Small Language Models (SLMs): A Case-Study Review and Recommendations

Nicholas Laureano

University of Oregon

laureano@uoregon.edu

Abstract:

This report examines the growing viability of Small Language Models (SLMs) as efficient, accessible alternatives to Large Language Models (LLMs) for modern AI applications, particularly on edge and consumer devices. While LLMs have driven major advances in reasoning, coding, and natural language generation, their escalating computational, financial, and environmental costs limit broad deployment and raise concerns regarding sustainability, latency, privacy, and accessibility. In contrast, SLMs (typically ranging from 1 to 8 billion parameters) leverage advances in data quality, model compression, and optimized architecture to deliver competitive performance under strict resource constraints. This study surveys the historical evolution of language models, analyzes core optimization methods such as quantization, pruning, and knowledge distillation, and evaluates three representative SLMs: Microsoft's Phi series, Google's Gemma 3, and the open-source TinyLLaMa. Findings highlight how data-optimal training, architectural innovations, and inference-efficient design enable these compact models to rival or approach the capabilities of much larger proprietary systems. This report also discusses the sociological, economic, environmental, and ethical implications of on-device AI, emphasizing improved privacy, reduced energy consumption, greater affordability, and democratized access to advanced machine intelligence. Overall, the evidence suggests that SLMs present a promising path toward sustainable, secure, and widely deployable AI in the near future.

I. Introduction

(Note: This report is designed to be informative and digestible to those familiar with machine learning and/or NLP research. With this audience in mind, this report is organized into self-contained sections, each of which can be understood independently.)

The landscape of Artificial Intelligence (AI) has been dramatically reshaped by the rapid progression and widespread application of advanced language models. Large Language Models (LLMs) showcase remarkable proficiency in reasoning, coding, and generating human-level text due to their training on extensive datasets and reliance on billions, sometimes trillions, of parameters/inputs. This influx of funding for AI R&D has spawned a category of models capable of running close to or at terminal/edge devices like smartphones for rapid response. Consequently, on-device AI models have emerged as its own distinct category, defined by their design to perform local data processing and inference, prioritizing characteristics such as enhanced data privacy, real-time performance, and operation under strict resource constraints [1].

Background and Motivation

LLMs represent one of, if not the latest and most promising technological and computational breakthroughs popularized over the past decade, demonstrating immense potential across various domains [2]. However, this scaling trend imposes serious challenges, notably the exponential increase in model sizes leading to considerable demands on essential resources like computation, memory, and energy. The costs associated with training and serving these massive generative models, which rely on massive and centralized computing infrastructures like cloud servers such as Amazon's AWS to Microsoft's Azure, are often prohibitively expensive, and the centralized architecture is deemed unsustainable and not environmentally friendly [3]. Furthermore, the usage of proprietary LLMs in the cloud raises critical concerns regarding user privacy and data safety. A custom tailored experience often requires uploading sensitive information, thus increasing the risk of data leakage. Hence, Small Language Models (SLMs) offer a vital alternative: being compact and lightweight enough to be feasibly self-hosted and deployed on consumer-grade hardware like laptops and smartphones. This capability for localized processing liberates this valuable technological asset from the oversight of corporate interests, mitigating the risk of data transmission, and placing access to high-performance AI directly into the hands of the public [4].

Research Question and Scope of Study

The central question addressed here is the viability of Small Language Models in the present or near-future. Viability is defined here as the capacity of SLMs to be competitive with LLMs in many main use cases, such as functioning as assistants, dynamic automation agents, and data interpreters. We use the term edge devices to define end-user devices like smartphones, tablets, and laptops. SLMs stand to be the only option to run on edge devices considering the inherent limitations in computation, storage, and memory within these tools. This report aims to better understand the innovative strategies required for effective deployment, including model comprehension, optimization, and many other adaptations. These methods, such as quantization, pruning, and knowledge distillation, are essential for reducing computational demands and memory usage just to run within the capabilities of edge devices. The scope includes analyzing the trade-offs involved, as optimizing for constrained environments may require sacrificing some accuracy or scalability. A large section of this report analyzes 3 selected case studies, namely Microsoft's Phi Series models, specifically Phi-3,

Google's Gemma Series (Gemma 3), and the open-sourced TinyLLaMa model derived from Meta's Llama 2. Finally, the study examines the significant socio-environmental benefits derived from SLMs, particularly focusing on how their energy efficiency reduces reliance on large, energy-intensive data centers, thereby aligning with Green AI goals and improving economic viability.

