Characterizing and Understanding the Performance of Small Language Models on Edge Devices

Md Romyull Islam, Nobel Dhar, Bobin Deng, Tu N. Nguyen, Selena He, Kun Suo

Presented by

Md Romyull Islam

Outlines

Background & Motivation

Methodology & Performance Metrics

Results, Observations, & Insights

Conclusion

Outlines

Background & Motivation

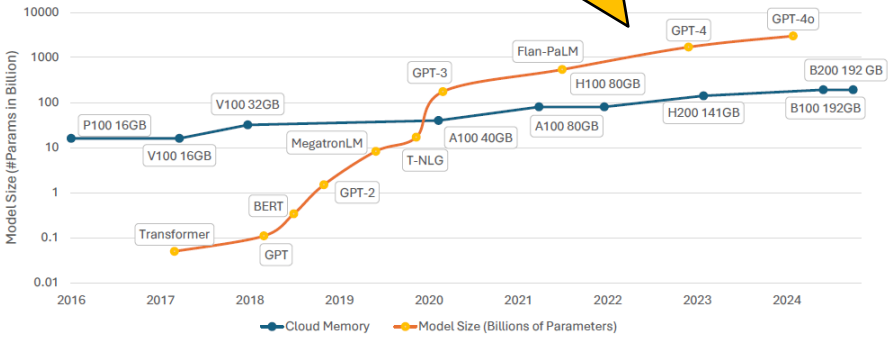
Methodology & Performance Metrics

Results, Observations, & Insights

Conclusion

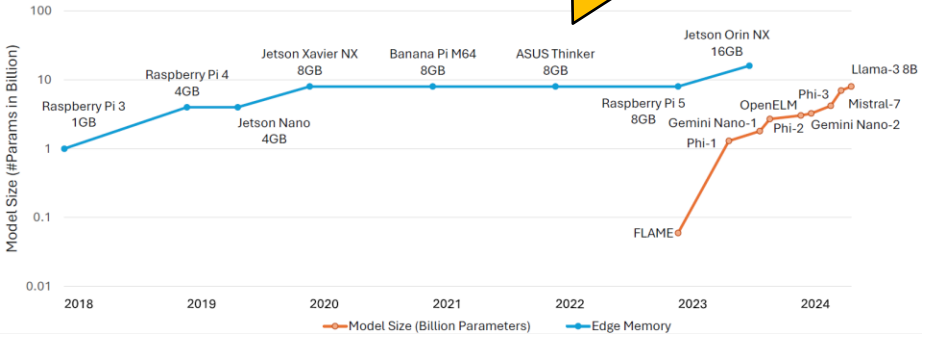
Why We Do this Research?

LLM sizes are growing exponentially, outpacing the growth of cloud GPU capacity



LLM vs Cloud GPU

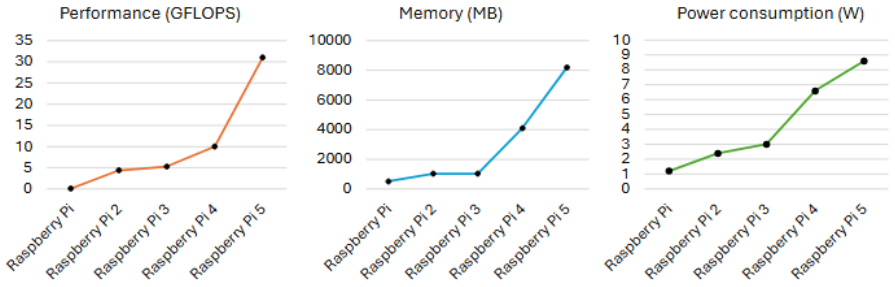
SLMs remain manageable within the GPU and memory limits of edge devices



SLM vs Edge Device

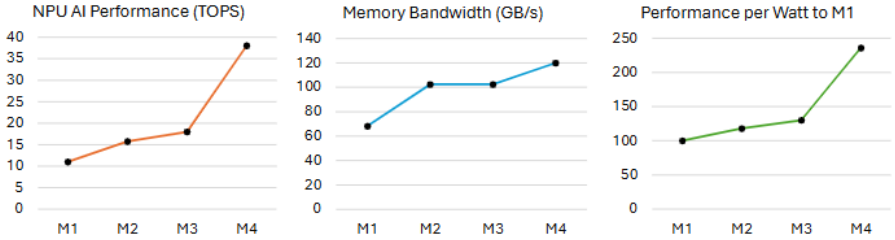
Edge AI Hardware over the Years

(a) Performance, Memory Capacity, and Power Consumption of Different Generations of Raspberry Pi

