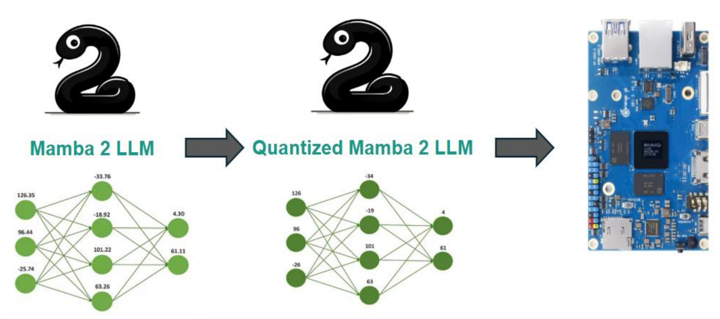
STATE SPACE MODEL OPTIMIZATION FOR EDGE AI

Objective

Compression of Mamba 2 state space model (SSM) for deployment on an Orange Pi 5 Ultra, while achieving <3% accuracy degradation and minimal latency for real world applications.

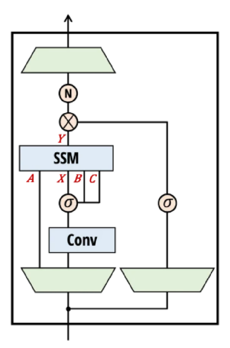


Mamba 2 & State Space Models

Key Metrix	Transformer	Mamba-2
Computational Complexity	O(n^2)	O(n)

Transformers have O(n^2) time complexity because each token compares itself with every other token in the sequence to compute attention. This means longer sequences require a lot more computation.

Mamba 2 has O(n) time complexity because it uses a fixed-size convolution to process each token by only looking at a few previous tokens. This makes it much faster and more scalable for long sequences.



The full model consists of 64 Mamba 2 SSM blocks, each of the which consist of the following layers:

- Input projection layer
  - Transforms input embeddings into a representation compatible with the SSM layer
- 1D convolutional layer + SILU activation layer
  - Introduces local dependencies prior to feeding the data into the SSM layer
- SSM specific layers
  - Efficiently maintain hidden state memory via a selective scan mechanism
  - Capture long-range dependencies
  - Optimize for low-resource hardware since the memory usage grows sub-linearly
- Normalization layer
  - Ensures stable training and convergence
- Activation layer + Output projection layer
  - Output projection maps features to logits for token prediction
  - Post-training activation clipping stabilize output values and improve robustness

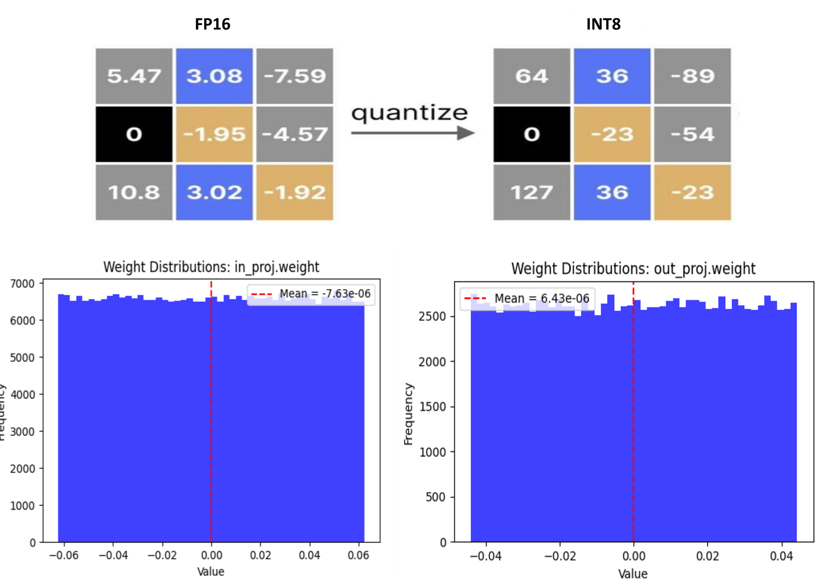
Hardware Requirements/Constraints : Orange Pi 5

