Mapping the Efficiency Landscape of Small Language Models

Fabian Reichwald[1] , **Lukas Schiesser**[1] , **Christiane Plociennik**[1] , **Leonhard Kunz**[2] , **Simon Pukrop**[1] , **Martin Ruskowski**[1,2] , **Oliver Thomas**[1,3]
[1]German Research Center for Artificial Intelligence (DFKI)
[2]RPTU University Kaiserslautern-Landau
[3]Osnabrück University
{fabian.reichwald, lukas.schiesser, christiane.plociennik, simon.pukrop, martin.ruskowski, oliver.thomas}@dfki.de, leonhard.kunz@rptu.de

Abstract

Large language models (LLMs) dominate both everyday and specialized applications, but their high computational demand, energy consumption, and privacy risks are increasingly critiqued. Small language models (SLMs) mitigate these drawbacks and are gaining momentum in scenarios where full LLM capabilities are not required, such as agents, industrial systems, or edge devices. Nevertheless, a systematic comparison of model capabilities, energy usage, and scaling behavior has not been conducted yet. We evaluate 70+ SLMs from 2023–2025 on five task-specific benchmarks and compare them with two popular LLMs, revealing key trade-offs between energy, performance, and model selection. Our findings challenge common assumptions: First, smaller models are not automatically more efficient, and energy increases do not guarantee performance gains. Second, newer SLMs show clear improvements in performance–energy trade-offs, though the progress begins to plateau. Last, the efficiency landscape forms a clear Pareto frontier: initial energy increases yield substantial gains, but the last percentage points of performance need orders of magnitude more energy. These results highlight diminishing returns of scaling and emphasize the need for informed, task-aware model selection rather than size-driven choices.

1 Introduction

In the last years, large language models (LLMs) have become ubiquitous. They have evolved to a state where they can understand and produce coherent natural language across diverse domains, and are able to perform a wide range of tasks that could previously only be accomplished by humans, such as summarizing technical texts [Käser *et al.*, 2025] or engaging in complex conversations [Maharana *et al.*, 2024]. LLMs are at the core of the emerging field of *agentic AI*, which encompasses AI applications performing complex tasks with a certain degree of autonomy [Acharya *et al.*, 2025; Mohammadi *et al.*, 2025]. LLMs are also increasingly being used in industrial settings [Raza *et al.*, 2025]. Companies use chatbots, for example, as an interface to customers, e.g. on their webpage [Szafarski *et al.*, 2025; Bhattacharyya, 2024].

However, the substantial resource consumption of LLMs [Berthelot *et al.*, 2025; Plociennik *et al.*, 2025; Plociennik *et al.*, 2026], in particular energy demand [Ji and Jiang, 2026], scarcity of compute power, especially GPUs [Strati *et al.*, 2024], fear of lock-in with cloud servers in different parts of the world [Wang *et al.*, 2025; Wang, 2025], along with privacy and sustainability concerns [Raza *et al.*, 2025], pose significant entrance barriers for companies to adopt LLMs for their business. Moreover, the broad knowledge encoded in LLMs is often not needed for industrial applications, which usually require domain- and company-specific information [Wang *et al.*, 2025]. Hence, in this paper we argue that small, locally deployable models can be an alternative to LLMs for most common industrial tasks. Since there is no consensus on what should be considered a *small language model (SLM)* yet [Wang *et al.*, 2025], we adopt a pragmatically-oriented definition: We consider every model with at most 15B parameters an SLM since those are the models that can run on consumer hardware.

Adopting SLMs in industry requires longer-term commitment and investment security. Companies must be able to choose a model that performs reasonably well over time with limited resources. This applies especially to SMEs [Bourdin *et al.*, 2024]. Hence, the question is whether SLM development has proceeded to a stable state for informed choices of models, and which benefits they offer over LLMs.

This paper aims to provide insights into the interactions between size, performance, and energy demand of SLMs. It therefore investigates the correlation between model parameter count, inference energy consumption and model performance for 75 SLMs over 5 different benchmark tasks to determine task-specific energy efficiency. The results are set into perspective with 2 LLMs for reference. In doing so, we particularly shed light on possible trade-offs between task performance and energy consumption. By including SLMs in a timeframe from 2023 to 2025, we also include a historical perspective on the development of operational energy efficiency. This question is particularly interesting for application developers who need to select a model and are still looking for the right time to commit to the effort of customiz-

ing a self-hosted SLM for lower operational costs. With this temporal analysis, we also analyze trends in the development of SLM performance in relation to the reference LLMs.

The main contributions of this paper are:

- It explores the relations between model size, performance, and energy consumption of SLMs.
- It presents the development of SLMs over time – from 2023 to 2025.
- It benchmarks a large number of SLMs (75) on 5 different tasks. The results can be viewed as a first guideline for task-specific model selection.

The results are presented comprehensively in Section 4, structured by the following research questions (RQs):

(RQ1) What relationships and trade-offs exist between model size, energy consumption and task performance in SLMs?

(RQ2) How have efficiency and task performance evolved across model sizes and over recent years, and how do they compare to reference LLMs?

The paper is structured as follows: Section 2 discusses related work. Section 3 introduces the methodology. Section 4 presents and discusses the results. Section 5 considers limitations, while Section 6 gives some general concluding remarks.

2 Related Work

Belcak et al. [2025] argue for the use of SLMs instead of LLMs in agentic AI applications. They point out that specialized or fine-tuned SLMs are sufficiently powerful, more operationally suitable, more flexible and more economical than LLMs in many usecases. They stress that well-designed SLMs can even outperform the task performance of much larger models. However, the choice between a customized SLM and a service LLM can be a multi-faceted decision in actual system design, as Koraag et al. [2025] emphasize in their report on the development of a specific application in financial reporting. They report the challenges regarding the implementation of both options and specifically regarding the selection of a suitable SLM. Corradini et al. [2025] present a systematic literature review on recent developments in the SLM field. They investigate the geographical distribution of SLM research and architectural trends, and find that MMLU-Redux is the most frequently used SLM benchmark in the literature, followed by GSM8K.

Recent studies have shown that the influence of model size on the performance of language models is not straightforward in the sense that "bigger is always better", challenging the LLM scaling law [Korga *et al.*, 2025; Varoquaux *et al.*, 2025]. Xiao et al. [2025] introduce the concept of capability density to characterize the relationship between model performance and efficiency. They find that inference costs decrease exponentially over time for equivalent performance. Lee et al. [2025] investigate the influence of different quantization methods on the performance of different size models of the Llama family. They find that performance varies by quantization method and the nature of the task, and performance degradation does not necessarily increase with task difficulty.

