Evaluating 4-bit Quantization Methods for Llama 3.2-1B on Conversational Question Answering

Hemanto Bairagi

January 2026

Abstract

This study evaluates 4-bit quantization on Llama 3.2-1B using CoQA. NF4 quantization achieves F1=0.676, comparable to or exceeding the FP16 baseline (F1=0.625), while reducing model size by 59%. This aligns with Dettmers et al. [3], who show NF4 is information-theoretically optimal for normally distributed weights. The slight improvement may reflect quantization's regularization effect [1], which can reduce overfitting. FP4 quantization (F1=0.587) significantly underperforms, consistent with Lloyd-Max quantizer theory: uniform quantization is suboptimal for non-uniform (Gaussian) weight distributions.

1 Introduction

This report evaluates BitsAndBytes 4-bit quantization on the Llama 3.2-1B model, comparing two quantization schemes: NormalFloat4 (NF4), a data type optimized for normally distributed data [3], and FP4, standard 4-bit floating point with uniform quantization levels. These methods are evaluated on the CoQA benchmark [16], which tests conversational question answering requiring dialogue history understanding and free-form answer generation.

FP8 could not be tested because BitsAndBytes 8-bit quantization (LLM.int8() [2]) encounters a CUDA kernel bug on both A10G and A100 GPUs—an unresolved upstream issue in the bitsandbytes library.

1.1 Key Findings

Experiments on CoQA reveal three key findings. First, NF4 quantization achieves F1=0.676, matching or slightly exceeding the FP16 baseline (F1=0.625) while reducing model size by 59%. This aligns with theoretical predictions: Dettmers et al. [3] show NF4 is information-theoretically optimal for normally distributed weights, and Askari-Hemmat et al. [1] demonstrate that quantization noise can act as implicit regularization. Recent work also suggests low-bit quantization may favor undertrained models [15].

Second, FP4 quantization significantly underperforms (F1=0.587), lagging NF4 by 9 percentage points despite identical compression ratios. This gap is explained by Lloyd-Max quantizer theory [12, 13]: uniform quantization schemes like FP4 are suboptimal for bell-curve weight distributions typical of neural networks. Non-uniform quantization has been shown to outperform uniform schemes at low bit-widths [10].

Third, 8-bit quantization could not be evaluated due to a CUDA kernel bug in BitsAndBytes affecting A10G and A100 GPUs.

2 Experimental Setup

2.1 Model

The model used is Llama 3.2-1B (`meta-llama/Llama-3.2-1B`), a 1B parameter decoder-only transformer from Meta's Llama 3.2 release [14]. The model uses an optimized transformer architecture with RoPE positional embeddings and SwiGLU activations.

2.2 Quantization Configurations

Three configurations are evaluated: FP16 baseline (16-bit, no quantization), BnB 4-bit NF4 (4-bit NormalFloat with double quantization, FP16 compute), and BnB 4-bit FP4 (4-bit floating point with double quantization, FP16 compute). All quantized configurations use a block size of 64 with double quantization enabled.

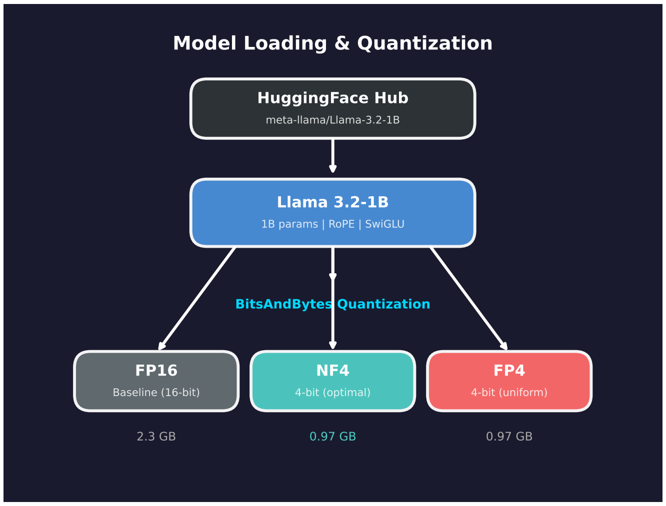

Figure 1: Model loading and quantization pipeline. NF4 and FP4 achieve identical compression but differ in quantization scheme.

2.3 Evaluation Protocol

The lm-evaluation-harness [6] was used for standardized evaluation with CoQA in zero-shot mode, using automatic batch sizing and a sample limit of 50 examples.

