intel labs

2025 Forty-Second International Conference on Machine Learning
Vancouver, Canada

Accelerating Linear Recurrent Neural Networks for the Edge with Unstructured Sparsity

Alessandro Pierro*[1,2], Steven Abreu*[1,3], Jonathan Timcheck [1], Philipp Stratmann [1], Andreas Wild [1], Sumit Bam Shrestha [1]

<alessandro.pierro@intel.com>

* Equal contribution
1 – Neuromorphic Computing Lab, Intel Corporation, USA
2 – Institute for Informatics, LMU Munich, Germany
3 – Bernoulli Institute & CogniGron, University of Groningen, Netherlands

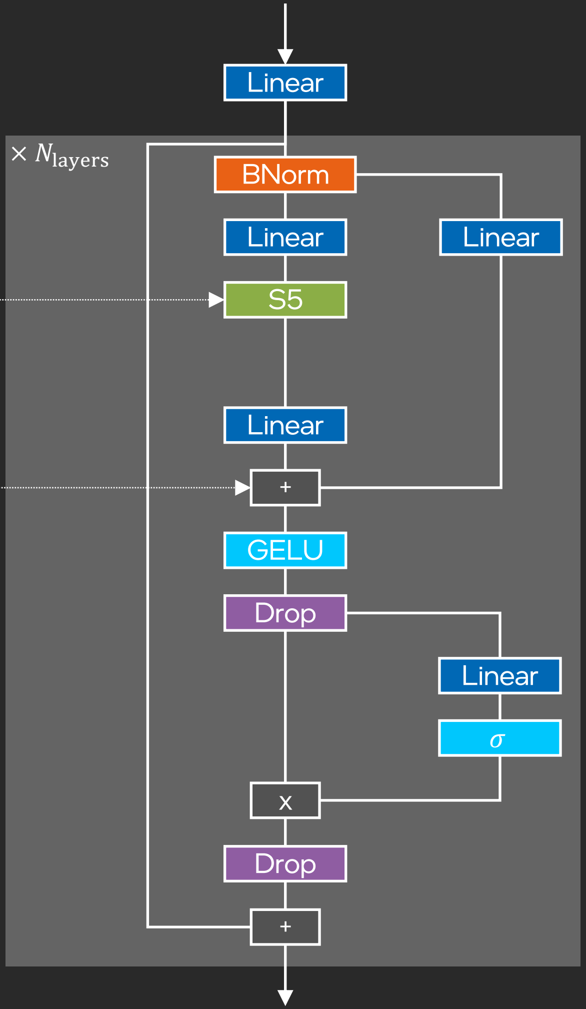
intel® ICML International Conference On Machine Learning

Linear RNNs enable efficient sequence modeling

- Competitive alternative to full self-attention
- Fast training with conv/parallel scan
- Low-latency inference in recurrent mode
- Strong signal processing capabilities
- Generalize to different inference rates

$$h_{t+1} = diag(A) \odot h_t + B^T u_{t+1}$$

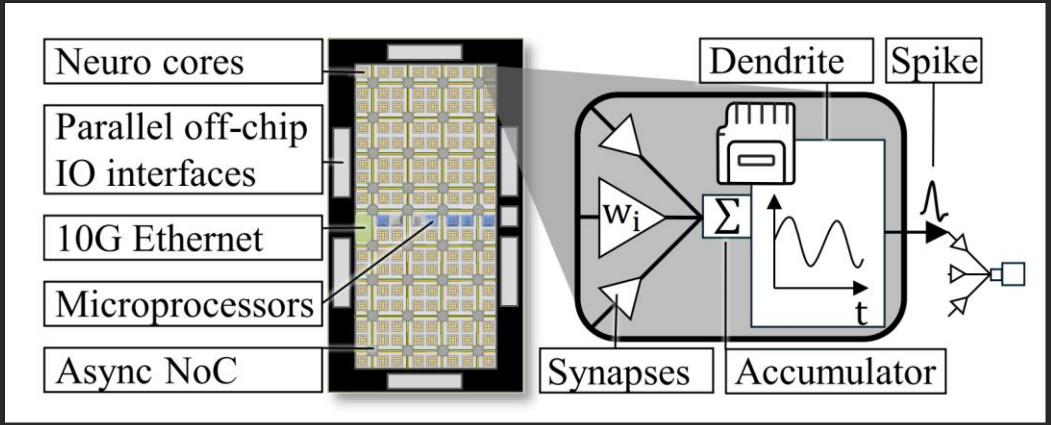
$$x_{t+1} = C^T h_{t+1} + diag(D) \odot u_{t+1}$$



