.

3.1 Prompt Variation Generation

For cognitive load property, we generate three variants for each base prompt, corresponding to intrinsic, extraneous, and germane load. We employ Gemini-2.5-Pro API (Comanici et al., 2025) as the generator. Given a base prompt sampled from public datasets, the generator produces candidate prompt variations that target a specific prompt sub-property while preserving semantic meaning. The generation process follows three rules: (i) *Single-property isolation*, where each variation instantiates only one prompt sub-property to avoid property overlap; (ii) *Semantic preservation*, where the original task intent and expected answer remain unchanged; and (iii) *Sub-property consistency*, where each individual variation strictly follows the precise definition of its intended target sub-property.

3.2 Prompt Variation Validation

To ensure that the generated prompt variants are valid and aligned with the intended cognitive load sub-property, all candidates are passed through a rigorous validation process: (i) *Property purity*: confirming that each candidate strictly reflects its target sub-property while avoiding overlap with other sub-properties. (ii) *Semantic similarity*: verifying that each candidate remains semantically aligned with the base prompt, with alignment quantified using embedding-based cosine similarity.

Prompt variation scoring. For prompt variation quality control, we adopt the rubric-based scoring system from (Long et al., 2025), which provides not only the definitions but also the standardized evaluation template for cognitive load properties (see Appendix C). Specifically, we invert the standard scoring direction for extraneous load, so that a higher score indicates a stronger presence of irrelevant and redundant information and better alignment with the intended extraneous load sub-property.

Following the evaluation template, each variation receives a score vector $\mathbf{s} = (s_i, s_e, s_g)$ over the three cognitive load sub-properties. To isolate the target property, we retain only variations whose target sub-property score is at least 8 while all non-target scores are at most 3 on the 1-10 scale.

Semantic consistency filtering. We further filter each prompt variation group to prevent semantic drift from the original task. Specifically, for each base prompt embedding $\mathbf{e}_0$ and variation embedding $\mathbf{e}_i$, we compute cosine similarity. Given the structural rewriting required by cognitive load instantiation, we adopt a threshold τ and retain a

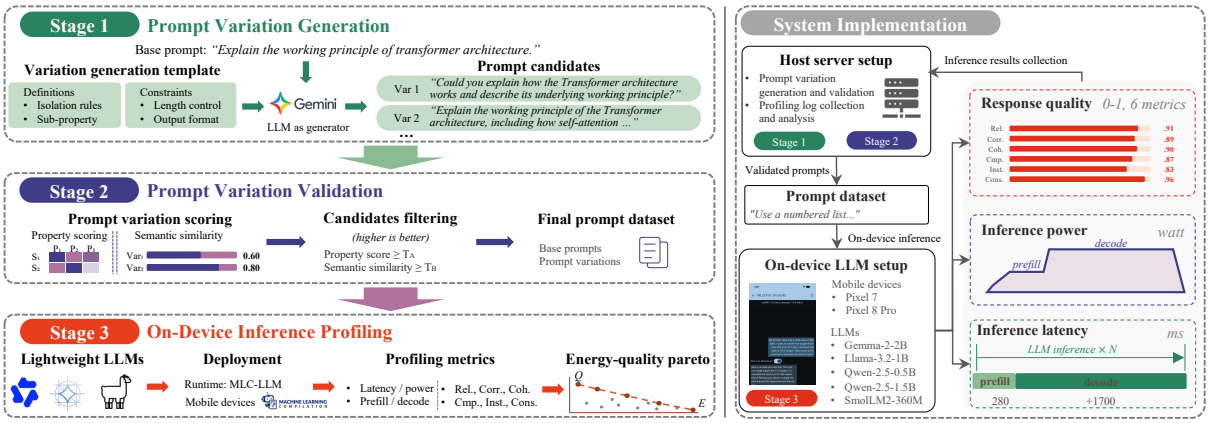


Figure 2: Empirical study pipeline. Prompt variants are generated from base prompts using property definitions and generation constraints, then validated through scoring and semantic similarity filtering. Validated prompts are deployed on lightweight LLMs (Gemma-2-2B (Team et al., 2024), Llama-3.2-1B (Grattafiori et al., 2024), Qwen-2.5-0.5B/1.5B (Team, 2024), and SmolLM2-360M (Allal et al., 2025)) via MLC-LLM to collect phase-level power, latency, and response quality measurements.

prompt variation group only if all its variations remain above this threshold:

$$\min_i \text{CosSim}(\mathbf{e}_0, \mathbf{e}_i) \geq \tau, \tag{1}$$

where we use a relaxed threshold $\tau = 0.6$ to accommodate added cognitive load information.

3.3 On-Device LLM Inference Profiling

We run the validated prompts on mobile devices using instruction-tuned, quantized models deployed locally through the MLC framework. All prompts are evaluated using a fixed inference configuration.

On-device power and latency profiling framework. Building on MLC-LLM (MLC-LLM Team, 2023) and LM-Meter (Wang et al., 2025), we develop a profiling framework for measuring power and latency during on-device LLM inference. The framework instruments the inference runtime to record the start (t_s) and end (t_e) timestamps of the prefill and decode phases, which define phase-level latency and the corresponding energy integration windows. We divide on-device inference into two phases. The prefill window spans from request arrival to first-token sampling, covering tokenization, queueing, embedding, the prefill forward pass, and first-token sampling; the decode window covers all subsequent token generation. This extended window is used instead of the isolated prefill forward pass to capture all energy consumed before the first output and cleanly separate the two phases.

Concurrently, we sample device-level current $I(t)$ and voltage $V(t)$ through Android Debug Bridge (ADB) to construct power traces. Hardware-specific interfaces and device settings are summarized in Table 3. Instantaneous power is computed as $P(t) = I(t) \cdot V(t)$. After aligning the inference traces with the power measurements, we estimate the phase-level energy E_{phase} by integrating power over the corresponding phase window $[t_s, t_e]$, approximated using discrete samples:

$$E_{\text{phase}} = \int_{t_s}^{t_e} P(t)\, dt \approx \sum_{i \in \mathcal{I}_{\text{phase}}} P_i \Delta t, \tag{2}$$

where $\mathcal{I}_{\text{phase}}$ denotes the set of power-sample indices whose timestamps fall within the phase window, and Δt is the effective sampling interval.

Response logging. For each run, we record the generated response and run-level metadata, including device, model, prompt ID, token counts, and timestamps. These records support traceable analysis across prompts, models, devices, and energy measurements. Detailed latency, energy, token, and quality statistics are provided in Appendix F.

3.4 Response Quality Evaluation

We evaluate response quality with task-specific metrics: accuracy for ground-truth tasks and reference-free assessment for open-ended tasks using DeepEval (Ip and Vongthongsri, 2026), with a randomly sampled subset manually verified.

Cognitive load tasks. For cognitive load datasets with ground-truth answers, we measure response quality with Exact Match (EM) accuracy. Since prompt variations preserve semantic intent, models should produce the same correct answer across variants. EM helps ensure that energy differences are not attributable to degraded correctness.

Phrasing pattern tasks. For phrasing pattern tasks, strict accuracy is unsuitable because the

prompts are open-ended and can elicit multiple valid responses. In addition, the reference answers are LLM-generated rather than objective ground truth. We score each open-ended response along six dimensions: *Relevance*, *Correctness*, *Coherence*, *Completeness*, *Instruction Adherence*, and *Internal Consistency*. Each dimension is scored on a 0-1 rubric, with the dimension definitions and evaluation rubric provided in Appendix E.

4 Results and Analysis

Prefill processes the input prompt, while decode generates the response. We therefore report phase-level energy alongside total energy to analyze how prompt variations affect each phase. We compare per-token energy with token usage, indicating the cost of processing each token and the number of tokens processed or generated:

$$E_{\text{phase}} = e_{\text{phase}} \cdot T_{\text{phase}}, \tag{3}$$

where E_{phase} denotes the energy consumption of a given inference phase, e_{phase} denotes the per-token energy, and T_{phase} denotes the number of tokens processed or generated in that phase.

To understand how prompt properties affect the *energy cost per token* and the *number of tokens processed and generated*, we organize this section around the following three research questions:

- ***RQ1: To what extent is per-token energy shaped by prompt properties relative to device and model factors?***
- ***RQ2: How do prompt properties affect token usage across the prefill and decode phases?***
- ***RQ3: Can prompt properties shift the energy-quality frontier across model architectures?***

4.1 RQ1: Per-token Energy Analysis

To answer RQ1, we first examine whether prompt properties change the energy cost of processing each token, as shown in Figure 3. Due to space constraints, we provide the remaining per-token energy analyses across prompt properties, devices, and models in Appendix G.

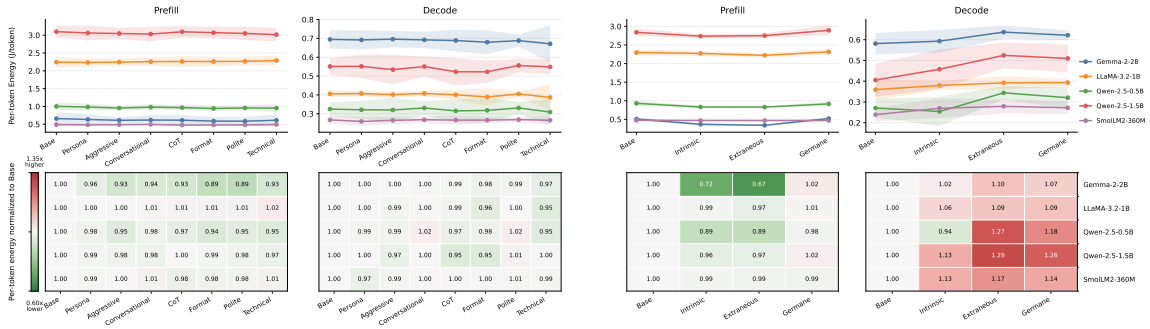
Observation 1: Prefill costs more energy per token than decode on most models. On LLaMA-3.2-1B and both Qwen models, per-token prefill energy is 3-6× that of decode, while Gemma-2-2B shows the opposite. This inversion relative to server-scale inference is known for on-device runtimes, reflecting limited compute, memory bandwidth, and the absence of parallelized prefill support (Wang et al., 2025). Part of the prefill window (e.g., tokenization, queueing, embedding) does not scale with prompt length; thus on-device prefill exhibits a pronounced amortization effect: longer prompts yield lower per-token energy. The effect is largest on Gemma-2-2B, which has the lowest marginal per-token prefill energy: its normalized prefill values fall to 0.67× under extraneous load.

Observation 2: Absolute per-token energy is primarily model-dependent. Across both prompt properties, absolute per-token energy is primarily determined by the executed model rather than the prompt sub-property. Across both phrasing pattern and cognitive load, the separation between model curves is much larger than the variation across prompt sub-properties within the same model. This indicates that model architecture and scale remain the dominant factors in per-token execution cost.

Observation 3: Phrasing pattern exhibits limited normalized per-token energy variation. For phrasing pattern variants, the normalized heatmaps remain close to the base prompt across most models and sub-properties. Most values fluctuate around 1.0 in both prefill and decode, indicating that surface-level changes in tone, persona, formatting, politeness, or technical wording do not substantially alter the cost of processing each token. While some models show small deviations for specific sub-properties, these changes are modest compared with the absolute differences across models.

Observation 4: Cognitive load substantially affects per-token decode energy. Compared with phrasing pattern, cognitive load variants produce more visible changes in per-token energy in decode phases. During decoding, extraneous and germane load increase per-token energy for Qwen-2.5-0.5B, Qwen-2.5-1.5B, and SmolLM2-360M, with extraneous load reaching 1.29× on Qwen-2.5-1.5B. This indicates that cognitive load affects per-token execution cost in a phase- and model-dependent manner.

Finding. Per-token energy is primarily model-driven, but cognitive load introduces variation in per-token inference cost that surface phrasing largely does not. This suggests that energy analysis should distinguish reasoning demand from linguistic form, rather than relying only on aggregate energy or token-count explanations.



(a) Per-token energy across phrasing pattern for Pixel 8 Pro. **(b)** Per-token energy across cognitive load for BoolQ.

Figure 3: Per-token energy analysis across prompt properties. (a) Phrasing pattern results on Pixel 8 Pro across prefill and decode phases, and (b) cognitive load results on BoolQ across prefill and decode phases. The first row shows absolute per-token energy for the prefill and decode phases, with shaded bands denoting ± 1 standard deviation across prompts within each sub-property; the second row reports each sub-property's per-token energy normalized by the corresponding base prompt.