Berthelot et al. [2025] find that energy usage is not evenly distributed across the language model lifecycle: Over time, the major share of energy is required for inference, not for training. Hence, it is key to investigate the energy consumption of language models (LMs) during inference, which is being done in several studies. Samsi et al. [2023] study the energy consumption per second of different size models (7B to 65B) from the Llama family and investigate the energy consumption per decoded token for the largest model (65B parameters). They find that energy consumption per second correlates positively with model size, and that energy consumption per decoded token for the 65B model is higher if more GPUs are used for inference. This is due to the fact that only 20-25% of GPU memory is being used at any given time. Argerich and Patiño-Martínez [2024] find that the energy consumption per token of a 3B parameter model is only about $2/3$ of that of a 7B model. A comparison of the total energy consumption per prompt is done by Jeanquartier et al. [2026] with two SLMs. They suggest that the energy consumption should be reported based on task outcome instead of per token. However, their study is limited regarding the number of included models. They also argue to include the application context in the sustainability assessment of the model. Maliakel et al. [2025] investigate the trade-offs of performance vs. energy across 5 models of different sizes and families and 4 different benchmarks. They find that larger models consume more energy while being more accurate, and that both model architecture and input characteristics influence efficiency. Saad-Falcon et al. [2025] introduce a metric called intelligence per watt (IPW). They argue that computing transitioned from mainframes to PCs due to efficiency improvements, and that language models are about to face a similar shift – from cloud to local inference. They benchmark 20+ local LMs in terms of accuracy and energy consumption, and find that local LMs have become 5 times more energy-efficient in the last two years. Pham et al. [2025] introduce SLM-Bench, a benchmark for SLMs. They evaluate 15 SLMs on 9 NLP tasks using 23 datasets from different domains, including metrics for energy consumption and CO2 emissions. They find that there are trade-offs among the dimensions correctness, computational efficiency, and energy consumption. Hence, accuracy gains come at the cost of a higher energy usage. These are interesting results, despite the fact that the choice of benchmarks does not include coding or function calling tasks. Furthermore, the limited model selection hinders a systematic analysis of SLMs, which differ in capabilities and behavior.

3 Method

To systemically evaluate the trade-offs between task performance, model size and energy consumption across SLMs, we conducted controlled inference experiments on a large collection of SLMs and selected LLMs. The setup allows comparison across diverse tasks and model families including the analysis of efficiency differences.

3.1 Models

In the experiments we included 75 open-source SLMs. The models either do not reason, or have reasoning disabled to allow for fair comparison. The evaluated SLMs span the time