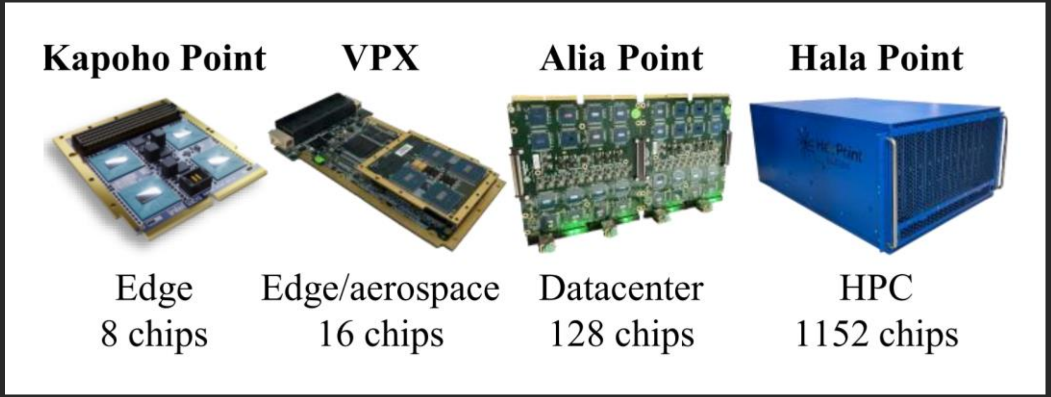
Tri Dao and Albert Gu. Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. ICML 2025, Vol. 235. JMLR.org, Article 399, 10041–10071.
Smith, J. T. H., Warrington, A., and Linderman, S. W.Simplified state space layers for sequence modeling, ICLR 2023, Kigali, Rwanda, May1-5, 2023

Neuromorphic hardware offers a compelling match for sparse linear RNNs

- Emerging neuromorphic chips provide energy-efficient support for:
  - Sparse matrix-vector multiply
  - Sparse activations with event-driven computation
  - Stateful neurons via compute-memory integration
- Example use cases
  - Audio denoising
  - Signal processing
  - Language modeling



Intel Loihi 2 computational model



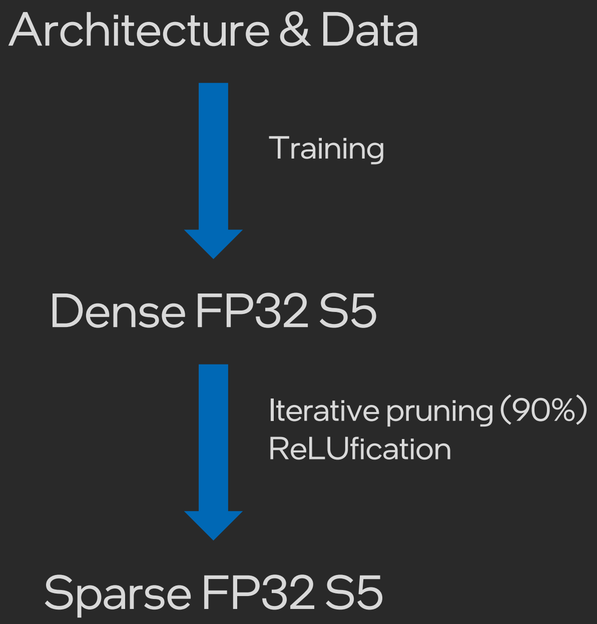
Intel Loihi 2 neuromorphic processor form factors

Research Questions

1. Do highly sparse linear RNNs outperform dense linear RNNs across different inference compute budgets?
2. Can fixed-point quantization compress sparse linear RNNs without damaging the network’s performance?
3. Can unstructured sparsity and fixed-point quantization be translated into latency and energy advantages on neuromorphic hardware?