II. History of Language Models

The Rise of Large Language Models (LLMs):

The history of language models has its roots in statistical language and neural language models which has culminated in the current era of LLMs [2]. These models, notably the generative pre-trained transformer (GPT) series, have driven immense progress in Natural Language Processing (NLP), the branch of machine learning focused on reading, writing, and overall conversing like humans [5]. LLMs distinguish themselves by extending previous pre-trained language models through the incorporation of massive data, computation, and sophisticated algorithms, resulting in highly expressive and adaptable language models capable of "understanding" and generating human-level text [2]. The key caveat is that these models do not truly "understand" in the conventional sense, but are built to replicate patterns they have learned, convincingly mimicking the knowledge and experience. For example, GPT-3, a key breakthrough in LLM technology, features 175 billion parameters, or inputs, and demands approximately 800GB of storage [1]. The amount of computation for global AI training has doubled roughly every 3.43 months since 2012. [2] This trend toward continually increasing model size and computational prowess enables LLMs to demonstrate versatile capabilities across domains like coding and writing [4].

Challenges of Scale: Cost, Energy, and Accessibility:

Due to the massive scale infrastructure required to run LLMs, there are increasing challenges related to fulfilling these demands and supporting its energy consumption, all while keeping it accessible [4]. Models like GPT-4 are proprietary, prohibitively expensive to train and serve, posing substantial financial barriers, especially for academic institutions and resource-constrained companies [5]. The rapid expansion of Generative AI models, which has recently experienced consistent doubling in size about every six months, far surpasses the slowing rate of improvement of hardware like CPUs and GPUs (about every two years). This is leading to a growing misalignment between computing supply and demand [3]. Moreover, the substantial computational intensity of LLMs requires large data centers and significant energy sources, leading to concerns about their environmental impact [4]. If these data centers are run using unclean and underprepared energy grids, it's not unreasonable to suspect the public may experience modifications to their energy bills including increased rates. Accessibility is limited by the LLMs' substantial size, which poses limitations for direct deployment on devices, and their common deployment via privately owned cloud APIs, which can introduce unreliability, inconsistent latency, and high, accumulating usage costs [1].

Emergence of Small Language Models (SLMs):

In direct response to the complexity and resource constraints inherent to LLMs, the emergence of Small Language Models offers an innovative direction [4]. SLMs are generally defined as language models typically in the 1 to 8 billion parameter range, often leveraging open-source frameworks, making them more feasible for self-hosting and specialized use cases [5]. The development of SLMs is often driven by adopting a "data optimal" approach, emphasizing the quality of training data through meticulous filtering and the generation of synthetic, high-quality content, rather than solely relying on raw data quantity [6]. In other words, only the most informative and valuable data is used to train these models, cutting the time required to create and run these models which would be bogged down by redundant or poor-grade training material. This strategy has yielded SLMs like Phi-3-mini (3.8 billion parameters), which demonstrate capabilities rivaling much larger LLMs such as GPT-3.5 [7]. SLMs can be created either through training efficiently from scratch using streamlined architectures or by leveraging model compression techniques like quantization, pruning, and knowledge distillation on existing LLMs [5].

Comparative Overview: LLMs vs. SLMs

LLMs and SLMs serve distinct yet complementary roles in the AI ecosystem. While LLMs maintain a leading edge in generalization and high-level reasoning due to their vast parameter counts, SLMs offer decisive practical advantages in edge deployments [5] [4]. SLMs require significantly fewer computational resources and energy compared to large models like GPT-4 or Gemini 1.5 Ultra, contributing to efforts toward building "green AI" [3] [4]. For commercial operations, self-hosting SLMs can provide more predictable shorter latency and lead to substantial cost reductions, showing potential savings ranging from 5x up to 29x against proprietary LLM APIs [5]. Furthermore, SLMs are inherently better suited for enhancing data privacy and security through localized processing [1]. Overall, SLMs are ideal for resource-constrained settings as mentioned before, such as edge computing, mobile devices, and IoT devices, thereby addressing the crucial challenges

of accessibility and efficiency posed by their larger counterparts [4].