Model family	Model	Date	BFC			GSM8K			MMLU-Redux			XSUM			BigCodeBench		
			Acc. (%)	Len.	Energy (J)	Acc. (%)	Len.	Energy (J)	Acc. (%)	Len.	Energy (J)	Similarity	Len.	Energy (J)	Acc. (%)	Len.	Energy (J)
Apertus	Apertus-8B	09-2025	67.7	65.68	628.88	64.6	212.29	1868.97	60.8	13.80	179.88	0.57	50.42	514.59	21.1	362.53	3259.44
Gemma	Gemma-1-2B	02-2024	13.4	110.15	525.87	13.1	110.83	516.19	39.1	40.12	222.57	0.47	38.66	223.06	5.3	247.23	1104.94
	Gemma-1-7B	02-2024	14.9	178.89	1949.41	3.6	191.22	1972.96	47.2	54.88	591.53	0.43	46.80	545.82	0.0	328.96	3535.54
	Gemma-2-2B	06-2024	53.8	96.61	530.30	60.7	171.84	875.28	57.1	8.54	96.67	0.57	38.77	246.54	8.4	627.58	3147.02
	Gemma-2-9B	06-2024	75.9	89.49	1117.98	82.4	157.05	1844.50	72.6	12.01	197.50	0.60	36.92	504.46	39.1	525.09	6453.02
	Gemma-3-0.270B	03-2025	0.0	284.55	989.72	6.4	211.94	737.85	22.9	3.52	70.53	0.46	53.65	236.89	1.5	166.76	592.27
	Gemma-3-1B	03-2025	15.0	82.08	468.88	46.1	319.00	1613.38	41.8	2.94	71.11	0.53	54.89	326.04	8.8	299.64	1555.56
	Gemma-3-4B	03-2025	60.8	92.25	771.79	85.8	315.48	2348.20	59.4	3.49	84.66	0.57	41.82	383.00	35.7	337.42	2739.30
	Gemma-3-12B	03-2025	77.1	93.65	1457.35	90.4	236.22	3456.53	73.9	2.05	93.57	0.60	43.52	713.92	49.8	368.97	5447.95
Granite	Granite-3.0-1B	10-2024	25.3	165.40	992.26	26.1	268.39	1569.34	28.6	22.98	181.76	0.51	89.88	561.06	6.4	355.05	2125.00
	Granite-3.0-3B	10-2024	51.5	80.20	571.89	53.6	232.79	1532.12	52.2	30.78	250.27	0.54	80.25	568.20	20.0	306.54	2015.18
	Granite-3.0-8B	10-2024	69.3	103.72	1115.26	72.6	234.74	2352.06	63.7	43.90	485.07	0.57	72.07	791.58	35.7	287.97	2980.72
	Granite-3.1-1B	12-2024	51.5	96.73	497.18	34.5	277.85	1265.97	20.2	6.64	91.84	0.54	92.16	477.41	13.5	334.99	1556.37
	Granite-3.1-2B	12-2024	71.1	74.94	424.39	67.0	253.71	1244.51	54.0	10.97	110.13	0.57	79.07	436.91	27.3	339.72	1693.46
	Granite-3.1-3B	12-2024	63.7	79.54	560.87	58.3	234.22	1482.59	52.1	11.58	134.55	0.56	102.64	705.64	24.9	285.96	1825.09
	Granite-3.1-8B	12-2024	73.5	70.64	737.40	79.9	219.59	2057.42	64.0	9.13	144.34	0.58	72.68	747.22	38.6	244.25	2344.58
	Granite-3.2-2B	02-2025	70.4	72.47	395.70	70.6	266.49	1262.15	54.0	8.21	92.60	0.57	79.18	422.70	26.1	368.28	1761.98
	Granite-3.2-8B	02-2025	73.4	74.95	783.53	81.1	219.63	2075.52	64.0	3.12	89.54	0.57	70.88	735.93	37.3	228.74	2210.31
	Granite-3.3-2B	04-2025	69.7	58.30	334.08	77.6	307.46	1470.85	54.2	15.24	126.19	0.56	79.64	433.42	28.1	379.32	1839.14
	Granite-3.3-8B	04-2025	73.8	82.40	927.86	84.0	320.23	3292.64	63.6	17.71	233.56	0.58	86.64	963.76	37.4	233.54	2474.77
	Granite-4-0.35B	10-2025	30.2	51.14	592.60	37.3	170.35	828.88	25.4	2.55	161.61	0.49	35.01	455.68	8.8	302.79	1492.85
	Granite-4-1B	10-2025	49.8	53.59	941.23	72.2	235.17	1665.99	51.1	2.04	235.54	0.53	42.96	757.45	30.1	227.61	1858.54
	Granite-4-3B	09-2025	60.0	54.17	1111.46	79.6	320.88	2453.59	67.3	2.21	287.97	0.56	47.33	900.83	39.7	346.60	2898.57
	Granite-4-7B	09-2025	60.2	68.26	1351.68	72.3	381.63	4205.20	58.8	17.88	423.59	0.56	101.42	1598.99	30.5	223.25	2816.96
Llama	Llama-2-7B	07-2023	32.1	287.32	2542.19	26.1	221.48	1883.82	42.3	101.42	904.93	0.57	86.44	850.97	7.2	618.32	5196.25
	Llama-2-13B	07-2023	37.3	282.45	4379.21	36.3	236.67	3526.94	47.9	107.97	1572.78	0.55	84.71	1340.48	8.7	707.12	11453.17
	Llama-3-8B	04-2024	66.2	114.61	1095.19	77.8	163.61	1504.53	65.1	14.14	175.94	0.58	52.83	534.16	36.0	287.21	2666.94
	Llama-3.1-8B	07-2024	56.1	59.23	642.17	80.8	251.12	2366.33	69.5	29.43	331.08	0.59	43.88	486.29	35.3	516.65	4848.99
	Llama-3.2-1B	09-2024	2.7	93.45	306.71	44.5	202.27	586.10	47.0	87.81	279.37	0.54	51.96	190.96	10.5	515.57	1445.67
	Llama-3.2-3B	09-2024	1.3	52.39	326.52	76.2	203.48	1063.12	62.5	48.90	291.40	0.57	45.71	285.53	26.5	515.68	2694.72
Mistral	Ministral-3-3B	12-2025	49.9	45.62	258.11	85.3	448.45	1946.04	67.1	6.64	83.63	0.56	53.06	286.32	38.2	577.25	2529.92
	Ministral-3-8B	12-2025	41.8	52.49	523.09	89.4	354.32	3059.06	76.4	2.00	74.66	0.57	60.51	585.83	46.1	661.25	5790.60
	Ministral-3-14B	12-2025	53.1	45.91	714.19	91.8	388.21	5579.33	77.7	2.01	88.81	0.58	63.42	951.87	50.7	628.76	8917.60
	Mistral-Nemo-12B	07-2024	76.7	65.47	976.15	83.1	208.34	2835.66	64.6	38.30	570.84	0.58	37.13	585.01	41.6	291.74	3997.33
	Mistral-v0.1-7B	09-2023	58.5	122.94	1053.80	41.2	226.61	1839.82	50.6	12.40	155.47	0.55	46.51	438.98	15.5	411.41	3399.38
	Mistral-v0.2-7B	11-2023	55.3	94.19	815.29	45.7	266.99	2160.53	57.1	101.88	861.65	0.57	59.01	537.82	20.9	373.33	3072.01
	Mistral-v0.3-7B	05-2024	68.6	93.56	893.14	55.3	242.81	2137.82	59.1	28.35	292.10	0.58	72.82	690.60	24.6	376.80	3385.71
OpenELM	OpenELM-1-0.27B	04-2024	0.0	232.00	489.00	1.5	874.97	1704.95	15.3	880.74	1689.85	0.11	253.05	528.31	0.0	673.71	1349.02
	OpenELM-1-0.45B	04-2024	0.0	1200.60	3256.42	1.3	806.10	2066.84	15.3	1377.61	3585.45	0.22	641.77	1698.07	0.0	271.41	787.34
	OpenELM-1-1.1B	04-2024	0.3	980.91	5062.03	1.5	922.71	4428.61	22.5	751.34	3623.02	0.25	258.56	1288.25	0.1	825.11	4232.99
	OpenELM-1-3B	04-2024	0.7	1170.32	9609.54	0.3	1515.66	11874.41	18.3	976.89	7706.56	0.11	206.57	1675.40	0.1	954.21	7655.46
Phi	Phi-3-4B	04-2024	60.5	98.97	660.90	80.0	225.46	1354.48	69.3	13.63	129.27	0.53	43.62	323.23	31.3	495.38	3106.80
	Phi-3-14B	04-2024	68.0	116.71	1906.62	83.0	266.86	4371.46	75.5	18.97	350.56	0.54	55.30	934.58	37.4	417.92	6760.86
	Phi-3.5-4B	11-2024	61.8	172.74	1144.73	78.5	237.47	1447.39	69.0	110.31	711.41	0.53	47.35	353.45	28.8	544.07	3506.97
	Phi-4-4B	02-2025	61.2	81.59	576.12	82.3	186.02	1211.00	70.5	57.10	408.48	0.53	43.81	336.55	34.3	406.80	2653.42
	Phi-4-15B	02-2025	68.8	133.72	2328.29	91.9	287.99	4820.58	85.0	105.26	1800.76	0.57	68.38	1211.41	50.0	519.57	9056.52
Qwen	Qwen-1-1.8B	09-2023	5.3	270.04	1062.91	30.7	133.39	523.37	43.5	55.84	248.93	0.52	56.08	259.65	5.2	311.69	1178.80
	Qwen-1-7B	09-2023	48.0	138.80	1309.15	46.3	126.19	1078.24	53.9	63.09	568.88	0.58	44.47	449.82	15.8	291.13	2597.75
	Qwen-1-14B	09-2023	53.4	125.01	2081.34	61.6	115.76	1827.70	64.8	36.67	623.49	0.60	45.27	806.27	27.8	315.51	5069.41
	Qwen-1.5-0.5B	01-2024	0.8	460.55	1132.86	7.8	201.72	499.34	34.3	17.35	94.23	0.45	48.08	163.37	0.4	433.94	1050.51
	Qwen-1.5-1.8B	01-2024	10.6	424.61	1376.22	28.9	202.54	626.08	45.7	52.92	208.85	0.54	103.26	358.52	2.4	640.85	1999.63
	Qwen-1.5-4B	01-2024	44.0	110.61	706.38	41.9	206.00	1174.56	54.4	7.14	95.18	0.56	39.88	286.44	14.1	315.62	1876.47
	Qwen-1.5-7B	01-2024	53.0	102.18	1015.55	57.1	228.48	2073.14	62.9	37.57	380.85	0.58	49.19	523.59	13.7	363.52	3416.89
	Qwen-1.5-14B	01-2024	71.4	82.38	1403.14	68.8	238.34	3713.49	70.1	31.53	545.92	0.60	47.00	832.12	21.7	371.25	5974.47
	Qwen-2-0.5B	06-2024	6.5	63.55	187.01	32.3	246.19	581.80	40.4	8.56	71.71	0.50	51.88	162.43	5.8	262.69	613.85
	Qwen-2-1.5B	06-2024	51.5	58.27	243.68	56.9	238.94	824.94	54.8	6.38	75.79	0.54	43.70	196.87	16.3	203.89	714.86
	Qwen-2-7B	06-2024	71.3	60.80	549.49	82.8	250.57	2026.44	71.3	2.12	77.30	0.58	69.18	615.65	33.6	409.64	3346.73
	Qwen-2.5-0.5B	09-2024	49.7	61.98	189.24	45.1	295.70	700.01	33.1	2.42	61.20	0.52	45.91	155.75	9.7	323.38	761.55
	Qwen-2.5-1.5B	09-2024	67.0	56.83	240.02	65.6	248.10	851.51	61.1	5.46	73.85	0.56	40.70	187.03	24.0	256.52	886.63
	Qwen-2.5-3B	09-2024	74.1	58.40	430.21	81.0	302.81	1942.41	65.7	2.06	62.22	0.57	42.85	328.98	32.6	300.38	1938.23
	Qwen-2.5-7B	09-2024	77.8	57.46	576.15	91.1	260.61	2280.12	73.0	2.00	67.33	0.58	41.75	435.81	41.8	398.77	3483.55
	Qwen-2.5-14B	09-2024	76.5	64.32	1132.58	92.5	258.01	4200.17	80.9	2.00	88.02	0.60	45.74	825.57	48.4	514.28	8407.74
	Qwen-3-0.6B	04-2025	58.0	53.21	231.75	53.6	167.13	609.47	25.8	21.04	120.14	0.50	42.12	193.03	13.5	232.46	822.95
	Qwen-3-1.7B	04-2025	71.2	54.82	292.39	77.3	242.78	1025.27	63.2	183.92	805.44	0.54	49.70	271.56	30.3	487.87	2091.38
	Qwen-3-4B	04-2025	78.4	58.31	427.00	93.2	297.79	1868.53	79.0	120.56	810.78	0.57	58.33	419.56	49.5	338.57	2164.67
	Qwen-3-8B	04-2025	79.1	56.05	620.97	91.2	259.10	2568.26	80.1	104.49	1092.16	0.55	42.80	490.94	49.2	426.99	4287.02
	Qwen-3-14B	04-2025	80.3	55.38	1031.44	92.5	210.89	3430.91	81.2	3.88	132.42	0.57	46.68	879.07	51.8	387.55	6146.20
RNJ	RNJ-1-8B	12-2025	76.7	182.26	1942.96	86.9	172.44	1805.02	67.6	223.16	2400.61	0.54	73.25	835.54	55.6	384.44	4036.56
SmolLM	SmolLM-1-0.135B	07-2024	0.0	921.13	2208.93	1.8	1007.49	2336.98	18.5	419.23	1018.37	0.38	1250.07	2934.67	0.9	589.73	1438.56
	SmolLM-1-0.36B	07-2024	0.0	876.84	2601.61	2.5	1234.66	3167.46	18.4	511.48	1518.46	0.39	670.32	1998.19	3.3	354.67	1077.90
	SmolLM-1-1.7B	07-2024	0.0	746.58	2479.79	5.4	514.09	1597.79	20.6	115.70	408.99	0.47	300.01	1018.94	3.7	413.44	1363.06
	SmolLM-2-0.135B	11-2024	0.0	734.97	1612.66	2.5	242.49	558.19	22.2	51.94	159.80	0.51	227.71	523.57	1.0	670.85	1476.80
	SmolLM-2-0.36B	11-2024	0.9	574.28	1651.65	9.3	234.84	679.63	22.3	11.02	76.54	0.53	138.55	422.78	4.2	374.35	1074.81
	SmolLM-2-1.7B	11-2024	28.0	75.90	392.53	42.0	230.83	1043.78	47.4	6.68	78.16	0.56	95.69	477.52	15.0	404.85	1870.60
	SmolLM-3-3B	07-2025	66.5	54.39	322.45	82.7	212.09	1041.31	64.0	220.25	1100.19	0.56	71.91	401.90	28.7	479.78	2398.85
LLMs	MistralLarge-3-675B	12-2025	47.3	51.02	14432.05	95.0	287.94	60488.13	87.1	2.00	1087.31	0.60	46.48	10740.19	57.0	527.79	114246.98
	DeepSeek-V3.1-685B	08-2025	72.5	62.15	11017.13	93.8	217.31	34909.59	88.2	27.04	5356.05	0.59	42.65	8076.04	58.6	313.25	51009.28