S5 Compression Pipeline: Sparsity

- Weight sparsity
  - **Iterative Magnitude Pruning** gradually updates the sparsity masks during training to reach a target sparsity
  - It works better than one-shot pruning at high sparsity levels
- Activation sparsity
  - **ReLUfication**: replaced GELU non-linaerity with ReLU and introduced additional ReLUs before key linear layers
- Both interventions are applied with a single fine-tuning run, starting from a pre-trained dense model



Architecture & Data

Training

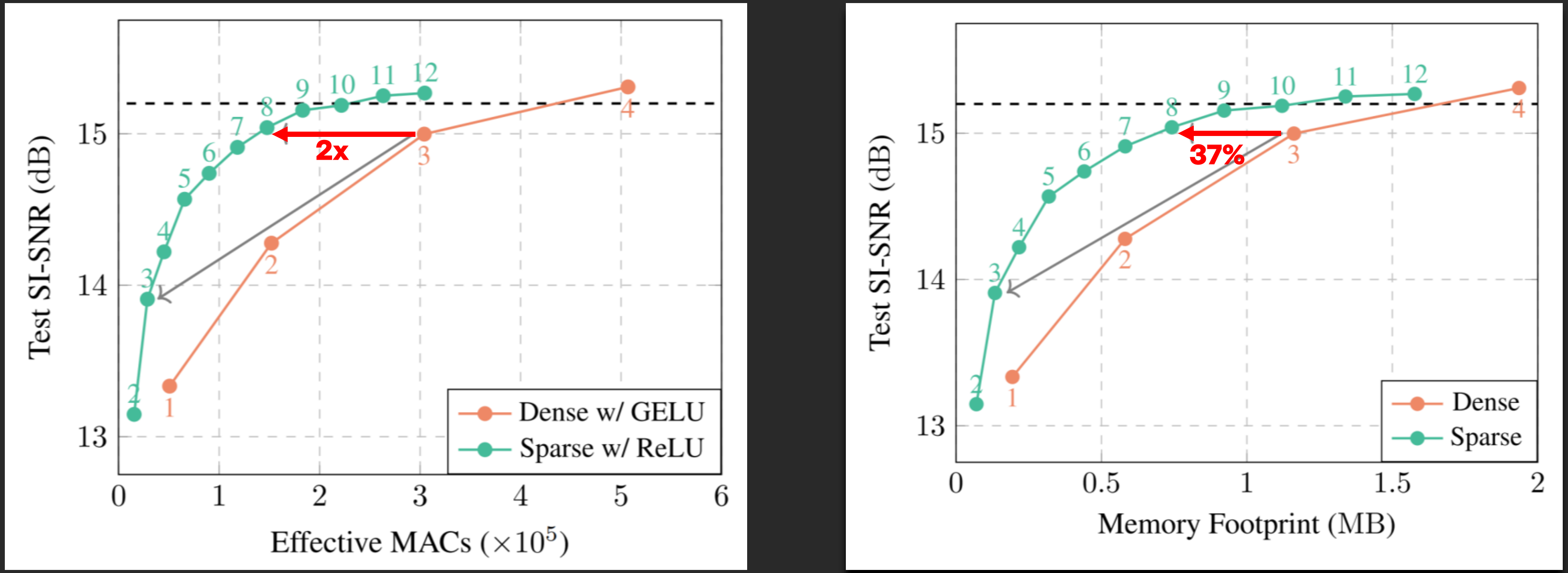
Dense FP32 S5

Iterative pruning (90%)
ReLUfication

Sparse FP32 S5

[2310.04564] ReLU Strikes Back: Exploiting Activation Sparsity in Large Language Models
[2304.14082] JaxPruner: A concise library for sparsity research
[1710.01878] To prune, or not to prune: exploring the efficacy of pruning for model compression