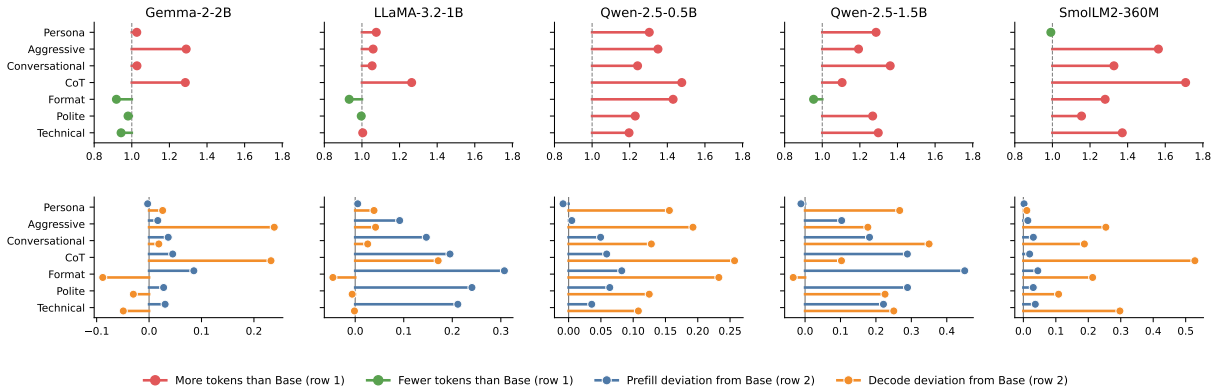


Figure 4: Token length and fixed-baseline phase-level energy burden for phrasing pattern on Pixel 8 Pro. Top: total token ratio relative to Base, computed as each sub-property token count divided by that of the corresponding Base prompt (>1: more tokens; <1: fewer tokens). Bottom: fixed-baseline prefill/decode burden relative to Base, normalized by base prompt/completion tokens, respectively and after subtracting the corresponding Base burden; positive/negative values indicate higher/lower burden.

4.2 RQ2: Token Footprint Analysis

As shown in Figure 4, we analyze how phrasing patterns affect token usage and phase-level burden. Due to space constraints, we provide the remaining token usage and phase burden results across properties, devices, and models in Appendix H.

Observation 1: Phrasing patterns substantially reshape token usage. The top row shows that CoT and aggressive prompts consistently increase token usage relative to Base, while format, polite, and technical prompts stay closer to Base and sometimes reduce tokens. Therefore, surface-level phrasing alone can alter inference computation, even when per-token energy shifts are small.

Observation 2: Token usage effects are model dependent. The same phrasing pattern can lead to different token expansion across models. For example, CoT produces much larger total token ratios on SmolLM2-360M and Qwen-2.5-0.5B than on Gemma-2-2B or LLaMA-3.2-1B. This suggests that phrasing patterns affect not only the input prompt but also model generation behavior.

Observation 3: Phase burden differs across phrasing patterns. The bottom row shows that token-related energy burden is not distributed uniformly across inference phases. CoT and aggressive prompts often increase decode burden, indicating stronger effects on response generation. In contrast, format prompts can increase prefill burden while reducing or weakly increasing decode burden, suggesting that some phrasing patterns shift cost toward input processing rather than generation.

> ***Finding.*** Phrasing patterns shape energy behavior primarily by affecting token usage during inference, rather than the per-token processing cost. This suggests that energy consumption does not simply scale with total token usage; energy analysis should also consider how phrasing patterns alter model-specific generation behavior.

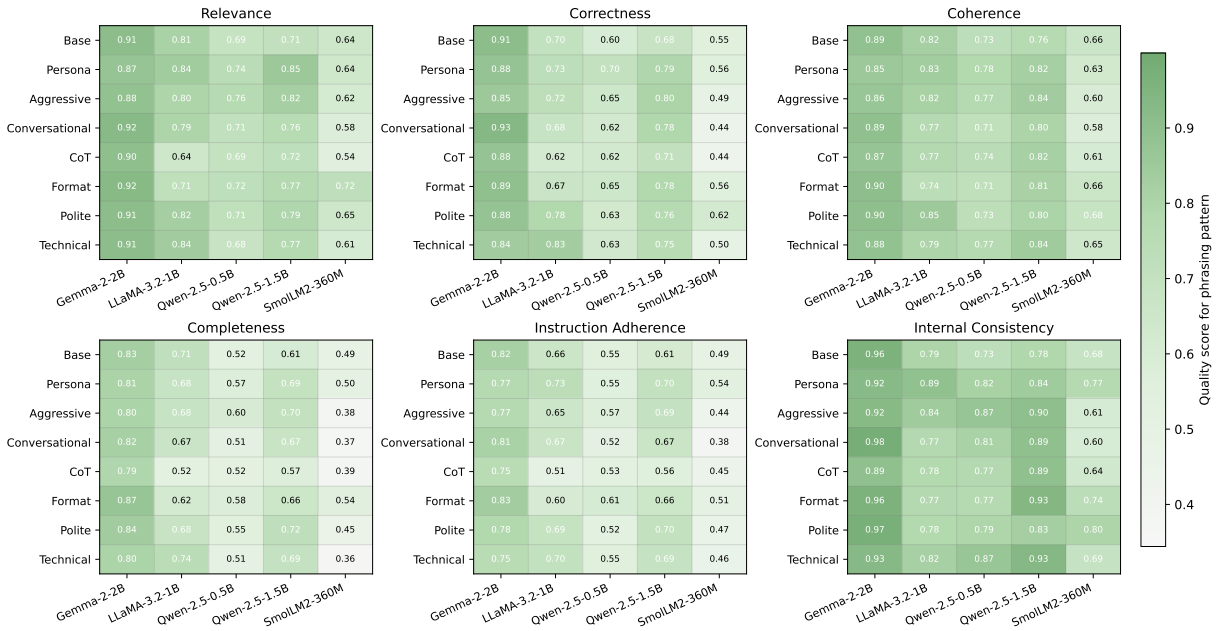


Figure 5: Response quality across phrasing pattern sub-properties and models. Heatmaps show quality varies across metrics, phrasing patterns, and models. Each heatmap shows one metric; rows are sub-properties and columns are models. Cells report mean scores, with darker green indicating higher quality.

4.3 RQ3: Energy-quality Trade-off Analysis

For RQ3, we analyze the energy-quality trade-off by first examining response quality across phrasing pattern sub-properties and metrics, as shown in Figure 5. We then evaluate how each phrasing pattern shifts the energy-quality frontier under fixed model weights, as shown in Figure 6. Due to space constraints, we provide the remaining energy-quality results across models and metrics in Appendix I.

Observation 1: Response quality varies across models and metrics. Figure 5 shows that response quality is strongly model dependent. Larger or more capable models generally achieve higher scores across relevance, correctness, coherence, completeness, instruction adherence, and internal consistency, whereas smaller models show lower and less stable quality. This indicates that phrasing patterns interact with model capacity rather than producing uniform quality shifts.

Observation 2: Phrasing patterns affect quality dimensions unevenly. Phrasing patterns show relatively stable performance on relevance, coherence, and internal consistency, but larger variation on correctness, completeness, and instruction adherence. This indicates that stylistic changes can preserve the apparent fluency and alignment of a response, while improvements in task correctness and completeness remain less consistent.

Observation 3: Phrasing patterns reshape the energy-quality frontier. Figure 6 shows that phrasing patterns can substantially alter each model's energy-quality profile. Red markers denote sub-properties achieving both lower energy and higher quality than Base. For LLaMA-3.2-1B, Qwen-2.5-1.5B, and SmolLM2-360M, these lie on the Pareto frontier, showing that prompt selection improves both sides of the trade-off under fixed weights and decoding settings. For Gemma-2-2B, the format and conversational prompts both improve on Base along both axes; the format prompt attains the highest quality and is the better operating point, though the technical prompt reaches slightly lower energy.

Observation 4: Energy-quality trade-offs are model-dependent. The energy-quality effect of a phrasing pattern does not transfer uniformly across models. A sub-property that improves the energy-quality balance for one model may lead to a different trade-off profile for another. This model dependence suggests that prompt phrasing style selection should be model-aware, rather than assumed to generalize uniformly across models.

Finding. Phrasing patterns can substantially shift the energy-quality frontier, enabling lower energy consumption and higher response quality, while efficiency gains are model-dependent. This suggests that energy-efficient prompting is not governed by a uniform optimal phrasing style, but requires highly specific, model-aware phrasing optimization strategies.

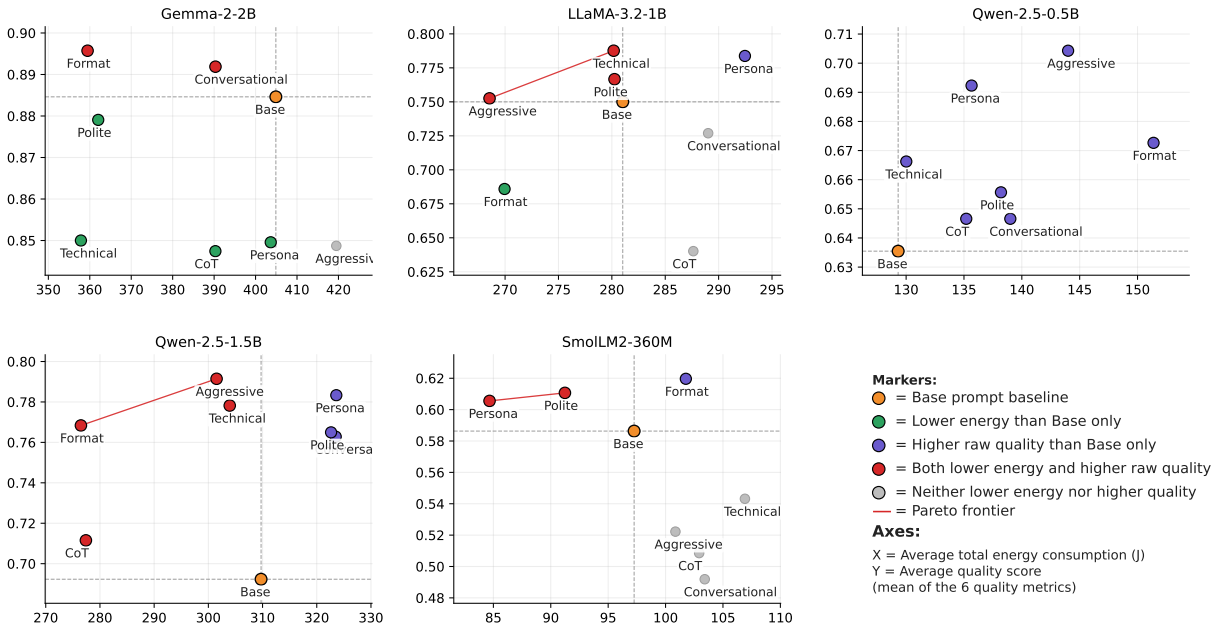


Figure 6: Energy-quality trade-off and pareto frontier for phrasing pattern. Each point represents a phrasing pattern sub-property evaluated under the same model weights and decoding configuration. Prompt sub-properties shift the energy-quality operating point in a model-dependent manner, revealing that stylistic variation can change both energy cost and response quality.

5 Related Work

Prompt categorization and linguistic analysis. Prior work has extensively studied prompt properties from linguistic and cognitive perspectives, including phrasing patterns (Brown et al., 2020; Reynolds and McDonell, 2021; Zhao et al., 2021), syntactic and discourse structures (Schick and Schütze, 2021; Jiang et al., 2020; Min et al., 2022; Liu et al., 2022), reasoning requirements (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023), and cognitive complexity (Fu et al., 2023). Surveys and taxonomies further systematize these properties along functional and cognitive axes (Liu et al., 2023; Vatsal and Dubey, 2024; Liu et al., 2026), while recent frameworks model prompt structure to support interpretability and prompt engineering (Jeoung et al., 2026). However, this line of work focuses on output quality and robustness, with little attention to how prompt properties affect system-level behavior such as on-device energy consumption during inference.

Prompt datasets and benchmarks. Large prompt and instruction datasets, such as Natural Instructions (Mishra et al., 2022), Super-NaturalInstructions (Wang et al., 2022), and the Flan Collection (Longpre et al., 2023), have been widely used for instruction tuning and evaluation. Tools like PromptSource further standardize prompt templates across tasks (Bach et al., 2022). While these resources enable broad coverage and performance-driven evaluation, prompts often vary simultaneously along multiple dimensions (e.g., semantics, length, and reasoning), making them unsuitable for isolating the impact of specific prompt properties on model or system behavior.

