University of St Andrews

Multi-DNN Inference on Edge SoCs

Jiawei Luo, Di Wu, Simon Dobson, Blesson Varghese
University of St Andrews

What is an Edge SoC?

- Edge **System-on-Chip** (SoC) is a highly **integrated**, **power-efficient** semiconductor chip designed for edge devices

- Integrates **heterogeneous** processors:
  - CPU, GPU, and NPU

- Each processor has different performance **characteristics**:
  - CPU: flexible, general-purpose
  - GPU: high parallelism
  - NPU: efficient for DNN inference

- **Shared** memory and **limited** resources

→ Requires efficient **scheduling** across processors

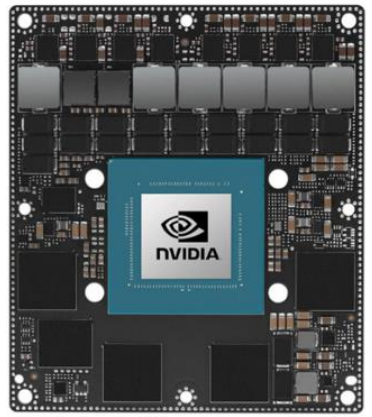


Fig. 1: NVIDIA Jetson AGX Orin.

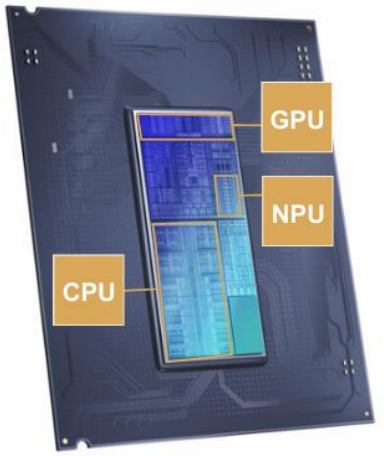


Fig. 2: Intel desktop.

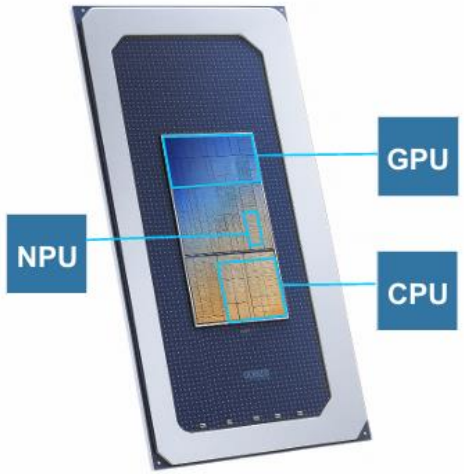
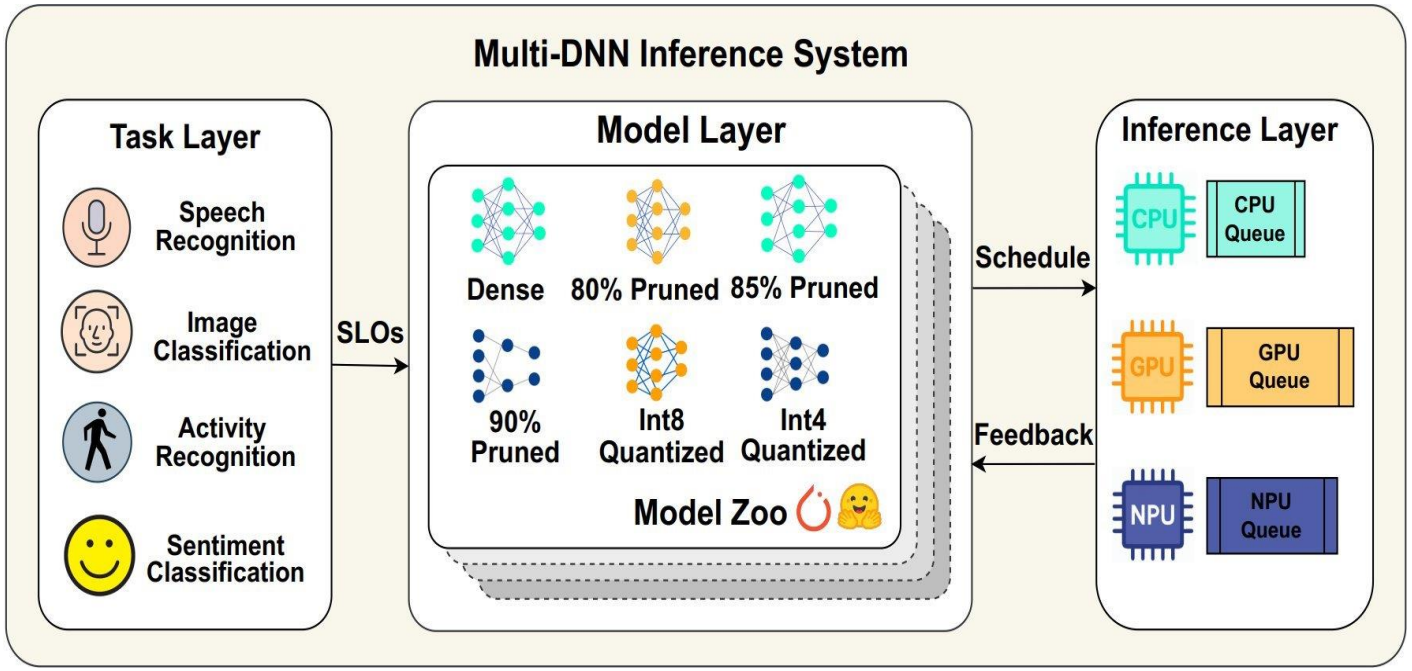