III. Core Methods for Model Efficiency

Quantization

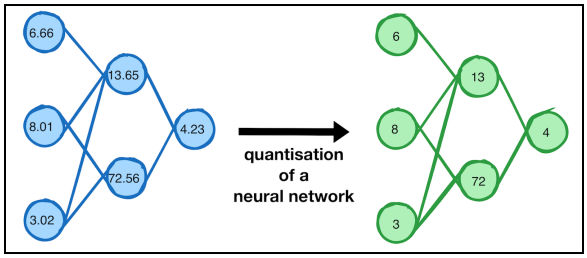

Fig. 1. Quantization Visualized (reproduced from [11]).

Quantization is a fundamental model compression technique that significantly reduces the size and memory footprint of models by lowering the precision of the numbers used to represent a model's weights, activations, or both [4]. It is critical for optimizing neural networks for on-device AI models, as it enhances computational efficiency, reduces memory/storage demands, and lowers power consumption. In practice, high-bit representations (like 16-bit floating-point) are converted to lower-bit formats (like 8-bit integers) [1]. This process speeds up inference times and allows deployment on resource-constrained devices. For example, the Phi-3-mini model is quantized to 4-bits, reducing its memory usage to approximately 1.8GB to run natively and fully offline on an iPhone 14 [7]. However, aggressive precision reductions can degrade model accuracy, requiring careful selection of quantization levels to maintain performance [3]. More on these trade-offs are discussed later.

Knowledge Distillation

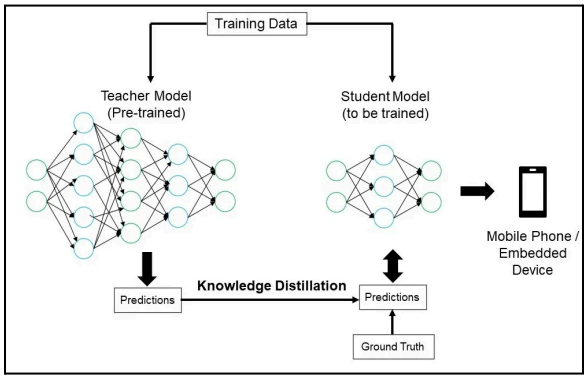

Fig 2. Knowledge Distillation Visualized (Reproduced from [12]).

Knowledge Distillation (KD) is a pivotal model optimization technique that trains a smaller, more efficient "student" model to mimic the performance and output distribution of a large, complex "teacher" model [6]. This approach effectively transfers the complex knowledge of the teacher, allowing the student model to achieve comparable high accuracy with significantly fewer parameters, making it ideal for resource-constrained edge devices. KD can be implemented in a black-box scenario, relying only on the teacher model's outputs (predictions, chain-of-thought, instruction following), or a white-box scenario, where the student model accesses the internal mechanisms or intermediate representations of the teacher model during training [4].

Pruning

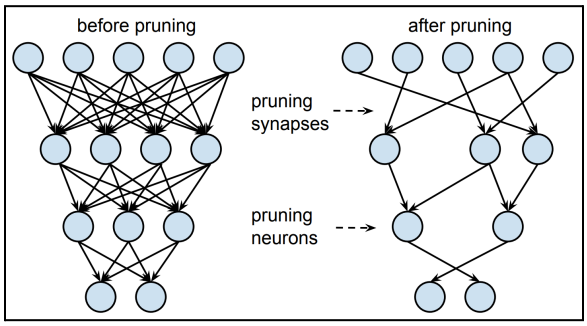

Fig. 3. Pruning Process Visualized (reproduced from [13]).

Pruning is a vital technique focused on optimizing model structure by identifying and eliminating redundancy within a neural network's parameters, weights, or entire neurons. By systematically removing less important weights, pruning effectively reduces computational demands, lowers memory usage, and enables faster inference [1]. It can essentially be understood to be trimming the fat off a model while aiming to keep as much of the quality as possible,just at a reduced size. Pruning is categorized into unstructured pruning, which removes individual weights (often those near zero) resulting in a sparser matrix, and structured pruning, which removes entire groups of parameters (like neurons, channels, or layers) to simplify the architecture for deployment optimization [4].