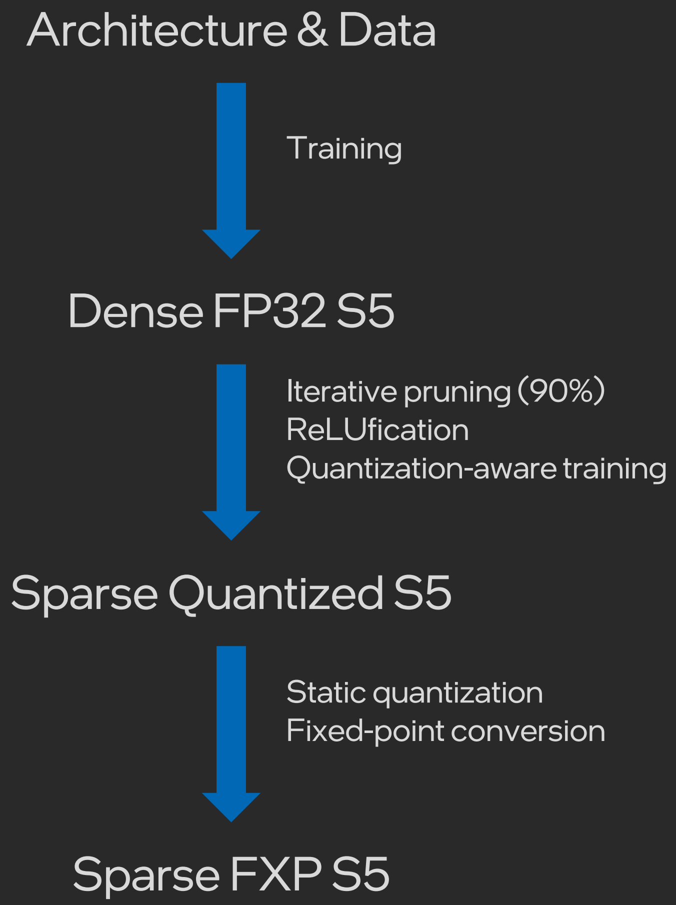
Key Result 1: unstructured sparse models are at the efficiency-performance Pareto front



Pareto fronts for S5 network audio denoising quality (SI-SNR) as a function of effective compute (left) and memory footprint (right) on the Intel N-DNS test set. S5 networks with weight and activation sparsity (green) exhibit a large domain of Pareto optimality versus dense S5 networks (orange). Number annotations enumerate increasing S5 dimensionality configurations, from 500 k to 4 M parameters. Dashed horizontal like marks SI-SNR of Spiking-FullSubNet XL, the previous state-of-the-art model. The horizontal arrows highlight models used for hardware deployment, the diagonal arrows highlight models of the same width.

S5 Compression Pipeline: Quantization

- Loihi 2 requires fully quantized computation
- Precision: 8bit for weights, 16bit for activations and diagonal components
- Three steps
  - Quantization-aware training (optional)
  - Conversion to static quantization
  - Fixed-point arithmetic (simulates execution on the chip)



Architecture & Data

Training

Dense FP32 S5

Iterative pruning (90%)
ReLUfication
Quantization-aware training

Sparse Quantized S5

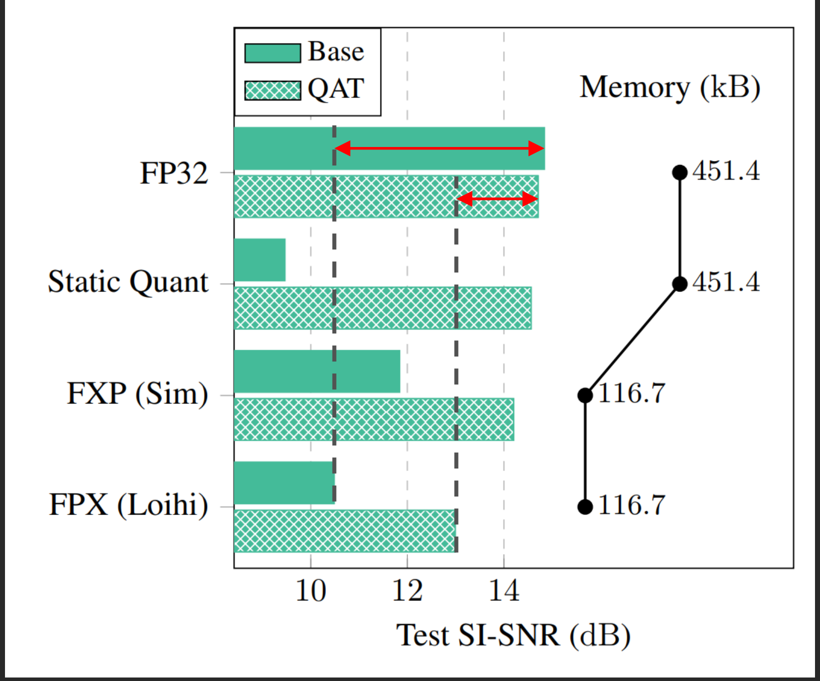
Static quantization
Fixed-point conversion