Fig. 3: Intel laptop.

What is a multi-DNN inference system?



What do users care about?

- Meeting Service Level Objectives (SLOs): **Latency and accuracy**
- Maximizing efficient resource utilization: **Throughput (tasks/s)**

Fig. 4: An Augmented Reality (AR) use case with a multi-DNN inference system.

Optimization Strategies in Existing Systems

Model-level

- Provide multiple sparse model variants
  - pruned: remove **less important** weights
  - quantized: use **lower-precision** weights
- Select a suitable variant for each SLO

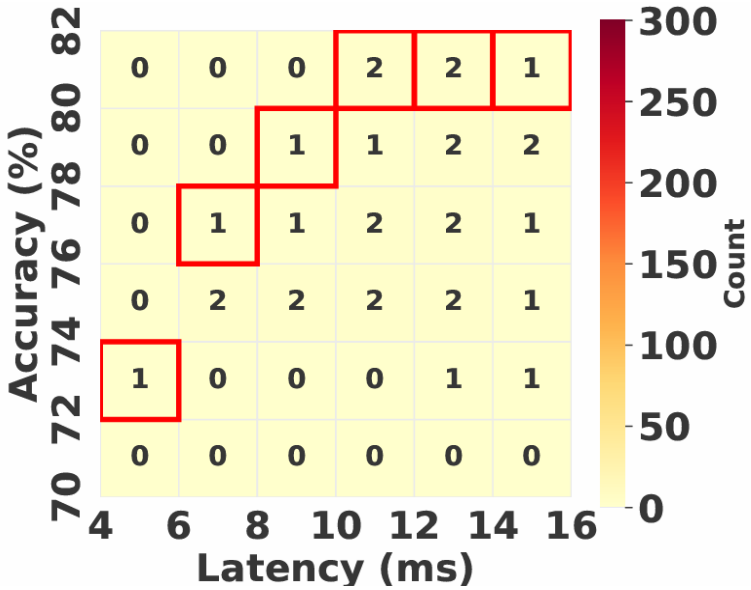


Fig. 5: ResNet101 variants in the accuracy–latency space

System-level

- **Partition** DNNs into subgraphs
- Execute subgraphs across different processors
- Use **pipeline-style** execution to improve concurrency

