Benchmarking the Energy Savings with Speculative Decoding Strategies

Rohit Dutta[1], Paramita Koley[2], Soham Poddar[1], Janardan Misra[3],
Sanjay Podder[3], Naveen Balani[3], Saptarshi Ghosh[1], Niloy Ganguly[1]
[1] Indian Institute of Technology, Kharagpur, India
[2] Indian Statistical Institute, Kolkata, India
[3] Accenture Labs, Bangalore, India

Abstract

Speculative decoding has emerged as an effective method to reduce latency and inference cost of LLM inferences. However, there has been inadequate attention towards the energy requirements of these models. To address this gap, this paper presents a comprehensive survey of energy requirements of speculative decoding strategies, with detailed analysis on how various factors – model size and family, speculative decoding strategies, and dataset characteristics – influence the energy optimizations.

1 Introduction

Large Language Models (LLMs) have witnessed rapid adoption across a wide range of applications. Despite their utility, the deployment of these models demands substantial computational resources, leading to considerable energy consumption (Wu et al., 2022; Patterson et al., 2022; Poddar et al., 2025). Recent work by Poddar et al. (2025) identifies token decoding latency and model complexity as critical determinants of inference-time energy consumption. While autoregressive models, forming the basis of most LLMs, generate tokens in a sequential manner, inherently resulting in higher decoding latency, *speculative decoding* (Leviathan et al., 2023; Chen et al., 2023) has emerged as a promising approach to reduce decoding time. This approach leverages a *lightweight 'assistant model'* to generate candidate token sequences, followed by a parallelized verification stage in which the larger '*target model*' evaluates multiple tokens in a single pass. This mechanism significantly reduces decoding time while offloading a substantial portion of the sequential generation to a smaller, more efficient model. Given these characteristics, we hypothesize that speculative decoding may also offer reductions in inference-time energy consumption.

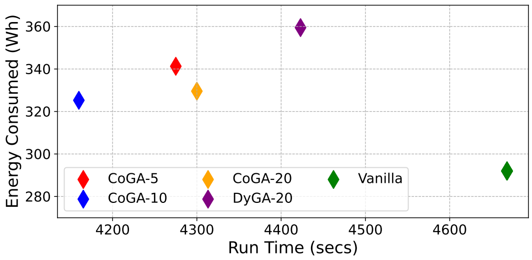

Figure 1: Speculative decoding approaches (CoGA, DyGA) with Vicuna-13B as target ends up in consuming more energy than vanilla decoding (applying only target model) despite exhibiting lower inference latency.

While the existing literature on speculative decoding (Leviathan et al., 2023; Cai et al., 2024; Li et al., 2024a, 2025) has predominantly focused on optimizing response time or latency, its implications for energy efficiency remain largely unexplored. To address this gap, this paper presents a comprehensive survey of speculative decoding techniques, with emphasis on their implications for energy efficiency during inference.

Motivation: Given that speculative decoding (SD) strategies yield speedup in inference latency/time, and response time is strongly correlated with energy consumption (Poddar et al., 2025), one might assume that speculative decoding inherently leads to improved energy efficiency. However, we argue that walltime speedup (Li et al., 2024a, 2025) does *not* necessarily translate into proportional energy savings. Furthermore, even when such a correlation exists, the relationship between runtime and energy consumption is often nonlinear and influenced by multiple interacting factors.

To substantiate this claim, we conducted a preliminary experiment utilizing two representative speculative decoding methods – CoGA (Leviathan et al., 2023) and heuristic (Hugging Face, 2025) – with Vicuna-13B (Chiang et al., 2023) as the target model and Vicuna-68M as the assistant model (details in later sections). Figure 1 reports the runtime versus energy consumption. It illustrates that *the vanilla decoding (i.e., target model alone) consumes less energy than the standard speculative*

Findings of the Association for Computational Linguistics: EACL 2026, pages 4737–4748
March 24-29, 2026 ©2026 Association for Computational Linguistics

decoding approaches, despite having longer runtime. This counterintuitive observation leads to a crucial insight: improvements in inference latency do not inherently yield corresponding reductions in energy consumption, emphasizing a clear need for a systematic analysis of SD strategies from the standpoint of energy efficiency.

In this study, we examine a diverse set of SD techniques across multiple architectures and benchmark datasets, toward elucidating the primary factors governing energy consumption in these approaches. We find that lower inference time may not always translate into proportional energy savings, and that model family and size difference have crucial roles in energy savings.

2 Experimental Setup

Tasks and Datasets: We conduct experiments on three standard inference tasks, common in speculative decoding (Leviathan et al., 2023; Li et al., 2024a, 2025), namely (1) code generation (HUMAN-EVAL (Chen et al., 2021)), (2) mathematical reasoning (GSM-8K (Cobbe et al., 2021)), and (3) summarization (CNN-DM (Nallapati et al., 2016)). We use 256 randomly sampled prompts from each dataset/task except HUMAN-EVAL which has only 164 samples. We employed model-specific chat templates; detailed prompts for the three tasks are illustrated in Table 4, Table 5, and Table 6 in Appendix A).

Target and Assistant Models: We consider LLMs from four families. Specifically, we evaluate: (i) VICUNA-7B and (ii) VICUNA-13B as target models paired with VICUNA-68M as the assistant model; (iii) LLAMA-8B and (iv) LLAMA-70B as target models paired with LLAMA-1B as the assistant model; (v) FLAN-T5-L and (vi) FLAN-T5-XL as target models paired with FLAN-T5-B as the assistant model; and (vii) QWEN3-4B and (viii) QWEN3-8B as target models paired with QWEN3-0.6B as the assistant model. All target–assistant model combinations, are listed in the first column of Table 1. Across all configurations, the target models have been loaded using NF4 4-bit quantization (Dettmers et al., 2023). Refer to Table 3 in Appendix B for model details.

Speculative Decoding strategies: We considered the following SD strategies, namely two variants of standard speculative decoding (Leviathan et al., 2023) – (1) Constant Generation by Assistant (COGA-x) and (2) Dynamic Generation by Assistant (DYGA-x) – and two state-of-the-art SD strategies, namely (3) EAGLE-2 (Li et al., 2024a), and (4) EAGLE-3 (Li et al., 2025). We used model-specific chat-templates with greedy decoding for reproducibility.

In COGA-x, the assistant generates fixed-length drafts (x tokens) per iteration, whereas in DYGA-x, the assistant generates dynamic length drafts, starting with x tokens and adjusting its length in each iteration (by increasing the length by 2 if all tokens in the last draft is accepted, otherwise reducing the draft length by 1). For COGA-x, the value of x was set to $5, 10$ and 20 in our experiments.

