Against FP32, FP16 Changes 3 and NVFP4 W+A 738 of 19,647 RoBERTa-large-MNLI Decisions (Eager Batch 32)

Dmitrii Lozhkin
AVELIN AI

Abstract

Reduced precision can leave average accuracy nearly unchanged while altering individual decisions. On all 19,647 MultiNLI development examples, we run RoBERTa-large-MNLI in eager mode at batch size 32 on one GPU and compare seven precision recipes with FP32, keeping agreement with FP32 separate from correctness against labels. FP16 changes 3 decisions (1 correction, 2 new errors); BF16 changes 21, seven times as many. The FP8 weights-and-activations (W+A) recipe changes 147 decisions with a net gain of 17 correct answers (+0.09 pp; 95% premise-cluster interval −0.03 to +0.20, so no accuracy gain is established). NVFP4 W+A changes 738 decisions and lowers accuracy by 0.77 pp; against BF16 it still changes 737. NVFP4 W+A decisions also depend on which rows share a batch.

1 Introduction

Quantization has been studied for transformer encoders: Q8BERT uses quantization-aware fine-tuning [1], and FP8 work specifies numerical formats for deep learning [2]. Compression can also change individual predictions while aggregate accuracy stays close: Hooker et al. found that pruning and quantization disproportionately affect a subset of examples [3], and Dutta et al. showed that compressed language models with near-baseline accuracy flip many answers between correct and incorrect [4]. We confirm this effect on a public encoder classifier evaluated without further training or calibration [5,6], and report paired transitions with premise-cluster intervals.

2 Setup

The task is natural language inference: decide whether a hypothesis contradicts, follows from, or is neutral toward a premise. We use all MultiNLI development rows: 9,815 matched and 9,832 mismatched examples [7,8]. Model and dataset revisions are pinned in References; class-label orders are explicitly mapped.

All runs use one cloud instance with an NVIDIA RTX PRO 6000 Blackwell Server Edition (96 GB, sm_120), driver 595.112.07. Software is Python 3.12.3, PyTorch 2.14.1+cu130, CUDA 13.0, torchao 0.18.0, Transformers 5.21.0 and NumPy 2.5.4. Rows are processed in dataset order (matched, then mismatched) in consecutive batches of 32. Each batch is truncated at 512 tokens and right-padded to its longest row, rounded up to a multiple of 16.

FP16 and BF16 convert the whole encoder; the final classifier stays FP32 in every recipe. TF32, automatic mixed precision and reduced-precision FP16/BF16 reductions are disabled. All quantized recipes use the torchao 0.18.0 APIs [9]: they first convert the encoder to BF16, then quantize its Linear layers only, with no calibration. INT8 weights uses `Int8WeightOnlyConfig()`, with symmetric per-channel weights. INT8 W+A uses `Int8DynamicActivationInt8WeightConfig()`, with symmetric per-token activations and per-channel weights. FP8 W+A uses `Float8DynamicActivationFloat8WeightConfig(granularity=PerRow())`: E4M3fn values, rowwise scales and fast accumulation. NVFP4 weights uses `NVFP4WeightOnlyConfig(use_dynamic_per_tensor_scale=True)`. NVFP4 W+A uses `NVFP4DynamicActivationNVFP4WeightConfig(use_triton_kernel=False, use_dynamic_per_tensor_scale=True)`: E2M1 values, E4M3 scales for blocks of 16 and an FP32 per-tensor scale. For activations, that per-tensor scale is computed from the whole padded batch, so a row's quantization depends on the other rows in its batch.

Definitions

A **decision** is the argmax over the three class scores. **Changed** means a different decision from the stated reference; **corrected**, **newly incorrect** and **wrong-to-different-wrong** mean wrong→right, right→wrong and wrong→another wrong label. **Accuracy** is the fraction matching dataset labels; **pp** denotes percentage points. **W+A** quantizes weights and activations; **weights** quantizes weights only.

3 Results

FP16 preserves almost all reference decisions in this cohort. FP32 answers 17,753 examples correctly (90.36%). FP16 changes 3 (0.02%), leaving one fewer correct answer.

Table 1. Paired results on 19,647 MultiNLI rows, eager batch 32, classifier FP32. Corrected, newly incorrect and accuracy difference use FP32; remaining changes are wrong-to-different-wrong. The BF16 column is a separate comparison against the BF16 recipe.

GPU recipe	Changed vs FP32	Changed vs BF16	Corrected	Newly incorrect	Accuracy Δ, pp
FP16	3	24	1	2	−0.01
BF16	21	0	11	9	+0.01
INT8 weights	74	71	39	30	+0.05
FP8 W+A	147	154	76	59	+0.09
INT8 W+A	291	286	133	145	−0.06
NVFP4 weights	501	498	192	279	−0.44
NVFP4 W+A	738	737	272	424	−0.77

A small net accuracy change can conceal many changed decisions. FP8 corrects 76 answers, introduces 59 errors and exchanges 12 wrong answers. NVFP4 W+A corrects 272, introduces 424 errors and exchanges 42 wrong answers, reaching 89.59% accuracy. Using BF16 as the reference shifts each quantized recipe's count by at most 7; this comparison does not isolate the causal effect of casting.