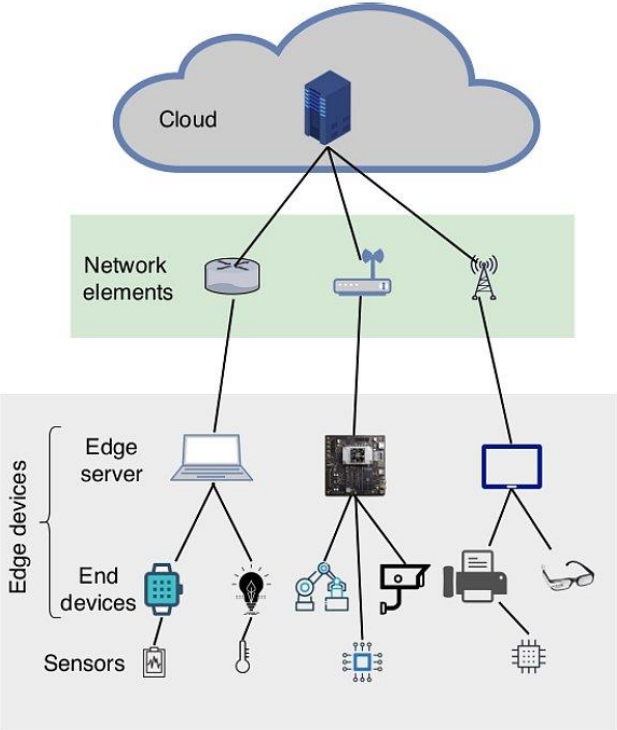


(b) Performance, Memory Capacity, and Power Consumption of Different Generation Nvidia Edge Devices

(c) NPU Performance, Memory Bandwidth, and Relative Performance Per Watt for Apple M Series Chips



When Edge meets SLM



Llama 3	Meta	8 billion parameters
Phi-3	Microsoft	3.8 billion - 7 billion parameters
Gemma	Google	2 billion - 7 billion parameters
Mixtral 8x7B	Mistral AI	7 billion parameters
OpenELM	Apple	0.27 billion - 3 billion parameters

Related Works

- **Resource Utilization on Edge Devices:**

  Studies have evaluated resource demands for edge AI applications, focusing on CPU, memory, and latency constraints

- **Optimizing Language Models for Edge:**

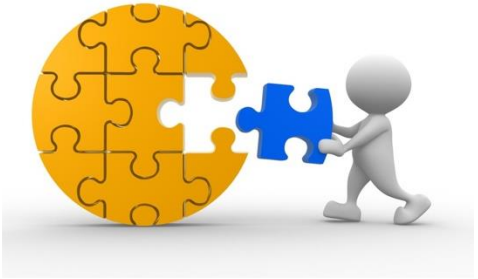
  Techniques like model pruning, quantization, and knowledge distillation have been developed to reduce model size and enhance efficiency

- **Performance in Custom Edge-AI Systems:**

  Research on specialized edge AI systems has highlighted the need for low-latency and energy-efficient models for IoT and mobile applications

These works here has focused on optimizing large models or specific tasks, few studies provide a comparative analysis of small language models across various edge devices.

Our work aims to fill this gap.



What this Research Is All About?

- **In-Depth Studies:** This research delves into balancing efficiency and performance in Small Language Models (SLMs).

- **Empirical Evaluation:** Systematic collection and analysis of data to evaluate SLMs' real-world performance on edge devices.

- **Identifying Bottlenecks:** Focus on understanding limitations that hinder SLM deployment on various edge devices.

- **Insights and Recommendations:** Findings provide practical recommendations for hardware and software optimizations in edge computing tailored to the specific requirements of SLMs.