MEDUSA (Cai et al., 2023) mitigates autoregressive bottlenecks by augmenting the target model with multiple decoding heads that predict future tokens in parallel, verified through a tree-based attention mechanism. EAGLE-2 (Li et al., 2024a) builds on standard SD by introducing a context-aware dynamic draft tree for generating dynamic drafts. EAGLE-3 (Li et al., 2025) further introduces direct token prediction, allowing the draft model to integrate multi-layer fused features from the target model. Refer Appendix C for further details.

For the purpose of inferences, we employ HugginFace implementations of models (Wolf et al., 2019; Hugging Face, 2025).

Hardware and Energy metrics: Most experiments are performed on a single NVIDIA A5000 GPU with 24GB VRAM hosted in a local server with Intel Xeon Silver 4210R processor and 128GB RAM, running Ubuntu 20.04-LTS with Pytorch v2.6 (with CUDA 12.8) and Huggingface transformers v4.51. The experiments comprising 70B models are performed on a single NVIDIA A6000 GPU with 48GB VRAM and other identical configurations as stated above.

We use the popular Code Carbon (Schmidt et al., 2021) package to measure the energy consumed. During inference, we provide test samples sequentially at batch size 1 to the LLM. Refer to Appendix D for further details.

Evaluation Metrics for SD performance: We employ the following metrics for measuring energy and time performance of SD strategies:

(i) GPU energy saving factor (γ_e^{GPU}) denotes the ratio of GPU energy consumed by the target model under vanilla autoregressive decoding and GPU energy consumed under the speculative decoding (SD) strategy. $\gamma_{\mathbf{e}}^{\mathbf{GPU}} = \text{Energy}_{\text{Target}}^{\text{GPU}} / \text{Energy}_{\text{SD}}^{\text{GPU}}$, $SD \in \{\text{COGA}, \text{DYGA}, \text{EAGLE}, \text{MEDUSA}\}$.

(ii) Total energy saving factor (γ_e^{Total}) denotes the ratio between the total energy consumed by the target model under vanilla autoregressive decoding and the total energy consumed under the SD strategy. In this context, total energy refers to the cumulative energy drawn by the GPU, CPU, and RAM. $\gamma_{\mathbf{e}}^{\mathbf{Total}} = \text{Energy}_{\text{Target}}^{\text{Total}}$ / $\text{Energy}_{\text{SD}}^{\text{Total}}$, SD $\in \{\text{CoGA}, \text{DyGA}, \text{EAGLE}, \text{MEDUSA}\}$.
For both energy gain metrics, we consider the average energy consumed to generate $1K$ tokens.

(iii) Speedup (γ_t) denotes the ratio of inference time under vanilla autoregressive decoding of the target model and the inference time under speculative decoding (SD) strategy. $\gamma_t = \text{Time}_{\text{Target}}$ / Time_{SD}, SD $\in \{\text{CoGA}, \text{DyGA}, \text{EAGLE}, \text{MEDUSA}\}$. For both cases, we consider the average time to decode $1K$ tokens for the same task.
Refer to Appendix E for additional metrics (reported in Table 8).

3 Energy Consumption of SD Strategies

In this section, we analyze the energy consumption of SD strategies to identify the factors driving energy efficiency and optimization. Table 1 reports the speedup and energy saving factor relative to the vanilla autoregressive decoding setting (where only the target model is run for the whole task) for all datasets and for all the settings described earlier. We observe that the simpler SD strategies CoGA and DyGA achieve energy reduction primarily for LLAMA and FLAN-T5 family (up to 2.0×), on all datasets except CNN-DM. But SOTA SD strategies EAGLE-2 and EAGLE-3 achieve notable energy reduction (1.34× - 2.51×) on all four model settings across all datasets except (LLAMA-70B, CNN-DM) model-dataset pair. An explanation can be that high speedup in EAGLE methods results in notable energy saving. For CoGA and DyGA, relatively smaller speedup fails to reflect into useful energy saving.

Model-specific trends: Model family plays a crucial role in energy reduction. Among decoder-only models, LLAMA models generally achieve moderate to high energy savings (up to 2.09× for LLAMA-8B and up to 1.36× for LLAMA-70B), VICUNA models achieve energy savings only on EAGLE-(2,3), with no reduction for CoGA and DyGA. Among encoder-decoder family, FLAN-T5 models achieve high energy savings (up to 2.02× for FLAN-T5-L and upto 1.86× for FLAN-T5-XL). On the other hand, energy savings is minimal for QWEN3 family (1.15× for QWEN3-4B and 0.91× for QWEN3-8B), with no reduction for many cases.

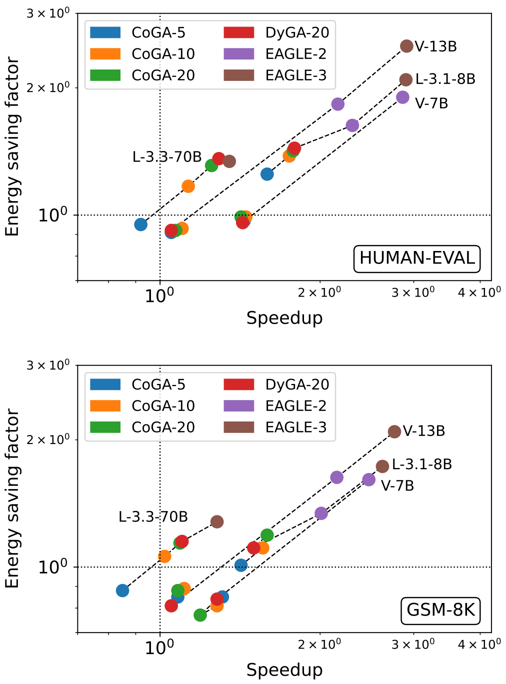

Figure 2: Speedup (γ_t) and total energy savings (γ_e^{Total}) for speculative decoding methods across model pairs on HUMAN-EVAL and GSM-8K. EAGLE consistently achieves the highest speedup and energy efficiency. LLAMA-based pairs under both CoGA and DyGA exhibit positive runtime and energy gains, while VICUNA-based pairs frequently underperform, with values often below unity.

Dataset-specific trends: Energy reduction substansially varies across datasets, with maximum for HUMAN-EVAL and minimum for CNN-DM. CoGA and DyGA achieve maximum energy saving of 2.0× on HUMAN-EVAL with FLAN-T5-L. EAGLE performed the best both latency and energy-wise - achieving highest energy saving of 2.5× times on HUMAN-EVAL with VICUNA-13B model. Surprisingly, LLAMA-70B, that achieves notable energy saving in most other SD setups, ends up with increased energy consumption with SD in case of the CNN-DM dataset (relative to vannila decoding). Similarly, QWEN3-4B that achieves 1.05× times energy gain in HUMAN-EVAL, drops to 0.85× in CNN-DM. Thus, energy optimization may vary significantly depending on the task/dataset.