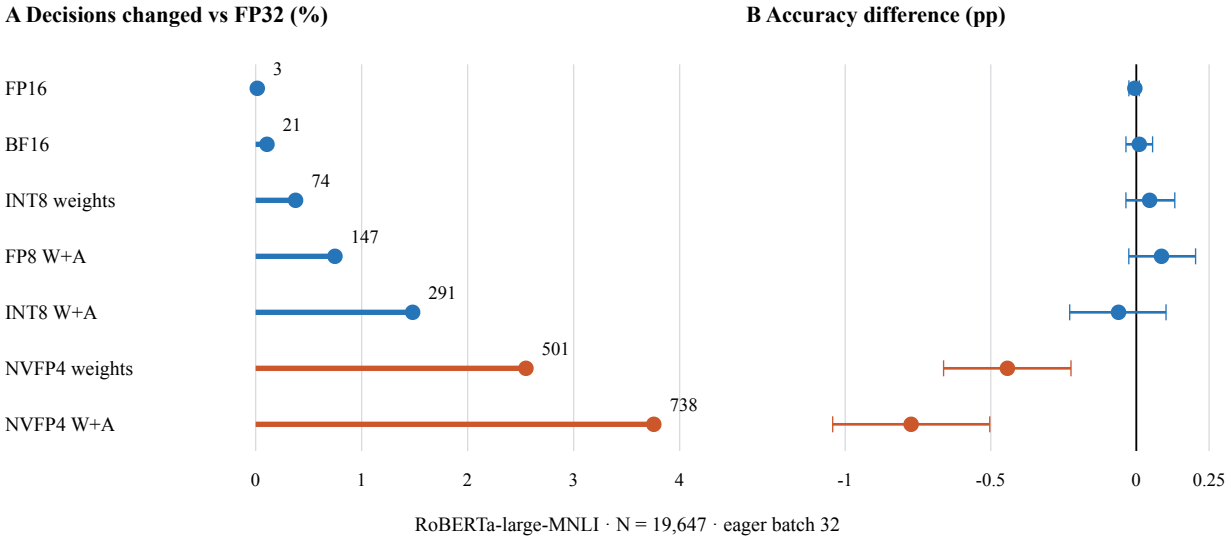


Figure 1. Seven GPU recipes, 19,647 MultiNLI rows, eager batch 32, classifier FP32. Left: changes relative to FP32, with count annotations. Right: accuracy differences and marginal 95% premise-cluster intervals; W+A means weights and activations.

NVFP4 loses labeled accuracy under this protocol. Its interval is [−1.04, −0.50] pp; FP8 is [−0.03, +0.20]. We use 20,000 paired bootstrap draws with a fixed seed for each comparison, resampling identical-premise groups within each split, with original split weights and row-weighted accuracy. The 3,334 matched and 3,335 mismatched groups address repeated premises.

4 Limitations

This is one fine-tuned checkpoint on development data, not an untouched test population. Two premise-hypothesis pairs occur twice (four rows); excluding all four rows leaves every net accuracy difference unchanged. Cluster intervals are marginal, not simultaneous, and do not capture every dependence; an interval crossing zero establishes neither equivalence nor superiority.

Attention backend selection can depend on numerical type, so we compare complete execution paths, not an isolated quantizer. Batch composition and padding also matter. On a balanced 1,200-row subset (200 rows per split and label, chosen by a seeded hash, seed 1729), evaluating each recipe at batch 1 and at batch 32 changes 26 NVFP4 W+A decisions, 3 BF16 decisions, 1 each for INT8 weights and NVFP4 weights, and none for the other recipes. Rerunning

six recipes (FP32, FP16, BF16, INT8 W+A, FP8 W+A and NVFP4 W+A) at batch 1 on that subset in a separate process reproduced the saved logits bitwise; the full batch-32 run and the two weights-only recipes were not repeated. These results do not transfer automatically to compiled execution, other hardware or applications. Planning, reasoning, downstream utility and speed are not evaluated.

5 Recommendations

1. Report changed decisions and labeled transitions together.
2. Compare quantized recipes with their surrounding-precision reference as well as FP32; here BF16 moves each quantized count by at most 7.
3. Validate the exact deployed batching and execution path; define acceptable disagreement before claiming equivalence.

Disclosure.
Experiments and analysis were produced with AI coding agents under the author's direction. All numbers were recomputed from the saved per-example predictions. The model card specifies the MIT license; MultiNLI combines sources with different licenses, listed in its card [6,8].

References

1. O. Zafrir, G. Boudoukh, P. Izsak, M. Wasserblat. Q8BERT: quantized 8-bit BERT. arXiv:1910.06188, 2019.
2. P. Micikevicius, D. Stosic, N. Burgess, et al. FP8 formats for deep learning. arXiv:2209.05433, 2022.
3. S. Hooker, A. Courville, G. Clark, Y. Dauphin, A. Frome. What do compressed deep neural networks forget? arXiv:1911.05248, 2019.
4. A. Dutta, S. Krishnan, N. Kwatra, R. Ramjee. Accuracy is not all you need. NeurIPS 2024; arXiv:2407.09141, 2024.
5. Y. Liu, M. Ott, N. Goyal, et al. RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692, 2019.
6. FacebookAI. FacebookAI/roberta-large-mnli. Hugging Face model card, revision 2a8f12d27941090092df78e4ba6f0928eb5eac98. Accessed October 28, 2026.
7. A. Williams, N. Nangia, S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. NAACL-HLT 2018, pp. 1112–1122. DOI: 10.18653/v1/N18-1101; arXiv:1704.05426, 2018.
8. NYU Machine Learning for Language. nyu-mll/multi_nli. Hugging Face dataset card, revision da70db2af9d09693783c3320c4249840212ee221. Accessed October 28, 2026.
9. PyTorch contributors. TorchAO 0.18.0 (released August 3, 2026): quantization APIs and source code. Git revision 5f2baf9d575cf732362594c998c399902942531f.