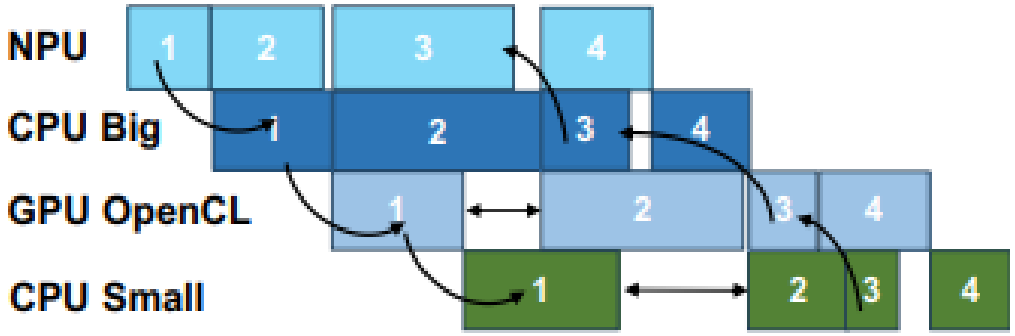


Fig. 6: Multi-DNN pipeline execution

Limitations of Existing Systems

Limited model variants

- **Insufficient** latency–accuracy trade-off space
- Cannot satisfy stricter SLOs

Sparsity–Hardware Mismatch

- Subgraph placement **ignores** sparsity support
- Eliminates potential latency benefits

Resulting in:

- **High SLO violation rate**
- **Low overall throughput**

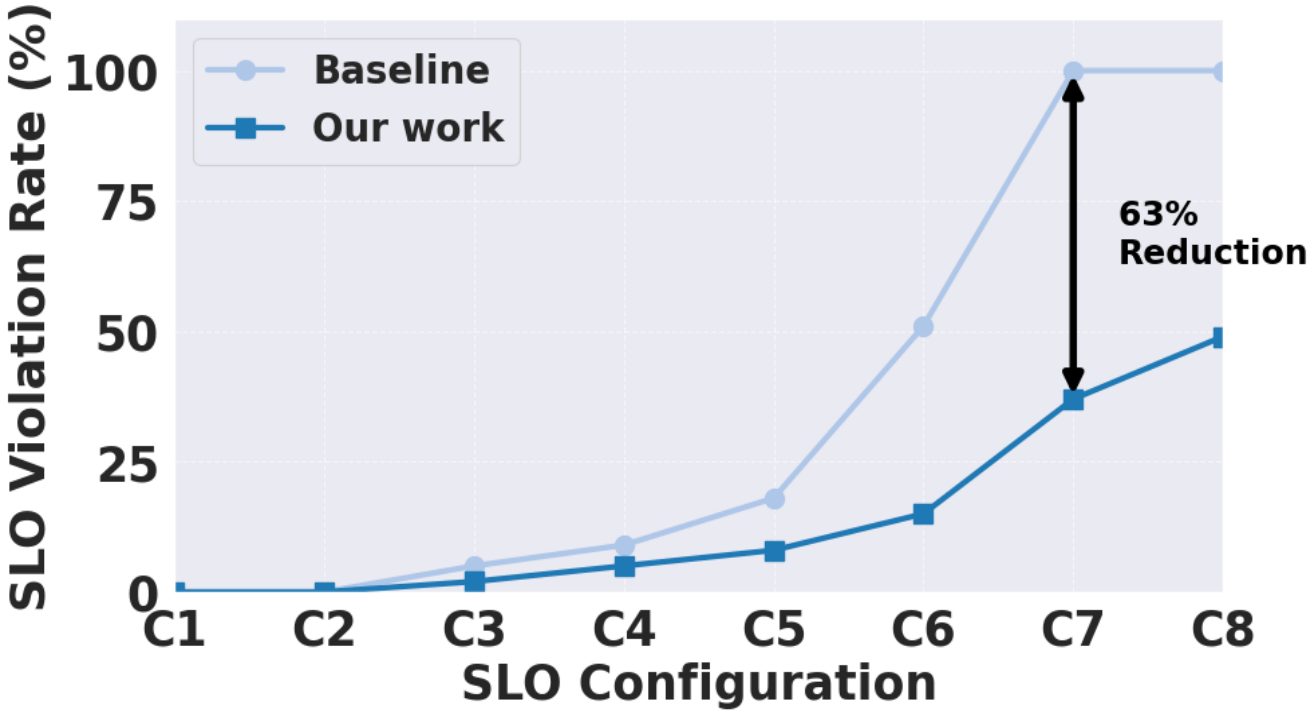


Fig. 7: SLO Violation Rate.
(**larger** indices (e.g., C8) correspond to **stricter** requirements.)