3.1 Analysis of Speedup and Energy Savings

Figure 2 shows speedup (γ_t) vs energy saving factor (γ_e^{Total}) for all models and SD strategies for the HUMAN-EVAL and GSM-8K datasets. A value

		Human-Eval			GSM-8k			CNN-DM		
		Speedup	Energy Saving Factor		Speedup	Energy Saving Factor		Speedup	Energy Saving Factor	
Target vs Assistant	**SD Method**	γ_t	γ_e^{GPU}	γ_e^{Total}	γ_t	γ_e^{GPU}	γ_e^{Total}	γ_t	γ_e^{GPU}	γ_e^{Total}
Vicuna-7B vs Vicuna-68M	CoGA-20	1.42×	0.83×	0.99×	1.19×	0.72×	0.77×	1.44×	0.89×	1.03×
	DyGA-20	1.43×	0.8×	0.96×	1.28×	0.73×	0.84×	1.34×	0.82×	0.94×
	Eagle 2	2.86×	1.57×	1.90×	2.47×	1.37×	1.61×	1.98×	1.17×	1.38×
	Medusa	1.87×	1.78×	1.76×	2.09×	1.85×	1.90×	1.26×	1.21×	1.20×
Vicuna-13B vs Vicuna-68M	CoGA-20	1.07×	0.93×	0.92×	1.08×	0.75×	0.88×	1.02×	0.89×	0.96×
	DyGA-20	1.05×	0.88×	0.92×	1.05×	0.71×	0.81×	0.98×	0.85×	0.91×
	Eagle 2	2.16×	1.79×	1.83×	2.15×	1.46×	1.63×	1.52×	1.29×	1.47×
	Eagle 3	2.91×	2.40×	2.51×	2.76×	1.87×	2.09×	2.17×	1.86×	2.10×
	Medusa	2.24×	2.09×	2.11×	2.10×	2.05×	2.03×	1.47×	1.46×	1.43×
Llama-8B vs Llama-1B	CoGA-20	1.78×	1.23×	1.42×	1.59×	1.10×	1.19×	0.96×	0.79×	0.83×
	DyGA-20	1.79×	1.26×	1.44×	1.50×	1.06×	1.11×	0.99×	0.80×	0.85×
	Eagle 2	2.30×	1.35×	1.63×	2.01×	1.19×	1.34×	1.48×	1.03×	1.21×
	Eagle 3	2.90×	1.74×	2.09×	2.62×	1.58×	1.73×	1.84×	1.31×	1.52×
Llama-70B vs Llama-1B	CoGA-20	1.25×	1.33×	1.31×	1.09×	1.12×	1.14×	0.61×	0.64×	0.63×
	DyGA-20	1.29×	1.38×	1.36×	1.10×	1.14×	1.15×	0.62×	0.64×	0.64×
	Eagle 3	1.35×	1.34×	1.34×	1.28×	1.26×	1.28×	0.68×	0.78×	0.77×
Flan-T5-L vs Flan-T5-B	CoGA-20	2.01×	2.02×	2.00×	1.22×	1.22×	1.22×	1.47×	1.39×	1.40×
	DyGA-20	1.97×	1.95×	1.94×	1.30×	1.29×	1.29×	1.51×	1.40×	1.42×
Flan-T5-XL vs Flan-T5-B	CoGA-20	1.86×	1.82×	1.81×	0.92×	0.92×	0.92×	1.69×	1.45×	1.45×
	DyGA-20	1.86×	1.85×	1.86×	1.03×	1.00×	1.00×	1.44×	1.40×	1.41×
Q-4B vs Q-0.6B	DyGA-20	1.10×	1.05×	1.05×	1.23×	1.14×	1.15×	0.93×	0.84×	0.85×
Q-8B vs Q-0.6B	DyGA-20	1.09×	0.79×	0.80×	1.19×	0.90×	0.91×	0.91×	0.66×	0.67×

Table 1: Comparative analysis of speedup (γ_t), GPU (γ_e^{GPU}) and Total (γ_e^{Total}) energy saving factor for various SD strategies. Here, Eagle 2 and Eagle 3 employ draft models provided by the source repositories. $\gamma_e^{GPU/Total} \geq 1.0\times$ and $\gamma_t \geq 1.0\times$ indicate reduction in energy and time respectively, relative to vanilla decoding.[1] For each dataset–model combination, the best-performing value is highlighted using green. In cases where performance degradation persists even under the best configuration, the corresponding metrics are indicated using red.

greater than 1.0 indicates energy/time savings. In general, we observe that energy saving factor varies more or less linearly with walltime speedup factor, with the slope varying with the target model chosen. We also observe that larger difference in target and assistant model size results in higher energy optimization, resulting in a steeper slope. Closer proximity of the lines corresponding to Llama-8B and Vicuna-7B further validates this claim. However, this trend is somewhat less prominent if the larger target model is Llama-70B due to the Llama-1B assistant model's outputs not aligning perfectly. Also, for Llama-8B, Eagle achieves higher energy saving relative to speedup, while comparing with CoGA, and DyGA, showing that even for same model, the slope can vary across speculative decoding approaches. For Human-Eval, methods with speedup around 1× (CoGA and DyGA) generally translates in equivalent energy consumption with the vanilla (target model only) setup. However, for GSM-8k, methods with even higher speedup (around 1.25×) turns out to consume higher energy than Vanilla setup, again showing the dataset to be a influential factor deciding the interplay between energy optimization and walltime speedup.

Target vs Assistant	Method	Assistant Time (mins)	Target Time (mins)	Total Time (mins)	Total Energy (Wh)
Human-Eval					
Vicuna-7B vs Vicuna-68M	CoGA-5	03:50	20:01	25:53	120.38
	CoGA-10	05:05	19:30	26:16	126.48
	CoGA-20	06:31	19:25	27:23	125.38
GSM-8k					
Vicuna-13B vs Vicuna-68M	CoGA-5	07:03	45:16	1:12:54	365.47
	CoGA-10	09:14	44:02	1:11:55	360.49
	CoGA-20	11:09	40:45	1:12:46	357.97
CNN-DM					
Vicuna-13B vs Vicuna-68M	CoGA-5	03:18	25:00	42:27	212.63
	CoGA-10	03:45	23:33	41:11	202.32
	CoGA-20	04:12	23:11	41:20	196.58

Table 2: Assistant and target model run time in CoGA on Vicuna models. We see instances where a setup with lesser total time (colored in red) ends up in higher total energy consumption than another setup with higher total time (colored in blue).

[1] The results for Qwen3-4B and Qwen3-8B are reported using only DyGA-20 as an implementation of CoGA-x is not available for Qwen3 models in the HuggingFace framework.