On-device LLM systems and energy profiling. Recent studies benchmark latency, memory, and throughput of LLM inference on mobile and edge devices (Li et al., 2024; Murthy et al., 2024) and propose hardware-aware inference optimizations (Yin et al., 2025; Xue et al., 2024). Energy profiling efforts typically analyze model architectures, runtimes, or prompt-level workloads on server-class hardware (Husom et al., 2024; Caravaca et al., 2025). Consequently, existing work provides limited insight into how prompt properties themselves influence on-device energy consumption under controlled conditions.

In contrast, we study prompt properties through direct measurements of on-device energy consumption, linking linguistic prompt analysis with system-level energy profiling. Our analysis is measurement-based and characterizes system-level energy behavior rather than isolating low-level execution mechanisms.

6 Conclusion

This paper studies how prompt properties shape the energy behavior and response quality of on-device LLM inference. Through a broad empirical study

covering models, devices, and prompt variants, we show that prompt-level variants can substantially affect mobile inference efficiency. In particular, cognitively demanding prompts tend to increase decode energy cost to process each token, while surface-level phrasing patterns can change token usage, phase-level energy burden, and response quality without altering the underlying task. Our analysis further shows that these effects are not uniform: different prompts can shift cost between prefill and decode phases, and the resulting energy-quality trade-offs vary across models and metrics.

These findings suggest that prompt design is not only a usability or quality concern, but also a practical optimization lever for energy-aware LLM deployment. Rather than treating energy consumption as a fixed property of a model, future systems should account for how prompts shape generation behavior and inference cost, especially in mobile and resource-constrained settings. Future work will extend this analysis to a broader prompt taxonomy and more diverse hardware platforms.

Limitations

Limited prompt property coverage. Our study primarily focuses on two prompt properties, cognitive load and phrasing pattern. However, the design space of prompts is vast, encompassing numerous other semantic and structural properties (e.g., emotional valence, stylistic constraints, or adversarial framings) that remain unexplored in our energy profiling. Future research incorporating a broader taxonomy of prompt properties would provide a more complete view of how diverse linguistic variables influence on-device energy consumption.

Limited control over prompt length. Extraneous load is defined by redundant or irrelevant information and intrinsic load by explicit step-by-step guidance; a prompt cannot carry these without becoming longer. Consequently, prompt length is not independently controlled, making it difficult to completely decouple the effects of cognitive load framing from length-related effects. Future work should construct length-matched prompt variants to better isolate these factors.

Evaluation methodology limitations. We use Gemini-2.5-Pro as the primary evaluator for prompt property scoring and response-quality assessment. Although this helps maintain internal consistency, relying on a single LLM-as-a-judge may introduce model-specific biases in scoring. In addition, embedding-based similarity and DeepEval metrics provide automatic proxies for response quality, but may not fully capture the nuanced utility perceived by human users. To strengthen evaluation reliability, we manually verify randomly sampled subsets of both prompt variation and response quality evaluations. Automated judgments that do not pass human review are replaced with human judgments.

Limited hardware coverage and profiling granularity. Our energy measurements are conducted on a limited set of mobile SoC architectures. Although these devices are representative of on-device inference settings, power profiles may differ across NPUs, GPUs, CPUs, memory systems, and runtime backends. In addition, our analysis focuses on end-to-end and phase-level energy behavior, rather than lower-level mechanisms such as KV-cache behavior, hardware counters, or operator-level traces. Future work should evaluate a wider range of mobile hardware and incorporate finer-grained profiling to explain hardware-specific energy variation.

Acknowledgments

We thank the reviewers and the area chairs for their insightful comments. This research was supported by funds from Toyota Motor North America. In addition, this research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-23-2-0224. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, and 1 others. 2025. SmolLM2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, and 8

others. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Francisco Caravaca, Ángel Cuevas, and Rubén Cuevas. 2025. From prompts to power: Measuring the energy footprint of LLM inference. *arXiv preprint arXiv:2511.05597*.

Yanjiao Chen, Baolin Zheng, Zihan Zhang, Qian Wang, Chao Shen, and Qian Zhang. 2020. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. *ACM Comput. Surv.*, 53(4).

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. 2025. Security and privacy challenges of large language models: A survey. *ACM Comput. Surv.*, 57(6).

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Erik Johannes Husom, Arda Goknil, Lwin Khin Shar, and Sagar Sen. 2024. The price of prompting: Profiling energy use in large language models inference. *arXiv preprint arXiv:2407.16893*.

Jeffrey Ip and Kritin Vongthongsri. 2026. deepeval. `https://github.com/confident-ai/deepeval`. Accessed: 2026-08-24.

Mohamed Insaf Ismithdeen, Muhammad Uzair Khattak, and Salman Khan. 2025. Promptception: How sensitive are large multimodal models to prompts? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23950–23985, Suzhou, China. Association for Computational Linguistics.

Sullam Jeoung, Yueyan Chen, Yi Zhang, Shuai Wang, Haibo Ding, and Lin Lee Cheong. 2026. PromptPrism: A linguistically-inspired taxonomy for prompts. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1168–1192, Rabat, Morocco. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jussi Karlgren, Marie Isabel Engels, Maria Barrett, Rohit Raj Gunti, Mohanna Hoveyda, Bruno Nadalic Sotic, Jaap Kamps, Mika Koistinen, and Elaine Zosa. 2025. Overview and joint report of the robustness and consistency task at the ELOQUENT 2025 lab for evaluating generative language model quality. *Working Notes of CLEF*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Xiang Li, Zhenyan Lu, Dongqi Cai, Xiao Ma, and Mengwei Xu. 2024. Large language models on mobile devices: Measurements, analysis, and insights. In *Proceedings of the Workshop on Edge and Mobile Foundation Models*, EdgeFM '24, page 1–6, New York, NY, USA. Association for Computing Machinery.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yao-Yang Liu, Zhen Zheng, Feng Zhang, Jin-Cheng Feng, Yi-Yang Fu, Ji-Dong Zhai, Bing-Sheng He, Xiao Zhang, and Xiao-Yong Du. 2026. A comprehensive taxonomy of prompt engineering techniques for large language models. *Frontiers of Computer Science*, 20(3):2003601.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. MobileLLM: optimizing sub-billion parameter language models for on-device use cases. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Do Xuan Long, Duy Dinh, Ngoc-Hai Nguyen, Kenji Kawaguchi, Nancy F. Chen, Shafiq Joty, and Min-Yen Kan. 2025. What makes a good natural language prompt? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5835–5873, Vienna, Austria. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

MLC-LLM Team. 2023. MLC-LLM: A universal deployment solution for large language models. `https://github.com/mlc-ai/mlc-llm`. Accessed: 2026-08-24.

Rithesh Murthy, Liangwei Yang, Juntao Tan, Tulika Manoj Awalgaonkar, Yilun Zhou, Shelby Heinecke, Sachin Desai, Jason Wu, Ran Xu, Sarah Tan, and 1 others. 2024. Mobileaibench: Benchmarking LLMs and LMMs for on-device use cases. *arXiv preprint arXiv:2406.10290*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM*, 63(12):54–63.

Bruno N Sotic and Jaap Kamps. 2025. University of amsterdam at the CLEF 2025 eloquent track. In *CLEF (Working Notes)*, pages 1435–1442.

John Sweller and Paul Chandler. 1991. Evidence for cognitive load theory. *Cognition and Instruction*, 8(4):351–362.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different NLP tasks. *arXiv preprint arXiv:2407.12994*.

Haoxin Wang, Xiaolong Tu, Hongyu Ke, Huirong Chai, Dawei Chen, and Kyungtae Han. 2025. lmmeter: Unveiling runtime inference latency for on-device language models. In *Proceedings of the Tenth ACM/IEEE Symposium on Edge Computing*, SEC'25, New York, NY, USA. Association for Computing Machinery.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, and 16 others. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.

Zhenliang Xue, Yixin Song, Zeyu Mi, Xinrui Zheng, Yubin Xia, and Haibo Chen. 2024. Powerinfer-2: Fast large language model inference on a smartphone. *arXiv preprint arXiv:2406.06282*.

Wangsong Yin, Rongjie Yi, Daliang Xu, Gang Huang, Mengwei Xu, and Xuanzhe Liu. 2025. Elastic on-device LLM service. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*, ACM MOBICOM '25, page 984–999, New York, NY, USA. Association for Computing Machinery.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. 2019. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. 2024. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, page 57–68, New York, NY, USA. Association for Computing Machinery.

A Prompt Property Definitions

Sub-property	Definition
Aggressive/Authoritative Tone	*Prompts characterized by commanding or forceful language, often lacking politeness or courtesy.*
Conversational Tone	*Prompts that mimic natural human dialogue, often informal and friendly in nature.*
Chain-of-Thought (CoT)	*A prompting technique where the model is guided to generate intermediate reasoning steps before arriving at a final answer.*
Formatting Differences	*Variations in the structural presentation of prompts, such as the use of lists, bullet points, or different punctuation.*
Persona-Based	*Prompts that assign a specific role or identity to the model, such as "You are a helpful assistant."*
Polite Tone	*Prompts that employ courteous language, including phrases like "please" and "thank you."*
Technical/Jargon-Heavy	*Prompts that utilize domain-specific terminology or complex language.*

Table 1: Definitions of phrasing pattern sub-properties, quoted from Sotic and Kamps (2025).

Sub-property	Definition
Intrinsic Load	*Prompts in this property explicitly guide models to break complex tasks into actionable steps aligned with LLM skills.*
Extraneous Load	*Prompts in this property contain unnecessary complexity with intricate language, redundant or irrelevant information, resulting in increased unnecessary load.*
Germane Load	*Prompts in this property explicitly engage models with their prior knowledge or deep working memory to integrate it with existing and new knowledge for problem-solving.*

Table 2: Definitions of cognitive load sub-properties, quoted from Long et al. (2025).

B Cognitive Load Prompt Generation Template

```
PROPERTY = Cognitive Load

DESCRIPTION = Cognitive Load Theory categorizes cognitive load into three types: Intrinsic Load,
Extraneous Load, and Germane Load. This property defines cognitive load as a prompt attribute with
three corresponding sub-properties.

TASK INSTRUCTION = You are tasked with generating prompt variations that incorporate cognitive load
information based on the given prompt.

SUB-PROPERTY DEFINITIONS:
>Intrinsic Load
Prompts explicitly guide the model to break complex tasks into actionable steps aligned with LLM
skills.
>Extraneous Load
Prompts contain unnecessary complexity with intricate language, redundant or irrelevant information,
resulting in increased unnecessary cognitive load.
>Germane Load
Prompts explicitly engage the model with its prior knowledge or deep working memory to integrate
existing and new knowledge for problem-solving.

INPUT PROMPT FORMAT:
<Begin of the prompt> {input_prompt} <End of the prompt>

GENERATION REQUIREMENTS:
>Generate exactly one prompt variation for each cognitive load sub-property: Intrinsic Load,
Extraneous Load, and Germane Load.
>Preserve semantic intent: Each variation must preserve the original semantic intent (the question
and required answer) while differing in the expression and organization of cognitive load information.
You may drop story details, background descriptions, or numbers that are not necessary to determine
the answer.
>Length consistency: The three generated variations should be roughly comparable in length; however,
cognitive-load constraints take priority over strict length matching. Do not artificially extend
the Germane prompt to match the Extraneous prompt length.
```

```
SUB-PROPERTY ISOLATION RULES:
1. Intrinsic Load variation:
- MUST include only step-breaking or task-structuring guidance.
- Use explicit step markers such as “Step 1 / Step 2 / Step 3” or “First / Next / Then / Finally”.
- Do NOT add narrative details beyond the minimum required for solving the task.
2. Extraneous Load variation:
- MUST include only unnecessary complexity, redundancy, or irrelevant information.
- MUST NOT introduce step-by-step solution structure or meta-cognitive guidance.
3. Germane Load variation:
- MUST activate prior knowledge and conceptual reasoning.
- MUST NOT include explicit procedural or step-by-step guidance.

CRITICAL CONSTRAINT CHECKLIST:
>For Extraneous Load Variation (Goal: High Extraneous, Low Intrinsic, Low Germane):
- Noise Injection: You MUST insert clearly irrelevant numerical data or distinct entities (e.g., unrelated times, prices, objects, names, weather).
- Distractors MUST NOT change the correct answer.
- Structure: Use complex and convoluted sentences to obscure key information.
- No Guidance: Do NOT use phrases such as “First”, “Next”, “Think step-by-step”, or “Use your knowledge of”.
>For Germane Load Variation (Goal: Low Intrinsic, Low Extraneous, High Germane):
- ONE-SENTENCE / ONE-MAIN-VERB RULE: The prompt MUST be exactly one sentence with one main directive verb.
- Explicit prior-knowledge cue is REQUIRED (e.g., “Use your knowledge of the simple part-whole idea...”).
- Single simple concept ONLY: part-whole idea, part-part-whole idea, or difference idea.
- SEMANTIC SIMPLICITY: BANNED terms include “Function”, “Dependency”, “Integration”, “Framework”, “Schema”, “Mental model”.
- NUMBERS AND ROLES: Mention only quantities required to compute the answer.
- Target structure (soft template): “Use your knowledge of the simple [concept] idea to find [goal] from the given numbers.”

OUTPUT FORMAT (STRICT):
<Begin of response>
{ “Base prompt”: “{input_prompt}”,
“Intrinsic load prompt”: “”,
“Extraneous load prompt”: “”,
“Germane load prompt”: “” }
<End of response>

Any deviation from the required format or violation of sub-property isolation is strictly prohibited.
No additional explanation or commentary is allowed outside the JSON object.
```