Table 1: Results across benchmarks. The highest overall accuracy is highlighted in green, while the highest accuracy excluding large language models is highlighted in orange. If an SLM achieves the best overall accuracy, it is only highlighted in green. Acc. = Accuracy in %, Len. = Output Length in Tokens.

period July 2023 to December 2025 and were selected to represent widely adopted and actively developed model families from major industrial and academic contributors. These include the following model families: Qwen (Alibaba), Llama (Meta AI), Gemma (Google), Mistral and Ministral (Mistral AI), OpenELM (Apple), Apertus (ETH Zürich), RNJ (Essential AI), Granite (IBM), SmolLM (Hugging Face), and Phi (Microsoft). All SLMs were executed using Hugging Face's transformers library on a single NVIDIA RTXA6000, deployed in a SLURM cluster, with bfloat (BF16) precision and minor model-specific loading adjustments where required. The analysis is model-centric rather than architecture-isolated, reflecting complete deployed models.

To provide a reference point for comparison, we included the open-source LLMs DeepSeek V3.1 and Mistral Large 3. These models were selected for their recent release and strong performance. Both were deployed via vLLM on a cluster of 8 x NVIDIA H200 GPUs with INT8 quantization and standard vLLM hyperparameters. This method was used as it is technically not possible to deploy the LLMs like the SLMs and this represents a standard deployment for LLMs. The comparison of the single-GPU setup of the SLMs and the multi-GPU setup of the LLMs is therefore based on a realistic deployment scenario. All models were drawn from Hugging Face and are publicly available for full reproducibility.

3.2 Datasets

Models were evaluated on five benchmark datasets representing diverse tasks: General Knowledge (MMLU-Redux) [Gema *et al.*, 2025], Math (GSM8K) [Cobbe *et al.*, 2021], Function Calling (Berkeley Function Calling V3, in the following BFC) [Patil *et al.*, 2025], Coding (BigCodeBench) [Zhuo *et al.*, 2025] and Text Summarization (XSum) [Narayan *et al.*, 2018]. From each benchmark 1000 single-turn tasks were randomly sampled. For MMLU-Redux, GSM8K, BFC and BigCodeBench, performance was measured using accuracy, with BigCodeBench requiring full pass/fail execution of provided unit tests. Accuracy is defined as the fraction of correct answers over all inputs. XSum was evaluated using mean cosine similarity between generated and reference summaries, computed with the all-MiniL6-v2 embedding model [Wang *et al.*, 2020]. All model answers, except for XSum, were post-processed using custom, benchmark-specific answer extraction rules; primarily using regular expressions.

3.3 Experimental Setup

We evaluated single model inference on a single input, and each input was evaluated only once. All models received identical user prompts and custom, benchmark-specific system prompts describing the task. In case that models do not support native tool-calling, the tool specifications were added to the system prompt with a concise usage description. This allows us to analyze older models without native function calling as well. Inference was performed deterministically with fixed random seeds and without sampling. Models may generate output until a model-specific stop sequence occurs with a maximum of 2048 output tokens.