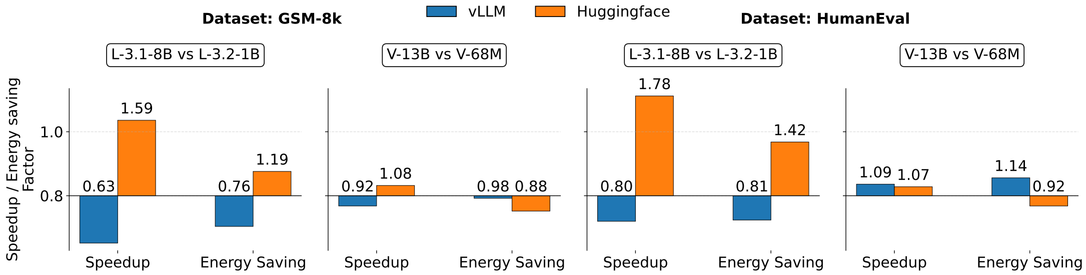

Figure 3: Comparison of speedup (γ_t) and energy savings (γ_e^{Total}) achieved by speculative decoding across different implementational platforms – HuggingFace and vLLM evaluated on the GSM-8K and HUMAN-EVAL. HuggingFace consistently outperforms vLLM in both metrics across almost all configurations.

3.2 Effect of Individual Model Runtime

In this section we separately analyze runtime and energy contributions of the target and assistant models. Intuitively, the much larger target model accounts for the majority of the total energy consumption in the SD setup. Thus, we examine how the runtimes of both models influence time and energy.

Table 2 presents the assistant run time, target run time, and total run time separately. The total energy consumption of both models is also shown.We see several instances where a setup having *lower total time* consumes *higher total energy* compared to another setup. For instance, COGA-20 setup often takes more total time but lower total energy than COGA-10 setup, primarily due to its reduced target run time (underlined in Table 2).

Overhead. We also notice that an important factor that influences energy saving in SD is the *presence of overheads*, i.e., additional computation and orchestration costs arising beyond direct token generation. These overheads primarily involve CPU and memory activity (i.e. managing cache/memory transfers, token verification, control flow logic), that, unlike GPU-bound computations, yield lower throughput and contribute disproportionately to energy consumption. As a result, configurations that offer marginal speedups may end up with higher energy usage than vanilla decoding. Our analysis reveals that these overheads can contribute approximately 6–10% of total runtime in certain configurations. For instance, in the HUMAN-EVAL dataset with VICUNA-7B and COGA-10, the combined assistant and target runtimes sum to ≈24.5 minutes, while the total measured time is ≈26.2 minutes, indicating ≈1.7 minutes of overheads.

3.3 Trends across Implementation Platforms

Figure 3 reports the runtime speedup and energy savings obtained via SD on HuggingFace and vLLM backends. Across all configurations, HuggingFace consistently outperforms vLLM in both metrics.

This disparity is primarily attributable to differences in quantization strategy: HuggingFace employs NF4 4-bit quantization for target models, whereas vLLM uses GPTQ 4-bit quantization. The superior performance of HuggingFace suggests that NF4 better preserves SD efficiency by lowering verification overhead and improving compute–memory trade-offs. In contrast, GPTQ exacerbates speculative overheads, limiting achievable speedup and energy savings, indicating that the sustainability gains of speculative decoding are sensitive to backend design.

4 Concluding Discussions

This study is the first attempt towards benchmarking energy-consumption of various SD strategies under diverse model and task settings. Our primary takeaways are as follows: (1) Lower inference time does not always correlate with proportional energy savings. Vanilla autoregressive decoding, despite higher latency, can sometimes be more energy-efficient than SD approaches (COGA and DYGA). (2) Larger target-assistant model size gap generally results in better energy optimization. (3) Dataset characteristics play a critical role in energy reduction (4) Correlation between runtime speedup and energy savings is affected by both the model architecture and decoding strategy; e.g., LLAMA models are more suited for energy savings than VICUNA models.

In summary, our findings highlight that speculative decoding is a promising approach for energy-efficient inference. But it is not a silver bullet, and we conclude by urging the community to consider the several factors needed to achieve energy-efficient inference in SD setup.

Limitations

While our study provides valuable insights into the energy characteristics of speculative decoding strategies, several limitations must be acknowledged:

- **Hardware Constraints:** All experiments were conducted on a fixed set of hardware configurations - specifically, NVIDIA A5000 and A6000 GPUs in a controlled local server environment. While this ensured consistency across measurements, it limits the generalizability of our results to other deployment environments, such as edge devices, multi-GPU clusters, or power-optimized cloud instances.
- **Scope of batch size:** Our evaluation primarily focuses on SD strategies with a batch size of one, a setting commonly adopted in academic research. However, in real-world deployment scenarios, the impact of larger batch sizes becomes critical, which may limit the generalizability of our findings.

Future work addressing these limitations can further refine our understanding of energy-efficient inference and contribute to the development of more generalizable and sustainable LLM deployment strategies.

Ethical Considerations

One of the main ethical issues was the substantial energy consumption and carbon emissions generated by our experimente. We performed inferences over 3 datasets in several speculative decoding configurations, necessitating multiple repetitions of the inferences, along with several pilot experiments to finalize the experimental setup. This led to an approx total energy consumption of 1000 kWh. To reduce our environmental impact, we limited our experiments to only 256 test examples sampled from the datasets. We hope that the insights from this study will lead the community towards a much larger reduction of the energy consuption of LLMs.

References

Lucía Bouza, Aurélie Bugeau, and Loïc Lannelongue. 2023. How to estimate carbon footprint when training deep learning models? a guide and review. *Environmental Research Communications*, 5(11):115014.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, and Tri Dao. 2023. Medusa: Simple framework for accelerating llm generation with multiple decoding heads. *arXiv preprint*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms, 2023. *URL https://arxiv. org/abs/2305.14314*, 2.

Hugging Face. 2025. Transformers documentation. https://huggingface.co/docs/transformers/en/main_classes/text_generation. Accessed: 2025-05-10.

Mathilde Jay, Vladimir Ostapenco, Laurent Lefèvre, Denis Trystram, Anne-Cécile Orgerie, and Benjamin Fichel. 2023. An experimental comparison of software-based power meters: focus on cpu and gpu. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 106–118. IEEE.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024a. Eagle-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28.

Soham Poddar, Paramita Koley, Janardan Misra, Niloy Ganguly, and Saptarshi Ghosh. 2025. Towards sustainable nlp: Insights from benchmarking inference energy in large language models. *arXiv preprint arXiv:2502.05610*.

Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. 2021. Codecarbon: estimate and track carbon emissions from machine learning computing. *Cited on*, 20.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813.