Outlines

Background & Motivation

Methodology & Performance Metrics

Results, Observations, & Insights

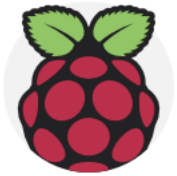
Conclusion

Used Software and Hardware

- Nmon, Nmon Analyzer, NmonChart
- Debian OS
- Psutil
- subprocess

BASIC SPECIFICATIONS OF USED EDGE DEVICES

Device Name	Memory	CPU Freq.	CPU #	Disk Size
Raspberry Pi 5B	4GB	2.4GHz	4	128GB
Jetson AGX Orin	32GB	2.2GHz	12	64GB
Mac mini	16GB	3.23GHz	8	494.38 GB



Language Models and Their Parameters

Model Name	Model Size	Type	Tokens Trained on
TinyLlama	1.1B	SLM	3T
Phi-3 mini	3.8B	SLM	3.3T
OpenELM	270M	SLM	1.8T
Llama3	8B	LLM	15T

TinyLlama

apple/OpenELM

Performance Metrics

- **Perplexity** is calculated as an exponent of the loss obtained from the model.
- Latency (Time To First Token)
- Total Generation Time
- Throughput (Tokens/Second)
- Resource Utilization (CPU, Memory, Disk, Swap)

Outlines

Background & Motivation

Methodology & Performance Metrics

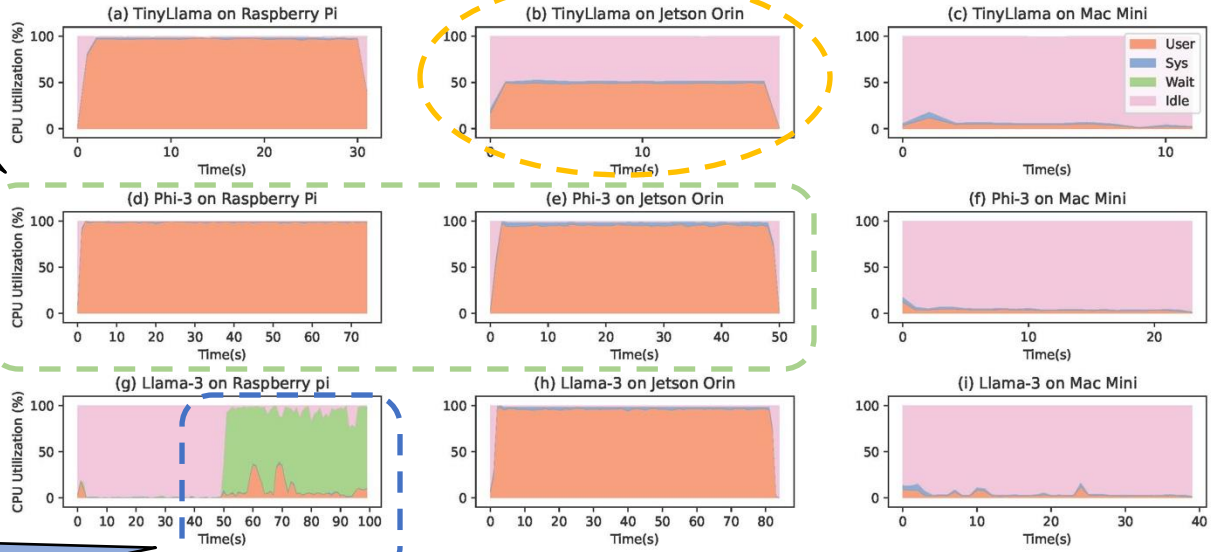
Results, Observations, & Insights

Conclusion

CPU Utilization

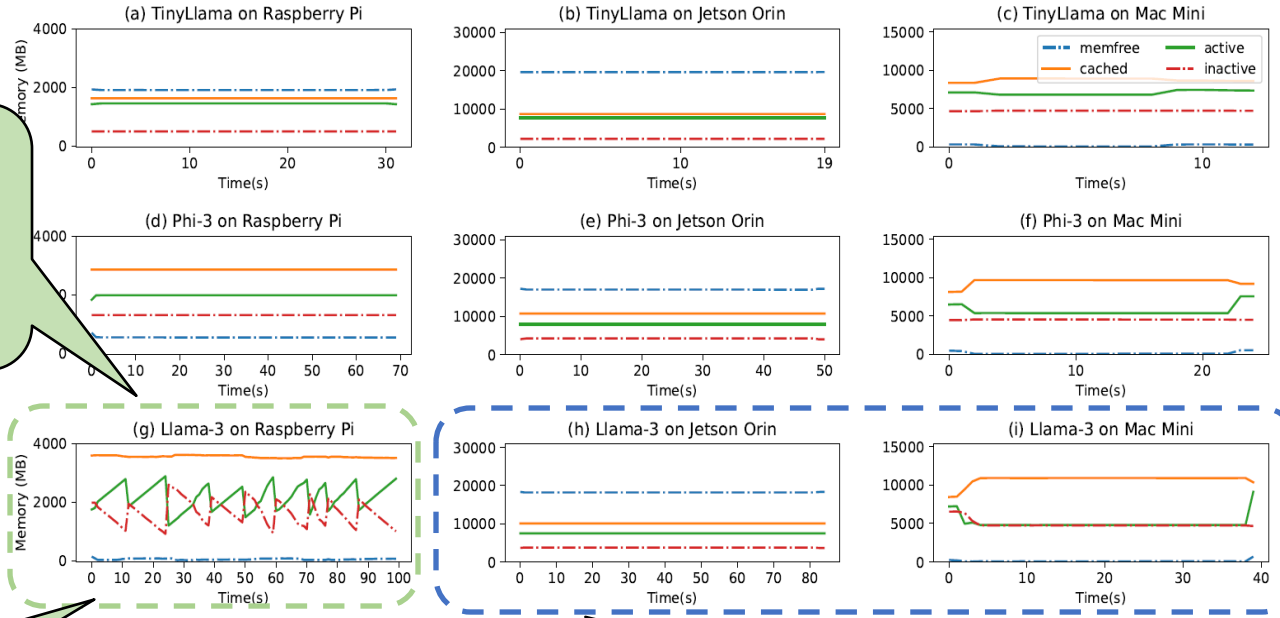
TinyLlama on Jetson Orin: Uses 49% CPU, showing efficient resource utilization

Pushes CPU to 99% on Raspberry Pi and 95% on Jetson Orin, with Orin managing more stably.



Llama-3 on Raspberry Pi: High CPU fluctuations and I/O waits due to frequent memory swapping.

Memory Utilization



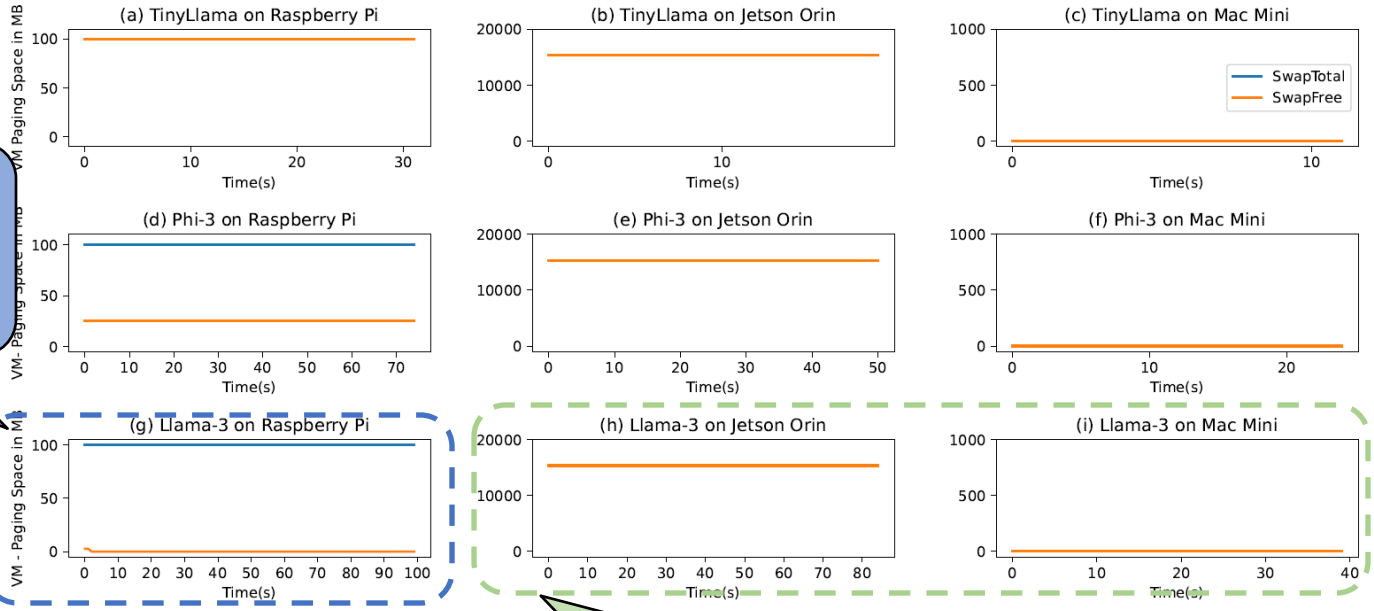
Larger models like Llama-3 require more memory resources, which causes strain on devices with limited RAM, such as the Raspberry Pi, leading to frequent memory swapping.