- Orange Pi 5 Ultra
  - CPU: 8 Cores ARM64, 2.4GHZ
  - NPU (Neural Processing Unit): 6TOPS computing power
  - GPU: Mali-G610
  - DRAM: 32GB
- Deployment Tool: Llama.cpp
  - Efficient inference engine
  - Support: OpenCL and Vulkan
  - Supports .gguf model formats (C/C++ ecosystem)
- Challenges
  - Need to create custom Mamba 2 specific operators for NPU deployment
  - Support INT8 operation for Mali-G610

Model Compression and Deployment

Post-Training Weight-Only Quantization

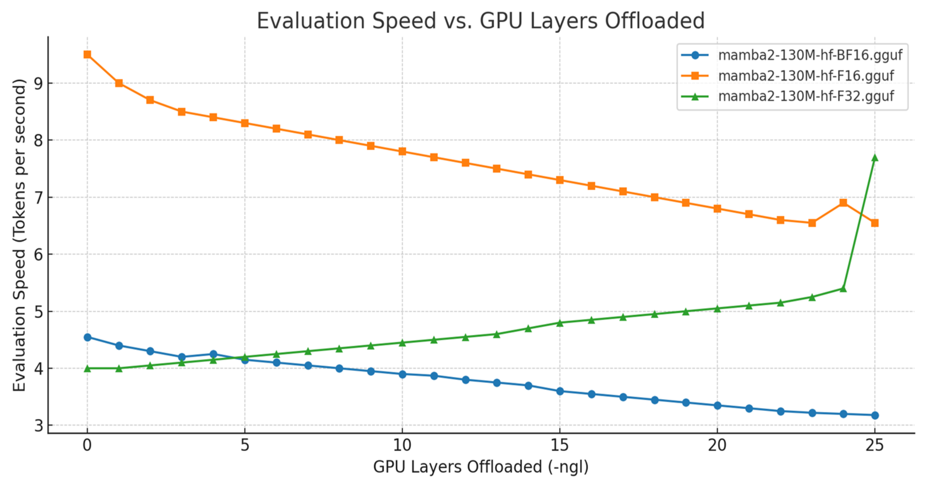
- A model compression technique that reduces the precision of neural network weights after training is complete.
- Quantized the linear layers (input and output projection layers) in all Mamba SSM blocks.
- Applied symmetric weight-only quantization because of uniform weight distribution in input and output projection layers.



Llama.cpp Model Deployment

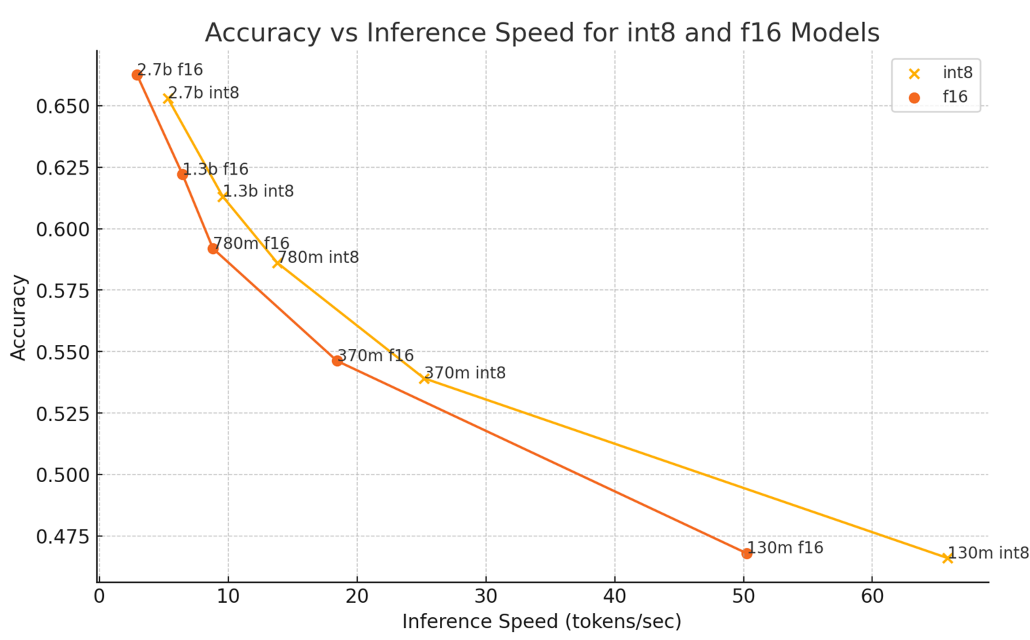
- Llama.cpp is used to convert our quantized models into GGUF format, so that it is compatible with the inference engine.