Appendix

A Sample prompts

For sample prompts from each dataset, refer to Table 4, 5 and 6.

B Model configurations

Refer to Table 3 for models used and their aliases.

Model	Alias
double7/vicuna-68m	VICUNA-68M / V-68M
lmsys/vicuna-7b-v1.3	VICUNA-7B / V-7B
lmsys/vicuna-13b-v1.3	VICUNA-13B / V-13B
meta-llama/Llama-3.2-1B-Instruct	LLAMA-1B
meta-llama/Llama-3.1-8B-Instruct	LLAMA-8B
meta-llama/Llama-3.3-70B-Instruct	LLAMA-70B
google/flan-t5-base	FLAN-T5-B
google/flan-t5-large	FLAN-T5-L
google/flan-t5-xl	FLAN-T5-XL
Qwen/Qwen3-0.6B	QWEN3-0.6B / Q-0.6B
Qwen/Qwen3-4B	QWEN3-4B / Q-4B
Qwen/Qwen3-8B	QWEN3-8B / Q-8B

Table 3: Model names and their alias

C Speculative Decoding strategies

In the speculative decoding framework, COGA-x and DYGA-x are widely used variations for assistant generation. In COGA-x, the assistant model generates x number of tokens in each iteration. Whereas, in DYGA-x, the assistant model generates x tokens initially. After the target verification phase, if all the assistant tokens are accepted (by the target model), then in the next iteration, the assistant model generates $x + 2$ tokens; otherwise, $x - 1$ tokens are generated in the next iteration. For COGA-x, the value of x was set to $5, 10$ and 20 in our experiments.

MEDUSA (Cai et al., 2023) introduces an alternative design to mitigate the sequential bottleneck of autoregressive decoding. Instead of relying on a separate draft model, MEDUSA augments the target model with multiple decoding heads that predict several future tokens in parallel. These predictions are verified simultaneously using a tree-based attention mechanism, substantially reducing inference latency. The framework provides two fine-tuning modes: MEDUSA-1, which fine-tunes only the added heads for lossless acceleration, and MEDUSA-2, which jointly fine-tunes both the heads and the backbone for higher gains. By incorporating a typical acceptance scheme and an optional self-distillation procedure for data generation, MEDUSA achieves $2.3 - 2.8\times$ speedups on models like Vicuna and Zephyr with negligible quality degradation - demonstrating its simplicity, adaptability, and effectiveness in modern LLM systems.

EAGLE-2 (Li et al., 2024a) builds on standard speculative decoding by introducing a context-aware dynamic draft tree for generating dynamic drafts. , which adapts draft token generation based on the confidence scores of a smaller draft model. This approach allows EAGLE-2 to increase the number of accepted tokens per cycle, leading to faster, lossless inference. Unlike prior methods using fixed tree structures (e.g., EAGLE (Li et al., 2024b), Medusa (Cai et al., 2023)), EAGLE-2 dynamically adjusts the tree shape without requiring additional training, achieving state-of-the-art speedups while preserving the original LLM's output distribution.

EAGLE-3 (Li et al., 2025) further eliminates the feature prediction constraint used in earlier versions like EAGLE and EAGLE-2. It introduces direct token prediction, allowing the draft model to integrate multi-layer fused features from the target model, significantly enhancing its expressiveness and scalability.

D Hardware and Energy metrics

We use the popular Code Carbon (Schmidt et al., 2021) package to measure the energy consumed in different experiments. Jay et al. (2023) and Bouza et al. (2023) demonstrated the suitability and accuracy of CodeCarbon across various software-based power meter setups. This package measure the GPU-power usage using pynvml and CPU-power using Intel RAPL files every $\mathcal{X}$ seconds, and integrates it over time, which we set as $1secs$. Codecarbon also adds an estimate of the RAM-power being used depending on the RAM size. Power Usage Effectiveness (PUE) is set to 1.0 as all experiments are performed on the same server, indicating the actual energy usage may be different than reported. During inference, we provide test samples sequentially at batch size 1 to the LLM and report the average energy usage per $1K$ tokens in Watt-hour (Wh).

In our study, we omit the amount of carbon emission because we perform all the experiments in a single region where the carbon intensity is fixed and therefore, energy consumed is closely related with the amount of CO_2 emission. Furthermore, the CO_2 emission strongly varies depending on the region and the type of electricity source. Thus, we prefer to report the total energy consumed instead of the amount of CO_2 emission.

E Additional metrics

We employ the following additional metrics to evaluate speculative decoding approaches in Table 8:

- **Total energy per $1K$ tokens:** The total energy consumed by target and assistant model measured in Watt (Wh) to generate $1K$ tokens.
- **Total time per $1K$ tokens:** Total time in minutes required by both to generate $1K$ tokens.

F Extended Results of COGA-$\mathcal{X}$

Table 7 presents an extended evaluation of the Constant Generation by Assistant (COGA-x) strategy by varying the draft length $x \in 5, 10, 20$ across all target–assistant model pairs and datasets. These results provide deeper insight into how the choice of draft length influences runtime speedup (γ_t) and energy savings (γ_e).

Effect of draft length: Across most model families, increasing the draft length generally improves speedup, as larger drafts allow the target model to verify more tokens per iteration. However, this improvement does not always translate into proportional energy savings. In several cases, particularly for VICUNA-7B and VICUNA-13B, changes in x yield only marginal variation in $\gamma_e^{GPU/Energy}$, with values often remaining close to or below unity despite moderate gains in γ_t. This suggests that verification and orchestration overheads dominate the energy profile for these models, limiting the benefit of longer drafts.

Model-family trends: LLAMA-8B exhibits a consistent increase in both speedup and energy savings as x increases, achieving up to $1.42\times$ total energy savings on HUMAN-EVAL with DYGA-20. In contrast, LLAMA-70B shows modest gains in speedup but persistent energy degradation on CNN-DM for all values of x, indicating that larger target models are more sensitive to dataset characteristics and overhead costs. Encoder-decoder models (FLAN-T5-L and FLAN-T5-XL) benefit most from larger drafts, where COGA-20 consistently achieves the highest speedup and energy savings across datasets.

Dataset sensitivity: The impact of x is strongly dataset-dependent. HUMAN-EVAL generally benefits from larger drafts, while GSM-8K and CNN-DM often exhibit diminishing or negative energy returns as x increases. This highlights that longer drafts can increase assistant computation and verification overhead without sufficient token acceptance to offset the added energy cost.

Overall, Table 7 demonstrates that while increasing the COGA draft length can improve runtime speedup, energy efficiency gains are neither monotonic nor guaranteed. Optimal values of x depend jointly on the target–assistant model pair and the dataset, reinforcing the need for adaptive or dataset-aware draft-length selection when energy efficiency is a primary objective.