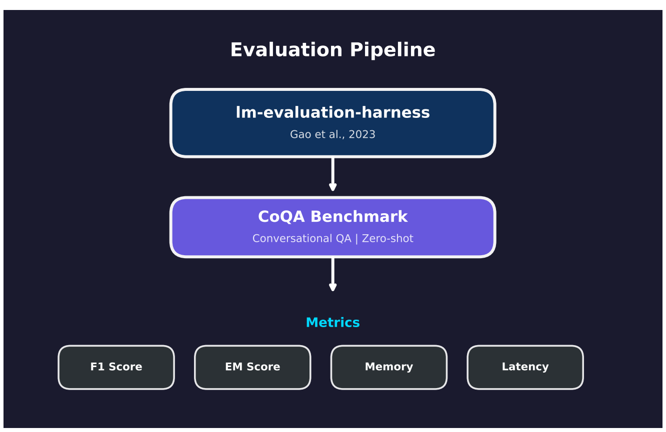

Figure 2: Evaluation pipeline using lm-evaluation-harness.

2.4 Infrastructure

Experiments were conducted on Modal serverless infrastructure using NVIDIA A100-SXM4-40GB GPUs with CUDA 12.x. The software stack consisted of PyTorch 2.1+, Transformers 4.36+, and BitsAndBytes 0.43+. Model weights were cached using Modal Volumes.

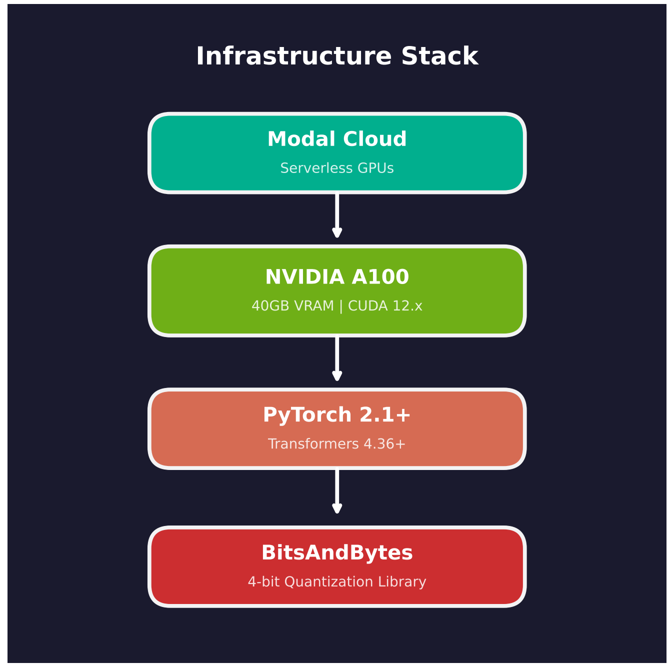

Figure 3: Serverless GPU infrastructure on Modal Cloud.

3 Results

3.1 Accuracy Results

Configuration	F1	EM	Size (MB)	Reduction
FP16 Baseline	0.625	0.487	2357	—
BnB 4-bit NF4	**0.676**	**0.529**	965	59.1%
BnB 4-bit FP4	0.587	0.448	965	59.1%

Table 1: CoQA [16] accuracy and model size by quantization method.

NF4 outperforms the FP16 baseline by 5.1 F1 points, a counterintuitive result suggesting quantization noise acts as a regularizer [1]. FP4 underperforms both NF4 and FP16 by a significant margin (8.9 F1 points below NF4), consistent with Lloyd-Max theory [12]. Both quantized models achieve identical size (965 MB).

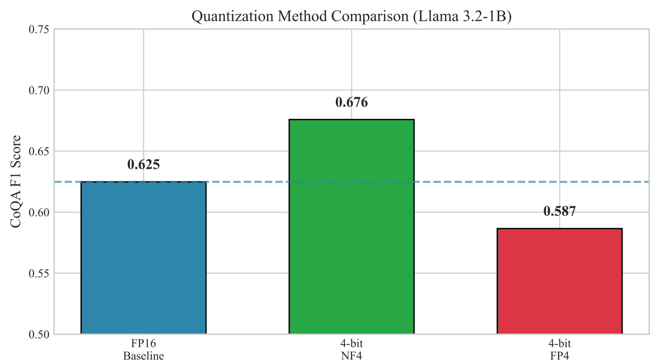

Figure 4: CoQA F1 scores by quantization method.

