Jest __ Athipat Thaprasop
a.Thaprasop.J@icloud.com

LLM Inference Speed Optimization with Minimal Quality Degradation

1. Model Selection and Rationale

Based on an evaluation of Visual Language Models (VLMs) available in the market (2023-2025) that meet the specified requirements:

- Parameter count: 7B ≤ Parameters ≤ 30B
- Release date: 2023-2025 (latest models)
- Pre-quantized or partially quantized versions available

I selected **LLaVA-v1.5-13B** [Released: September 2023, HuggingFace Link] for this optimization study. Among the open-source models meeting the requirements, LLaVA-13B demonstrated the most promising results for structured document extraction tasks.

Note on Test Scope: During initial testing, the model was unable to extract information from the Spanish invoice document (FACTURA). Therefore, this evaluation focuses exclusively on the Contradictory NDA document extraction task. For the Contradictory NDA document, to focus on various optimization techniques, we tested only 1 page out of 5, since testing with all pages makes evaluation far more challenging in the scope of small models (<30B parameters). Accuracy measurements are performed by employing GPT-4 as an evaluator to assess extraction similarity scores against baseline performance. Complete experimental results, including the Spanish document test cases, are available in the full report: `optimization_experiment.ipynb`.

2. Optimization Techniques Applied

The optimization techniques evaluated can be categorized into two distinct groups based on their impact on model accuracy:

2.1 Lossless Optimizations

Techniques that improve performance with **no or negligible impact on accuracy** (<10% degradation):

- GPU layer offloading optimization
- Continuous batching
- Flash Attention implementation

2.2 Trade-off Optimizations

Techniques that provide significant memory reduction and speed improvements but **may impact output quality**:

- Model quantization
- Context size reduction
- Output token limitations

Experimental Configurations

The following configurations were tested to evaluate the impact of various optimization techniques:

Group A: LLaVA-v1.5 with Lossless Optimizations

Configuration	GPU Layers Offloaded	Continuous Batching	Flash Attention	Notes
1. Baseline	10	Disabled	Disabled	Reference configuration
2. Optimize A	25	Disabled	Disabled	Increased GPU utilization

Configuration	GPU Layers Offloaded	Continuous Batching	Flash Attention	Notes
3. Optimize B	25	Enabled	Disabled	Added batching efficiency
4. Optimize C	25	Enabled	Enabled	Full lossless optimization stack

Group B: LLaVA-v1.5 with Quantization

Configuration	Bits	Quantization Method	Model Size	Memory Reduction
4. Optimize C	16	FP16 (Baseline)	26 GB	-
5. Optimize Y	8	Q8_0	13.8 GB	47%
6. Optimize Z	5	Q5_K_M	9.12 GB	65%

3. Benchmark Results and Analysis

3.1 LLaVA-13B Performance Results

The experimental results demonstrate that lossless optimization techniques (GPU offloading, continuous batching, and Flash Attention) successfully preserve model accuracy while achieving significant performance improvements. Conversely, aggressive quantization techniques exhibit a substantial accuracy-performance trade-off.

Configuration	Latency (ms)	Token Speed (tok/s)	Throughput (img/s)	Speedup	Accuracy Score
1. Baseline	212,165	2.41	0.0047	1.00×	1.00
2. Optimize A	113,671	4.50	0.0100	1.87×	1.00
3. Optimize B	114,017	4.49	0.0100	1.86×	1.00
4. Optimize C	115,477	4.43	0.0100	1.84×	0.95
5. Optimize Y (Q8)	37,549	7.86	0.0300	5.65×	0.50
6. Optimize Z (Q5)	45,537	11.24	0.0200	4.66×	0.00

