



Supercharging Generative AI: KleidiAI™, PyTorch, and Arm® Neoverse

Nikhil Gupta
Arm Ltd.

Arm Neoverse: Powering GenAI at Scale

- **Arm Neoverse and Armv9:** Foundation for Cloud, HPC and GenAI
- **Scalable Vector Extension 2 (SVE2):** Unlocking Data-Level Parallelism
- **Massive Compute** and Hardware Acceleration for AI and HPC
- **Widespread Cloud Adoption:** Neoverse in Major Hyperscalers

Clouds everywhere are deploying Arm-based servers



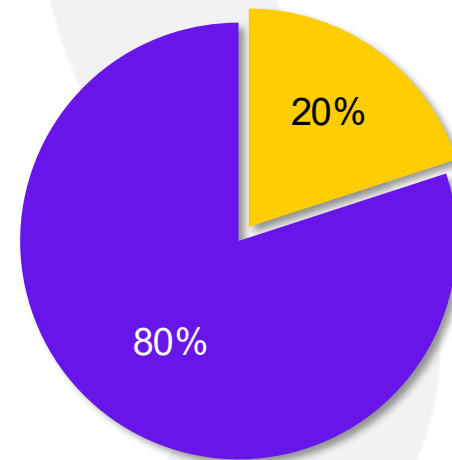
Inferencing is Key to GenAI Adoption (1)

Training is the tip of iceberg

- Training = Only **15-20%** of AI workloads

The Growth of AI is driven by inference

- Inference = **~80-85%** AI workloads*
- Real-world AI applications (GenAI, Recommender) are **inference heavy**



■ Training ■ Inference



Natural Language
Processing



Automatic Speech
Recognition



Object Detection



Recommender
System



Image Generation

Inferencing is Key to GenAI Adoption (2)

Efficient Inference needs **high throughput**, **low latency** and **scalable performance**

Challenges with Efficient Inferencing on CPUs

- **Memory bandwidth bottlenecks:** Large models require high memory access rates
- **Limited parallelism:** CPUs must maximize vectorization and threading efficiency
- **Handling large models:** Scaling inference efficiently across CPU cores

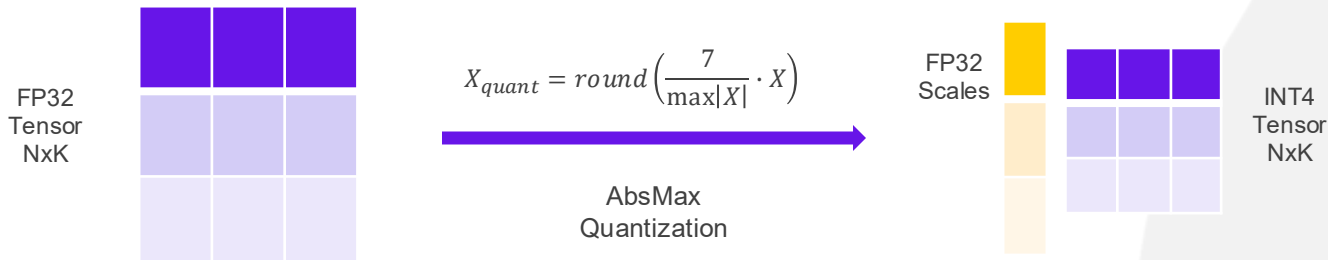
Solution Requirements

- **Fast and Memory-Efficient:** Optimize vectorization and bandwidth usage
- **CPU-optimized:** Leverage Arm Neoverse features (low precision instructions)
- **Scalable** across model types, sizes and newer platforms

Supercharging GenAI at scale

Low-bit Quantization

- Converts high-precision numbers (e.g., FP32) to lower-precision formats (e.g., INT4).
- Performs computation directly in the quantized form.
- **8× higher data density:** INT4 model can fit eight times more parameters into the same cache space as FP32.
- **Higher effective memory bandwidth:** more data moved per cycle, less traffic to main memory.
- Enables Arm **low-bit intrinsics (i8mm, dotprod)** to achieve much **higher compute throughput** than FP32 matmuls.
- Possible to **preserve near-original model accuracy** with LLMs



Scalability Through PyTorch Integration

- **PyTorch-native integration** ensures **seamless adoption** and **ease of use**.
- Leverages **PyTorch's massive ecosystem** — tools, libraries, and active community support.
- **Out-of-the-box compatibility** for newly trained GenAI models — no custom rework needed.
- PyTorch is **the Leading training framework** for GenAI (e.g., Llama, Gemma, Stable Diffusion).
- **Scalable** across model sizes and deployment targets — from raspberry-pies to arm neoverse.