Sparse FXP S5

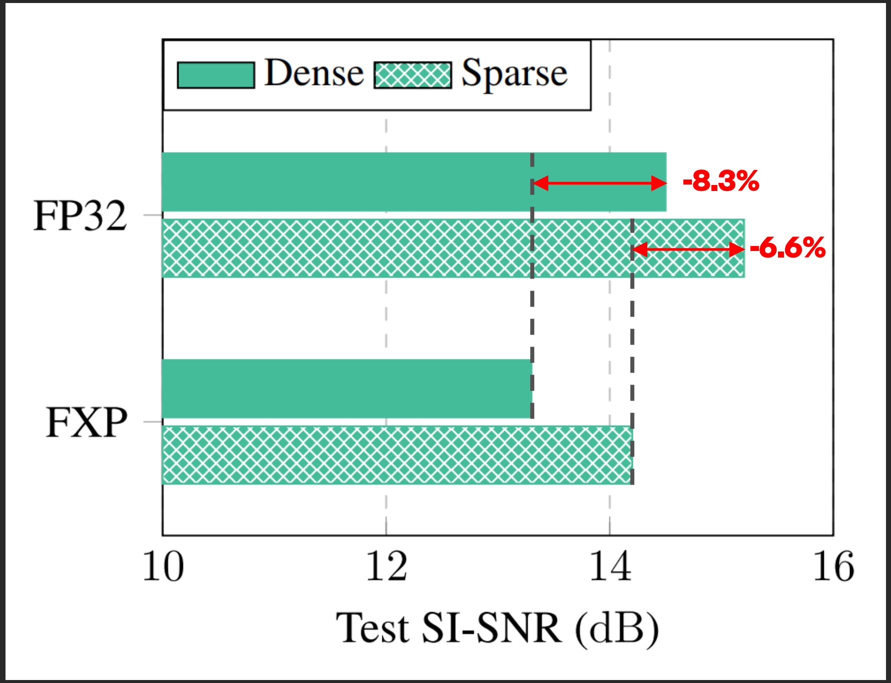
[2406.09477] Q-S5: Towards Quantized State Space Models

Key Result 2: sparse models can be converted to fixed-point with small accuracy degradation