Trade-offs: Accuracy, Interpretability, and Compute Cost

Implementing optimization techniques for smaller AI models requires addressing critical performance trade-offs. Optimizing models for constrained edge environments often involves sacrificing model accuracy or scalability to maintain reasonable functionality. Specifically, aggressive model compression techniques like quantization and pruning often lead to a reduction in model accuracy or output quality. While quantization and pruning drastically reduce compute cost and memory requirements, they also introduce the challenge of balancing resource efficiency with fidelity comparable with the original large model's capabilities. Some ongoing research in this field is focused on techniques like Quantization Aware Training (QAT)

used in Gemma 3 and adaptive compression to strike a sustainable balance, ensuring models remain efficient, accurate, and adaptable for diverse edge environments [8] [3].

IV. Case Studies of Modern Small Language Models

1. Phi Series (Microsoft)

According to the Phi-3 Technical Report, the Phi-3 series introduces a family of highly capable LMs ranging in scale from 3.8 billion to 14 billion parameters, built on the principle of prioritizing training data quality over sheer quantity. The foundational model, phi-3-mini, contains 3.8 billion parameters and was pre-trained on 3.3 trillion tokens. Despite its compact size, its performance, measured across academic benchmarks, rivals that of much larger models such as Mixtral 8x7B and GPT-3.5, achieving 69% on MMLU and 8.38 on MT-bench. The series also includes larger variants, phi-3-small (7B) and phi-3-medium (14B), which show even greater performance with up to 78% on MMLU, demonstrating that Phi 3 models achieve performance typically associated with models many times their size [7].

Model Architecture and Core Philosophy

The Phi-3 series primarily utilizes a transformer decoder architecture. The core philosophy involves designing models to be highly effective within specific inference constraints, particularly having small sizes. The phi-3-mini variant is structured with 32 layers, 32 heads, and a hidden dimension of 3072, and is built upon the same block structure and tokenizer as Meta's Llama-2. To handle extended inputs, a long-context version, phi-3-mini-128K, extends the default 4K context length up to 128K tokens using the LongRope method. In terms of efficiency, the phi-3-small model incorporates Grouped-Query Attention and features a novel blocksparse attention module to optimize KV cache memory usage and retrieval performance. The Phi-3.5 series further explores modularity via a Mixture-of-Experts (MoE) architecture, allowing the phi-3.5-MoE to activate only 6.6 billion parameters out of 42 billion total parameters, boosting efficiency and capability [7].

Unique Development Style - Quality over Quantity

The unique development style of the Phi series, known as the "Textbooks Are All You Need" philosophy, relies heavily on improving training data quality rather than merely increasing its volume [6]. The training methodology aims for a "Data Optimal Regime" by using a scaled-up, heavily filtered dataset composed of publicly available web data and synthetic, LLM-generated data. The web data is meticulously filtered based on its "educational level" to provide generalized knowledge, while the following phase uses synthetic data to teach specific logical reasoning and niche skills. This rigorous filtering process removes certain factual information deemed less critical for a compact model to preserve the model's limited capacity for reasoning ability. This focus ensures that the resulting small models can match the performance of much larger counterparts that rely on vast swaths of information [7].

Key Findings/Uniqueness

A critical finding of the Phi-3 project is the effectiveness of quality data in creating compact models that perform on par with models like GPT-3.5. This validates the design goal of optimizing for the inference within budget. This inherent efficiency makes the Phi-3 series exceptionally suitable for deployment on resource-constrained edge devices. To enable efficient on-device execution, the phi-3-mini model is optimized using 4-bit quantization, reducing its required memory to approximately 1.8GB. This quantized model was successfully tested running natively and fully offline on an iPhone 14, achieving an inference speed of more than 12 tokens per second [7]. This focus on optimizing model complexity, memory footprint, and inference speed is vital for meeting the demands of on-device AI applications, such as real-time processing and extended battery life [3] [1].

2. Gemma Series (Google DeepMind)

According to the Gemma 3 Technical Report, the Gemma 3 series represents the newest iteration of the Gemma family of lightweight open models, extending in scale from 1 billion to 27 billion parameters. Developed in conjunction with the Gemini frontier models, Gemma 3 introduces several novel capabilities to the family, including multimodality (vision understanding), long context (up to 128K tokens), and multilinguality, while maintaining or exceeding the performance of previous versions [8]. The models are powerful and versatile, demonstrating superior performance in downstream tasks after pre-training and specialized post-training recipes.