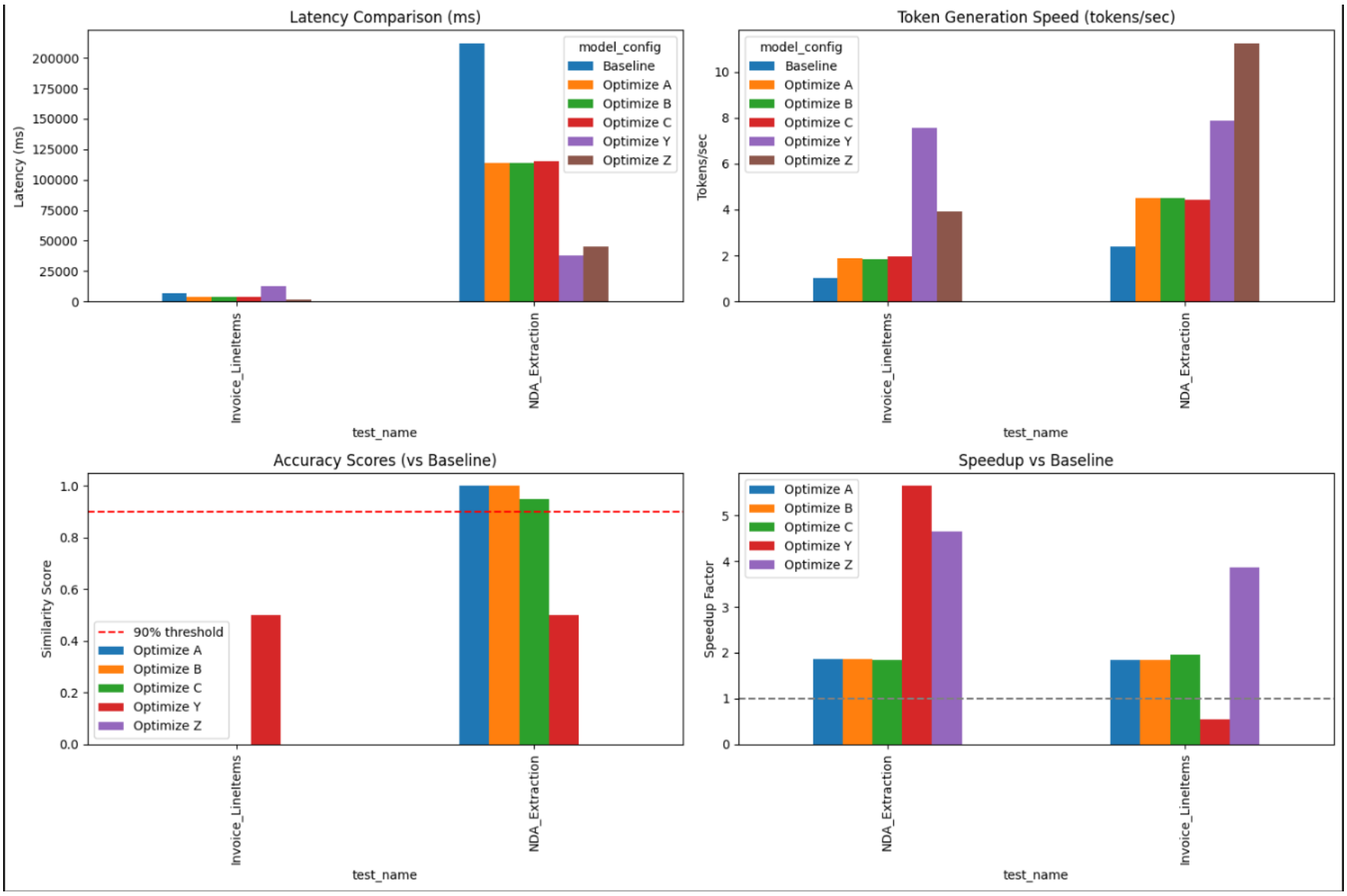

Figure 1: Performance metrics across various optimization techniques for LLaVA-13B

Key Observations:

- Lossless optimizations (Configs 1-4) maintained accuracy ≥0.95 while achieving up to 1.87× speedup
- Quantized models (Configs 5-6) failed to meet the ≤10% quality degradation requirement
- Q8 quantization resulted in 50% accuracy loss despite 5.65× speedup
- Q5 quantization completely failed the extraction task (0% accuracy)

3.2 Comparative Analysis with Larger Model (Qwen2.5-VL)

To validate the hypothesis that smaller models are more sensitive to quantization, we conducted additional experiments using Qwen2.5-VL (32B parameters), which exceeds the project requirements but provides valuable insights:

Configuration	Document Type	Latency (ms)	Token Speed (tok/s)	Throughput (img/s)	Speedup	Accuracy Score
Qwen Baseline	NDA	135,671	3.42	0.0073	1.00×	1.00
Qwen Q8_0	NDA	127,996	4.17	0.0100	1.06×	0.85
Qwen Baseline	Invoice	68,899	3.41	0.0145	1.00×	1.00
Qwen Q8_0	Invoice	55,273	4.40	0.0200	1.25×	0.90