QAT significantly reduces the FP to FXP gap



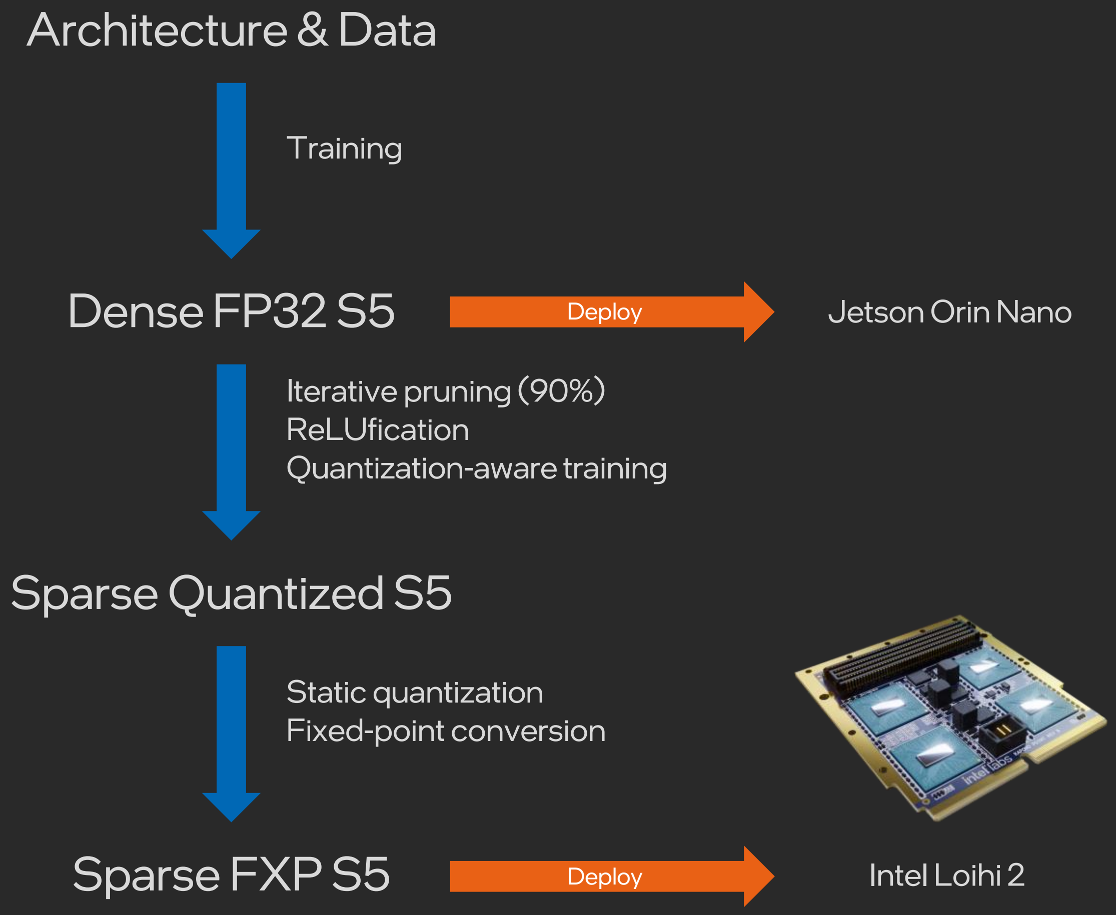
Sparse wider models are more resilient to quantization



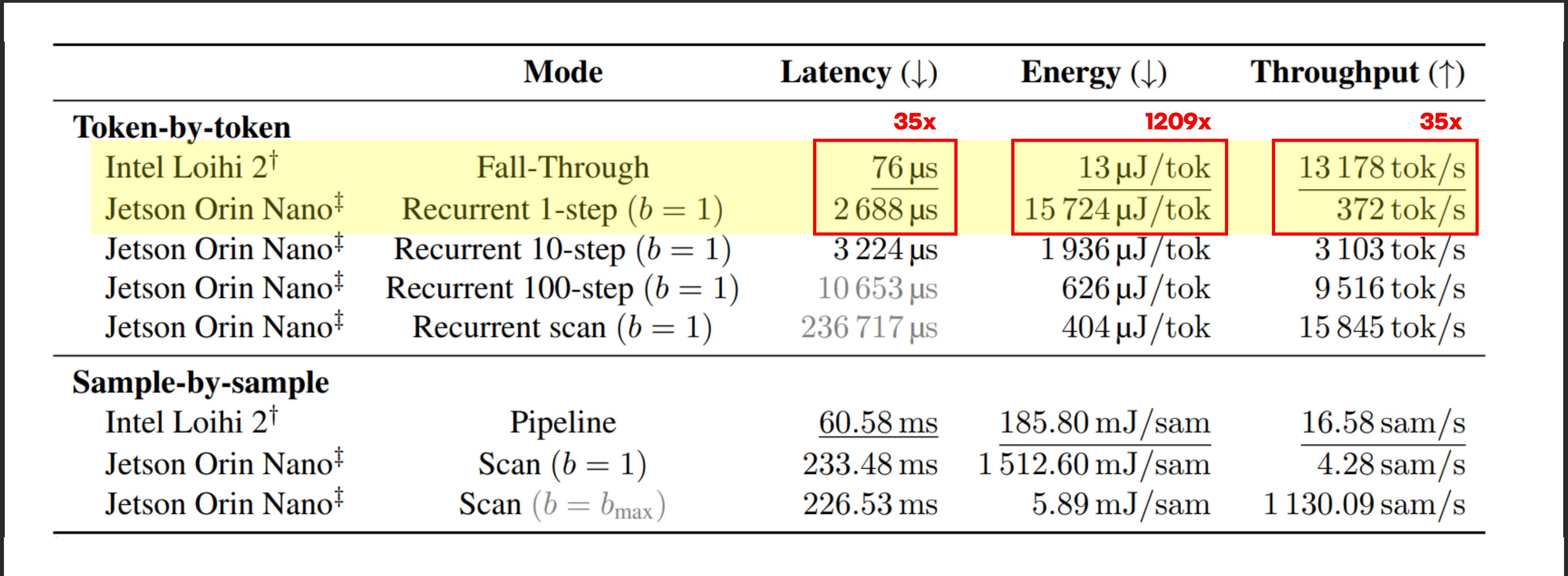
Impact of quantization interventions on Test SI-SNR and memory footprint, with and without quantization-aware training, for model variant sparse-6.