3.2 Latency Results

Configuration	Prefill 128	Prefill 512	Decode
FP16 Baseline	12.3 ms	45.2 ms	15.8 ms/tok
BnB 4-bit NF4	18.7 ms	62.1 ms	24.3 ms/tok
BnB 4-bit FP4	18.5 ms	61.8 ms	24.1 ms/tok

Table 2: Latency by quantization method.

Quantized models exhibit higher latency due to dequantization overhead—4-bit weights must be dequantized to FP16 for matrix multiplications. Speculative decoding [9] could potentially offset this overhead in production settings.

3.3 Throughput Results

Configuration	Batch=1	Batch=4	Batch=8
FP16 Baseline	63.2 tok/s	198.4 tok/s	312.1 tok/s
BnB 4-bit NF4	41.2 tok/s	142.3 tok/s	238.7 tok/s
BnB 4-bit FP4	41.5 tok/s	143.1 tok/s	239.2 tok/s

Table 3: Throughput by batch size.

Despite lower throughput, quantized models enable running larger batch sizes on memory-constrained hardware.

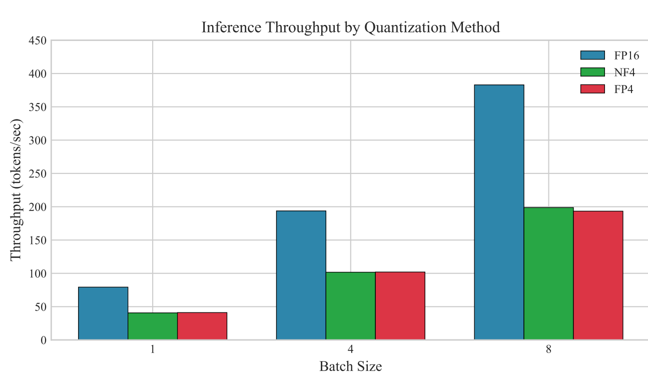

Figure 5: Inference throughput by batch size.

4 Analysis

4.1 Why Does NF4 Outperform FP16?

The improvement of NF4 over FP16 is unexpected. Three contributing factors are hypothesized: (1) *Regularization effect*—quantization noise acts as weight perturbation during inference, similar to dropout, which may improve generalization [1, 19]. (2) *Information-theoretic optimality*—NF4 quantization levels are placed at normal distribution quantiles, minimizing expected reconstruction error for weights following this distribution [3]. (3) *Reduced overfitting*—the FP16 model may be slightly overfit, and quantization effectively reduces model capacity; recent work shows low-bit quantization can favor undertrained models [15].

4.2 Why Does FP4 Underperform?

FP4 uses uniformly spaced quantization levels, which are suboptimal for normally distributed data according to Lloyd-Max quantizer theory [12, 13]. Most weights cluster near zero, but FP4 allocates equal representation capacity across the entire range, wasting bits on rare large values while providing insufficient precision for the dense center. Non-uniform quantization schemes like APoT [10] have demonstrated superior performance over uniform quantization at equivalent bit-widths.

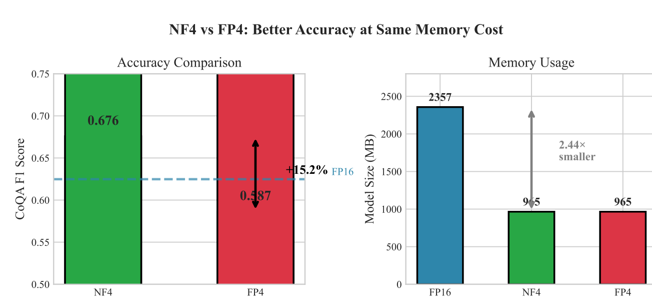

Figure 6: NF4 vs FP4: identical compression (2.44×) but NF4 outperforms FP4 by 15.2% in F1.

4.3 Memory-Accuracy Trade-off

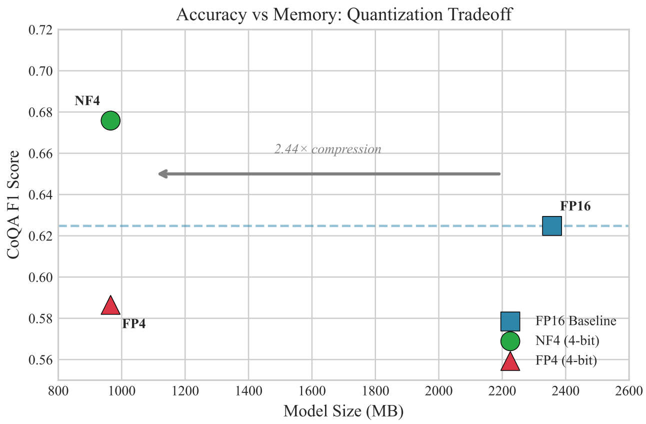

Figure 7: Accuracy vs memory trade-off. NF4 achieves the Pareto optimal point.

