ECE-GY 9143 B

Vision Model Optimization with Quantization and Efficient Attention

Team:

Jayraj Pamnani (jmp10051)

Puneeth Kotha (pk3058)

Motivation & Problem

- ViT-L/16 offers strong accuracy but is computationally heavy.
- Large model size and high VRAM usage limit deployment.
- Attention computation is a major bottleneck under standard SDPA.

Goal: Improve efficiency without altering ViT architecture.

Solution Approach

- Apply 4-bit & 8-bit weight quantization (bitsandbytes).
- Replace standard attention with FlashAttention-2 for efficient IO-aware attention.
- Use LoRA to stabilize fine-tuning under low precision.
- Evaluate 5 model variants using a unified benchmarking pipeline.

Value

Achieves 4× model compression, up to 40% latency reduction, and significant VRAM savings, while preserving accuracy suitable for production deployment.

TECHNICAL CHALLENGES

Heavy Memory Footprint

- ViT-L/16 baseline size ≈ 1157 MB
- High VRAM usage during inference

Accuracy Drop in Low Precision

- 4-bit quantization destabilizes deeper layers
- Requires selective precision retention

Expensive Attention Computation

- Multi-Head Self-Attention dominates FLOPs
- Inefficient kernels on standard SDPA

Kernel-Level Performance Bottlenecks

- Slower CUDA kernels in quantized models
- Need to isolate attention kernel cost

Approach (Model Design & Variants)

Model Baseline:

ViT-L/16:

- 24 encoder blocks
- Patch size: 16×16
- Hidden dimension: 1024
- Classification head for 102 classes

Variants:

- Full precision baseline
- 4-bit quantized (SDPA)
- 8-bit quantized (SDPA)
- 4-bit quantized + FlashAttention2
- 8-bit quantized + FlashAttention2

Notes:

- All models share the exact same architecture.
- Only weight precision + attention kernel change.
- Ensures fair comparison across all variants.

Approach (Training & Optimization)

Training Setup

- Dataset: 102-class Flower dataset
- Fine-tuning: LoRA (r=8, α=32) applied to attention and MLP layers
- Mixed precision training with fp16
- 6 epochs on A100 (~31s per epoch)

FlashAttention-2 Integration

- Optimized attention kernel
- Reduced memory reads/writes
- Effective on quantized models

4-bit / 8-bit Weight Quantization

- Reduces model size by 4–8×
- Weight-only quantization
- Helps lower VRAM and memory bandwidth

Parameter Efficient Fine-Tuning (LoRA)

- Fine-tuning only low-rank adapters
- Stabilizes low-precision models
- Low training cost

Results Summary

Model Accuracy

- Baseline ViT-L/16: 59.54%
- 8-bit (SDPA/FlashAttn2): 59.66–59.90%
- 4-bit variants: 53.3–53.4% (expected accuracy drop)

Latency / Runtime

- Fastest: Baseline 16.35 ms
- 4-bit SDPA: 56.7 ms
- 4-bit FlashAttn2: 62.0 ms
- 8-bit FlashAttn2: 113.8 ms

VRAM Usage

- Baseline: 1177 MB
- 4-bit SDPA: 173 MB (~85% reduction)
- 8-bit SDPA: 458 MB

Efficiency Improvements

- 4-bit quantization reduces model size from 1157 MB → 146 MB (≈ 8× smaller)
- 8-bit reduces to 290 MB (≈ 4× smaller)

- **8-bit quantization + SDPA provides the best accuracy–efficiency balance.**
- **4-bit achieves highest compression with moderate accuracy cost.**

Experimental Evaluation (1/4) - Benchmark Table

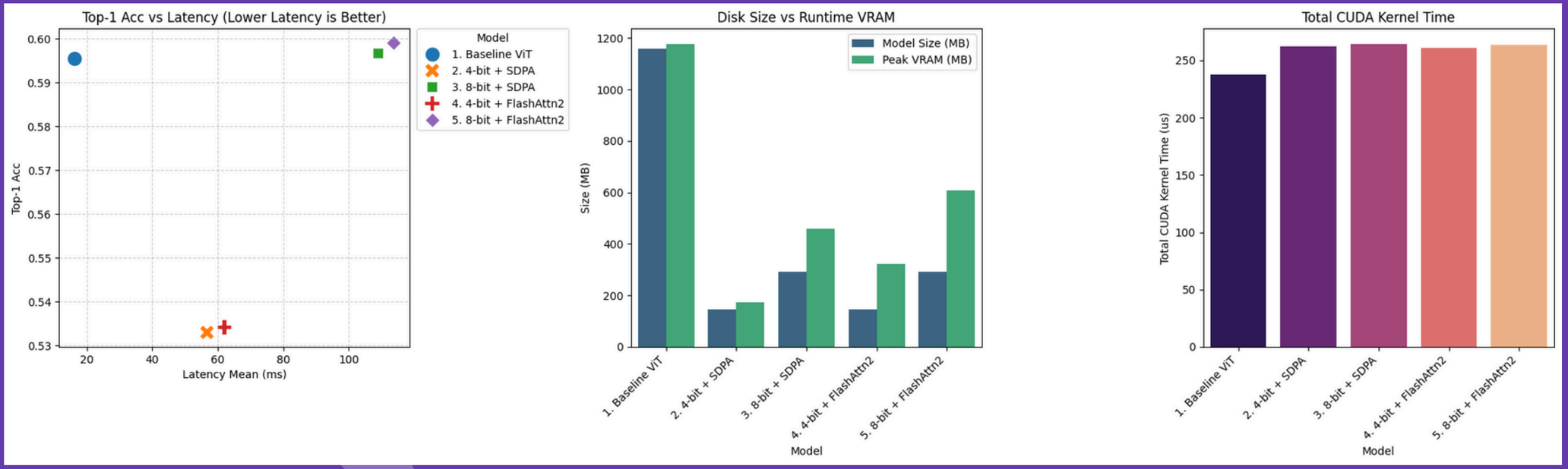
```
...
=== FINAL BENCHMARK RESULTS ===
```