G Acknowledgements

The authors thank the anonymous reviewers whose suggestions helped to improve the work. The research is partially supported by a research grant from Accenture Corporation. Paramita Koley is supported by SERB NPDF Fellowship.

<table>
<tr><th colspan="2">GSM-8K</th></tr>
<tr><th>model</th><th>prompt</th></tr>
<tr><td>LLAMA-3</td><td><|begin_of_text|><|start_header_id|>system<|end_header_id|>
Cutting Knowledge Date: December 2023 Today Date: 11 May 2025
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.<|eot_id|><|start_header_id|>user<|end_header_id|>
Solve the math problem and give a numeric solution Problem: Carol and Jennifer are sisters from Los Angeles who love collecting signatures from celebrities. During their summer break from school, the sisters spend every afternoon collecting signatures. After five weeks, Carol and Jennifer compare their autograph books, counting up the number of signatures each sister has collected. Carol has 20 signatures in her book, and Jennifer has 44. The sisters have three more weeks of summer vacation, and they decide they want to reach 100 signatures between them by the end of the summer. How many signatures do the sisters need to collect to reach their goal?<|eot_id|><|start_header_id|>assistant<|end_header_id|></td></tr>
<tr><td>VICUNA</td><td>A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. USER: Solve the math problem and give a numeric solution Problem: Carol and Jennifer are sisters from Los Angeles who love collecting signatures from celebrities. During their summer break from school, the sisters spend every afternoon collecting signatures. After five weeks, Carol and Jennifer compare their autograph books, counting up the number of signatures each sister has collected. Carol has 20 signatures in her book, and Jennifer has 44. The sisters have three more weeks of summer vacation, and they decide they want to reach 100 signatures between them by the end of the summer. How many signatures do the sisters need to collect to reach their goal? ASSISTANT:</td></tr>
<tr><td>QWEN3</td><td>$< |im_start| >$user You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information. Solve the math problem and give a numeric solution Problem: Carol and Jennifer are sisters from Los Angeles who love collecting signatures from celebrities. During their summer break from school, the sisters spend every afternoon collecting signatures. After five weeks, Carol and Jennifer compare their autograph books, counting up the number of signatures each sister has collected. Carol has 20 signatures in her book, and Jennifer has 44. The sisters have three more weeks of summer vacation, and they decide they want to reach 100 signatures between them by the end of the summer. How many signatures do the sisters need to collect to reach their goal?$< |im_end| > < |im_start| >$assistant</td></tr>
</table>

Table 4: Prompts for the GSM-8K dataset

<table>
<tr><th colspan="2">HUMAN-EVAL</th></tr>
<tr><th>model</th><th>prompt</th></tr>
<tr><td>LLAMA-3</td><td><|begin_of_text|><|start_header_id|>system<|end_header_id|>
Cutting Knowledge Date: December 2023 Today Date: 10 May 2025
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.<|eot_id|><|start_header_id|>user<|end_header_id|>
Complete the function(s) based on the given function prototype and the docstring:
def can_arrange(arr): """"""Create a function which returns the largest index of an element which is not greater than or equal to the element immediately preceding it. If no such element exists then return -1. The given array will not contain duplicate values.
Examples: can_arrange([1,2,4,3,5]) = 3 can_arrange([1,2,3]) = -1
""""""<|eot_id|><|start_header_id|>assistant<|end_header_id|></td></tr>
<tr><td>VICUNA</td><td>A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. USER: Complete the function(s) based on the given function prototype and the docstring:
def can_arrange(arr): """"""Create a function which returns the largest index of an element which is not greater than or equal to the element immediately preceding it. If no such element exists then return -1. The given array will not contain duplicate values.
Examples: can_arrange([1,2,4,3,5]) = 3 can_arrange([1,2,3]) = -1 """""" ASSISTANT:</td></tr>
<tr><td>QWEN3</td><td>$< |im_start| >$user You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information. Complete the function(s) based on the given function prototype and the docstring:
def can_arrange(arr): """"""Create a function which returns the largest index of an element which is not greater than or equal to the element immediately preceding it. If no such element exists then return -1. The given array will not contain duplicate values.
Examples: can_arrange([1,2,4,3,5]) = 3 can_arrange([1,2,3]) = -1 """"""$< |im_end| > < |im_start| >$assistant</td></tr>
</table>