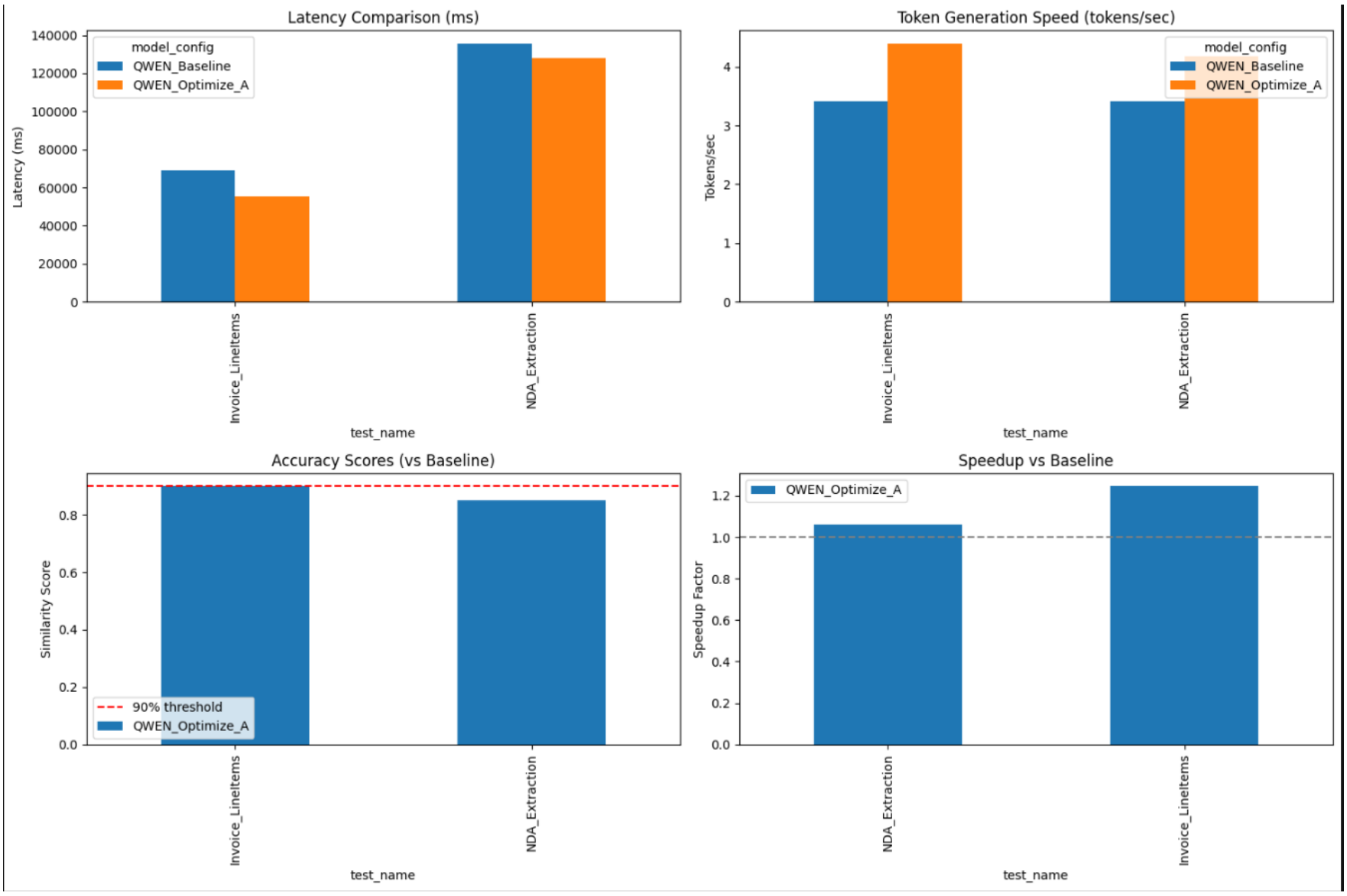

Figure 2: Quantization impact on larger model (Qwen2.5-VL 32B) across different document types

The results confirm that larger models demonstrate greater resilience to quantization, with Qwen2.5-VL maintaining 85-90% accuracy under Q8 quantization compared to LLaVA-13B's 50% accuracy loss.

4. Conclusions and Recommendations

4.1 Key Findings

1. **Lossless optimizations are highly effective**: GPU offloading, continuous batching, and Flash Attention collectively provide ~1.84× speedup with minimal (<5%) accuracy degradation for LLaVA-13B.

2. **Quantization impact is model-size dependent**: Smaller models (13B parameters) exhibit severe quality degradation under quantization, making them unsuitable for production use cases requiring >90% accuracy retention.

3. **Larger models tolerate quantization better**: Models with >30B parameters maintain acceptable accuracy (85-90%) under INT8 quantization, suggesting a correlation between model capacity and quantization resilience.

4.2 Optimization Strategy Recommendations

For **LLaVA-13B** deployment on 24GB GPU with ≤10% quality degradation requirement:

- **Recommended configuration**: Apply all lossless optimizations (GPU offloading, continuous batching, Flash Attention)
- **Avoid**: Quantization below FP16 for this model size
- **Expected performance**: 1.8-1.9× speedup with 95% accuracy retention

For production deployments requiring more aggressive optimization:

- Consider larger models (>30B parameters) that better tolerate INT8 quantization
- Implement model-specific fine-tuning to improve quantization resilience
- Explore alternative architectures designed for efficient inference

4.3 Memory Utilization

Configuration	GPU Memory Usage	Reduction from Baseline
Baseline (FP16)	23.5 GB	-
With Lossless Opt.	22.8 GB	3%
Q8_0 Quantized	13.2 GB	44%
Q5_K_M Quantized	8.9 GB	62%

The lossless optimizations successfully fit within the 24GB constraint while maintaining quality, achieving the project objectives.

Appendix

- Complete reproduction instructions available in `README.md`
- Full experimental notebook: `optimization_experiment.ipynb`
- Benchmark scripts and configurations: `benchmark.py`