C Prompt Variation Evaluator Template

```
COG_FORMAT = “{‘Intrinsic load’: 1-10, ‘Extraneous load’: 1-10, ‘Germane load’: 1-10}”

COG_JUDGING_PROMPT = f””” You are a highly experienced judge tasked with evaluating a prompt on criteria.
The prompt given to you is provided below: <begin of the prompt> {input_prompt} <end of the prompt>
Your task is to evaluate the above prompt on the following criteria on a scale of 1-10:
- Intrinsic load: This evaluates the prompts in explicitly guiding models to break complex tasks into actionable steps aligned with LM skills.
- Extraneous load: The extent to which prompts exclude irrelevant materials to reduce unnecessary load.
- Germane load: The degree to which prompts explicitly engage models with their prior knowledge or deep working memory (e.g., “ask itself”) to integrate it with existing and new knowledge for problem-solving.
The scoring system is provided below:
> Intrinsic load:
- 1-2 (Poor): The prompt provides little to no guidance on breaking down the task. It is overly vague, abstract, or assumes the model can handle complexity without guidance.
- 3-4 (Below Average): The prompt provides minimal guidance but fails to clearly break the task into actionable steps. The model is left to infer most of the process.
- 5-6 (Average): The prompt partially breaks down the task but lacks clarity or completeness in defining actionable steps. Some guidance is present, but it is inconsistent or incomplete.
- 7-8 (Good): The prompt effectively breaks the task into clear, actionable steps. It aligns well with the model’s skills but may lack some nuance or optimization.
```

```
- 9-10 (Excellent): The prompt perfectly breaks the task into logical, actionable steps. It is
highly aligned with the model's capabilities and ensures clarity and efficiency in execution.
> Extraneous load:
- 1-2 (Poor): The prompt is perfectly concise and excludes all irrelevant materials. It is optimized
to reduce extraneous load to the bare minimum.
- 3-4 (Below Average): The prompt is concise and mostly free of irrelevant information. It minimizes
extraneous load effectively, with only minor distractions.
- 5-6 (Average): The prompt includes some unnecessary details but generally stays focused on the
task. The extraneous load is moderate but not overly detrimental.
- 7-8 (Good): The prompt contains some irrelevant information, but the core task is still somewhat
discernible. The extraneous load is noticeable and distracting.
- 9-10 (Excellent): The prompt includes excessive irrelevant information, making it difficult for
the model to focus on the core task. It is cluttered or overly verbose.
> Germane load:
- 1-2 (Poor): The prompt does not engage the model's prior knowledge or working memory. It provides
no cues or instructions to leverage existing knowledge.
- 3-4 (Below Average): The prompt makes minimal attempts to engage prior knowledge but does so
ineffectively or inconsistently. The model is left to infer connections on its own.
- 5-6 (Average): The prompt partially engages the model's prior knowledge but lacks depth or clarity
in integrating it with new information. The engagement is superficial.
- 7-8 (Good): The prompt effectively engages the model's prior knowledge and encourages integration
with new information. It provides clear cues or instructions for leveraging existing knowledge.
- 9-10 (Excellent): The prompt perfectly engages the model's prior knowledge and deep working memory.
It explicitly guides the model to integrate existing and new knowledge for optimal problem-solving.
Your evaluations must focus on explicit instructions rather than implicit instructions.
For example, if the prompt does not say "Reflect on your prior knowledge" then you should not
assume that the prompt is effective in encouraging germane load.
Begin your evaluation by providing a short explanation for each. Be as objective, thorough, and
constructive as possible.
After providing your explanation, please rate the response on all the criteria on a scale of 1 to
10 by strictly following this format: <begin of explanation> ... <end of explanation> <begin of
ratings> {COG_FORMAT} <end of ratings> """
```

D Experimental Setup Details

Category	Item	Configuration / Value	Notes
Devices	Pixel 8 Pro	CPU: 1.32 GHz / 1.57 GHz / 2.04 GHz GPU: 810 MHz	Android on-device inference
	Pixel 7	CPU: 1.32 GHz / 1.49 GHz / 1.58 GHz GPU: 810 MHz	Same OS configuration
Hardware Control	CPU / GPU Frequency	Fixed (no DVFS)	Governors locked during inference
	Screen Brightness	Minimum brightness	Reduce display power noise
Models	Llama-3.2-1B-Instruct	1B params, q4f16_1, MLC	Instruction-tuned
	Qwen-2.5-0.5B-Instruct	0.5B params, q0f16, MLC	Lightweight model
	Qwen-2.5-1.5B-Instruct	1.5B params, q4f16_1, MLC	Medium-scale model
	Gemma-2-2b-it	2B params, q4f16_0, MLC	Google Gemma family
	SmolLM2-360M-Instruct	360M params, q4f16_1, MLC	Small-scale model
Prompt Setup	Base Prompt Datasets	SVAMP, BoolQ, AI2-ARC	Used as semantic anchors
	Prompt Variations	Property-controlled generation	Cognitive load, phrasing patterns
	Prompt Count	Three candidates per base prompt	See Appendix B
Inference Settings	Number of Runs	Three runs per (device, model, prompt)	Averaged for stability
	Temperature	$T = 0.6$ (fixed)	Same across all experiments
	Top-p	$p = 0.8$ (fixed)	Same across all experiments
	Max Tokens	Fixed across models	Prevent output-length bias
Measurement	Energy Profiling	On-device power measurement	See Section 3.3

Table 3: Experimental setup and controlled configurations for on-device prompt-energy evaluation. We conducted a carefully controlled study covering 2 major prompt properties, 10 sub-properties, 4 datasets, 404 prompts, 5 LLMs, 2 mobile devices (phrasing pattern prompts are profiled on both devices, cognitive load prompts are profiled on Pixel 8 Pro only), and 3 repeated runs per configuration, resulting in 7,620 total inference runs. To isolate the effect of prompt properties, CPU and GPU frequencies are fixed to disable DVFS, and all experiments use identical inference parameters across prompt variants.

E Phrasing Pattern Response Evaluation Metrics and Scoring Rubrics

Metric	Definition
Relevance	Whether the response directly addresses the user's request and remains aligned with the prompt's intent and scope, without extraneous or tangential content.
Correctness	Whether the response avoids factual errors or hallucinated claims unsupported by the prompt or commonly accepted knowledge.
Coherence	Whether the response is logically organized, structurally well formed, and semantically continuous, without abrupt topic shifts, disorganization, or repetitive generation.
Completeness	Whether the response covers all material components of the request, including key sub-questions, required elements, or expected deliverables.
Instruction Adherence	Whether the response follows explicit and implicit prompt constraints, including format, scope, quantity, style, and target audience.
Internal Consistency	Whether the response is free of internal contradictions, self-negating statements, or mutually inconsistent claims.

Table 4: Definitions of the six response-quality metrics used to evaluate linguistic and functional competence in open-ended phrasing tasks.

Table 5: Scoring Rubric for **Relevance** and **Correctness**

Score	Relevance	Correctness
1.00	**Completely Relevant.** Entirely focused on the user's prompt. Every sentence contributes to answering the specific question.	**No Errors.** All assertions and claims are factually correct and verifiable based on general knowledge.
0.75	**Mostly Relevant.** Core answer is relevant, but includes slight divergence, tangential examples, or minor unnecessary elaboration.	**Minor Inaccuracy.** Core answer is correct, but contains a trivial error (e.g., slightly off date) that does not mislead the user.
0.50	**Mixed Relevance.** Addresses the prompt but drifts significantly into unrelated topics for a large portion of the text.	**Mixed Accuracy.** Contains a mix of correct and incorrect statements. A significant claim is factually wrong.
0.25	**Mostly Irrelevant.** Acknowledges the topic but fails to address the specific question; high noise-to-signal ratio.	**Major Inaccuracy.** Primary conclusion is factually incorrect, though some minor supporting details might be right.
0.00	**Completely Irrelevant.** Unrelated to the prompt (e.g., hallucinated question, gibberish).	**Complete Hallucination.** Fundamentally wrong, fabricated, or contradicts well-established facts entirely.

Table 6: Scoring Rubric for **Coherence** and **Completeness**

Score	Coherence	Completeness
1.00	**Perfectly Coherent.** Flows logically with clear structure. Transitions are smooth and natural.	**Fully Complete.** Comprehensively addresses every aspect, including all sub-questions and implied requirements.
0.75	**Mostly Coherent.** Logic is sound, but contains minor awkward transitions or slightly confusing structures.	**Mostly Complete.** Addresses main points but misses a minor detail, example, or nuance.
0.50	**Somewhat Disjointed.** Understandable but requires effort to follow. Text may jump between ideas.	**Partial Completion.** Answers one part well but ignores another significant part (e.g., explains concept but omits examples).
0.25	**Incoherent.** Fragmented text. Sentences do not logically follow one another. Severe repetition occurs.	**Minimal Coverage.** Touches on the topic but fails to provide the substance required for a functional answer.
0.00	**Unreadable.** "Word salad," gibberish, or completely unstructured text.	**Incomplete.** Fails to answer the core request entirely.

Table 7: Scoring Rubric for **Instruction Adherence** and **Internal Consistency**

Score	Instruction Adherence	Internal Consistency
1.00	**Strict Adherence.** Follows every constraint, including negative constraints ("do not...") and stylistic requirements.	**Consistent.** Perfectly consistent. No logical contradictions exist within the text.
0.75	**Minor Deviation.** Follows main instructions but misses a minor formatting constraint or tone is slightly off.	**Mostly Consistent.** Consistent, but may contain slight ambiguity interpretable as a contradiction.
0.50	**Partial Adherence.** Ignores a major constraint (e.g., wrong format) but adheres to others.	**Contradictory.** Contains a clear contradiction on a minor point without context.
0.25	**Major Deviation.** Ignores most specific constraints, adhering only to the general topic.	**Heavily Contradictory.** Contradicts its own main thesis or conclusion.
0.00	**Non-Adherence.** Completely ignores specific instructions (e.g., wrong language, wrong format).	**Illogical.** Riddled with self-negating statements; impossible to determine the stance.

F Detailed Full Experimental Measurements

Note on per-token conventions. Token counts and energy values in the following tables are arithmetic means across questions within each condition. In contrast, the per-token energy reported in Fig. 3 is first computed separately for each question and then averaged across questions, giving equal weight to every question. As a result, the two approaches use different weighting schemes: taking the ratio of the aggregate means in the table implicitly gives greater weight to questions with more tokens. For example, for Qwen-2.5-1.5B on BoolQ (Extraneous, decode), the equal-weight convention used in Fig. 3 yields a relative per-token energy of 1.29×, while computing the ratio from the aggregate means in Table 11 gives $(93.43/166) \div (25.56/56) = 1.23\times$. Both values are derived from the same underlying measurements and differ only in how questions are weighted. All per-token quantities reported in the text and Fig. 3 follow the equal-weight, per-question convention.

Table 8: Phrasing pattern summary across five LLMs on Pixel 8 pro. Energy columns are shaded green, where darker indicates lower energy, and quality columns are shaded blue, where darker indicates higher quality.