Model Architecture and Core Philosophy

Gemma 3 employs a decoder-only transformer architecture, retaining many structural elements from previous Gemma versions, such as Grouped-Query Attention, which essentially focuses the "attention" of the model to clusters of attention queries, a middle ground between small memory usage and maintained attention by the model. A key architectural distinction is the implementation of long-context capabilities (up to 128K tokens for most variants). All Gemma 3 models are developed using knowledge distillation. This distillation method followed with Reinforcement Learning fine-tuning phases form the

improved post-training approach used to achieve performance gains across math, coding, reasoning, and instruction-following domains [8].

Unique Development Style - Model Ecosystem

The Gemma 3 family is released as open models, ranging in sizes (1B, 4B, 12B, 27B) comparable to the Gemma 2 series. These models use a larger pre-training token budget (e.g., 14T tokens for the 27B variant) and increased multilingual data compared to their predecessors. The resulting instruction-tuned Gemma 3 models show substantial improvements over older models; for example, on the LMSys Chatbot Arena, the Gemma 3 27B IT model scored an Elo rating of 1338, significantly higher than the Gemma 2 27B IT model at 1220. Furthermore, the Gemma 3 27B IT variant is competitive with, or comparable to, the powerful closed model Gemini-1.5-Pro across several industry benchmarks [8]. Although the 27B variant is likely too large for many handheld devices, it's not unreasonable for high-end laptops and desktops to run these sizes of model.

Key Findings/Uniqueness

The Gemma 3 models are explicitly co-designed to be lightweight and run efficiently on standard consumer-grade hardware, including phones and laptops. The necessity of developing efficient AI models for constrained edge environments 7]. Gemma 3 addresses exactly that with architectural modifications, such as the 5:1 interleaving of local and global attention layers, specifically designed to reduce the memory overhead of the KV cache during long-context inference. This memory optimization results in a significant reduction of memory usage compared to prior global-only attention models (like Gemma 1 or Llama). This focus aligns with the design of related SLMs in the ecosystem; for instance, the smaller Gemma 2B model is specifically intended for deployment on edge devices [6].

3. TinyLLaMa

TinyLLaMa is an open-source, compact SLM consisting of 1.1 billion parameters that was built upon the architecture and tokenizer of Meta's Llama 2 [9]. The model was pre-trained on an extensive body of up to 3 trillion tokens. Despite its size, TinyLlama exhibits remarkable performance on a range of downstream tasks, significantly surpassing existing open-source models of comparable scale, such as OPT-1.3B and Pythia-1.4B. The overall training speed of TinyLlama proved superior to other models, achieving a throughput of 24,000 tokens per second per A100-40G GPU and requiring significantly fewer GPU hours compared to models like Pythia-1.0B and MPT-1.3B [6] [9].

Model Architecture and Core Philosophy

TinyLLaMa utilizes a decoder-only Transformer architecture with 1.1 billion parameters, following the design principles of the Llama models [9]. The model is characterized by architectural hyperparameters including a hidden size of 2,048, 22 layers, 32 heads, and a vocabulary size of 32,000. To incorporate positional data, it employs Rotary Positional Embedding (RoPE), a widely adopted method in mainstream models which helps models quickly inference input positional data [6]. For stability and efficiency, TinyLlama uses pre-norm along with the RMSNorm normalization function, and it incorporates the SwiGLU activation function instead of the traditionally popular ReLU. Furthermore, optimization for inference speed and reduced memory bandwidth overhead is achieved through the use of Grouped-query Attention, as seen in all 3 of our case studies, which groups 32 query heads into 4 groups of key-value heads, allowing for the sharing of key and value representations [9].

Unique Development Style - Inference-Optimal Training

TinyLLaMa was developed based on an Inference-Optimal Training philosophy, distinguishing itself from the compute-optimal approach suggested by classical scaling laws. This unique strategy was motivated by findings suggesting that simply scaling up model and data size proportionally may not be optimal, and instead prioritizes achieving optimal performance given specific inference constraints, such as a compact model size suitable for mobile devices. The creators aimed to explore the behavior of a small model when trained with a significantly larger number of tokens than recommended by scaling laws, ultimately training the 1.1B parameter model using up to 3 trillion cumulative tokens (v1.0) or 2 trillion tokens (v1.1). This emphasis on almost over-training these models on data beyond their assumed recommended amounts seems to lead to beneficial performance on various metrics [6] [9].