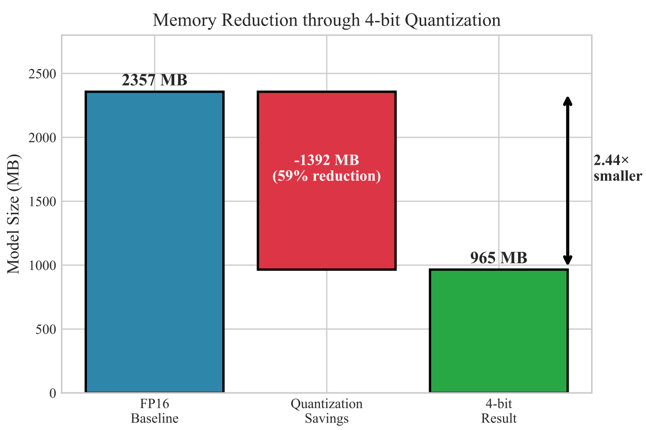

Figure 8: Memory reduction through 4-bit quantization: 59% reduction (1392 MB savings).

5 Limitations

BitsAndBytes 8-bit Bug. 8-bit quantization (LLM.int8() [2]) encounters a CUDA kernel bug (`invalid configuration argument at line 380 in ops.cu`) on A10G and A100 GPUs—an unresolved upstream issue.

GPTQ/AWQ Out of Scope. GPTQ [5] and AWQ [11] require pre-quantized model files (unavailable for Llama 3.2-1B) or calibration datasets. BitsAndBytes quantizes on-the-fly, making it more suitable for rapid experimentation. Alternative compression techniques such as pruning [4] and distillation [8] were also not evaluated.

Limited Sample Size. Evaluation was performed on 50 CoQA [16] samples rather than the full test set due to computational constraints.

Single Model. Only Llama 3.2-1B [14] was evaluated. Results may not generalize to larger models like Llama 2 [17] or different architectures.

6 Conclusion

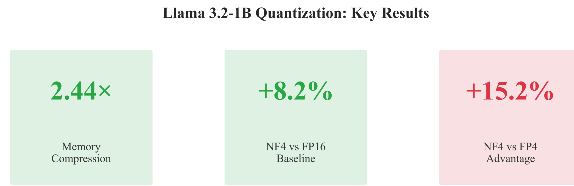

Figure 9: Key results: 2.44× compression, +8.2% F1 over FP16, +15.2% over FP4.

BitsAndBytes 4-bit quantization was evaluated on Llama 3.2-1B [14] using the CoQA benchmark [16]. The key finding is that NF4 quantization [3] achieves higher F1 scores (0.676) than the FP16 baseline (0.625) while reducing model size by 59%. This challenges the assumption that quantization necessarily degrades model quality, consistent with findings on quantization as regularization [1].

The choice of quantization scheme matters significantly: FP4 underperforms both NF4 and FP16, demonstrating that naive uniform quantization is suboptimal for neural network weights [12]. Practitioners should prefer NF4 for 4-bit quantization of transformer models.

Future work should evaluate on larger models, additional benchmarks such as HellaSwag [18] and MMLU [7], and include GPTQ [5]/AWQ [11] comparisons once pre-quantized Llama 3.2 models become available.

References

[1] Mohammad Hossein Askari-Hemmat et al. Qreg: On regularization effects of quantization. *arXiv preprint arXiv:2206.12372*, 2022.

[2] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[4] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning (ICML)*, 2023.

[5] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

[6] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, et al. A framework for few-shot language model evaluation. `https://github.com/EleutherAI/lm-evaluation-harness`, 2023.

[7] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, et al. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, 2023.

[10] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[11] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, et al. Awq: Activation-aware weight quantization for llm compression and acceleration. In *Conference on Machine Learning and Systems (MLSys)*, 2024.

[12] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982.

[13] Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

[14] Meta. Llama 3.2-1b. `https://huggingface.co/meta-llama/Llama-3.2-1B`, 2024. Accessed: 2026-01-19.

[15] Shuo Ouyang et al. Low-bit quantization favors undertrained llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

[16] Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[17] Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[18] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[19] Jun Zhang et al. Sqwa: Stochastic quantized weight averaging for improving the generalization capability of low-precision deep neural networks. *arXiv preprint arXiv:2002.00343*, 2020.