Model	Phrasing	Prompt Tok.	Completion Tok.	Prefill Power (W)	Decode Power (W)	Prefill Latency (s)	Decode Latency (s)	Total Latency (s)	Prefill Energy (J)	Decode Energy (J)	Rel.	Corr.	Coh.	Comp.	Instr. Adh.	Internal Cons.
Gemma-2-2B	Base	41	538	4.01	6.09	5.85	63.01	68.86	23.71	381.19	0.91	0.91	0.89	0.83	0.82	0.96
	Persona	41	533	4.01	6.06	5.75	63.62	69.37	23.31	380.36	0.87	0.88	0.85	0.81	0.77	0.92
	Aggressive	40	556	4.06	6.01	5.61	66.54	72.15	22.93	396.55	0.88	0.85	0.86	0.80	0.77	0.92
	Conversational	42	519	4.04	6.02	5.82	61.11	66.93	23.72	366.61	0.92	0.93	0.89	0.82	0.81	0.98
	CoT	44	518	4.01	6.02	6.02	60.57	66.59	24.32	365.98	0.90	0.88	0.87	0.79	0.75	0.89
	Format	45	476	4.03	6.00	6.06	55.49	61.55	24.54	334.93	0.92	0.89	0.90	0.87	0.83	0.96
	Polite	44	485	4.03	6.08	5.89	55.65	61.54	24.00	338.03	0.91	0.88	0.90	0.84	0.78	0.97
	Technical	43	473	4.03	5.93	5.85	55.49	61.34	23.73	334.13	0.91	0.84	0.88	0.80	0.75	0.93
LLaMA-3.2-1B	Base	56	380	3.57	6.36	34.99	24.45	59.45	125.08	155.94	0.81	0.70	0.82	0.71	0.66	0.79
	Persona	56	409	3.53	6.35	34.83	26.54	61.37	123.59	168.87	0.84	0.73	0.83	0.68	0.73	0.89
	Aggressive	56	351	3.55	6.32	35.21	22.56	57.76	125.68	142.87	0.80	0.72	0.82	0.68	0.65	0.84
	Conversational	58	385	3.57	6.39	36.31	24.78	61.09	130.57	158.45	0.79	0.68	0.77	0.67	0.67	0.77
	CoT	59	374	3.58	6.30	37.00	24.26	61.25	133.16	154.47	0.64	0.62	0.77	0.52	0.51	0.78
	Format	60	333	3.56	6.15	37.77	21.44	59.21	134.44	135.52	0.71	0.67	0.74	0.62	0.60	0.77
	Polite	60	352	3.57	6.37	38.03	22.54	60.58	136.44	143.81	0.82	0.78	0.85	0.68	0.69	0.78
	Technical	59	360	3.56	6.11	37.14	23.36	60.50	132.24	147.93	0.84	0.83	0.79	0.74	0.70	0.82
Qwen-2.5-0.5B	Base	50	234	3.85	5.22	12.83	15.04	27.87	49.06	80.25	0.69	0.60	0.73	0.52	0.55	0.73
	Persona	50	253	3.81	5.13	12.51	16.48	28.99	47.56	88.09	0.74	0.70	0.78	0.57	0.55	0.82
	Aggressive	50	275	3.78	5.05	12.57	17.99	30.55	47.28	96.72	0.76	0.65	0.77	0.60	0.57	0.87
	Conversational	52	256	3.82	5.28	13.13	16.56	29.68	49.89	89.12	0.71	0.62	0.71	0.51	0.52	0.81
	CoT	53	243	3.75	5.03	13.61	15.77	29.39	50.94	84.26	0.69	0.62	0.74	0.52	0.53	0.77
	Format	54	281	3.80	5.00	13.31	18.90	32.21	50.38	101.00	0.72	0.65	0.71	0.58	0.61	0.77
	Polite	54	253	3.82	5.27	13.46	16.29	29.76	51.12	87.08	0.71	0.63	0.73	0.55	0.52	0.79
	Technical	53	233	3.79	4.94	12.94	15.28	28.22	48.71	81.31	0.68	0.63	0.77	0.51	0.55	0.87
Qwen-2.5-1.5B	Base	50	272	3.54	5.12	43.47	30.70	74.17	154.18	155.54	0.71	0.68	0.76	0.61	0.61	0.78
	Persona	50	294	3.55	5.08	42.15	34.37	76.53	150.20	173.41	0.85	0.79	0.82	0.69	0.70	0.84
	Aggressive	50	259	3.55	5.00	42.92	29.50	72.42	152.59	148.93	0.82	0.80	0.84	0.70	0.69	0.90
	Conversational	52	287	3.56	5.07	43.94	32.93	76.87	157.11	166.39	0.76	0.78	0.80	0.67	0.67	0.89
	CoT	53	204	3.55	5.01	46.12	22.24	68.36	163.99	113.42	0.72	0.71	0.82	0.57	0.56	0.89
	Format	54	201	3.54	5.03	46.59	21.94	68.53	164.76	111.73	0.77	0.78	0.81	0.66	0.66	0.93
	Polite	54	275	3.55	5.14	46.50	30.83	77.33	165.73	156.91	0.79	0.76	0.80	0.72	0.70	0.83
	Technical	53	259	3.57	5.12	44.02	28.73	72.75	157.62	146.32	0.77	0.75	0.84	0.69	0.69	0.93
SmolLM2-360M	Base	62	243	3.82	4.80	7.92	13.55	21.48	30.03	67.24	0.64	0.55	0.66	0.49	0.49	0.68
	Persona	63	201	3.81	4.67	7.98	11.16	19.15	30.26	54.40	0.64	0.56	0.63	0.50	0.54	0.77
	Aggressive	62	258	3.83	4.77	7.87	14.50	22.37	29.96	70.90	0.62	0.49	0.60	0.38	0.44	0.61
	Conversational	63	262	3.79	4.82	8.28	14.66	22.94	31.20	72.21	0.58	0.44	0.58	0.37	0.38	0.60
	CoT	65	259	3.81	4.78	8.18	14.54	22.72	31.07	71.86	0.54	0.44	0.61	0.39	0.45	0.64
	Format	66	255	3.82	4.77	8.30	14.27	22.57	31.63	70.13	0.72	0.56	0.66	0.54	0.51	0.74
	Polite	66	218	3.81	4.84	8.32	12.09	20.41	31.42	59.81	0.65	0.62	0.68	0.45	0.47	0.80
	Technical	64	269	3.81	4.76	8.28	15.15	23.44	31.45	75.47	0.61	0.50	0.65	0.36	0.46	0.69

Table 9: Phrasing pattern summary across five LLMs on Pixel 7. Energy columns are shaded green, where darker indicates lower energy, and quality columns are shaded blue, where darker indicates higher quality.

Model	Phrasing	Prompt Tok.	Completion Tok.	Prefill Power (W)	Decode Power (W)	Prefill Latency (s)	Decode Latency (s)	Total Latency (s)	Prefill Energy (J)	Decode Energy (J)	Rel.	Corr.	Coh.	Comp.	Instr. Adh.	Internal Cons.
Gemma-2-2B	Base	41	538	2.79	3.13	14.95	107.76	122.71	43.87	337.80	0.89	0.90	0.89	0.84	0.81	0.95
	Persona	41	533	2.84	3.15	14.34	107.88	122.22	43.03	335.80	0.88	0.88	0.85	0.79	0.77	0.92
	Aggressive	40	556	2.75	3.13	13.58	112.61	126.19	38.96	350.58	0.89	0.86	0.86	0.81	0.80	0.94
	Conversational	42	519	2.84	3.15	14.64	103.74	118.38	43.31	324.21	0.92	0.92	0.89	0.82	0.81	0.98
	CoT	44	518	2.91	3.13	15.60	103.27	118.87	46.89	323.56	0.89	0.88	0.86	0.80	0.76	0.93
	Format	45	476	2.94	3.11	15.66	94.34	110.00	47.28	294.89	0.92	0.93	0.91	0.86	0.79	0.96
	Polite	44	485	2.92	3.17	14.88	95.43	110.31	44.91	301.51	0.91	0.87	0.88	0.79	0.83	0.95
	Technical	43	473	2.84	3.08	14.74	94.32	109.06	43.48	294.37	0.91	0.87	0.87	0.83	0.77	0.91
LLaMA-3.2-1B	Base	56	381	2.47	3.39	55.09	35.88	90.97	138.13	122.21	0.81	0.68	0.80	0.72	0.66	0.80
	Persona	56	410	2.48	3.40	54.50	38.94	93.44	136.55	133.09	0.83	0.77	0.81	0.71	0.73	0.89
	Aggressive	56	355	2.49	3.40	54.89	33.48	88.37	137.36	114.63	0.80	0.71	0.80	0.64	0.62	0.82
	Conversational	58	387	2.48	3.40	57.38	36.52	93.90	144.02	125.17	0.78	0.68	0.81	0.69	0.66	0.79
	CoT	59	376	2.49	3.36	58.56	35.74	94.30	147.47	122.24	0.63	0.64	0.77	0.54	0.56	0.79
	Format	60	334	2.49	3.33	60.16	31.61	91.77	151.09	107.99	0.72	0.68	0.74	0.66	0.60	0.80
	Polite	60	350	2.50	3.43	60.44	32.85	93.29	152.19	112.44	0.79	0.77	0.84	0.69	0.68	0.83
	Technical	59	360	2.48	3.29	58.93	34.11	93.04	147.78	116.49	0.84	0.86	0.82	0.74	0.69	0.86
Qwen-2.5-0.5B	Base	50	234	2.12	2.69	17.70	27.13	44.83	37.99	72.57	0.69	0.61	0.72	0.55	0.55	0.74
	Persona	50	253	2.13	2.66	17.16	29.66	46.82	36.78	78.62	0.74	0.71	0.77	0.56	0.58	0.85
	Aggressive	50	275	2.13	2.66	17.18	32.19	49.38	36.83	85.82	0.75	0.66	0.75	0.61	0.58	0.87
	Conversational	52	256	2.12	2.68	18.06	29.70	47.76	38.62	79.46	0.69	0.64	0.73	0.50	0.53	0.81
	CoT	53	243	2.14	2.65	18.88	28.44	47.32	40.71	75.84	0.72	0.68	0.74	0.55	0.55	0.77
	Format	54	281	2.15	2.64	18.42	34.05	52.47	39.97	90.27	0.73	0.67	0.73	0.56	0.56	0.77
	Polite	54	253	2.14	2.68	18.69	29.21	47.90	40.42	78.12	0.71	0.65	0.73	0.55	0.54	0.75
	Technical	53	233	2.12	2.61	17.78	27.52	45.30	38.13	73.23	0.65	0.62	0.77	0.48	0.55	0.86
Qwen-2.5-1.5B	Base	50	272	2.33	2.86	65.33	44.36	109.69	154.26	125.04	0.77	0.67	0.77	0.62	0.64	0.76
	Persona	50	294	2.34	2.83	62.71	49.72	112.44	148.01	138.44	0.81	0.75	0.78	0.69	0.69	0.84
	Aggressive	50	259	2.33	2.82	63.87	42.68	106.55	150.01	119.82	0.84	0.79	0.84	0.71	0.71	0.90
	Conversational	52	287	2.35	2.87	66.03	47.59	113.62	156.63	134.89	0.77	0.78	0.81	0.69	0.65	0.86
	CoT	53	204	2.35	2.81	70.27	32.11	102.39	166.26	90.27	0.73	0.70	0.81	0.61	0.54	0.88
	Format	54	201	2.34	2.83	70.20	31.73	101.93	165.84	89.56	0.76	0.79	0.85	0.64	0.69	0.92
	Polite	54	275	2.35	2.85	70.63	44.64	115.26	166.84	125.90	0.81	0.74	0.81	0.71	0.68	0.85
	Technical	53	259	2.34	2.87	66.27	41.53	107.80	156.72	118.12	0.80	0.74	0.85	0.70	0.70	0.93
SmolLM2-360M	Base	62	243	2.05	2.45	11.07	16.89	27.96	22.85	41.36	0.64	0.58	0.63	0.48	0.51	0.69
	Persona	63	201	2.04	2.43	11.06	13.65	24.71	22.73	33.03	0.64	0.60	0.64	0.46	0.49	0.73
	Aggressive	62	271	2.07	2.47	10.93	19.58	30.51	22.72	47.68	0.61	0.44	0.58	0.39	0.42	0.61
	Conversational	63	265	2.05	2.49	11.55	18.53	30.08	23.83	45.88	0.55	0.45	0.54	0.38	0.36	0.58
	CoT	65	268	2.04	2.47	11.36	19.13	30.50	23.31	46.52	0.53	0.47	0.62	0.37	0.46	0.61
	Format	66	255	2.05	2.46	11.59	17.85	29.44	23.79	43.87	0.74	0.54	0.70	0.56	0.46	0.68
	Polite	66	218	2.09	2.44	11.63	14.80	26.43	24.20	36.19	0.64	0.63	0.67	0.45	0.48	0.73
	Technical	64	278	2.07	2.47	11.55	20.10	31.65	23.97	49.03	0.64	0.51	0.65	0.38	0.46	0.69

Table 10: Cognitive load summary on AI2-ARC across five LLMs on Pixel 8 Pro. Within each model block, the energy columns are shaded green (darker = lower / better) and the accuracy column is shaded blue (darker = higher / better).