Table 5: Prompts for the HUMAN-EVAL dataset

<table>
<tr><th colspan="2">CNN-DM</th></tr>
<tr><th>model</th><th>prompt</th></tr>
<tr><td>LLAMA-3</td><td><|begin_of_text|><|start_header_id|>system<|end_header_id|>
Cutting Knowledge Date: December 2023 Today Date: 11 May 2025
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don’t know the answer to a question, please don’t share false information.<|eot_id|><|start_header_id|>user<|end_header_id|>
Summarize the following news article in about 50 words: ARTICLE: Down Augusta way they say the azaleas are in full bloom, which is more than can be said for England’s Justin Rose. A bruising Florida swing last month saw the Englishman fall outside the world’s top 10. For a player who has been virtually a fixture in the top five for the last three years it was certainly a dent to the ego, with the Masters now just around the corner. Rose’s solution to his miserable form — three missed cuts and a 55th-place finish at the Cadillac Championship in four PGA Tour starts — was the time-honoured one. For the past two weeks, the 34-year-old has spent long hours on the practice ground. Justin Rose hit 17 out of 18 greens in regulation and signed for a 69 at the Shell Houston Open . In the first round of the Shell Houston Open on Thursday there were encouraging signs his decline will prove temporary. Rose hit 17 out of 18 greens in regulation and signed for a 69, the same score as his playing partner, the ever- consistent Jordan Spieth. ‘It’s certainly a welcome return to the sixties, for it had been a while,’ said Rose, smiling. On a day when American Scott Piercy went round in 63 and Phil Mickelson enjoyed his best round in months with a 66, it was hardly surprising the only reporter waiting to talk to Rose was this one. But under the radar is never a bad place to be going to the Masters. The boom and bust years that characterised the first half of Rose’s career meant there was never going to be any feelings of panic following his unusually poor run in the Sunshine State. ‘There’s no doubt I lost my game there but the Florida swing can be unforgiving if you’re slightly off,’ he said. ‘Over the past two weeks I feel like I’ve done some good work and whether I finish well or not here I feel like I’m going in the right direction again. ‘Basically I was getting ahead of the ball at impact, and shots were going left or right, the irons were not solid and the new putter was not working. So we’ve corrected the faults and I’ve gone back to the old putter.’ Phil Mickelson enjoyed his best round in months with a 66 on Thursday . Does he pay much attention to the world rankings? ‘You notice, for sure,’ he said. ‘I’m very proud of the fact I’ve been in the world’s top five for practically the whole of the last three years. It’s a nice ego thing, so by the end of the year I’m hoping there won’t be any slippage. ‘But right now, I’ve got to focus on my game in the knowledge that the rankings change fast when you’re playing well. ...... ‘The Masters has probably been less on my mind this year because I am trying to find some form,’ he admitted. ‘But I think the fact I’ve had a number of great rounds there will always stand me in good stead. Regardless of what happens here, I feel comfortable on that course and know I can manage my game even if it’s not 100 per cent. You draw off the energy of the place.’ Mickelson has certainly done that over the years and perhaps the veteran lefty, a three-time Masters champion, is gearing himself up for another run at the green jacket. ‘It was a good start to the tournament and now I’m looking for three more good rounds,’ he said. ’This is a big week for me. I felt the game was close last week. The only thing missing was chipping and short game.’ Paul Casey, like Mickelson another former winner of this event, celebrated his last-gasp Masters invitation with a fine round of 68 notable for two eagle threes. In the afternoon wave, Padraig Harrington and Lee Westwood both made good starts to play their first six holes in two under.<|eot_id|><|start_header_id|>assistant<|end_header_id|></td></tr>
<tr><td>VICUNA</td><td>A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user’s questions. USER: Summarize the following news article in about 50 words: ARTICLE: Down Augusta way they say the azaleas are in full bloom, which is more than can be said for England’s Justin Rose. A bruising Florida swing last month saw the Englishman fall outside the world’s top 10. For a player who has been virtually a fixture in the top five for the last three years it was certainly a dent to the ego, with the Masters now just around the corner. Rose’s solution to his miserable form — three missed cuts and a 55th-place finish at the Cadillac Championship in four PGA Tour starts — was the time-honoured one. For the past two weeks, the 34-year-old has spent long hours on the practice ground. Justin Rose hit 17 out of 18 greens in regulation and signed for a 69 at the Shell Houston Open . In the first round of the Shell Houston Open on Thursday there were encouraging signs his decline will prove temporary. Rose hit 17 out of 18 greens in regulation and signed for a 69, the same score as his playing partner, the ever-consistent Jordan Spieth. ‘It’s certainly a welcome return to the sixties, for it had been a while,’ said Rose, smiling. On a day when American Scott Piercy went round in 63 and Phil Mickelson enjoyed his best round in months with a 66, it was hardly surprising the only reporter waiting to talk to Rose was this one. But under the radar is never a bad place to be going to the Masters. The boom and bust years that characterised the first half of Rose’s career meant there was never going to be any feelings of panic following his unusually poor run in the Sunshine State. ‘There’s no doubt I lost my game there but the Florida swing can be unforgiving if you’re slightly off,’ he said. ‘Over the past two weeks I feel like I’ve done some good work and whether I finish well or not here I feel like I’m going in the right direction again. ‘Basically I was getting ahead of the ball at impact, and shots were going left or right, the irons were not solid and the new putter was not working. So we’ve corrected the faults and I’ve gone back to the old putter.’ Phil Mickelson enjoyed his best round in months with a 66 on Thursday . Does he pay much attention to the world rankings? ‘You notice, for sure,’ he said. ‘I’m very proud of the fact I’ve been in the world’s top five for practically the whole of the last three years. It’s a nice ego thing, so by the end of the year I’m hoping there won’t be any slippage. ‘But right now, I’ve got to focus on my game in the knowledge that the rankings change fast when you’re playing well. .... . ‘The Masters has probably been less on my mind this year because I am trying to find some form,’ he admitted. ‘But I think the fact I’ve had a number of great rounds there will always stand me in good stead. Regardless of what happens here, I feel comfortable on that course and know I can manage my game even if it’s not 100 per cent. You draw off the energy of the place.’ Mickelson has certainly done that over the years and perhaps the veteran lefty, a three-time Masters champion, is gearing himself up for another run at the green jacket. ‘It was a good start to the tournament and now I’m looking for three more good rounds,’ he said. ’This is a big week for me. I felt the game was close last week. The only thing missing was chipping and short game.’ Paul Casey, like Mickelson another former winner of this event, celebrated his last-gasp Masters invitation with a fine round of 68 notable for two eagle threes. In the afternoon wave, Padraig Harrington and Lee Westwood both made good starts to play their first six holes in two under. ASSISTANT:</td></tr>
<tr><td>QWEN3</td><td>$< |im\_start| >$ user You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don’t know the answer to a question, please don’t share false information. Summarize the following news article in about 50 words: ARTICLE: Down Augusta way they say the azaleas are in full bloom, which is more than can be said for England’s Justin Rose. A bruising Florida swing last month saw the Englishman fall outside the world’s top 10. For a player who has been virtually a fixture in the top five for the last three years it was certainly a dent to the ego, with the Masters now just around the corner. Rose’s solution to his miserable form — three missed cuts and a 55th-place finish at the Cadillac Championship in four PGA Tour starts — was the time-honoured one. For the past two weeks, the 34-year-old has spent long hours on the practice ground. Justin Rose hit 17 out of 18 greens in regulation and signed for a 69 at the Shell Houston Open . In the first round of the Shell Houston Open on Thursday there were encouraging signs his decline will prove temporary. Rose hit 17 out of 18 greens in regulation and signed for a 69, the same score as his playing partner, the ever- consistent Jordan Spieth. ‘It’s certainly a welcome return to the sixties, for it had been a while,’ said Rose, smiling. On a day when American Scott Piercy went round in 63 and Phil Mickelson enjoyed his best round in months with a 66, it was hardly surprising the only reporter waiting to talk to Rose was this one. But under the radar is never a bad place to be going to the Masters. The boom and bust years that characterised the first half of Rose’s career meant there was never going to be any feelings of panic following his unusually poor run in the Sunshine State. ‘There’s no doubt I lost my game there but the Florida swing can be unforgiving if you’re slightly off,’ he said. ‘Over the past two weeks I feel like I’ve done some good work and whether I finish well or not here I feel like I’m going in the right direction again. ‘Basically I was getting ahead of the ball at impact, and shots were going left or right, the irons were not solid and the new putter was not working. So we’ve corrected the faults and I’ve gone back to the old putter.’ Phil Mickelson enjoyed his best round in months with a 66 on Thursday . Does he pay much attention to the world rankings? ‘You notice, for sure,’ he said. ‘I’m very proud of the fact I’ve been in the world’s top five for practically the whole of the last three years. It’s a nice ego thing, so by the end of the year I’m hoping there won’t be any slippage. ‘But right now, I’ve got to focus on my game in the knowledge that the rankings change fast when you’re playing well. .... ‘The Masters has probably been less on my mind this year because I am trying to find some form,’ he admitted. ‘But I think the fact I’ve had a number of great rounds there will always stand me in good stead. Regardless of what happens here, I feel comfortable on that course and know I can manage my game even if it’s not 100 per cent. You draw off the energy of the place.’ Mickelson has certainly done that over the years and perhaps the veteran lefty, a three-time Masters champion, is gearing himself up for another run at the green jacket. ‘It was a good start to the tournament and now I’m looking for three more good rounds,’ he said. ’This is a big week for me. I felt the game was close last week. The only thing missing was chipping and short game.’ Paul Casey, like Mickelson another former winner of this event, celebrated his last-gasp Masters invitation with a fine round of 68 notable for two eagle threes. In the afternoon wave, Padraig Harrington and Lee Westwood both made good starts to play their first six holes in two under.$< |im_end| > < |im_start| >$assistant</td></tr>
</table>