Leveraging KleidiAI & PyTorch on Arm Neoverse

Introduction to KleidiAI



- **Optimized AI Micro Kernels:** Provide low-level, highly optimized microkernels designed for ARM CPUs
- **Quantized Matmul Kernels:** Designed for GenAI use-cases
- **Fast Packing Routines:** Offers fast weight & input packing kernels for memory efficient computations
- **Independent & Stateless:** No memory allocation & dependencies

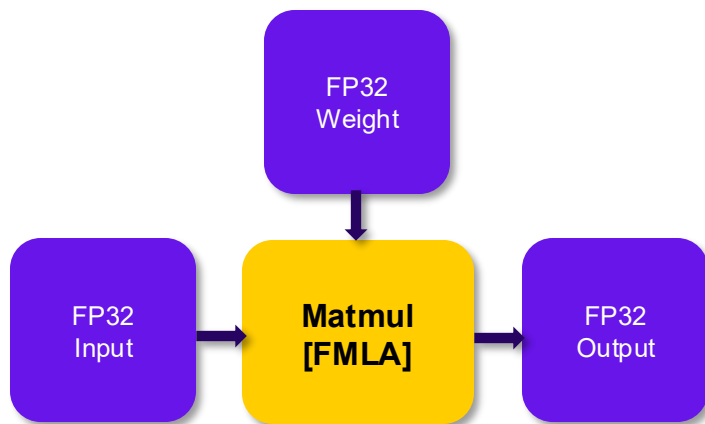
Target Users

- AI Framework Developers
- AI SDK Developers