S5 Compression Pipeline: Hardware Deployment

- Real-time audio de-noising requires each token to be processed within 8ms
  - Parallelization on sequence length not possible!
- We implement the sparse S5 model on Loihi using the new NxKernel API
- Benchmark latency, energy, and throughput against a dense FP32 JAX implementation on a Jetson Orin Nano



Key Result 3: theoretical gains from sparsity can be translated in power and performance advantages



	Mode	Latency (↓)	Energy (↓)	Throughput (↑)
Token-by-token		35x	1209x	35x
Intel Loihi 2[†]	Fall-Through	76 µs	13 µJ/tok	13 178 tok/s
Jetson Orin Nano[‡]	Recurrent 1-step ($b = 1$)	2 688 µs	15 724 µJ/tok	372 tok/s
Jetson Orin Nano[‡]	Recurrent 10-step ($b = 1$)	3 224 µs	1 936 µJ/tok	3 103 tok/s
Jetson Orin Nano[‡]	Recurrent 100-step ($b = 1$)	10 653 µs	626 µJ/tok	9 516 tok/s
Jetson Orin Nano[‡]	Recurrent scan ($b = 1$)	236 717 µs	404 µJ/tok	15 845 tok/s
Sample-by-sample				
Intel Loihi 2[†]	Pipeline	60.58 ms	185.80 mJ/sam	16.58 sam/s
Jetson Orin Nano[‡]	Scan ($b = 1$)	233.48 ms	1 512.60 mJ/sam	4.28 sam/s
Jetson Orin Nano[‡]	Scan ($b = b_{\text{max}}$)	226.53 ms	5.89 mJ/sam	1 130.09 sam/s

Power and performance results∗ . The Loihi 2 is running a sparse and quantized S5 model, while the Jetson Orin Nano is running a smaller dense S5 model that reaches similar test performance. All measurements are averaged over 8 random samples from the test set, each containing 3750 steps.

[†] Loihi 2 workloads were characterized on an Oheo Gulch system with N3C1-revision Loihi 2 chips running NxCore 2.5.8 and NxKernel 0.2.0 with on-chip IO unthrottled sequencing of inputs. Researchers interested to run S5 on Loihi 2 can gain access to the software and systems by joining Intel's Neuromorphic Research Community. [‡] Jetson workloads were characterized on an NVIDIA Jetson Orin Nano 8GB running Jetpack 6.2, CUDA 12.4, JAX 0.4.32, using the MAXN SUPER power mode; energy values are computed based on the TOT power as reported by jtop 4.3.0. The batch size bmax = 256 was chosen to be the largest that fits into memory. ∗Performance results are based on testing as of January 2025 and may not reflect all publicly available security updates; results may vary