Model Deployment: CPU vs. GPU vs. NPU

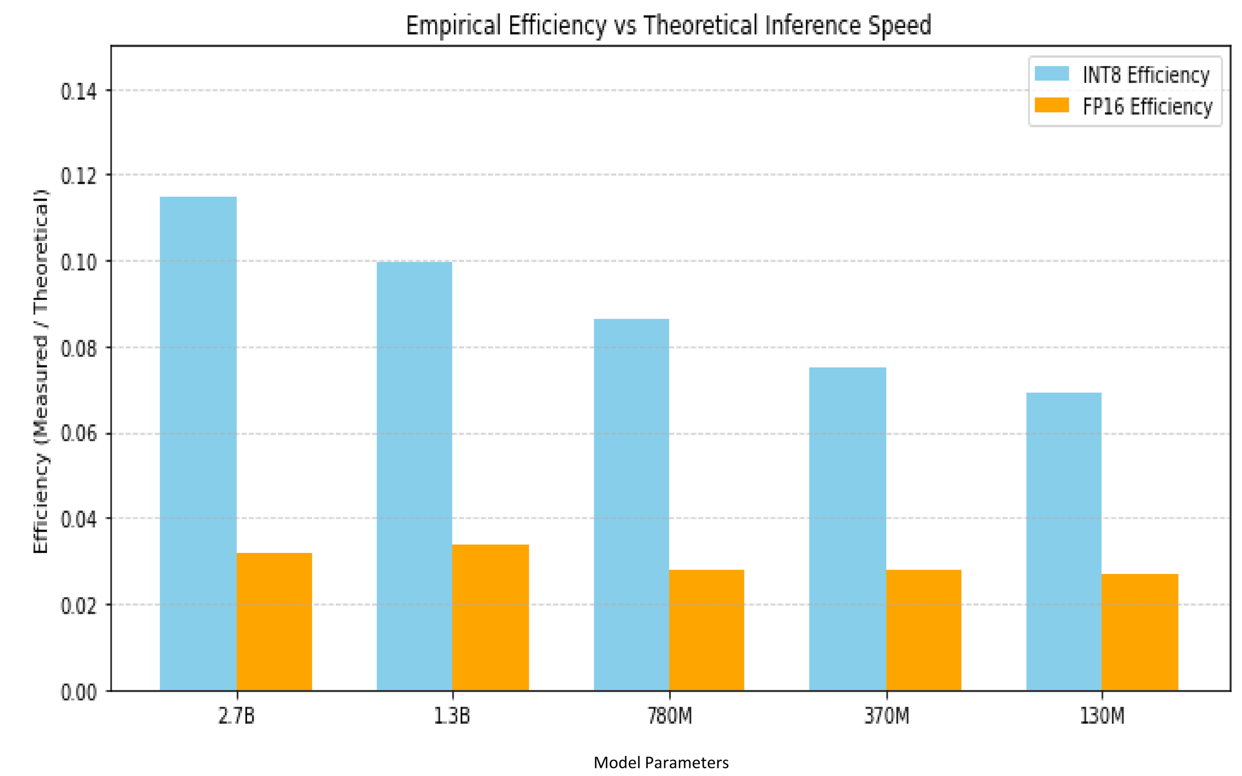


- Computation can be split between the Orange Pi's CPU and GPU.
- However, this experiment shows that only full-precision data (FP32) is processed faster by the GPU. For other data types, inference runs faster the more that computation is kept on the CPU only.
- With these results, we decide to run inference on the CPU only for our final product.
- The NPU does not support the Mamba 2 architecture and custom optimized operations.