Model	Load	Prompt Tokens	Completion Tokens	Prefill Power (W)	Decode Power (W)	Prefill Latency (s)	Decode Latency (s)	Total Latency (s)	Prefill Energy (J)	Decode Energy (J)	Accuracy
Gemma-2-2B	Base	63	174	3.95	6.37	6.63	16.66	23.28	26.27	106.04	0.84
	Intrinsic	129	258	4.49	6.19	8.90	26.58	35.49	40.05	164.25	0.84
	Extraneous	167	254	4.70	6.18	10.41	26.48	36.89	48.98	163.37	0.64
	Germane	70	218	4.01	6.33	6.81	21.18	28.00	27.45	133.97	0.88
LLaMA-3.2-1B	Base	79	197	3.59	6.04	47.06	11.95	59.00	169.74	72.67	0.64
	Intrinsic	143	305	3.69	6.09	81.33	19.13	100.46	300.84	117.01	0.47
	Extraneous	179	240	3.71	5.92	100.11	15.09	115.21	372.55	90.85	0.56
	Germane	86	265	3.61	6.12	50.44	16.24	66.67	182.63	99.56	0.59
Qwen-2.5-0.5B	Base	73	190	3.34	5.20	18.26	12.03	30.29	60.95	65.41	0.48
	Intrinsic	138	207	3.40	5.27	31.38	13.60	44.98	106.70	73.30	0.44
	Extraneous	179	258	3.41	5.33	39.15	17.30	56.44	133.53	92.93	0.48
	Germane	80	238	3.35	5.34	19.71	15.44	35.15	66.10	84.44	0.52
Qwen-2.5-1.5B	Base	73	125	3.59	5.65	58.55	12.23	70.79	210.88	71.92	0.80
	Intrinsic	138	253	3.67	5.69	107.14	27.03	134.18	393.95	155.01	0.72
	Extraneous	179	272	3.73	5.65	131.42	30.65	162.07	491.30	173.18	0.77
	Germane	80	191	3.61	5.94	64.02	18.90	82.92	231.55	111.60	0.76
SmolLM2-360M	Base	85	72	3.49	4.01	11.89	4.60	16.49	41.63	21.61	0.24
	Intrinsic	150	160	3.46	4.63	21.09	10.31	31.40	73.13	49.85	0.17
	Extraneous	190	122	3.47	4.46	26.94	7.89	34.83	93.58	37.92	0.24
	Germane	92	115	3.49	4.45	12.85	7.40	20.25	44.83	34.94	0.20

Table 11: Cognitive load summary on BoolQ across five LLMs on Pixel 8 Pro. Within each model block, the energy columns are shaded green (darker = lower / better) and the accuracy column is shaded blue (darker = higher / better).

Model	Load	Prompt Tokens	Completion Tokens	Prefill Power (W)	Decode Power (W)	Prefill Latency (s)	Decode Latency (s)	Total Latency (s)	Prefill Energy (J)	Decode Energy (J)	Accuracy
Gemma-2-2B	Base	51	119	4.15	6.15	6.23	11.56	17.79	25.86	72.90	0.87
	Intrinsic	100	137	4.57	6.11	7.97	13.86	21.83	36.45	86.23	0.72
	Extraneous	132	236	4.86	6.21	9.14	24.91	34.05	44.54	154.16	0.57
	Germane	48	215	4.06	6.37	6.10	21.49	27.59	24.80	136.30	0.69
LLaMA-3.2-1B	Base	65	116	3.68	5.94	40.47	7.10	47.57	149.01	44.48	0.40
	Intrinsic	113	145	3.78	6.14	67.84	9.10	76.93	256.49	56.83	0.49
	Extraneous	141	236	3.81	6.17	82.20	15.12	97.31	312.80	94.92	0.32
	Germane	63	220	3.69	6.36	39.75	13.77	53.52	146.77	88.42	0.37
Qwen-2.5-0.5B	Base	61	66	3.54	4.54	15.94	3.98	19.92	56.39	19.20	0.60
	Intrinsic	108	78	3.52	4.19	25.56	4.92	30.48	89.92	25.07	0.65
	Extraneous	142	189	3.56	5.35	33.02	12.42	45.44	117.64	68.56	0.44
	Germane	58	190	3.56	5.16	14.93	12.10	27.03	52.96	67.07	0.44
Qwen-2.5-1.5B	Base	61	56	3.59	4.90	47.75	4.84	52.59	171.70	25.56	0.88
	Intrinsic	108	101	3.64	5.11	81.40	9.38	90.78	296.11	52.01	0.67
	Extraneous	142	166	3.67	5.46	106.24	16.81	123.06	389.55	93.43	0.65
	Germane	58	192	3.63	5.69	45.98	18.08	64.06	166.74	105.24	0.79
SmolLM2-360M	Base	74	45	3.65	4.11	9.60	2.59	12.19	35.03	12.30	0.49
	Intrinsic	122	104	3.62	4.52	15.88	6.36	22.23	57.37	30.75	0.53
	Extraneous	155	122	3.58	4.68	20.22	7.34	27.56	72.48	36.20	0.36
	Germane	69	91	3.61	4.67	9.00	5.28	14.28	32.52	26.42	0.40

Table 12: Cognitive load summary on SVAMP across five LLMs on Pixel 8 Pro. Within each model block, the energy columns are shaded green (darker = lower / better) and the accuracy column is shaded blue (darker = higher / better).

Model	Load	Prompt Tokens	Completion Tokens	Prefill Power (W)	Decode Power (W)	Prefill Latency (s)	Decode Latency (s)	Total Latency (s)	Prefill Energy (J)	Decode Energy (J)	Accuracy
Gemma-2-2B	Base	45	88	4.20	6.40	6.21	8.31	14.52	25.96	53.69	0.68
	Intrinsic	77	90	4.47	6.44	7.55	8.61	16.16	33.62	55.67	0.80
	Extraneous	128	140	4.77	6.45	9.69	14.19	23.88	46.20	91.98	0.40
	Germane	46	122	4.17	6.51	6.28	11.80	18.08	26.12	77.56	0.72
LLaMA-3.2-1B	Base	61	100	3.77	5.51	37.72	6.10	43.83	142.21	34.76	0.76
	Intrinsic	92	97	3.78	5.59	56.37	5.94	62.31	212.80	34.03	0.80
	Extraneous	140	199	3.85	5.81	82.94	12.76	95.70	319.15	78.16	0.24
	Germane	60	133	3.76	5.81	38.37	8.17	46.54	144.13	48.78	0.56
Qwen-2.5-0.5B	Base	56	118	3.71	5.11	14.48	7.01	21.49	53.59	39.04	0.64
	Intrinsic	88	115	3.64	5.41	21.57	6.78	28.35	78.48	37.88	0.76
	Extraneous	139	318	3.66	5.76	31.98	20.94	52.92	116.91	121.37	0.29
	Germane	56	173	3.74	5.68	13.91	10.32	24.23	51.91	59.58	0.67
Qwen-2.5-1.5B	Base	56	141	3.42	5.20	44.02	13.89	57.91	150.46	70.80	0.76
	Intrinsic	88	112	3.49	5.11	64.32	11.37	75.69	224.40	56.86	0.76
	Extraneous	139	264	3.55	4.90	99.14	29.75	128.89	352.72	142.60	0.56
	Germane	56	182	3.40	5.23	42.98	18.34	61.33	146.29	93.25	0.64
SmolLM2-360M	Base	67	154	3.69	4.89	9.15	9.07	18.22	33.74	45.73	0.28
	Intrinsic	99	110	3.67	4.85	13.18	6.52	19.70	48.35	32.09	0.36
	Extraneous	150	179	3.63	4.88	20.25	10.55	30.80	73.33	54.44	0.12
	Germane	67	120	3.69	4.79	8.83	7.11	15.94	32.60	35.31	0.20

G Additional Figures for RQ1

Overview. This appendix provides additional per-token energy results supporting the analysis in Section 4.1. These figures complement Figure 3 by covering additional datasets and device. Each figure reports prefill and decode per-token energy. The top row shows absolute values, while the bottom row normalizes each sub-property by the corresponding Base prompt.

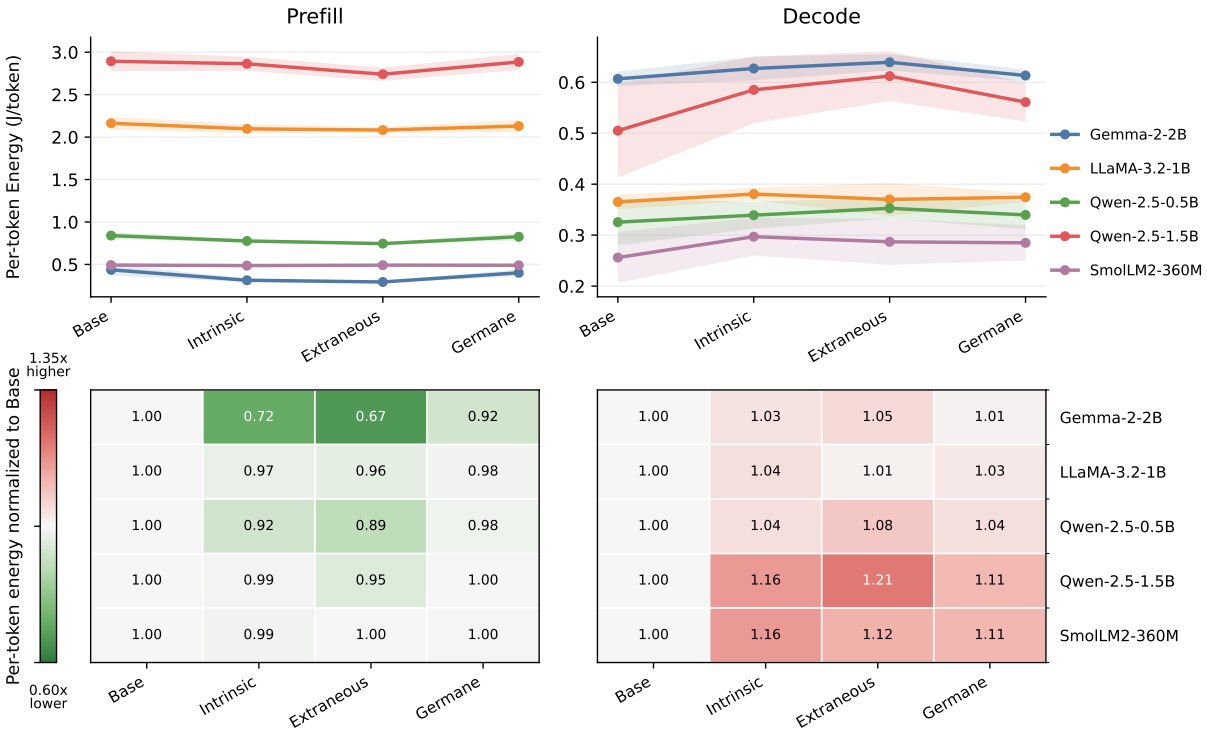


Figure 7: Additional per-token energy results for cognitive load on AI2-ARC on Pixel 8 Pro. The figure reports prefill and decode per-token energy across intrinsic, extraneous, and germane load variants.

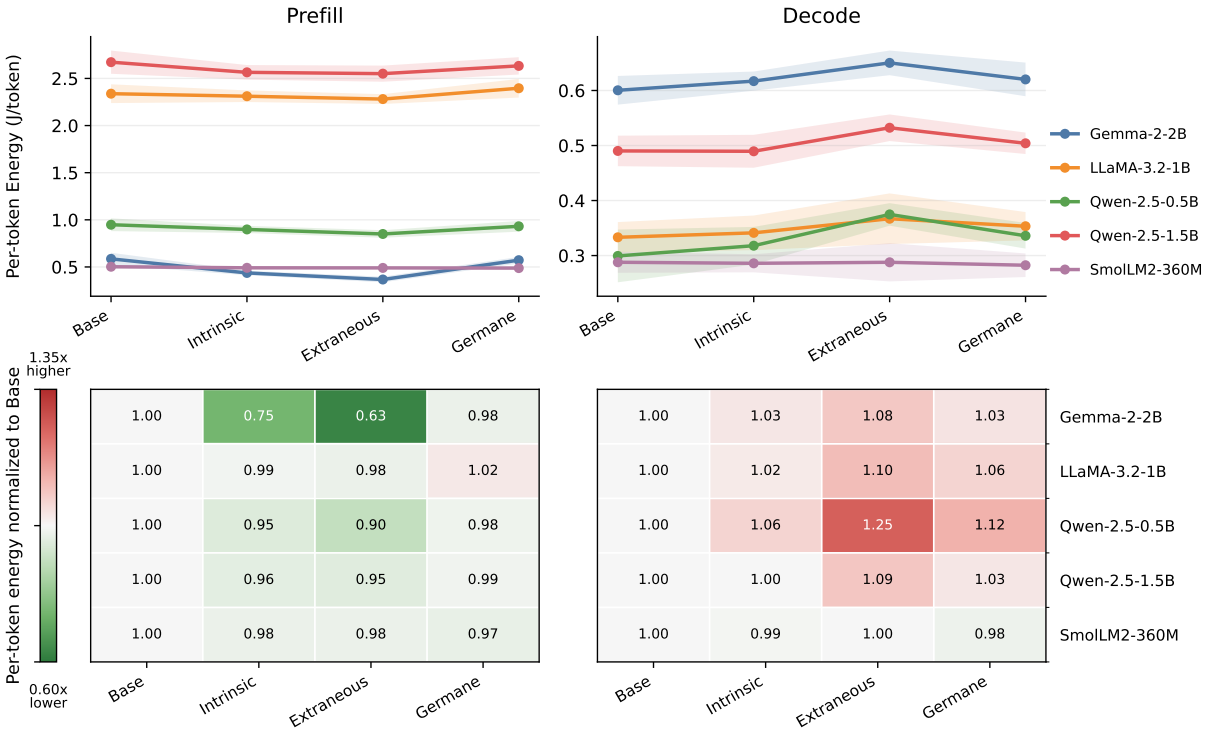


Figure 8: Additional per-token energy results for cognitive load on SVAMP on Pixel 8 Pro. The figure reports prefill and decode per-token energy across intrinsic, extraneous, and germane load variants.