How do we address these limitations?

Expanding the variant space via **model stitching**

- Construct stitched variants by recombining subgraphs from sparse variants

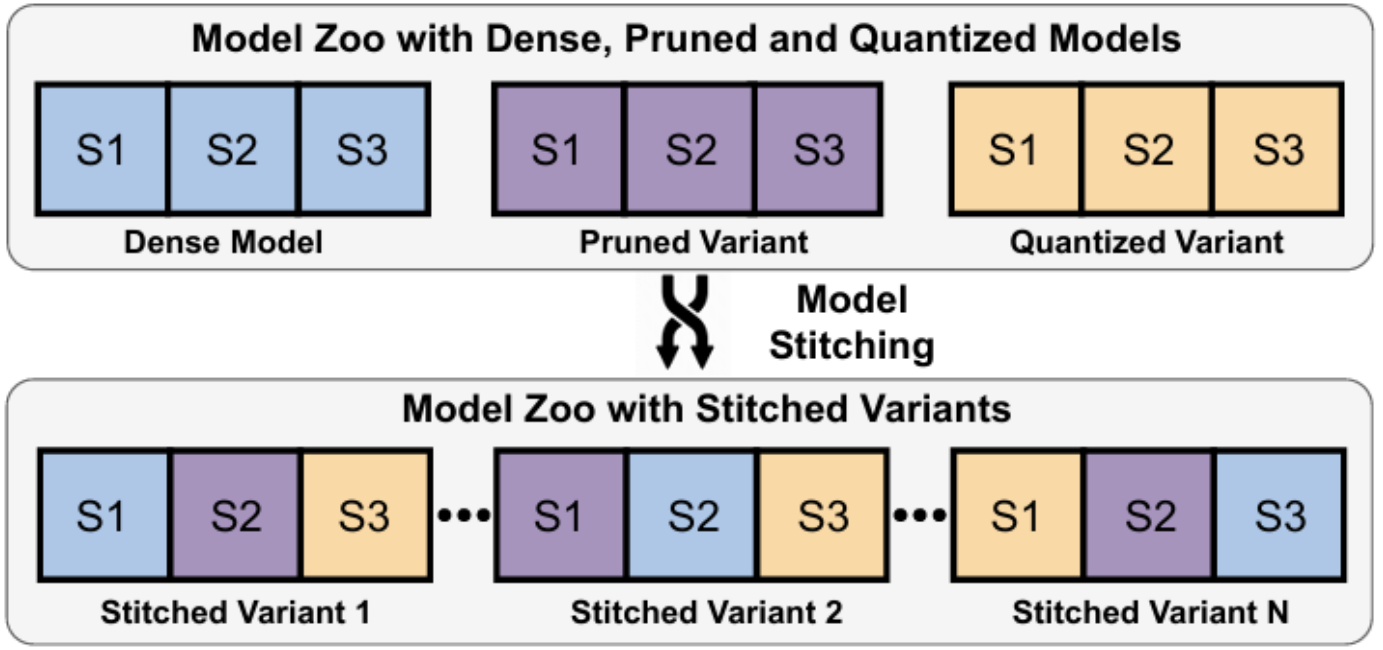
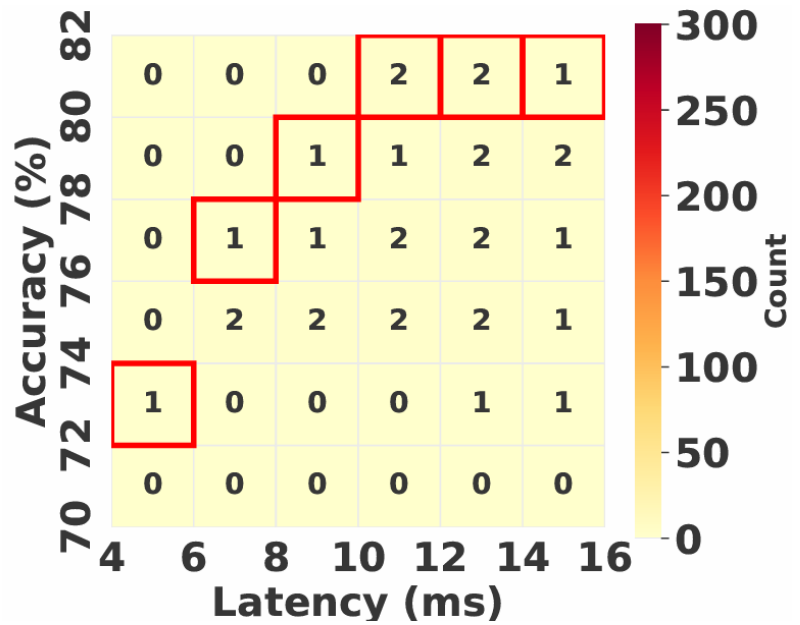