3.4 Energy Measurements

Inference-time energy consumption was measured per input, assuming an already loaded model, using CodeCarbon [Lottick *et al.*, 2019]. The measurement window covered only the prefilling and generation step (as this uses most energy), excluding tokenization, pre- and post-processing. Analyses were focused on GPU, as it dominates energy consumption and this approach minimizes interference from background processes.

4 Results & Analysis

In this section, we present our results, structured by focused research questions.

4.1 RQ1: What relationships and trade-offs exist between model size, energy consumption and task performance in SLMs?

To analyze the relation of those three dimensions, we look at the reported performance and energy measurements as shown in Table 1. The performance varies substantially among the SLMs and while there is a general trend of better performance from bigger models, there are notable exceptions. For example, in GSM8K and BigCodeBench, the best performing models are Qwen-3-4B (93.2%) and RNJ-1-8B (55.6%), respectively, and not the most recent, largest SLMs. In addition, the differences across model sizes in the same model family are in several cases minor, e.g. Qwen-3. This suggests diminishing performance gains with increasing model sizes for some tasks and model families.

The energy consumption varies heavily among SLMs and model sizes. As expected, if models with the same number of parameters have a similar mean output length, the energy consumption is similar as well. However, the output lengths necessary to complete a task can differ widely among SLMs resulting in drastically varying energy consumption. This can result in smaller SLMs that consume more energy than bigger ones. Taking each SLM and comparing it to each strictly larger SLM shows that the smaller model consumes more energy than the larger one in 37.73% (BFC), 24.01% (GSM8K), 42.97% (MMLU-Redux), 32.12% (XSum) and 14.41% (BigCodeBench) of cases of each benchmark. This indicates that model size or joules per token alone are not sufficient parameters to determine efficient SLMs. It is necessary to consider output length or in the best case, the energy consumption as well.

Examining the relationship between energy consumption and performance reveals that higher energy demands do not reliably translate into better performance. There is no benchmark where the best-performing SLM is also the one with the highest consumption. In GSM8K, for example, the best-performing SLM, namely Qwen-3-4B, consumes less energy than the next eight best contenders. Similar examples are scattered over all benchmarks and performance levels. However, there is an interesting phenomenon in that the highest consuming SLMs are mostly among the worst performing ones. This appears to be due to the extremely high output length. This is especially pronounced for the model family OpenELM, e.g. in XSum the one-sentence summarization

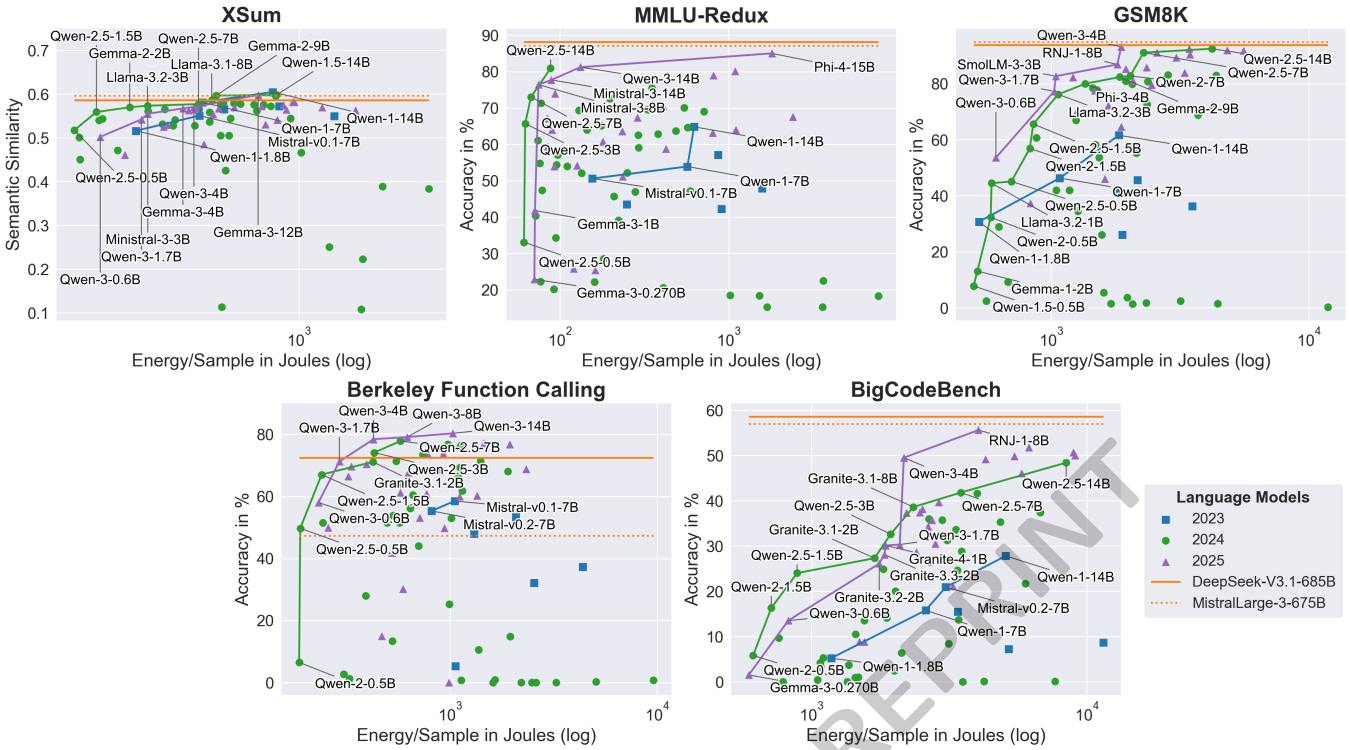

Figure 1: Performance over mean energy consumption per benchmark, color-coded by release year. The solid line connects, respectively, the next model in energy consumption with a higher performance.

has a mean output length of up to 641.77 tokens (OpenELM-1-0.450B), drastically exceeding the typical length of one sentence. The high output length and low performance indicate minor instruction following as well as task capabilities. This further underlines the role of output length and energy consumption in model selection.

Given the inconsistent relation between the three dimensions, comparing models along a single dimension is insufficient. In practice, efficiency in SLMs is a multi-objective problem in which the goal is high performance and low energy consumption. To systematically capture these trade-offs and to identify models with balanced performance-energy combinations, we analyze the Pareto frontier in Figure 1. This frontier exhibits a non-linear relationship across all benchmarks. While initial increases in energy yield clear performance gains, they start to saturate quickly resulting in noticeable diminishing returns, which set in after a moderate energy budget is reached; achieving further improvements requires a substantial amount of additional energy. For example, regarding BFC, improving accuracy by 1 percentage point from Qwen-3-8B to Qwen-3-14B nearly doubles inference energy consumption. This pattern of diminishing returns is particularly expressed in XSum and BFC, where near-maximum performance is reached at relatively low energy budgets, and additional energy consumption yields little benefit. In more reasoning-intensive tasks such as GSM8K and MMLU-Redux, the performance continues to improve with higher energy consumption, although diminishing returns remain clearly visible.