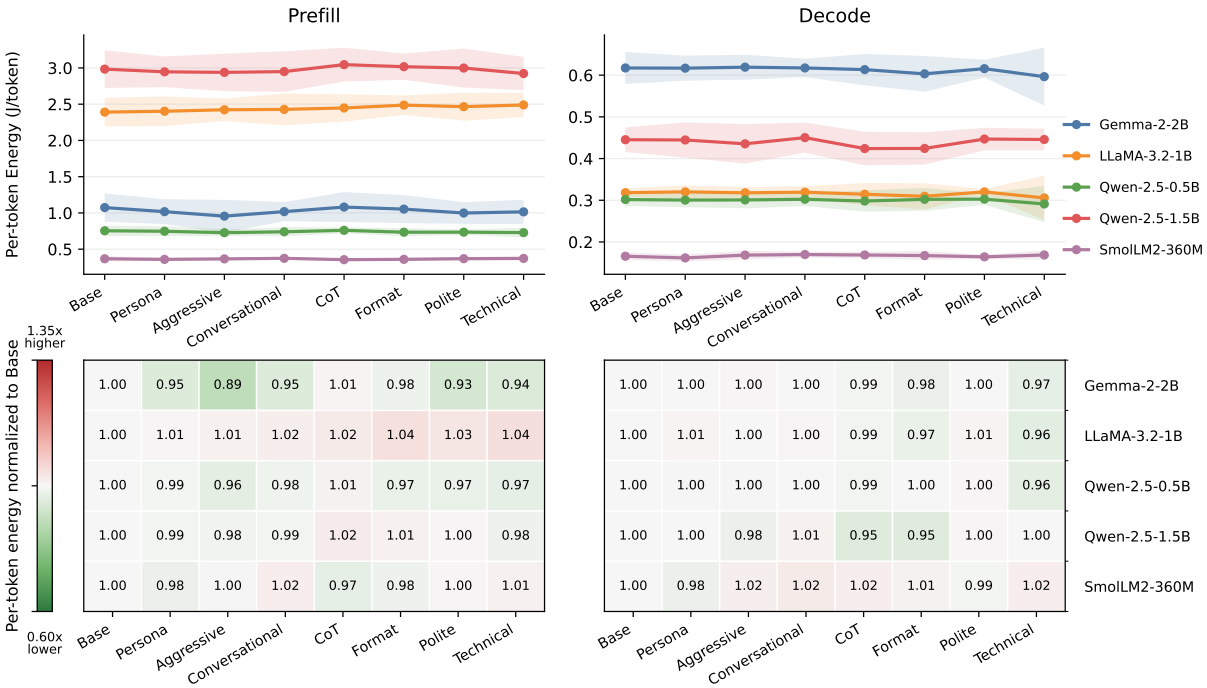


Figure 9: Additional per-token energy results for phrasing pattern on Pixel 7. The figure reports prefill and decode per-token energy across phrasing-pattern sub-properties.

H Additional Figures for RQ2

Overview. This appendix provides additional token-usage and fixed-baseline burden results supporting the analysis in Section 4.2. These figures complement Figure 4 by covering additional datasets and prompt-property settings. The top row reports total token ratio relative to Base. The bottom row reports fixed-baseline prefill and decode burden relative to Base. Bottom-row values are normalized by Base prompt/completion token counts, respectively, after subtracting the corresponding Base burden.

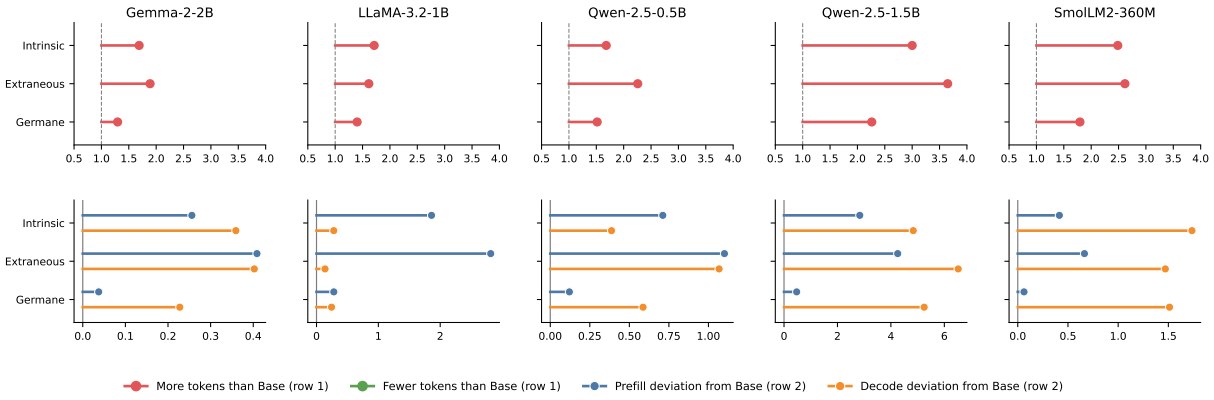


Figure 10: Additional token usage results for cognitive load on AI2-ARC on Pixel 8 Pro. The top row reports total token ratio relative to Base across cognitive load variants. The bottom row reports fixed-baseline prefill and decode burden relative to Base.

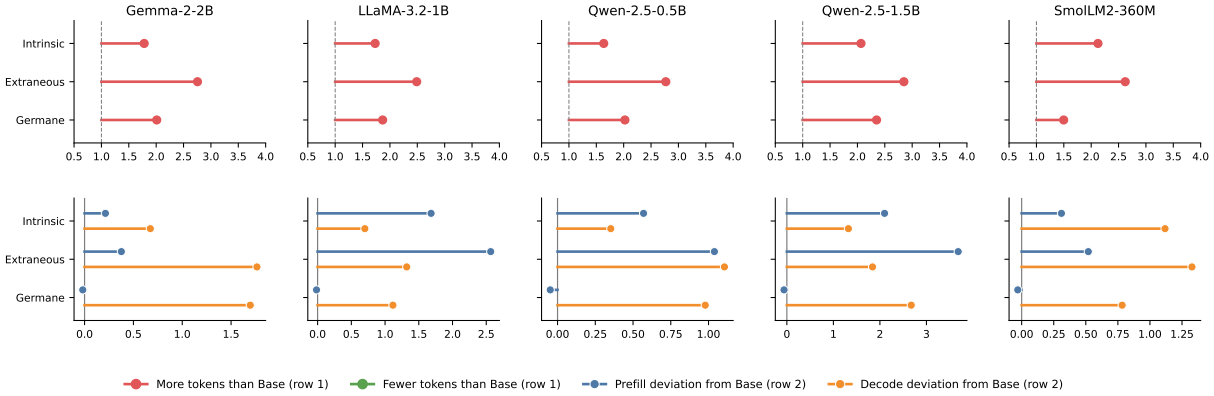


Figure 11: Additional token usage results for cognitive load on BoolQ on Pixel 8 Pro. The top row reports total token ratio relative to Base across cognitive load variants. The bottom row reports fixed-baseline prefill and decode burden relative to Base.

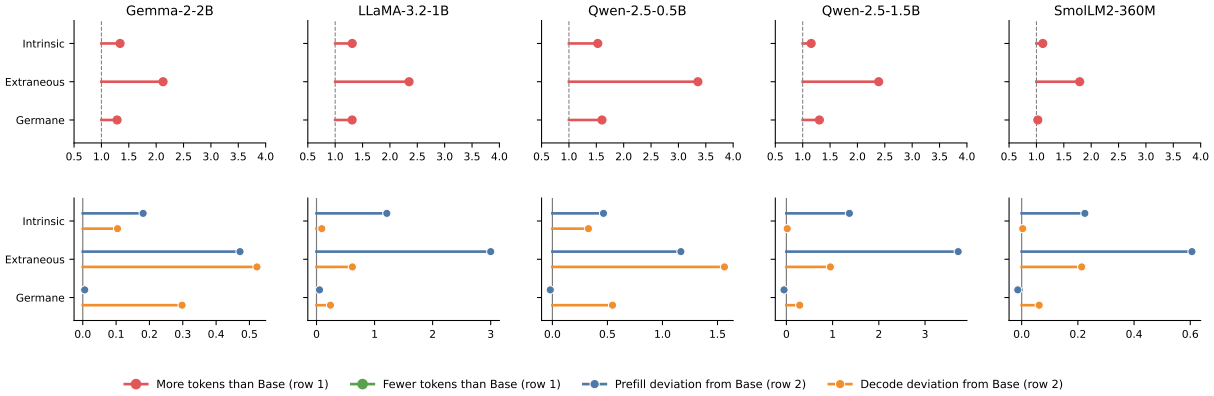


Figure 12: Additional token usage results for cognitive load on SVAMP. The top row reports total token ratio relative to Base across cognitive load variants. The bottom row reports fixed-baseline prefill and decode burden relative to Base.

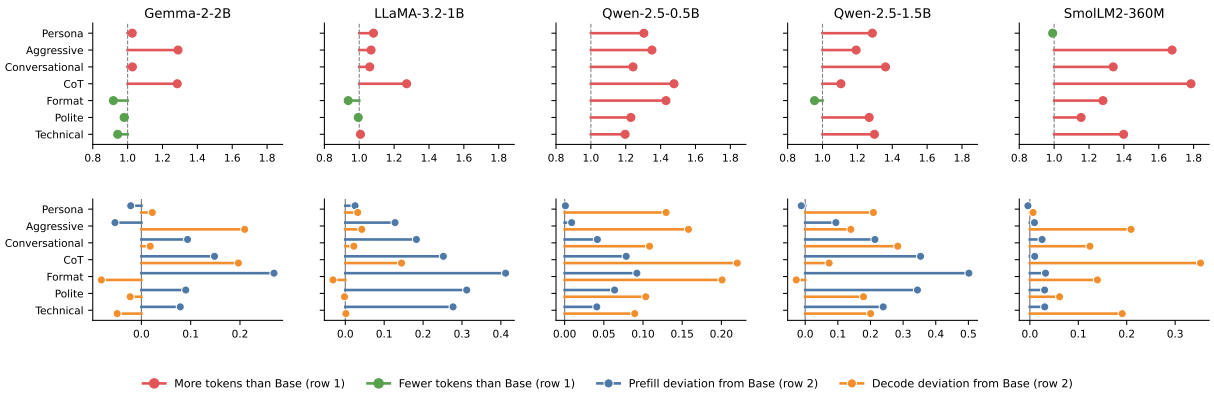


Figure 13: Additional token usage results for phrasing pattern on Pixel 7. The top row reports total token ratio relative to Base across phrasing-pattern sub-properties. The bottom row reports fixed-baseline prefill and decode burden relative to Base.

I Additional Figures for RQ3

Overview. This appendix provides additional response-quality and energy-quality trade-off results supporting the analysis in Section 4.3. These figures complement the main-text analysis by covering cognitive-load accuracy results and additional trade-off settings.