Fig. 8: Model stitching.

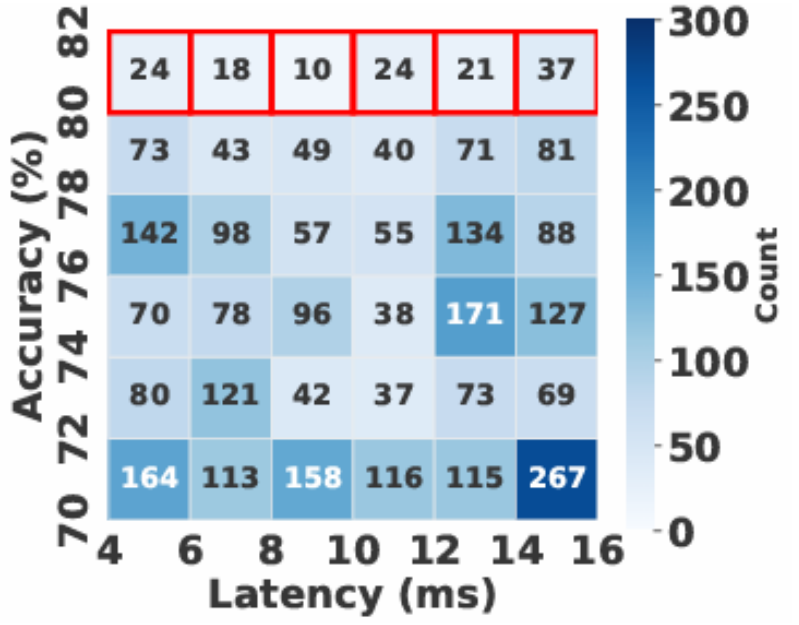
How do we address these limitations?

Expanding the variant space via **model stitching**

- Significantly expand the latency–accuracy trade-off space



(a) Without Model Stitching



(b) With Model Stitching

Fig. 9: ResNet101 variants in the accuracy–latency space.