The Pareto frontier analysis also illustrates these relations by identifying the SLMs that show the best combination of performance and energy consumption. Interestingly, only a small subset of models lies on the Pareto frontier for any of the five benchmarks, and the majority are Pareto dominated. This means that alternative models exist that can achieve higher performance with lower or comparable energy consumption. Importantly, we can again identify several larger SLMs that outperform their smaller counterparts in terms of efficiency. Although this is counterintuitive, it can be explained by differences in output length as already described.

Despite the focus on SLMs, it is important to place the results regarding efficiency in the broader context of SLMs vs. LLMs. As Table 1 shows, the efficiency benefits of SLMs become particularly apparent. LLMs achieve marginal performance differences while consuming dramatically more energy in nearly all benchmarks. For example, in GSM8K, comparing the best-performing LLM, MistralLarge-3-675B, to the best-performing SLM, Qwen-3-4B, the LLM consumes approximately 32× the energy for only about a 2% absolute accuracy improvement. This indicates huge efficiency benefits of SLMs in contrast to LLMs. However, it must be noted that due to the differences in deployment, the LLMs' results may vary, though this nonetheless remains a strong indication.

RQ1 Findings: The relations between model size, performance, and energy consumption are complex and, in

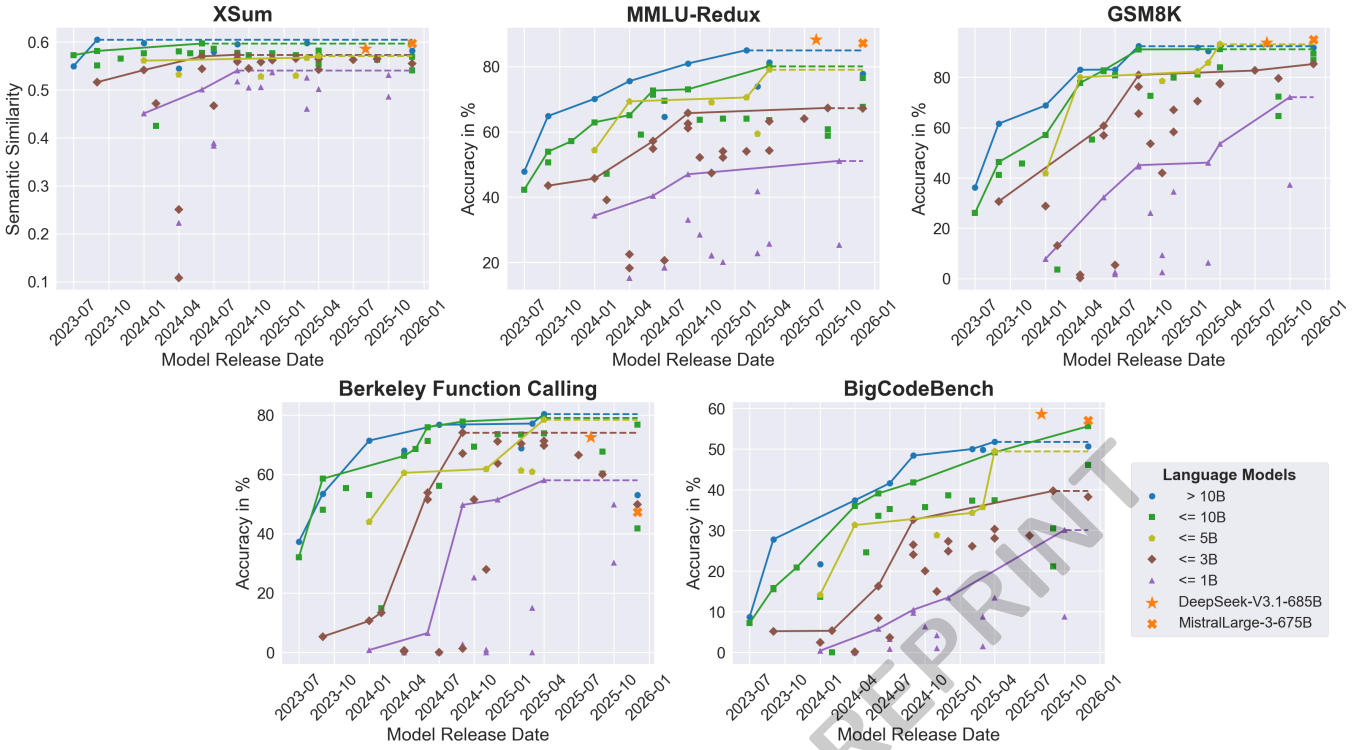

Figure 2: Historical development of model performance, color-coded by model parameter count per benchmark. The solid line connects, respectively, the next model in time with a higher performance. The figure with model names can be found in the GitHub-Repository.

some cases, unintuitive. Model size alone is a weak predictor for energy consumption and performance, as smaller models can outperform larger ones but can also consume more energy. The main driver for this effect is the output length necessary for a specific SLM to complete a task, which can cause substantial efficiency differences even among SLMs with similar model sizes. The Pareto frontier highlights that only a few SLMs achieve optimal performance-energy trade-offs, while the majority are heavily Pareto dominated. Furthermore, it shows that initial increases in energy yield substantial performance gains, but diminishing returns occur with higher energy consumption. These results reveal that efficient SLM selection requires informed, task-aware model selection rather than simple size-driven decisions.

4.2 RQ2: How have efficiency and task performance evolved across model size and over recent years, and how do they compare to reference LLMs?

To answer this research question, we analyze both the temporal evolution of task performance across model sizes and the changes in inference efficiency. Figure 1 shows the yearly Pareto frontiers from 2023 to 2025 regarding energy vs. performance. Figure 2 depicts the temporal development of task accuracy across different model size categories. Together, these perspectives allow us to assess not only whether SLMs improve over time, but whether such improvements translate into more favorable energy–performance trade-offs. For the analysis, we focus on models that are Pareto optimal in regard to the trade-off accuracy to energy consumption in their release year (Figure 1) or in regard to accuracy to release date in a given model size group (Figure 2).

Across all benchmarks, we observe substantial efficiency improvements from 2023 to 2024, reflected by a clear outward shift of the Pareto frontier in Figure 1. For reasoning- and code-intensive benchmarks like MMLU-Redux, BFC, and BigCodeBench, this shift is primarily driven by the emergence of new top-performing models rather than by uniform efficiency gains across all energy budgets. In particular, models with very low energy consumption exhibit only limited changes over time, whereas the high-accuracy end of the frontier expands noticeably. For all three benchmarks, the highest-accuracy models originate from 2025, indicating that recent progress mainly manifests through frontier extensions rather than broad efficiency improvements.