Experimental Results: Inference

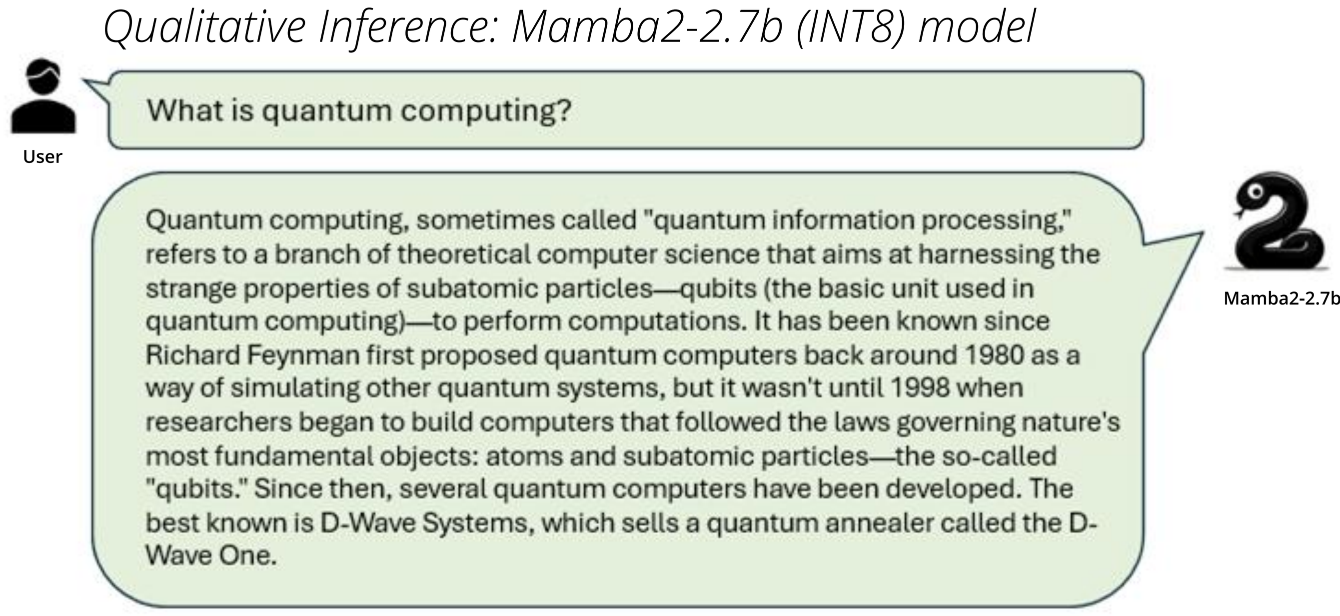


- Both quantized and unquantized models exhibit a clear accuracy vs. inference-speed trade-off.
- Larger models (more parameters and higher precision) deliver higher accuracy but run inference more slowly.
- Our final product is Mamba 2 with 2.7 billion parameters, quantized to INT8.
- This configuration achieves near-maximal accuracy while inferring at human reading speeds (4–7 tokens/sec).
- All model accuracy evaluations completed on LM Harness benchmark tasks: arc_easy (multiple-choice, commonsense reasoning), hellaswag (multiple-choice, sentence completion), lambada_openai (free response, model's text understanding), piqa (multiple-choice, physical commonsense reasoning)



- Theoretical upper bound for inference speed vs. our measured data shows that our INT8 quantized model has higher efficiency than the un-quantized FP16 model.
- It implies that there's still a gap between the theoretical bound and our model performance, which we can explore further to optimize.

Qualitative Inference: Mamba2-2.7b (INT8) model



Conclusion/Future Work

Summary of Results

- Quantized Mamba2-2.7b (INT8) model showed less than 3% accuracy degradation across LM Harness benchmark datasets.
- Quantized Mamba2-2.7b (INT8) model had an inference latency of around 185 ms/tokens (5.4 tokens/second).
- Quantized Mamba2-2.7b (INT8) model had peak memory usage of 2.87GB.

Future Work

- Exploration of lower precision quantization such as INT4 or INT2
- Exploration of mixed precision quantization to increase inference speed, while minimizing accuracy degradation.

References/Acknowledgements

[1] Dao T, Gu A. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. arXiv preprint arXiv:2405.21060. 2024 May 31.
[2] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752. 2023 Dec 1.
[3] Chiang HY, Chang CC, Frumkin N, Wu KC, Marculescu D. Quamba: A post-training quantization recipe for selective state space models. arXiv preprint arXiv:2410.13229. 2024 Oct 17.
[4] Das A, Raha A, Kundu S, Ghosh SK, Mathaikutty D, Raghunathan V. XAMBA: Enabling Efficient State Space Models on Resource-Constrained Neural Processing Units. arXiv preprint arXiv:2502.06924. 2025 Feb 10.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: Shashwat Verma, Sankalp Dayal, Radha Poovendran

SPONSOR: Amazon