- AI pipeline Developers

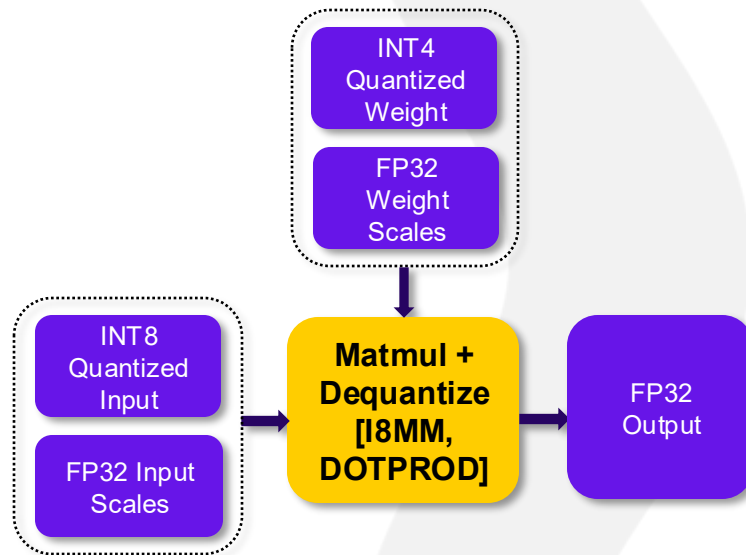
KleidiAI Domain

- Classic ML
- GenAI
- Agentic AI

KleidiAI 4-bit Matmul

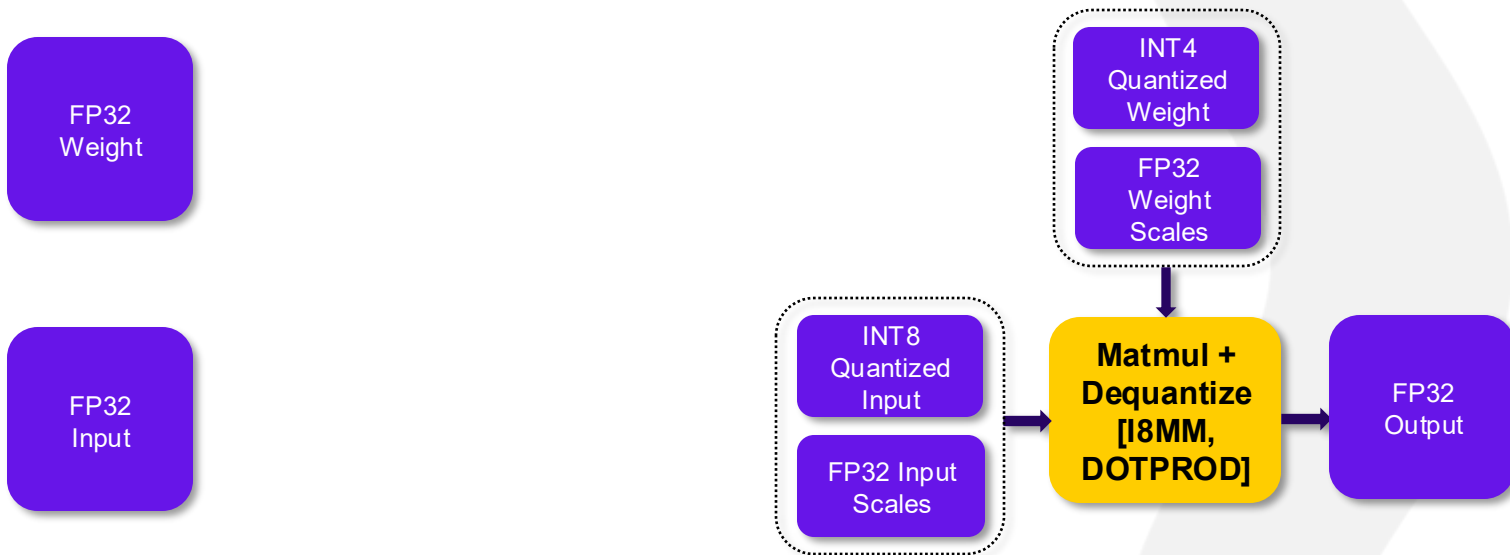


FP32 Matmul

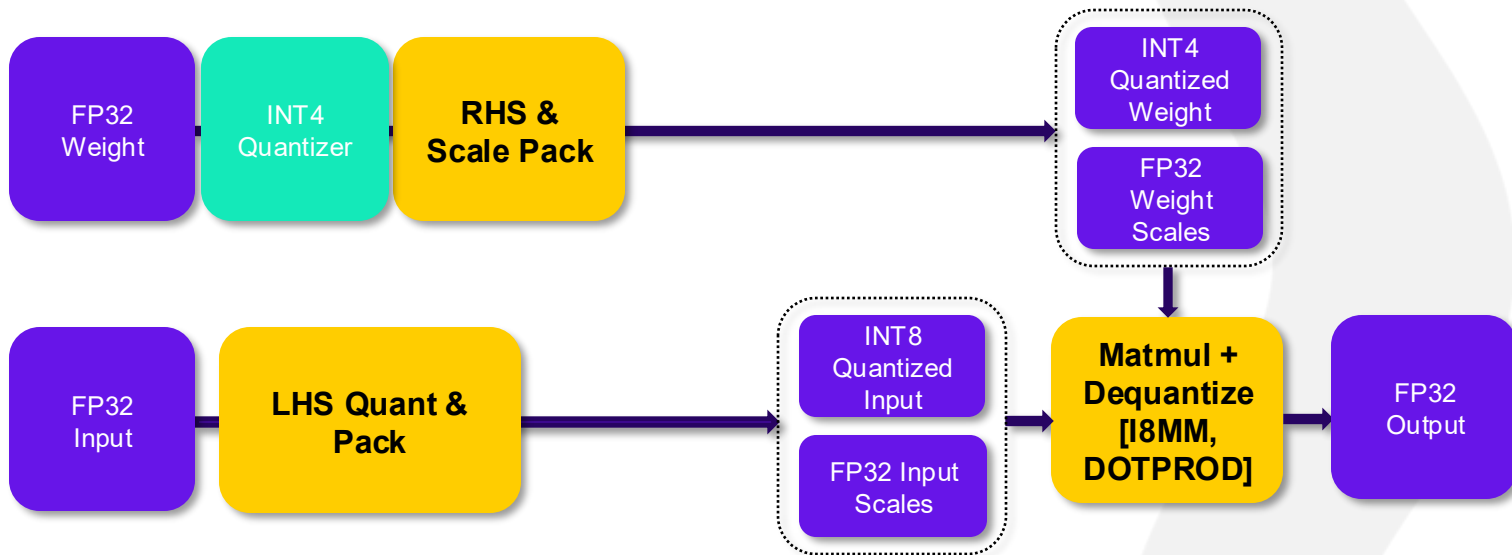


INT4 KleidiAI Matmul

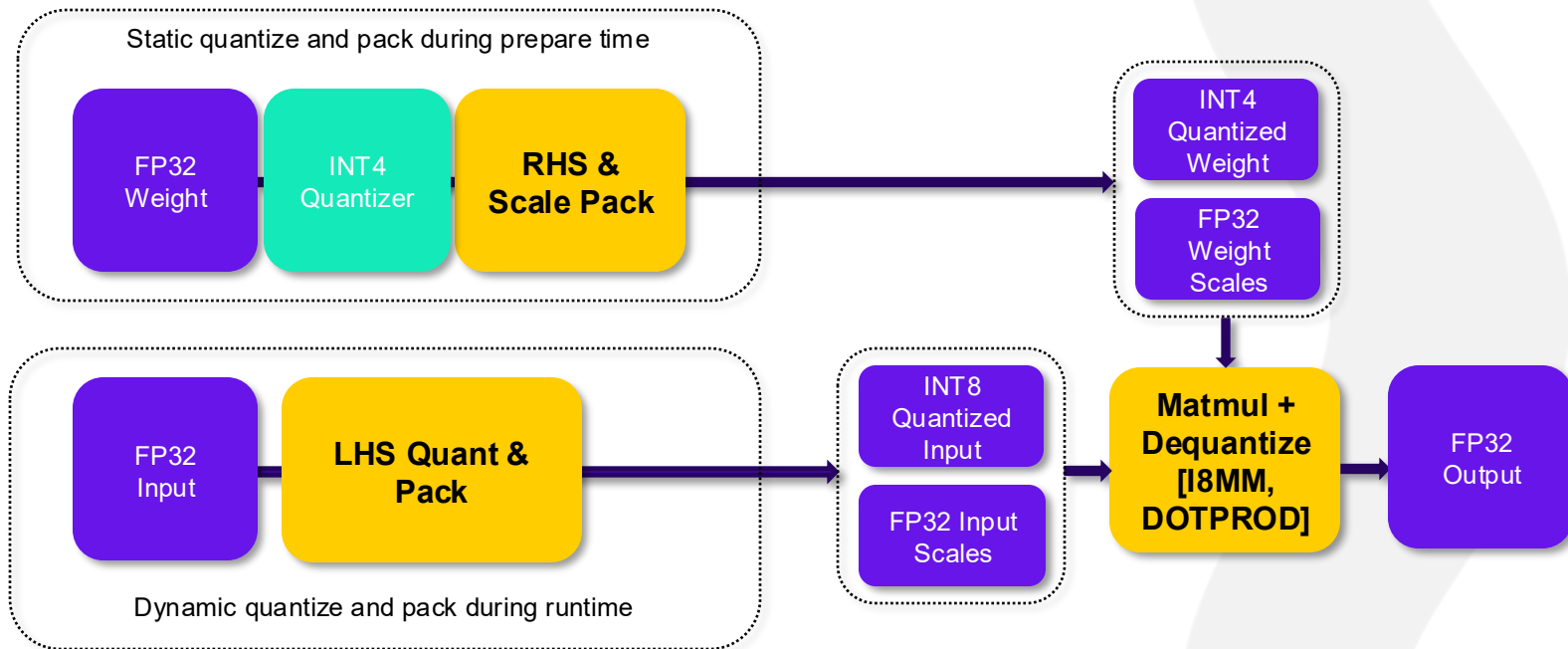
Input and Weight Packing



Input and Weight Packing



Input and Weight Packing

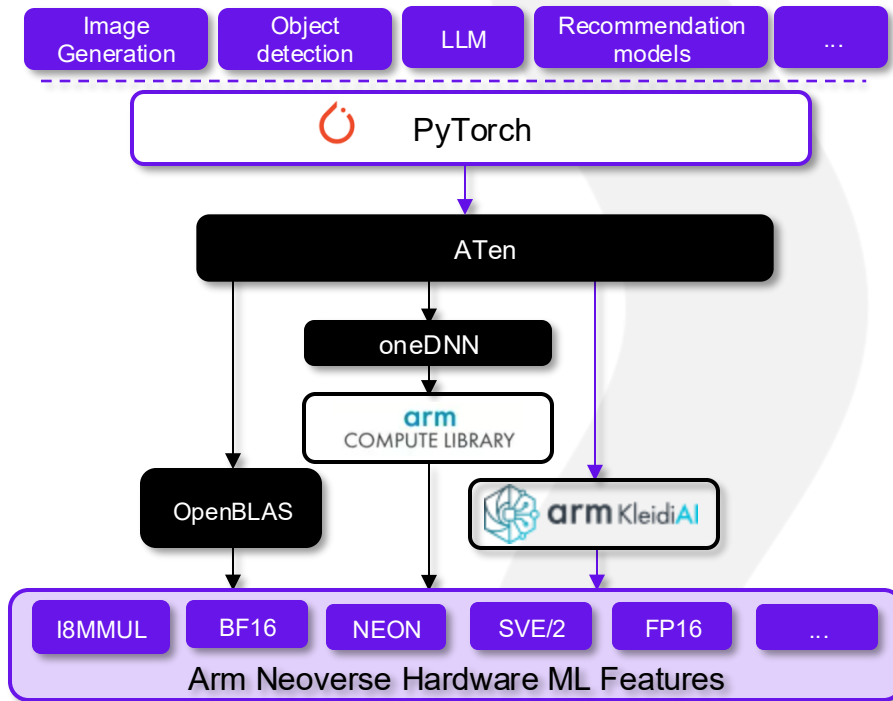


KleidiAI Integration into PyTorch

- **Integrated directly into PyTorch backend** via custom ATen ops.
- Introduces **two new 4-bit quantized ops**:

`torch.ops.aten._dyn_quant_pack_4bit_weight`

`torch.ops.aten._dyn_quant_matmul_4bit`



TorchAO - Simplifying Quantization

- **One API call** to quantize any PyTorch model
- Seamless **Integration with PyTorch + KleidiAI**
- Precise **control over model accuracy**
- **Layer-wise quantization** support

```
torchao_quantize.py

quantize_(
    my_model,
    int8_dynamic_activation_intx_weight(
        weight_dtype=torch.int4,
        granularity=PerGroup(32),
        has_weight_zeros=True,
        weight_mapping_type=MappingType.SYMMETRIC_NO_CLIPPING_ERR,
        layout=PackedLinearInt8DynamicActivationIntxWeightLayout(target="aten"),
    ),
)
```

Text Generation Speedups

```
User: The lemon tree produces a pointed oval yellow fruit. Botanically this is a hesperidium, a modified berry with a tough, leathery rind. The rind is divided into an outer colored layer or zest, which is aromatic with essential oils, and an inner layer of white spongy pith.
```

```
Assistant: [thinking]
```