	Model	Model Size (MB)	Latency Mean (ms)	Latency P50 (ms)	Latency P95 (ms)	Latency P99 (ms)	Throughput (img/s)	Peak VRAM (MB)	CPU Latency (ms)	Top-1 Acc	Top-5 Acc	F1 Score	Total CUDA Kernel Time (us)	Attention Kernel Time (us)
0	1. Baseline ViT	1157.4	16.35	15.83	18.56	20.23	61.16	1177.38	328.94	0.5954	0.7751	0.5765	237.508149	37.187958
1	2. 4-bit + SDPA	146.7	56.72	55.99	59.13	64.28	17.63	173.87	N/A	0.5330	0.7237	0.5142	261.838826	0.000000
2	3. 8-bit + SDPA	290.7	109.00	108.49	111.08	116.78	9.17	458.39	N/A	0.5966	0.7775	0.5822	264.379607	0.000000
3	4. 4-bit + FlashAttn2	146.7	62.02	61.32	64.72	66.25	16.12	322.71	N/A	0.5342	0.7225	0.5147	260.547492	15.591958
4	5. 8-bit + FlashAttn2	290.7	113.80	112.51	122.13	123.64	8.79	607.60	N/A	0.5990	0.7812	0.5844	263.377440	12.610708

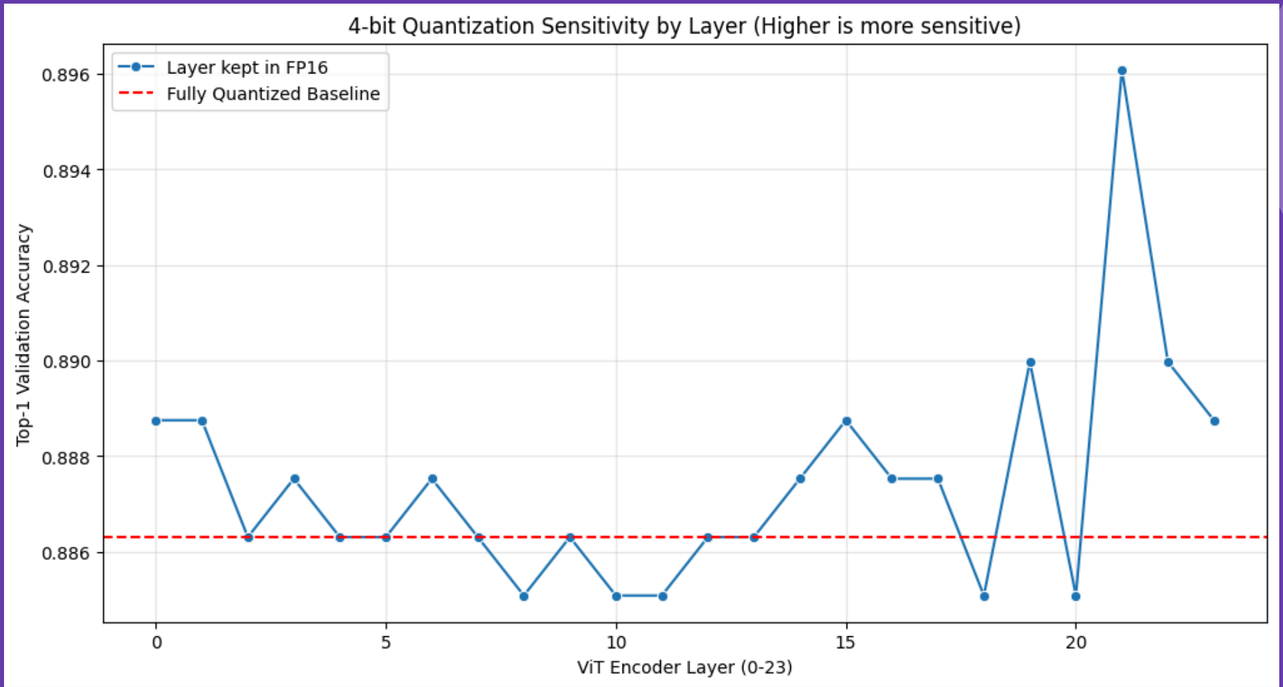
Key metrics included:

- Model size
- Latency (mean, P50, P95, P99)
- VRAM
- Throughput
- Top-1 / Top-5 / F1
- CUDA kernel time

Experimental Evaluation (2/4) - Metrics Overview

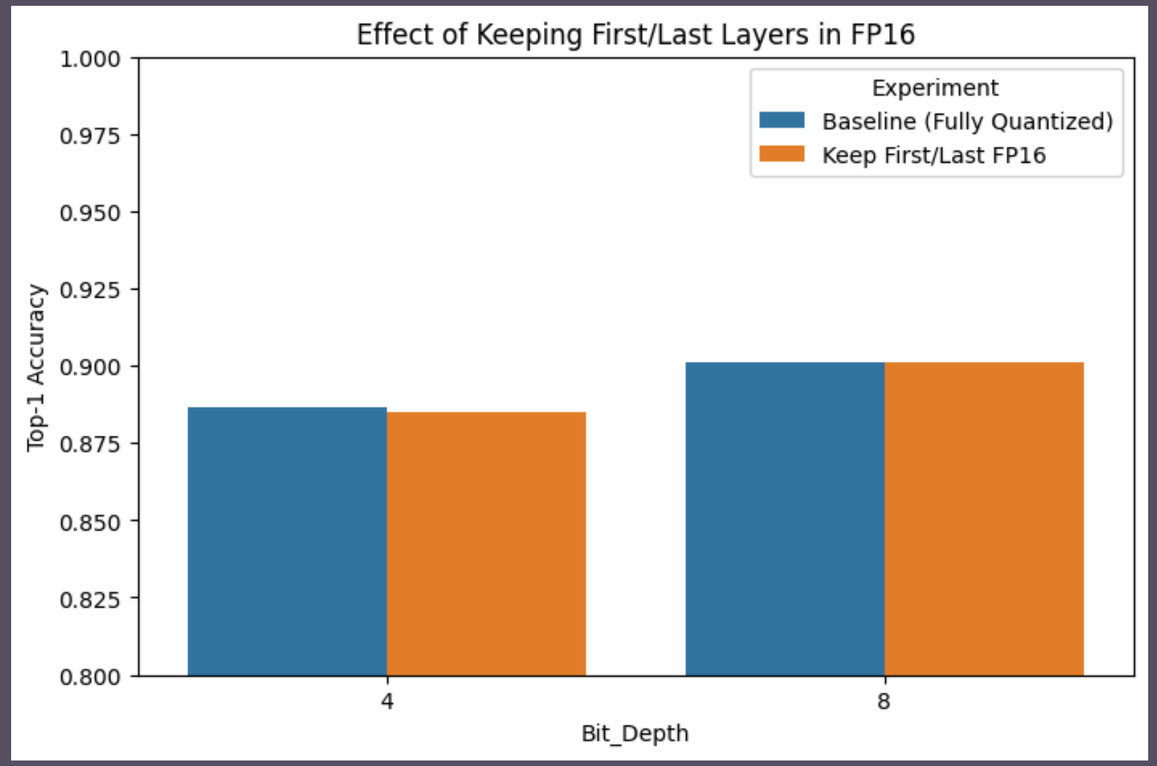


Experimental Evaluation (3/4) - Layer Sensitivity



- Layer 21 is the most sensitive (Top-1 ↑ from 88.6% → 89.6%).
- Layers 22–23 also show noticeable gains.
- Early & middle layers less sensitive → safe to quantize.

Experimental Evaluation (4/4) - First/Last Layer Ablation



- Keeping first/last layers FP16 does not significantly improve accuracy.
- Critical sensitivity exists within encoder blocks, not embedding/classifier.

Observations

- 8-bit quantization preserves nearly full-precision accuracy
- 4-bit gives largest compression (8× smaller) with moderate accuracy loss
- FlashAttention-2 reduces attention kernel time (more efficient compute)
- Layer 21–23 are the most sensitive to low-bit quantization
- First/last layers do not significantly impact accuracy when kept FP16

Conclusion

- 4-bit with FlashAttention-2 provides the best efficiency
- 8-bit quantization preserves accuracy closest to full precision
- 4-bit models are ideal for **memory-constrained deployment**
- LoRA successfully stabilizes all quantized variants
- Quantization + Efficient Attention enables practical **ViT-L/16 deployment**
- Pipeline is ready for future extensions (activation quantization, pruning)

https://github.com/jayrajpamnani/Vision-Model-Optimization-with-Quantization-Efficient-Attention

REPO

THANK YOU