Table 6: Prompts for the CNN-DM dataset

		Human-Eval			GSM-8k			CNN-DM		
		Speedup	Energy Saving Factor		Speedup	Energy Saving Factor		Speedup	Energy Saving Factor	
Target vs Assistant	SD Method	γ_t	γ_e^{GPU}	γ_e^{Total}	γ_t	γ_e^{GPU}	γ_e^{Total}	γ_t	γ_e^{GPU}	γ_e^{Total}
Vicuna-7B vs Vicuna-68M	CoGA-5	1.44×	0.81×	0.97×	1.31×	0.75×	0.85×	1.38×	0.84×	0.96×
	CoGA-10	1.45×	0.83×	0.99×	1.28×	0.75×	0.81×	1.34×	0.83×	0.95×
	CoGA-20	1.42×	0.83×	0.99×	1.19×	0.72×	0.77×	1.44×	0.89×	1.03×
Vicuna-13B vs Vicuna-68M	CoGA-5	1.05×	0.89×	0.91×	1.08×	0.73×	0.85×	0.99×	0.86×	0.92×
	CoGA-10	1.10×	0.95×	0.93×	1.11×	0.76×	0.89×	1.02×	0.89×	0.96×
	CoGA-20	1.07×	0.93×	0.92×	1.08×	0.75×	0.88×	1.02×	0.89×	0.96×
Llama-8B vs Llama-1B	CoGA-5	1.59×	1.06×	1.25×	1.42×	0.96×	1.01×	1.10×	0.86×	0.92×
	CoGA-10	1.75×	1.20×	1.38×	1.56×	1.06×	1.11×	1.05×	0.83×	0.89×
	CoGA-20	1.78×	1.23×	1.42×	1.59×	1.10×	1.19×	0.96×	0.79×	0.83×
Llama-70B vs Llama-1B	CoGA-5	0.92×	0.94×	0.95×	0.85×	0.87×	0.88×	0.62×	0.64×	0.63×
	CoGA-10	1.13×	1.18×	1.17×	1.02×	1.04×	1.06×	0.63×	0.65×	0.65×
	CoGA-20	1.25×	1.33×	1.31×	1.09×	1.12×	1.14×	0.61×	0.64×	0.63×
Flan-T5-L vs Flan-T5-B	CoGA-5	1.73×	1.74×	1.70×	1.32×	1.32×	1.32×	1.38×	1.32×	1.32×
	CoGA-10	1.90×	1.79×	1.80×	1.38×	1.36×	1.38×	1.51×	1.40×	1.42×
	CoGA-20	2.01×	2.02×	2.00×	1.22×	1.22×	1.22×	1.47×	1.39×	1.40×
Flan-T5-XL vs Flan-T5-B	CoGA-5	1.64×	1.62×	1.60×	1.06×	1.02×	1.01×	1.43×	1.32×	1.33×
	CoGA-10	1.71×	1.67×	1.70×	1.03×	1.01×	1.00×	1.43×	1.37×	1.38×
	CoGA-20	1.86×	1.82×	1.81×	0.92×	0.92×	0.92×	1.69×	1.45×	1.45×

Table 7: Comparative analysis of speedup (γ_t), GPU (γ_e^{GPU}) and Total (γ_e^{Total}) energy saving factor across multiple CoGA-x speculative decoding strategies for varying values of $x \in 5, 10, 20$.

		Human-Eval		GSM-8k		CNN-DM	
Target vs Draft	Method	GPU energy (Wh/1K tokens)	Total energy (Wh/1K tokens)	GPU energy (Wh/1K tokens)	Total energy (Wh/1K tokens)	GPU energy (Wh/1K tokens)	Total energy (Wh/1K tokens)
Vicuna-7B vs Vicuna-68M	CoGA-5	2.24	3.03	2.37	3.13	2.74	3.63
	CoGA-10	2.19	2.98	2.37	3.30	2.76	3.65
	CoGA-20	2.19	2.98	2.46	3.46	2.59	3.38
	DyGA-20	2.25	3.07	2.44	3.18	2.8	3.68
	Eagle 2	1.15	1.55	1.30	1.66	1.97	2.51
Vicuna-13B vs Vicuna-68M	CoGA-5	3.83	5.02	4.00	5.27	4.55	6.12
	CoGA-10	3.60	4.87	3.85	5.02	4.39	5.9
	CoGA-20	3.65	4.93	3.90	5.07	4.38	5.88
	DyGA-20	3.86	4.94	4.13	5.55	4.62	6.20
	Eagle 2	1.91	2.49	2.01	2.75	3.04	3.85
	Eagle 3	1.42	1.81	1.57	2.14	2.11	2.69
Llama-8B vs Llama-1B	CoGA-5	2.03	2.72	2.25	3.22	3.11	4.31
	CoGA-10	1.80	2.46	2.03	2.93	3.20	4.45
	CoGA-20	1.75	2.38	1.95	2.74	3.40	4.76
	DyGA-20	1.71	2.35	2.02	2.92	3.34	4.65
	Eagle 2	1.60	2.08	1.81	2.43	2.58	3.28
	Eagle 3	1.24	1.62	1.36	1.88	2.03	2.60
Llama-70B vs Llama-1B	CoGA-5	12.33	14.84	13.62	16.74	23.66	28.95
	CoGA-10	9.87	11.95	11.33	13.92	23.1	28.16
	CoGA-20	8.78	10.71	10.56	13.02	23.53	28.71
	DyGA-20	8.44	10.29	10.36	12.92	23.34	28.53
	Eagle 3	8.68	10.46	9.40	11.53	19.28	23.71

Table 8: Comparative evaluation of inference energy consumption in various speculative decoding strategies for four standard target models on Human-Eval, GSM-8k and CNN-DM datasets. Energy is measured in Watt-Hour per 1000 tokens (Wh/1K). In each case, corresponding draft model is mentioned. However, Eagle 2 and Eagle 3 employ draft model provided by the source repositories.