Key Challenges in Stitched Multi-DNN Inference

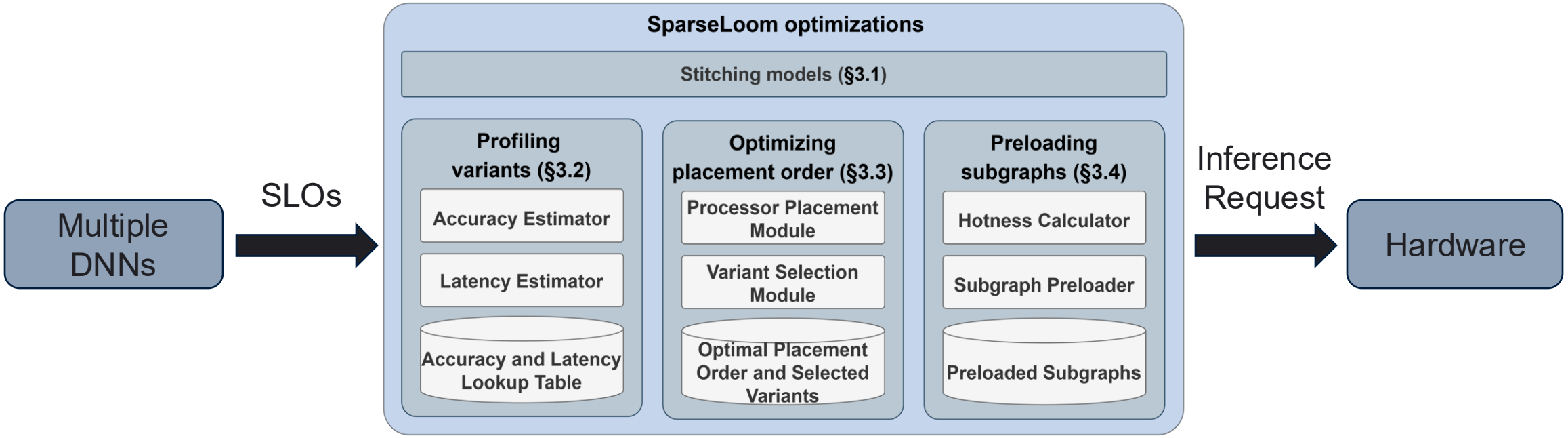


Fig. 10: The SparseLoom architecture.

Link to paper: https://arxiv.org/abs/2603.09642
Link to code: https://github.com/blessonvar/SparseLoom

Experimental setup

- **Edge SoCs:**
  - Intel Core Ultra 7 265K with **CPU, GPU, NPU** (Desktop)
  - Intel Core Ultra 5 135U with **CPU, GPU, NPU** (Laptop)
  - Jetson AGX Orin with **CPU, GPU**.
- **Tasks:**
  - Speech recognition
  - Image classification
  - Activity classification
  - Sentiment classification.

