HPML Project: BERT on SQuADv2 – Experiment Summary and Metrics

TODO: Your Name(s)

December 10, 2025

1 Overview

This document collects the core quantitative results for all experiments run on `bert-base-squadv2` on SQuADv2, including:

- Baseline vs SDPA vs `torch.compile` vs AOT-eager variants (A-series, D1),
- Parameter-efficient finetuning (B1/B2/B3),
- Memory-oriented configurations (C1, C2-8, C2),
- Training-time VRAM measurements for selected configs (A3, C1, C2).

The tables are meant to be the canonical record of all numeric results. You can add narrative analysis around them in the main report.

Throughout, E_{exact} and E_{f1} denote overall dev-set exact match and F1. Runtimes and throughput are taken from the Hugging Face `Trainer` logs. VRAM is measured with a dedicated script using `torch.cuda.max_memory_allocated()` and `max_memory_reserved()` on a single training step after warmup.

ID	Directory	Key configuration
A1	bert-base-squadv2-baseline-t4	Baseline BERT, no SDPA, no `torch.compile`
A2	bert-base-squadv2-a2-sdpa-t4	SDPA attention, no `torch.compile`
A3	bert-base-squadv2-a3-sdpa-compile-t4	SDPA + `torch.compile` (backend = inductor)
A4-nb	bert-base-squadv2-a4-final-nobucket	SDPA + `torch.compile` (inductor), no bucketing (pad batching)
A4-b	bert-base-squadv2-a4-sdpa-compile-bucket-t4-v2	SDPA + `torch.compile` (inductor), bucketing (buggy / failed ablation)
B1	bert-base-squadv2-b1-lora-compile-t4	LoRA adapters + `torch.compile` (inductor)
B2	bert-base-squadv2-b2	LoRA only (BitFit-style), lr $= 10^{-3}$, no `torch.compile`
B3	bert-base-squadv2-b3	LoRA only (BitFit-style), lr $= 5 \times 10^{-5}$, `torch.compile` (inductor)
C1	bert-base-squadv2-c1-8bit	8-bit optimizer (`adamw_8bit`), `torch.compile` (inductor)
C2-8	bert-base-squadv2-c2-8bit	8-bit optimizer only (no GC, no compile) for VRAM profiling
C2	bert-base-squadv2-c2-gc	Full precision + gradient checkpointing + `torch.compile` (inductor)
D1	bert-base-squadv2-d1-aot-eager-lr3e-5	AOT-eager backend: `torch.compile` (aot_eager)

Table 1: Summary of main experimental configurations.

ID	Dir	E_{exact}	E_{f1}	T_{eval} (s)	samp/s	Train loss	T_{train} (s)	train samp/s
A1	bert-base-squadv2-baseline-t4	73.21	76.62	66.97	181.19	0.796	9001.32	43.91
A2	bert-base-squadv2-a2-sdpa-t4-old	73.26	76.56	70.90	171.15	0.793	9411.91	42.00
A3	bert-base-squadv2-a3-sdpa-compile-t4	73.37	76.89	72.92	166.41	0.793	8729.56	45.28
A4-nb	bert-base-squadv2-a4-final-nobucket	73.78	77.23	44.66	271.70	0.789	7662.12	51.59
A4-b	bert-base-squadv2-a4-sdpa-compile-bucket-t4-v2	25.82	26.87	39.89	304.21	0.794	5769.40	68.51
D1	bert-base-squadv2-d1-aot-eager-lr3e-5	73.04	76.35	69.34	174.99	0.793	8677.81	45.55

Table 2: SQuADv2 dev metrics for baseline, SDPA, `torch.compile` and AOT-eager variants. A4-b is a bucketing experiment with broken accuracy (failed ablation).

ID	Dir	E_{exact}	E_{f1}	T_{eval} (s)	samp/s	Train loss	T_{train} (s)	train samp/s
B1	bert-base-squadv2-b1-lora-compile-t4	51.12	54.95	81.16	149.50	1.94	5613.03	70.42
B2	bert-base-squadv2-b2	54.61	58.00	71.64	169.37	1.86	5409.28	73.07
B3	bert-base-squadv2-b3	68.21	71.99	71.60	169.46	1.05	4403.16	89.77

Table 3: SQuADv2 dev metrics for parameter-efficient finetuning variants (LoRA, BitFit-style bias-only, layer freezing).

2 Baseline, SDPA, Compile, and Backend Variants (A-series, D1)

3 Parameter-Efficient Finetuning (LoRA, BitFit, Freezing)

4 Memory-Oriented Configurations (C-series)

ID	Dir	Key knobs	E_{exact}	E_{f1}	T_{eval} (s)	samp/s	Train loss	T_{train} (s)
C1	bert-base-squadv2-c1-8bit	8-bit optim + compile	72.64	76.27	75.06	161.65	0.81	8509.86
C2-8	bert-base-squadv2-c2-8bit	8-bit optim, no GC, no compile	72.67	76.14	72.91	166.43	0.81	9116.26
C2	bert-base-squadv2-c2-gc	Full precision + GC + compile	70.85	74.53	74.88	162.05	0.82	10645.29

Table 4: SQuADv2 dev metrics for memory-oriented configurations (8-bit optimizer, gradient checkpointing, `torch.compile`).

5 Training-Time VRAM Measurements

Measurement setup (summarised; expand in the report body as needed):

- Device: NVIDIA T4 (reported total VRAM $\approx$ 14.9 GiB).
- Batch size: 8, fp16, single forward+backward step.
- One warmup step (not measured), then one measured step.
- Metrics: `torch.cuda.max_memory_allocated()` and `torch.cuda.max_memory_reserved()`.

Config	Description	Peak allocated (MiB)	Peak reserved (MiB)	Device total (MiB)
A3	SDPA + `torch.compile` (inductor)	3087.0	3256.0	14915.7
C1	8-bit optimizer + `torch.compile` (inductor)	2465.9	2720.0	14915.7
C2	Full precision + gradient checkpointing + `torch.compile` (inductor)	2100.2	2344.0	14915.7

Table 5: Single-step training VRAM on a T4 GPU (batch size 8, fp16). Measured via `torch.cuda.max_memory_allocated()` and `torch.cuda.max_memory_reserved()` after one warmup step.

6 Placeholders for Narrative (to be filled)

6.1 LoRA and Other Parameter-Efficient Methods

6.2 C2 Trained vs C2 for VRAM Profiling

6.3 Bucketing Ablation (A4-b) as Failed Variant

6.4 VRAM Methodology and Limitations

6.5 Practical T4 "Recipes"

6.6 Proposal vs Midpoint vs Final: Promises and Outcomes