Inadequate memory for the workload or potential memory leaks where unused memory isn't being released back to the system effectively.

Devices with higher memory capacity, like Jetson Orin and Mac Mini, can handle intensive models more efficiently, indicating the importance of matching model size to device capability for optimal performance.

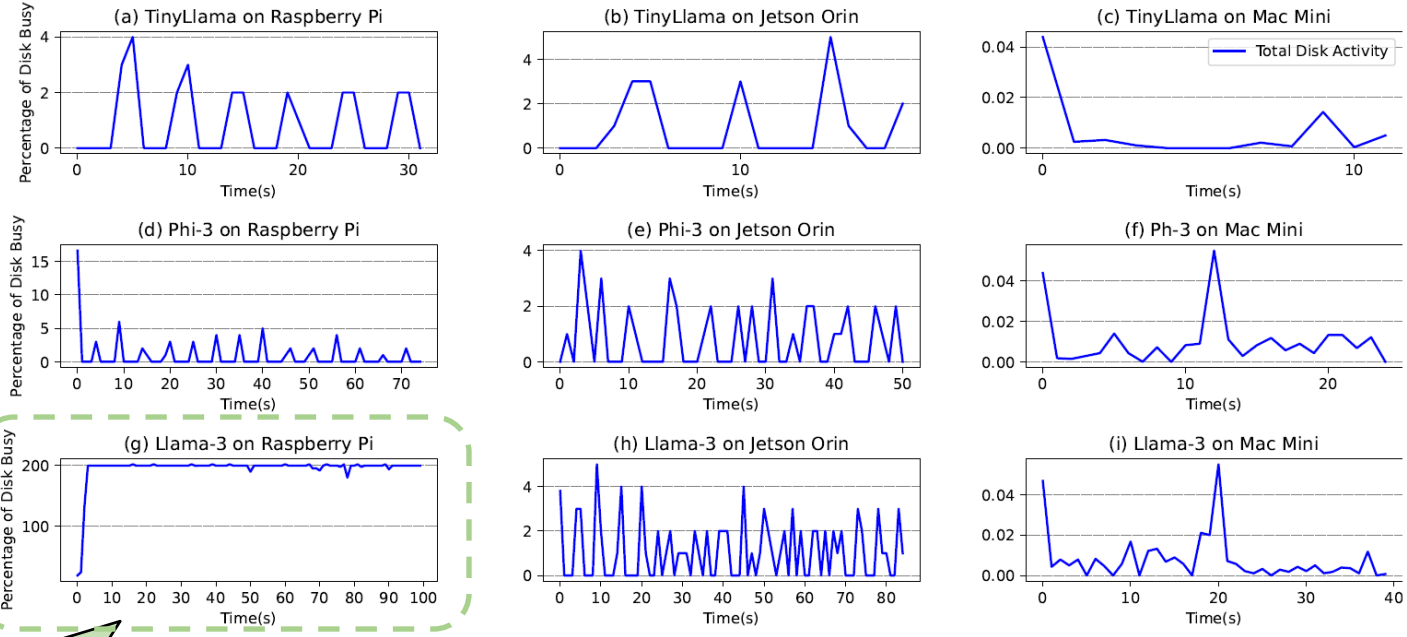
Swap Data

Swap usage is minimal across all models and devices, with slight paging in Llama-3 on Raspberry Pi due to higher memory demands.