GSM8K shows a similar pattern with a pronounced outward shift of the Pareto frontier from 2023 to 2024. Notably, the frontiers cross each other, indicating that one model from 2023 achieves Pareto optimality over a group of 2024 Pareto optimal models. In 2025, the outward shift of the Pareto frontier is less pronounced, indicating a transition from broad efficiency improvements toward more incremental gains.

XSum shows a distinct behavior. Performance saturates early across all size categories, with most SLMs already matching the LLM reference points by 2024. Efficiency improvements are less pronounced, with no clear gains from

2024 to 2025. This suggests a task- or metric-specific ceiling, where further improvements become increasingly difficult to realize, particularly at low energy budgets.

The temporal performance analysis across model sizes in Figure 2 shows that progress is not solely driven by scale, but manifests differently across benchmarks. For MMLU-Redux, GSM8K, and BFC, models with at least 3B parameters exhibit consistent performance gains from 2024 to 2025, converging toward similar accuracy levels and, in the case of GSM8K and MMLU-Redux, approaching the LLM reference points. Smaller models ($\leq$1B parameters) follow a similar trend over time but retain a considerable performance gap relative to larger models, indicating that methodical improvements do not fully eliminate scale effects for these tasks.

BigCodeBench exhibits a more differentiated pattern. While performance improves across all model size categories, clear gaps between sizes persist, and only a single mid-sized model (RNJ-1-8B) reaches performance comparable to the LLM reference points. This is in notable contrast to the previously mentioned convergence across sizes for XSum.

Overall, these results indicate a shift from broad efficiency improvements between 2023 and 2024 toward more frontier-driven progress in 2025. While some benchmarks show signs of saturation, particularly XSum, recent SLMs increasingly achieve state-of-the-art performance on reasoning, coding, and function-calling tasks – often at substantially lower energy consumption than reference LLMs.

RQ2 Findings: SLMs exhibit strong temporal improvements in both task performance and inference efficiency. From 2023 to 2024, most benchmarks show a clear outward shift of the energy–performance Pareto frontier, indicating substantial efficiency gains. From 2024 to 2025, improvements become more task and energy-budget dependent, with signs of saturation in tasks like summarization, suggesting that further frontier advances become increasingly difficult to realize under strict energy constraints, particularly for very low-energy models. Reasoning, coding, and function-calling benchmarks show continued progress, particularly driven by recent gains in smaller models. Overall, methodical advances increasingly offset scale disadvantages, allowing SLMs to approach or exceed LLM performance at significantly lower energy costs.

5 Limitations

Our study has the following limitations: First, energy measurements depend on the specific hardware configuration, runtime stack, batching behavior, and decoding parameters; absolute values may vary across systems, although relative trends should remain stable. Furthermore, surrounding pipeline components were excluded and may offer further optimization potential. Second, prompt design and evaluation protocols influence output length and accuracy and may indirectly affect energy measurements. Third, we evaluate a fixed snapshot of publicly available models and do not account for continued fine-tuning, distillation, or hardware-aware optimizations that could further shift the efficiency–performance frontier. Fourth, to enable a fair comparison between models, we only investigate non-reasoning LMs – reasoning models generate extra tokens for thinking. Hence, they tend to use more energy while exhibiting a better performance. Fifth, the quality of some of the benchmarks themselves is limited. For example, the ground truths in BFC can be flawed, and some of the tasks make certain assumptions without encoding them in the prompts/instructions for the models. And last, some benchmarks are several years old – XSum, for example, was published in 2018. This implies that some (especially newer) models might have been trained on some of the benchmark data. This phenomenon is known as Benchmark Data Contamination and is hard to eliminate [Xu *et al.*, 2024].

6 Conclusion and Future Work

In this paper, we have evaluated a wide range of SLMs from the years 2023 to 2025 in five different tasks in terms of performance, energy consumption, and efficiency with two popular open-source LLMs as reference points. We find that model size and energy consumption are not directly linked to performance, higher energy usage does not necessarily lead to better performance, and smaller models are not automatically more efficient. The main reason is that models need varying output lengths to answer tasks.

Analyzing the efficiency of SLMs, a non-linear Pareto frontier becomes apparent with diminishing returns at higher levels of energy consumption. Furthermore, improvements from 2023 to 2025 can be found regarding efficiency. However, this progress slows down, indicating practical limits and occasionally possible benchmark saturation. With respect to model size, performance gains can be observed among all sizes over recent years. SLMs are even approaching the performance of current frontier LLMs, making them viable replacements for some tasks. Future work may extend the comparison framework used in this study to experimental setups in which SLMs are first fine-tuned and optimized for concrete use cases. Additionally, a more complete investigation into the whole model life cycle would extend this line of research into the broader context of sustainable and circular AI development. Such strategies may reveal further performance and efficiency increases in SLMs, particularly if SLMs are customized for specific domains like agentic AI or industrial applications. Furthermore, application-oriented research may yield task-specific, efficient frontier SLMs. With more stable results, it is now also becoming easier to give industrial practitioners reliable advice as to which model to employ for a given use case and domain.

Overall, our paper demonstrates rapid advances in both efficiency and performance of SLMs. The results show that already today's SLMs are very capable, but model selection must be task-specific and efficiency-focused to fully benefit from the advantages of SLMs.

Reproducibility

All code to conduct the experiments is publicly available at https://github.com/DFKI/SLM-Efficiency-EvalSuite.

References

[Acharya *et al.*, 2025] Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey. *IEEE Access*, 13:18912–18936, 2025.

[Argerich and Patiño-Martínez, 2024] Mauricio Fadel Argerich and Marta Patiño-Martínez. Measuring and improving the energy efficiency of large language models inference. *IEEE Access*, 12:80194–80207, 2024.

[Belcak *et al.*, 2025] Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai, 2025.

[Berthelot *et al.*, 2025] Adrien Berthelot, Eddy Caron, Mathilde Jay, and Laurent Lefèvre. Understanding the Environmental Impact of Generative AI Services. *Commun. ACM*, 68(7):46–53, June 2025.

[Bhattacharyya, 2024] Som Sekhar Bhattacharyya. Study of adoption of artificial intelligence technology-driven natural large language model-based chatbots by firms for customer service interaction. *Journal of Science and Technology Policy Management*, 16(8):1367–1379, 05 2024.

[Bourdin *et al.*, 2024] Mathieu Bourdin, Thomas Paviot, Robert Pellerin, and Samir Lamouri. NLP in SMEs for industry 4.0: opportunities and challenges. *Procedia Computer Science*, 239:396–403, 2024.

[Cobbe *et al.*, 2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. arXiv:2110.14168 [cs].

[Corradini *et al.*, 2025] Flavio Corradini, Matteo Leonesi, and Marco Piangerelli. State of the art and future directions of small language models: A systematic review. *Big Data and Cognitive Computing*, 9(7), 2025.