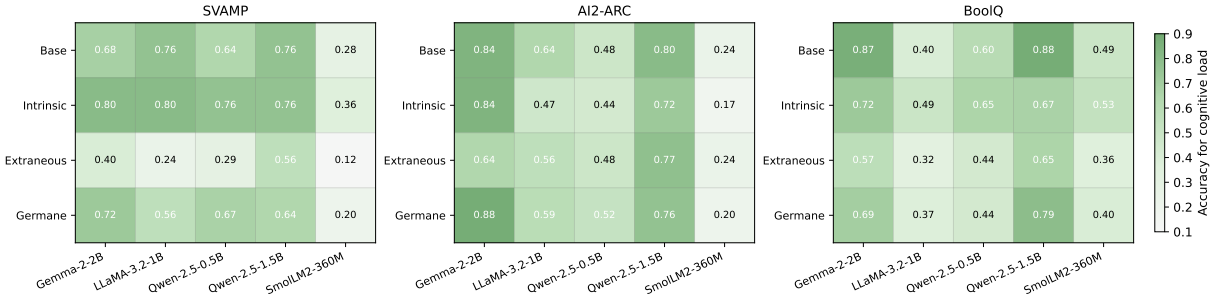


Figure 14: Additional RQ3 results for cognitive load response accuracy. Each heatmap reports accuracy across cognitive load sub-properties, datasets, and models. These results complement the response quality analysis in the main text by showing how intrinsic, extraneous, and germane load affect task correctness under objective ground-truth evaluation.

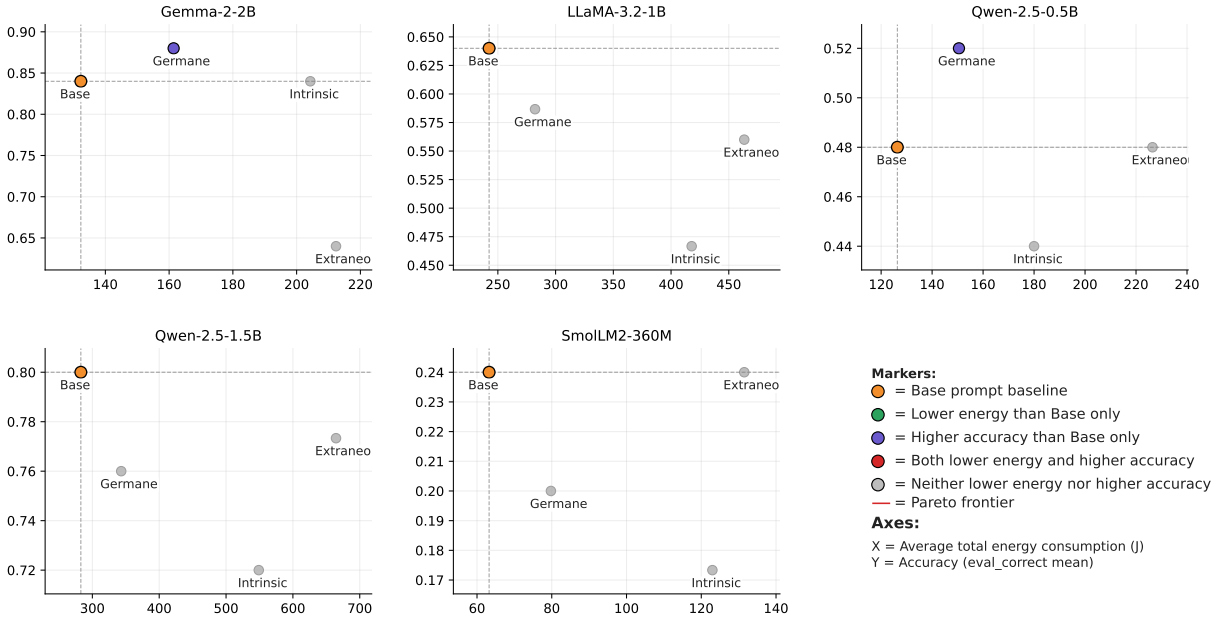


Figure 15: Additional RQ3 energy-accuracy trade-off results for cognitive load on AI2-ARC. Each point represents a cognitive load sub-property under fixed model weights and decoding settings. The x-axis reports average total energy consumption, and the y-axis reports accuracy.

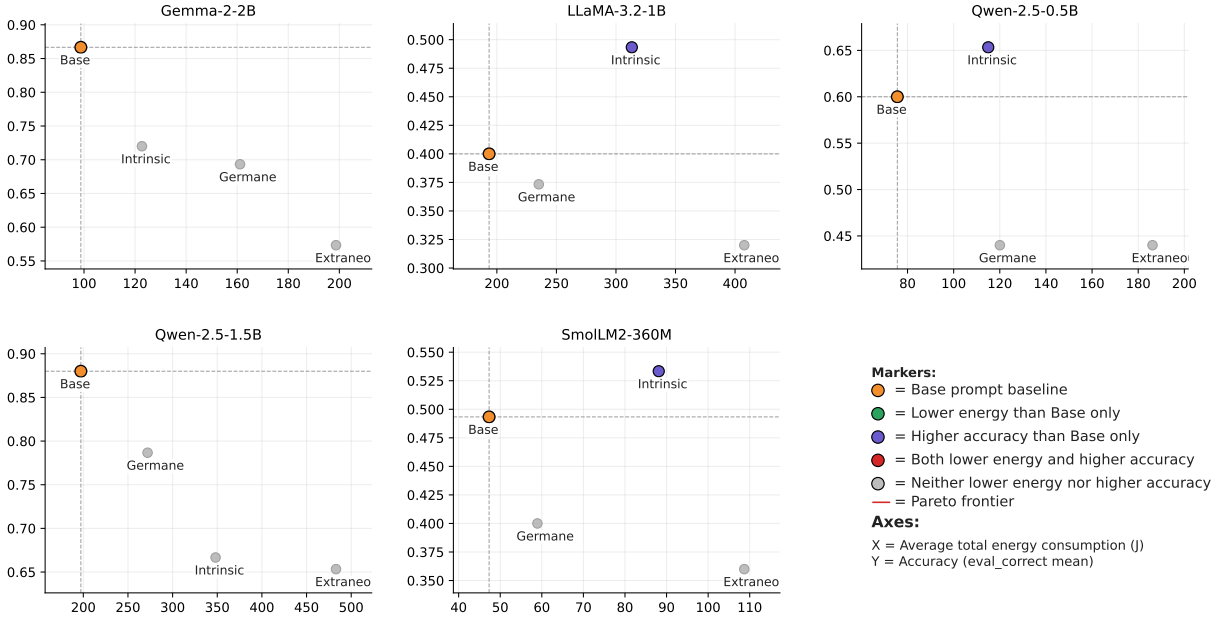


Figure 16: Additional RQ3 energy-accuracy trade-off results for cognitive load on BoolQ. Each point represents a cognitive load sub-property under fixed model weights and decoding settings. The x-axis reports average total energy consumption, and the y-axis reports accuracy.

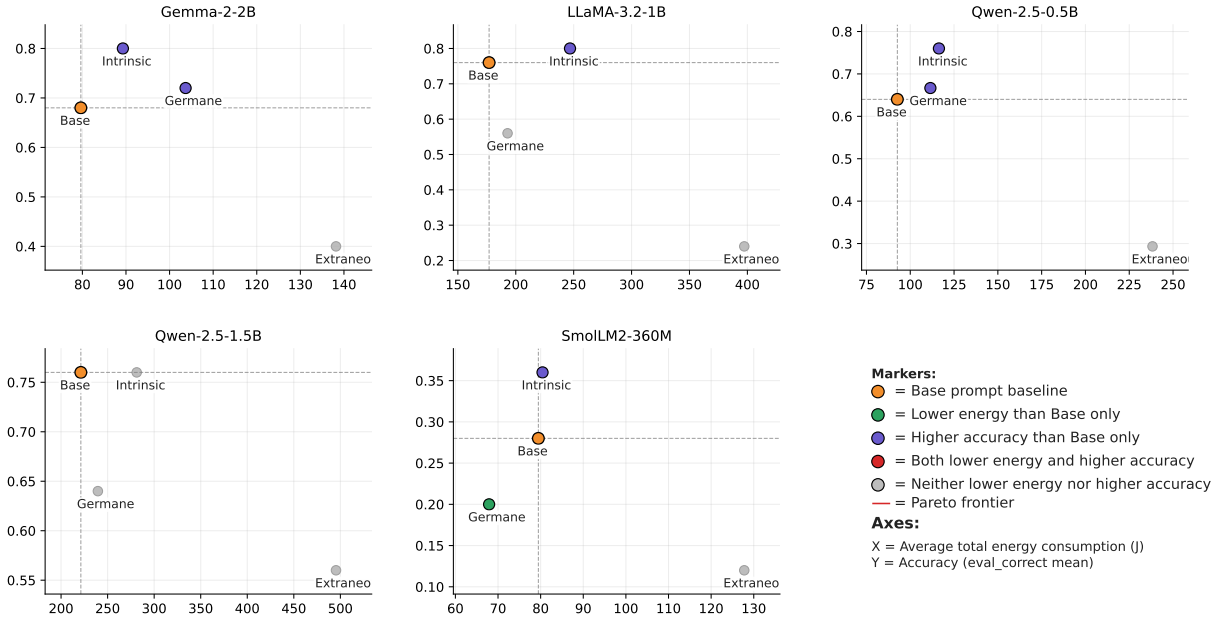


Figure 17: Additional RQ3 energy-accuracy trade-off results for cognitive load on SVAMP. Each point represents a cognitive load sub-property under fixed model weights and decoding settings. The x-axis reports average total energy consumption, and the y-axis reports accuracy.

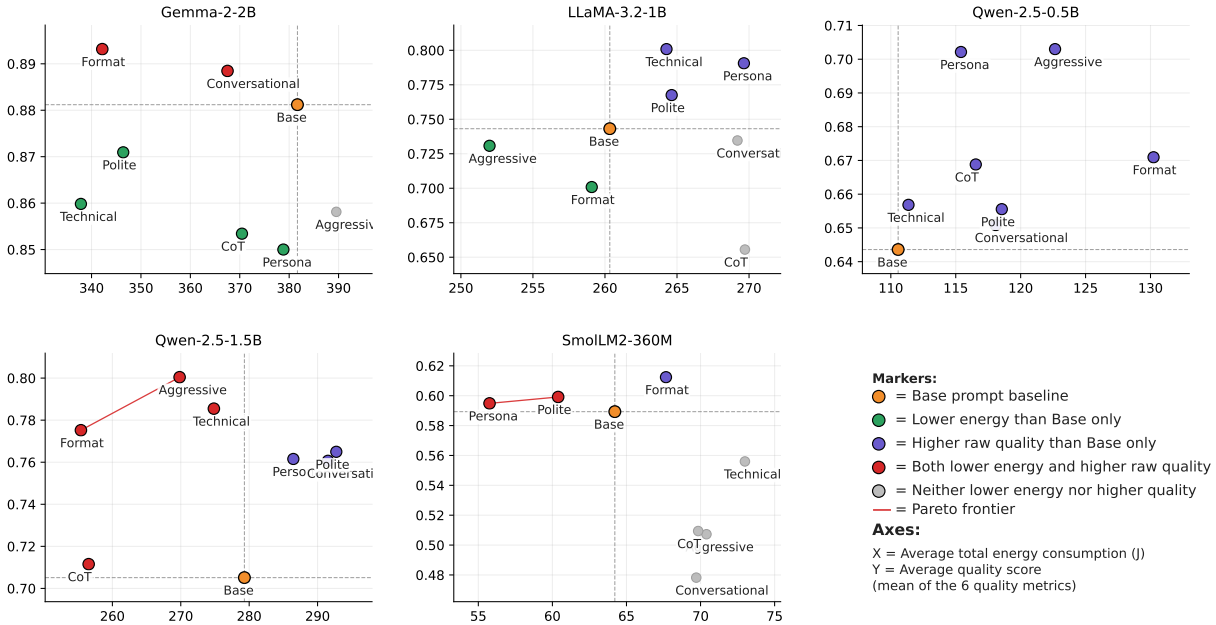


Figure 18: Additional RQ3 energy-quality trade-off results for phrasing on Pixel 7. Each point represents a phrasing pattern sub-property under fixed model weights and decoding settings. The x-axis reports average total energy consumption, and the y-axis reports the averaged response quality score across the six reference-free evaluation dimensions.

J License

Datasets. The datasets used in this study are governed by their respective licenses and data-use terms. SVAMP is released under the *MIT License*; BoolQ under *Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*; AI2-ARC under *Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)*; and CLEF 2025 data under the applicable *Data Use Agreements*.

Software, models, and hardware. DeepEval is used under the *Apache-2.0*, and MLC-LLM under the *Apache-2.0*. The open-source models Llama-3.2, Qwen-2.5, Gemma-2, and SmolLM2 are used under their respective community licenses. Gemini-2.5-Pro is accessed via the *Google AI applicable terms*. Hardware profiling on Pixel 7 and Pixel 8 Pro follows standard consumer and developer terms.