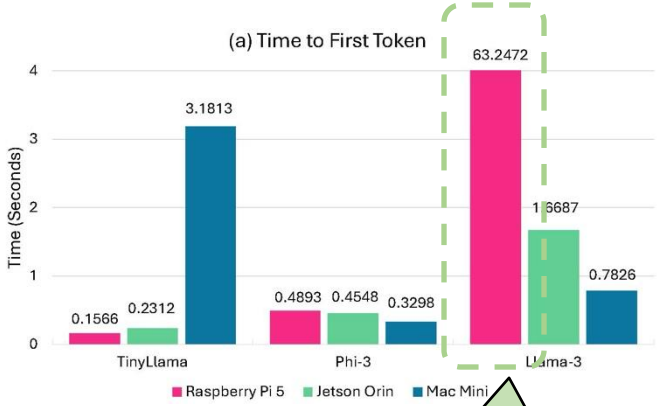
Edge devices with sufficient RAM, like Jetson Orin and Mac Mini, avoid swap usage entirely, highlighting the advantage of ample memory in reducing disk I/O load and enhancing performance stability.

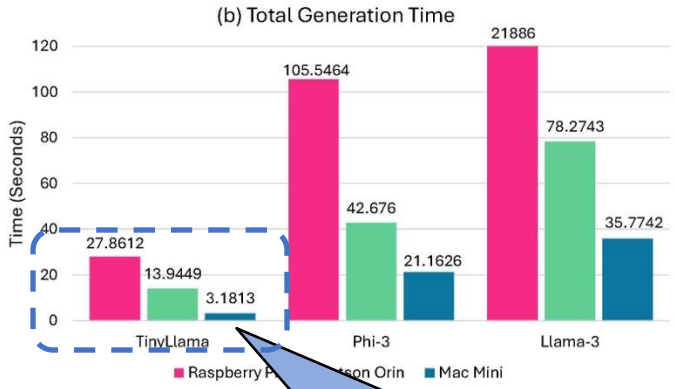
Disk Usage



Prolonged high disk usage risks slower response times, increased wear, and higher failure rates, highlighting the need to align model complexity with disk capacity.

Latency Comparison among Language Models





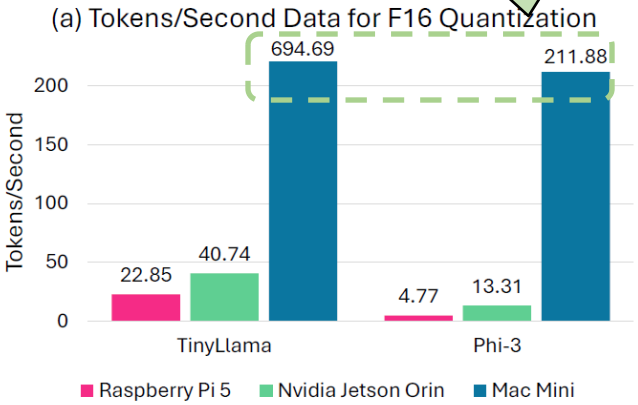
Exceptional latency because of High CPU fluctuations and I/O waits due to frequent memory swapping.

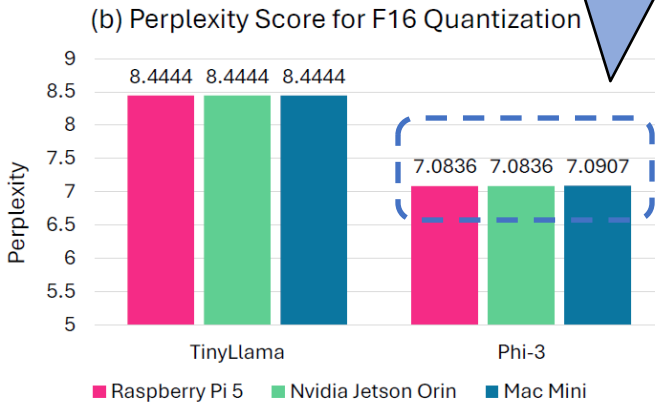
Powerful processors can generate a standard length of the prompt in a shorter time.