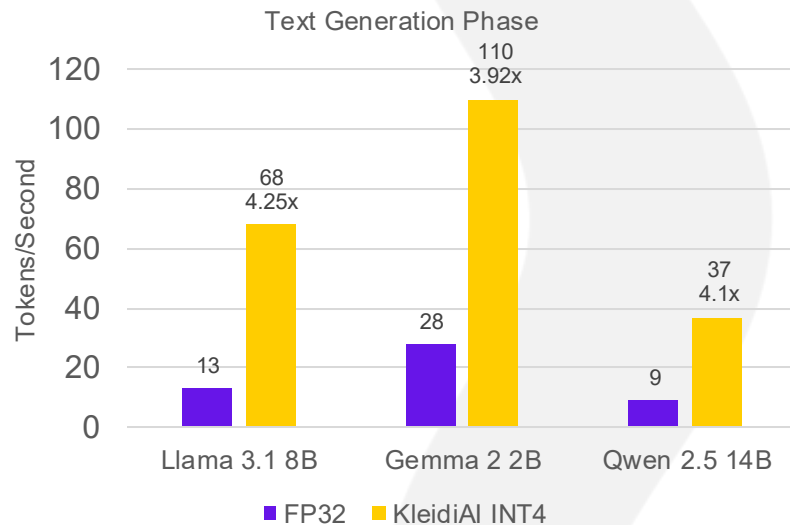
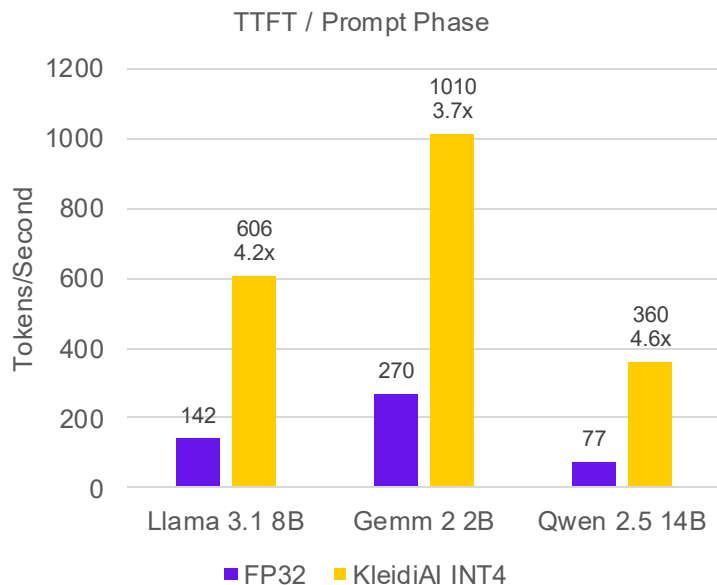
13 tok/s | Llama 3.1 8B FP32
PyTorch

```
User: The lemon tree produces a pointed oval yellow fruit. Botanically this is a hesperidium, a modified berry with a tough, leathery rind. The rind is divided into an outer colored layer or zest, which is aromatic with essential oils, and an inner layer of white spongy pith.
```

```
Lemons need a minimum temperature of around 7 °C, so they are not hardy year-round in temperate climates, but become hardier as they mature. Citrus require minimal pruning by trimming overcrowded branches, with the tallest branch cut back to encourage bushy growth. Throughout summer, pinching back tips of the most vigorous growth assures more abundant canopy development. As mature plants may produce unwanted, fast-growing shoots called water shoots, these are removed from the main branches at the bottom or middle of the plant. There is reputed merit in the tradition of urinating near a lemon tree. Lemons need a minimum temperature of around 7 °C, so they are not hardy year-round in temperate climates, but become hardier as they mature. Citrus require minimal pruning by trimming overcrowded branches, with the tallest branch cut back to encourage bushy
```

68 tok/s | Llama 3.1 8B INT4
PyTorch + KleidiAI

Performance Impact with KleidiAI



Arm Neoverse 2 | 32 Threads | PyTorch Compile

Key Takeaways

- **Arm Neoverse:** Designed for AI at Scale
- **Inferencing:** The Driver of GenAI
- **Low-Bit Quantization:** Core to Performance Gains for GenAI
- **PyTorch Integration:** Mature Ecosystem with ease of Use at Scale

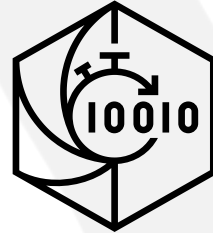
Resources & References

- **Arm Neoverse 2** <https://www.arm.com/products/silicon-ip-cpu/neoverse/neoverse-v2>
- **KleidiAI** <https://gitlab.arm.com/kleidi/kleidiai>
- **PyTorch** <https://github.com/pytorch/pytorch>
- **TorchAO** <https://github.com/pytorch/ao>
- **Arm Learning Path** <https://learn.arm.com/learning-paths/servers-and-cloud-computing/pytorch-llama/>

Nikhil Gupta

Senior Software Engineer | Arm™

nikhil.gupta2@arm.com



arm
Kleidi





Thank You!