Key Findings/Uniqueness

A core finding demonstrated that TinyLlama generally outperforms baseline models of comparable size on commonsense reasoning tasks like HellaSwag, PIQA, OpenBookQA, and ARC. The refinement of the initial training methodology introduced a three-stage pipeline (basic pre-training, continual pre-training, and a cooldown phase) for TinyLlama v1.1, resulting in specialized variants. These variant models included the general TinyLlama v1.1 foundational model, the TinyLlama v1.1 - Math&Code variant, and the TinyLlama v1.1 - Chinese variant. The specialization proved effective, as the Math&Code variant achieved a significant improvement in performance on problem-solving tasks such as HumanEval and DROP [9].

V. Discussion

Technical and Ethical Implications of SLMs

The shift toward deploying SLMs on devices like smartphones and laptops enhances the accessibility of advanced AI technologies for the public, companies, and researchers with limited resources. SLMs offer a feasible alternative to relying on proprietary cloud APIs from large corporations (like OpenAI's GPT-4) which often present issues such as unreliable uptime, inconsistent latency, and a lack of model control [5]. By enabling self-hosting and local inference, SLMs promote innovation and experimentation in NLP technologies within academic institutions and small/medium-sized companies [4]. This democratization of AI, shifting the power back to the masses through accessible, compact models, helps align with the advancement of modern technology and facilitates the broader application of AI in various domains. Furthermore, SLMs can be rapidly retrained or fine-tuned to cater to specific tasks, further enhancing their adaptability without the substantial costs associated with larger models [4].

Privacy and Security of On-Device AI

SLMs significantly enhance data privacy and security by facilitating localized data processing directly on your devices, eliminating the need to transmit sensitive data—such as health information or personal identifiers—to centralized cloud servers. This local approach reduces the risks associated with data transmission and potential breaches that cloud-based models face. By keeping data within the local environment, on-device AI helps organizations and individuals comply with stringent data privacy regulations, such as the General Data Protection Regulation (GDPR) [1]. Moreover, actively researched advanced techniques like Federated Learning enables decentralized model training across edge devices, ensuring that user data never leaves the device while still allowing the models to evolve and learn [3]. This decentralized processing acts as a robust solution, mitigating the risk of inadvertent data leakage or the misuse of user data for training models by large companies [1].

Environmental and Economic Impacts

The widespread reliance on traditional LLMs as of now requires powerful centralized computing infrastructure, which is inherently unsustainable, expensive, and not environmentally friendly due to substantial energy requirements it takes to run them [3]. In contrast, SLMs offer a solution by being designed to be both lightweight and computationally efficient for deployment on consumer-grade tech [4]. This shift toward efficient on-device processing reduces reliance on energy-intensive data centers and lowers the associated carbon footprint, aligning with the pursuit of Green AI [1]. Economically, self-hosting SLMs can lead to significant cost reductions for companies when compared to the pay-per-token pricing model of proprietary LLM APIs, with observed reductions ranging from 5x to 29x according to some sources [5]. This enhanced energy efficiency and economic viability are critical considerations for the future sustainability and development of AI technology.

VI. Recommendations for Future Research and Development

Model Specialization and On-Device Adaptation

Future research should focus intensely on model specialization and adaptation to on-device constraints. Due to the restricted nature of edge devices, AI hosted locally must be built strategically with these limitations in mind. Unlike the near limitless infrastructure expansion LLMs can rely on for their improvements, SLMs must find a niche to remain competitive. While LLMs pride themselves on their effectiveness in a variety of tasks and domains, lightweight models can flourish when built for hyper-specialized tasks. Research can expand in the direction of collections of small models (Like MoE models), strategically utilized only when the demands of the local user match. This could allow for SLMs to rival the capabilities of LLMs even at the local level in a seemingly general way.

Benchmark Standardization for SLM Evaluation

To fully establish the viability of SLMs, there is a recognized need for benchmark standardization for SLM evaluation to ensure fair, reproducible, and accurate comparisons between models. Current model assessment procedures, especially those requiring human judgment, can be labor-intensive and time-consuming. This requires automated and facilitated evaluation processes. Future investigation should prioritize the development of more comprehensive evaluation frameworks that can assess SLMs across diverse domains, including multimodal understanding and critical thinking. Furthermore, research must focus on strengthening the evaluation of ethical and societal impacts, such as fairness and explainability (XAI), through the establishment of robust evaluation standards and regulatory bodies. While ongoing research of AI’s improvement can overshadow the desire to pause and objectively score models, the need to critically differentiate the capabilities of models grows everyday as more and more models are produced.