[Gema *et al.*, 2025] Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are We Done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[Jeanquartier *et al.*, 2026] Fleur Jeanquartier, Claire Jean-Quartier, Paul Rieder, Vedad Misirlić, Christian Pasero, Richard Hohensinner, Heimo Müller, and Andreas Holzinger. Assessing the carbon footprint of language models: Towards sustainability in AI. *Resources, Conservation and Recycling*, 226:108670, 2026.

[Ji and Jiang, 2026] Zhenya Ji and Ming Jiang. A systematic review of electricity demand for large language models: evaluations, challenges, and solutions. *Renewable and Sustainable Energy Reviews*, 225:116159, 2026.

[Koraag *et al.*, 2025] Theo Koraag, Niklas Wagner, Felix Dobslaw, and Lucas Gren. Comparing Open-Source and Commercial LLMs for Domain-Specific Analysis and Reporting: Software Engineering Challenges and Design Trade-offs. *CoRR*, abs/2509.24344, 2025.

[Korga *et al.*, 2025] Alex Maximilian Korga, Sebastian Wefers, Keno Hanken, Raad Bin Tareaf, Ben Steemers, and Hunaida Avvad. Does Size Matter? Examining Sentence Similarity Performance in Large Language Models. In *2025 International Conference on Information Networking (ICOIN)*, pages 595–600, 2025.

[Käser *et al.*, 2025] Josua Käser, Thomas Nagy, Patrick Stirnemann, and Thomas Hanne. Multilingual text summarization in healthcare using pre-trained transformer-based language models. *Computers, Materials and Continua*, 83(1):201–217, 2025.

[Lee *et al.*, 2025] Jemin Lee, Sihyeong Park, Jinse Kwon, Jihun Oh, and Yongin Kwon. Exploring the Trade-Offs: Quantization Methods, Task Difficulty, and Model Size in Large Language Models From Edge to Giant. In *Proceedings of IJCAI-25*, pages 8113–8121, 8 2025.

[Lottick *et al.*, 2019] Kadan Lottick, Silvia Susai, Sorelle A. Friedler, and Jonathan P. Wilson. Energy Usage Reports: Environmental awareness as part of algorithmic accountability. *CoRR*, abs/1911.08354, 2019.

[Maharana *et al.*, 2024] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Maliakel *et al.*, 2025] Paul Joe Maliakel, Shashikant Ilager, and Ivona Brandic. Investigating Energy Efficiency and Performance Trade-offs in LLM Inference Across Tasks and DVFS Settings, 2025.

[Mohammadi *et al.*, 2025] Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. Evaluation and Benchmarking of LLM Agents: A Survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6129–6139, New York, NY, USA, 2025. Association for Computing Machinery.

[Narayan *et al.*, 2018] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[Patil *et al.*, 2025] Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and

Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Forty-second International Conference on Machine Learning*, 2025.

[Pham *et al.*, 2025] Nghiem Thanh Pham, Tung Kieu, Duc Manh Nguyen, Son Ha Xuan, Nghia Duong-Trung, and Danh Le-Phuoc. SLM-Bench: A Comprehensive Benchmark of Small Language Models on Environmental Impacts. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21369–21392, Suzhou, China, November 2025. Association for Computational Linguistics.

[Plociennik *et al.*, 2025] Christiane Plociennik, Ponnapat Watjanatepin, Karel Van Acker, and Martin Ruskowski. Life Cycle Assessment of Artificial Intelligence Applications: Research Gaps and Opportunities. *Procedia CIRP*, 135:924–929, 2025. 32nd CIRP Conference on Life Cycle Engineering (LCE2025).

[Plociennik *et al.*, 2026] Christiane Plociennik, René Reich, Adrien Berthelot, Martin Ruskowski, Karel Van Acker, and Daniel Schien. Artificial Intelligence and Circular Economy: An Exploration of the Ecological Infosphere. In *33rd CIRP Conference on Life Cycle Engineering (LCE)*, 2026.

[Raza *et al.*, 2025] Mubashar Raza, Zarmina Jahangir, Muhammad Riaz, and Muhammad Sattar. Industrial applications of large language models. *Scientific Reports*, 15, 04 2025.

[Saad-Falcon *et al.*, 2025] Jon Saad-Falcon, Avanika Narayan, Hakki Orhun Akengin, J. Wes Griffin, Herumb Shandilya, Adrian Gamarra Lafuente, Medhya Goel, Rebecca Joseph, Shlok Natarajan, Etash Kumar Guha, Shang Zhu, Ben Athiwaratkun, John Hennessy, Azalia Mirhoseini, and Christopher Ré. Intelligence per Watt: Measuring Intelligence Efficiency of Local AI, 2025.

[Samsi *et al.*, 2023] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9, 2023.

[Strati *et al.*, 2024] Foteini Strati, Paul Elvinger, Tolga Kerimoglu, and Ana Klimovic. ML Training with Cloud GPU Shortages: Is Cross-Region the Answer? In *Proceedings of the 4th Workshop on Machine Learning and Systems*, EuroMLSys '24, page 107–116, New York, NY, USA, 2024. Association for Computing Machinery.

[Szafarski *et al.*, 2025] Daniel Szafarski, Charlotte-Fé Radowski, and Dominik Jung. User Acceptance and Use Cases of LLM-Based Embodied Conversational Agents in Customer Interaction – A Case Study in the Luxury Automotive Industry. In *Artificial Intelligence in HCI*, pages 178–197, Cham, 2025. Springer Nature Switzerland.

[Varoquaux *et al.*, 2025] Gael Varoquaux, Sasha Luccioni, and Meredith Whittaker. Hype, Sustainability, and the Price of the Bigger-is-Better Paradigm in AI. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, page 61–75, New York, NY, USA, 2025. Association for Computing Machinery.

[Wang *et al.*, 2020] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

[Wang *et al.*, 2025] Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, TzuHao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, Qi He, Yao Ma, Ming Huang, and Suhang Wang. A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness. *ACM Trans. Intell. Syst. Technol.*, 16(6), November 2025.

[Wang, 2025] Zhenyu Wang. Generative AI-Making and State-Making: Sovereign AI Race and the Future of Digital Geopolitics. *Politics and Governance*, 13(0), 2025.

[Xiao *et al.*, 2025] Chaojun Xiao, Jie Cai, Weilin Zhao, Guoyang Zeng, Biyuan Lin, Jie Zhou, Zhi Zheng, Xu Han, Zhiyuan Liu, and Maosong Sun. Densing Law of LLMs. *Nature Machine Intelligence*, 7:1823–1833, 2025.

[Xu *et al.*, 2024] Cheng Xu, Shuhao Guan, Derek Greene, and Tahar Kechadi. Benchmark Data Contamination of Large Language Models: A Survey. 06 2024.

[Zhuo *et al.*, 2025] Terry Yue Zhuo, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, James Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. BigCodeBench: Benchmarking Code Generation with Diverse Function Calls and Complex Instructions. March 2025.