Performance of TinyLlama and Phi-3 for Half Precision(F16)

The Apple M1 vastly outperforms the ARM Cortex-A78AE in both 4-bit and 16-bit calculations due to its dedicated AI hardware (Neural Engine)

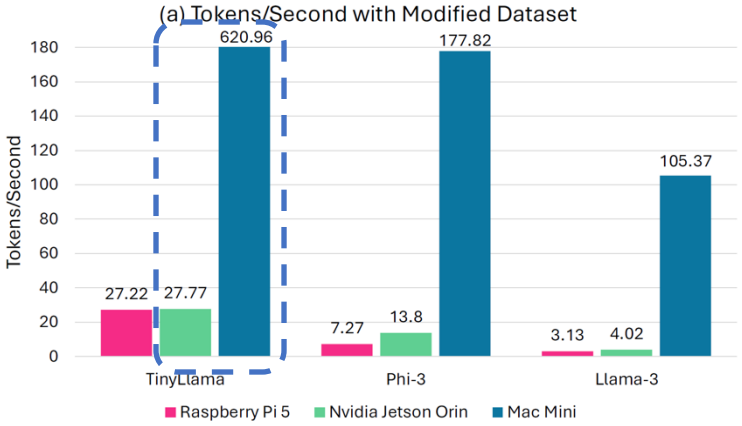
Comparatively newer SLM performs better in terms of perplexity



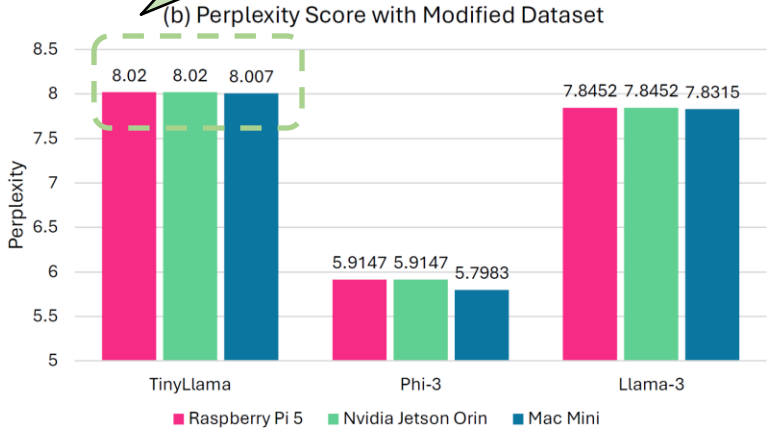


Throughput and Perplexity Comparison between SLM and LLM(4bit Quantization)

Processor that specially designed for AI application that perform relatively better in half precision.



The perplexity score doesn't change in terms of the environment, but 4-bit quantization reduces performance.