Sustainable AI Practices and Green Computing Goals

A critical direction for future development involves prioritizing sustainable AI practices and green computing goals, especially considering that traditional LLM infrastructure is often described as a monumental drain on resources and unprofitable. Future research must focus on energy efficiency optimization for both LLMs and SLMs by minimizing energy consumption through optimized algorithms and adopting low-power hardware designs. A key recommendation is to intensify research into the design of hardware and software, aiming to optimize the integration of AI models with specialized hardware technologies such as TPUs to achieve enhanced computational efficiency and improved energy management. By successfully moving toward efficient on-device processing via SLMs, researchers can significantly reduce reliance on energy-intensive data centers. Such advancements in Green AI align with broader societal objectives, enabling SLMs to play a pivotal role in achieving sustainable development goals.

VII. Closing Remarks

AI, as we know it, has proven itself time and time again to be a useful tool, when used properly. My own qualms with the technology are rooted in the philosophy that we should be capable as individuals with and without our aides. However, my further passion for ensuring power for the people extends to AI as it does for any other useful technology. I see SLMs as of now to be viable, yet infantile in their capacities. For most everyday purposes, the capacity for self-hosting these tools are available, yet the major barrier of entry is the fact that these models are not common knowledge, nor are the means to run them. My hope is that as research progresses, and as we push for environmentally friendly technology, SLMs will become more popular and accessible, even to those who are not as technologically informed, allowing for the population to thrive by any means necessary.

References

1. X. Wang, Z. Tang, J. Guo, T. Meng, C. Wang, T. Wang, and W. Jia, “Empowering edge intelligence: A comprehensive survey on on-device AI models,” arXiv preprint arXiv:2503.06027, 2025.
2. Z. Wang, Z. Chu, T. V. Doan, et al., “History, development, and principles of large language models: An introductory survey,” AI Ethics, vol. 5, pp. 1955–1971, 2025, doi: 10.1007/s43681-024-00583-7.
3. S. Sai, M. Prasad, G. Dashore, V. Chamola, and B. Sikdar, “On-device generative AI: The need, architectures, and challenges,” IEEE Consumer Electronics Magazine, vol. 14, no. 4, pp. 21–32, 2025, doi: 10.1109/MCE.2024.3518761.
4. Q. Zhang, Z. Liu, and S. Pan, “The rise of small language models,” IEEE Intelligent Systems, vol. 40, no. 1, pp. 30–37, 2025, doi: 10.1109/MIS.2024.3517792.
5. C. Irugalbandara, M. Prasad, G. Dashore, V. Chamola, and B. Sikdar, “Scaling down to scale up: A cost-benefit analysis of replacing OpenAI’s LLM with open source SLMs in production,” in Proc. 2024 IEEE Int. Symp. Performance Analysis of Systems and Software (ISPASS), 2024, pp. 280–291, doi: 10.1109/ISPASS61541.2024.00034.
6. S. Subramanian, V. Elango, and M. Gungor, “Small language models (SLMs) can still pack a punch: A survey,” arXiv preprint arXiv:2501.05465, 2025.
7. M. Abdin, J. Aneja, H. Awadalla, A. A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, … and others, “Phi-3 technical report: A highly capable language model locally on your phone,” arXiv preprint arXiv:2404.14219, 2024.
8. Gemma Team (A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, … and others), “Gemma 3 technical report,” arXiv preprint arXiv:2503.19786, 2025.
9. P. Zhang, G. Zeng, T. Wang, and W. Lu, “TinyLlama: An open-source small language model,” arXiv preprint arXiv:2401.02385, 2024.
10. H. Han, J. Liang, J. Shi, Q. He, and Y. Xiao, “Small language model can self-correct,” arXiv preprint arXiv:2401.07301, 2024.
11. S. McLeod, “Understanding AI/LLM Quantisation Through Interactive Visualisations,” smcleod.net, https://smcleod.net/2024/07/understanding-ai/llm-quantisation-through-interactive-visualisations (accessed Nov. 25, 2025).
12. vijendra.1893, “Knowledge distillation theory,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2022/01/knowledge-distillation-theory-and-end-to-end-case-study (accessed Nov. 25, 2025).
13. S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural networks,” arXiv.org, https://arxiv.org/abs/1506.02626.