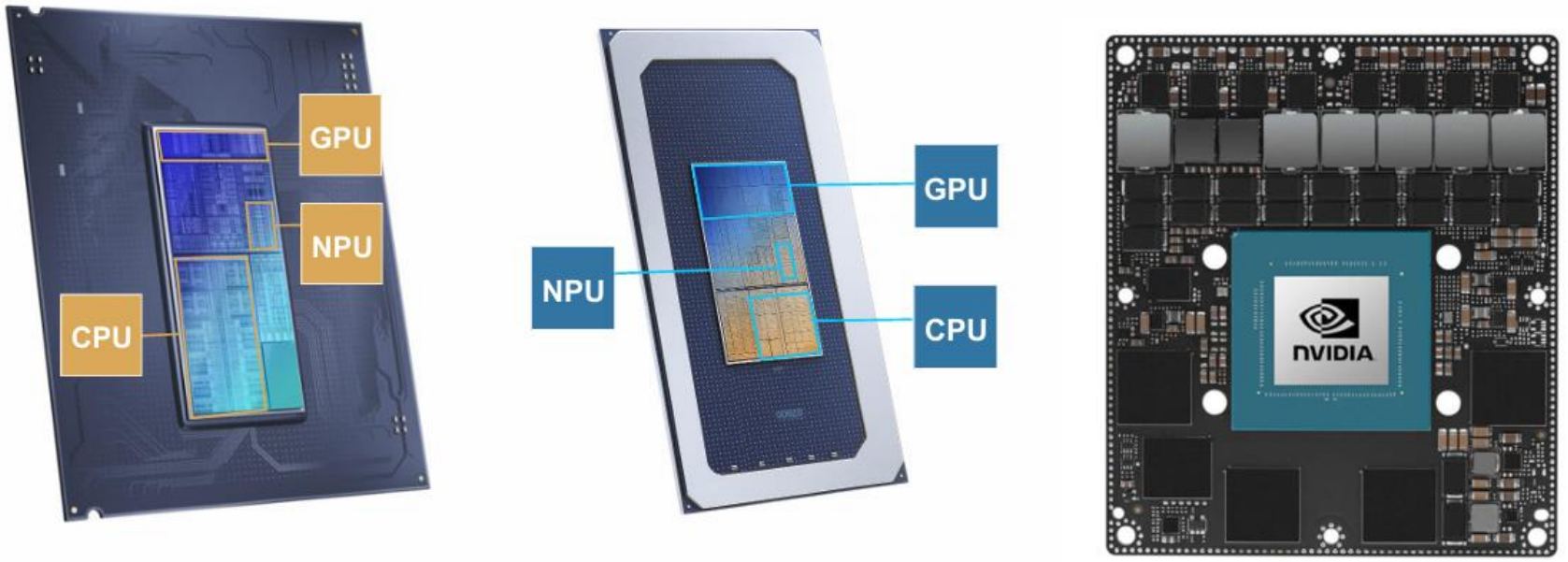


Fig. 11: Heterogeneous edge SoCs.
Intel desktop [Left], laptop [Middle], and NVIDIA Jetson AGX Orin [Right].

Results

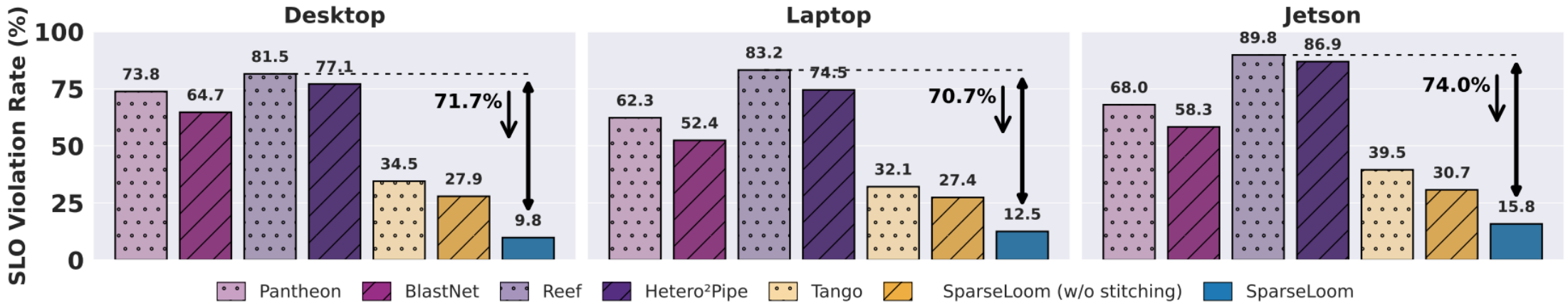


Fig. 12: **SLO violation rates** of SparseLoom and baselines across three SoCs.

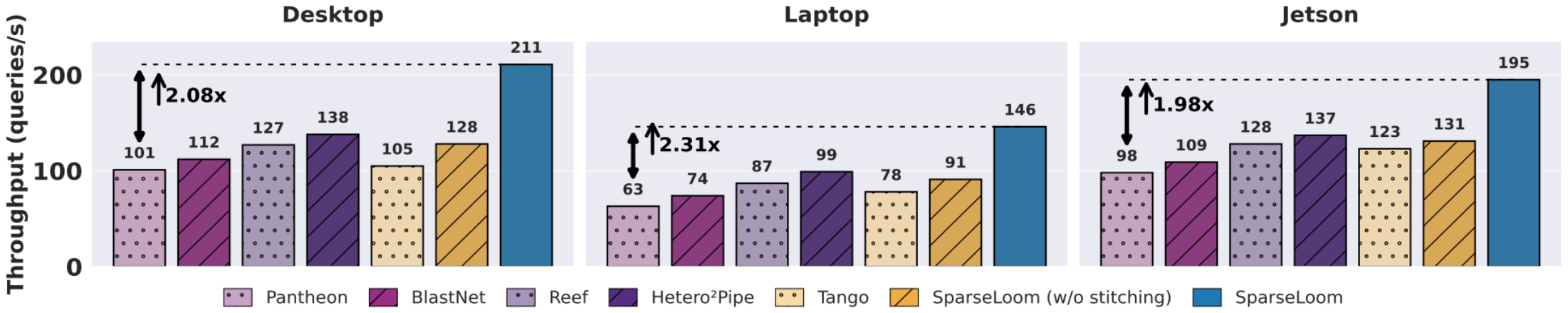


Fig. 13: **Inference throughput** of SparseLoom and baselines across three SoCs.


University of
St Andrews
Q & A
www.st-andrews.ac.uk