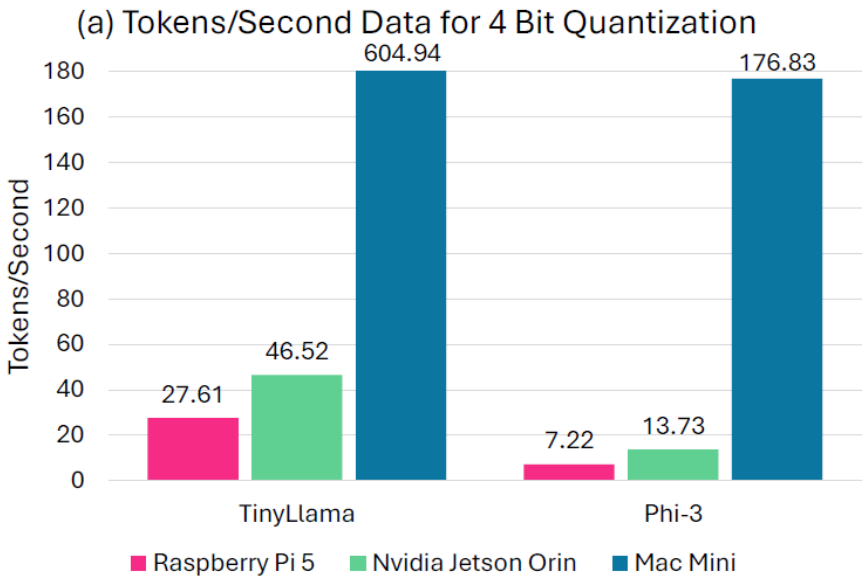
Outlines

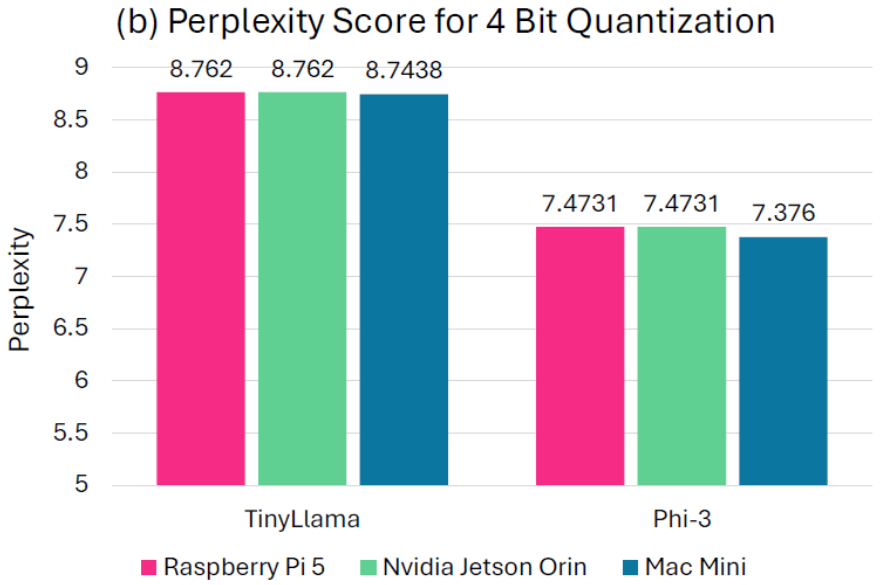
Conclusion and Takeaways

- SLMs provide feasible alternatives to LLMs for edge AI, with the choice of device influencing performance significantly.
- Proper memory and CPU management are crucial to prevent bottlenecks like high disk busy rates and frequent swapping.
- Insights from this study can guide developers in optimizing AI applications for diverse edge devices, improving latency, privacy, and energy efficiency.

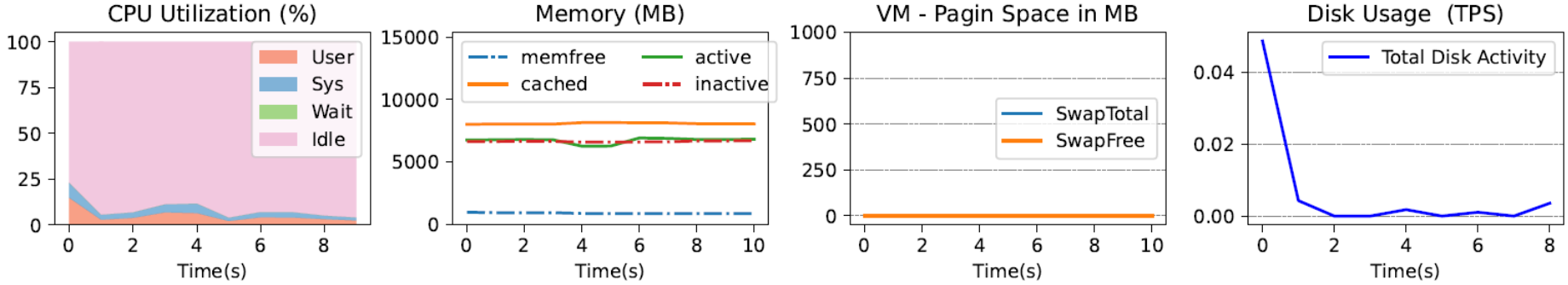

Q
&
A

Performance of TinyLlama and Phi-3 for 4-Bit Quantization





Performance Observation and Analysis of OpenELM



Results on Mac Mini: Stable CPU and memory usage with negligible swap and disk activity.

Insight: OpenELM's efficient resource footprint makes it well-suited for Apple devices and potentially mobile edge deployment