Efficient Fine-Tuning Methods for Portuguese Question Answering: A Comparative Study of PEFT on BERTimbau and Exploratory Evaluation of Generative LLMs

Mariela M. Nina and Caio Veloso Costa and Lilian Berton and Didier A. Vega-Oliveros
Institute of Science and Technology
Federal University of São Paulo (UNIFESP)
São José dos Campos, SP, Brazil
Correspondence: {mariela.nina, veloso.caio, lberton, didier.vega}@unifesp.br

Abstract

Although large language models have transformed natural language processing, their computational costs create accessibility barriers for low-resource languages such as Brazilian Portuguese. This work presents a systematic evaluation of Parameter-Efficient Fine-Tuning (PEFT) and quantization techniques applied to BERTimbau for Question Answering on SQuAD-BR, the Brazilian Portuguese translation of SQuAD v1. We evaluate 40 configurations combining four PEFT methods (LoRA, DoRA, QLoRA, QDoRA) across two model sizes (Base: 110M, Large: 335M parameters). Our findings reveal three critical insights: (1) LoRA achieves 95.8% of baseline performance on BERTimbau-Large while reducing training time by 73.5% (F1=81.32 vs 84.86); (2) higher learning rates (2e-4) substantially improve PEFT performance, with F1 gains of up to +19.71 points over standard rates; and (3) larger models show twice the quantization resilience (loss of 4.83 vs 9.56 F1 points). These results demonstrate that encoder-based models can be efficiently fine-tuned for extractive Brazilian Portuguese QA with substantially lower computational cost than large generative LLMs, promoting more sustainable approaches aligned with *Green AI* principles. An exploratory evaluation of Tucano and Sabiá on the same extractive QA benchmark shows that while generative models can reach competitive F1 scores with LoRA fine-tuning, they require up to 4.2× more GPU memory and 3× more training time than BERTimbau-Base, reinforcing the efficiency advantage of smaller encoder-based architectures for this task.

Index Terms— BERTimbau, Parameter-Efficient Fine-Tuning, LoRA, Quantization, Extractive Question Answering, Brazilian Portuguese

1 Introduction

Large language models (LLMs) based on the Transformer architecture (Vaswani et al., 2023) have achieved extraordinary capabilities in recent years, reaching state-of-the-art results across multiple natural language processing benchmarks (Devlin et al., 2019). However, in lower-resource language settings such as Brazilian Portuguese, the landscape faces significant limitations. Unlike English, which benefits from a proliferation of specialized models, Brazilian Portuguese has a more restricted ecosystem. The most relevant and widely used models, like Sabiá (7B parameters) (Pires et al., 2023), Tucano (1.1B parameters) (Corrêa et al., 2025), and BERTimbau (110M–335M parameters) (Souza et al., 2020), are primarily based on widely established architectures subsequently adapted for Portuguese. Although these models have demonstrated competitive performance, they are typically adapted using full parameter updates, which can demand significant computational resources (e.g., up to 7 hours and over 18 GB of GPU memory for Large models, as observed in our experiments), resulting in accessibility barriers for academic and industrial environments with constrained resources (Strubell et al., 2019).

To overcome these computational constraints, QLoRA (Dettmers et al., 2023) has emerged as a promising technique. Unlike standard post-training quantization used solely for model compression during inference, QLoRA enables fine-tuning of 4-bit quantized models with low-rank adapters while keeping the base model frozen, without significant performance degradation. While this technique has been widely adopted in English-language models, studies applying quantization techniques to Brazilian Portuguese question-answering models remain scarce and largely unsystematic. Parameter-Efficient Fine-Tuning (PEFT) methods provide an additional solution. Techniques such as LoRA (Hu et al., 2021), which injects low-rank matrices while updating only 0.1–1% of the parameters, and DoRA (Liu et al., 2024), which introduces magnitude–direction decomposition, have

Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)
Vol. 1, pages 917–926
April 13-16, 2026

been shown to achieve performance close to full fine-tuning in English. Nevertheless, empirical evidence remains overwhelmingly concentrated on English-language models, leaving unanswered whether these techniques retain their effectiveness when applied to Brazilian Portuguese models for tasks requiring deep language understanding.

Driven by the limitations of current hardware and the lack of Portuguese benchmarks, this study is guided by three core hypotheses: **(H1) Low-Resource Efficiency:** We hypothesize that PEFT methods can match full fine-tuning performance on Portuguese QA while significantly reducing computational overhead. **(H2) Scale Robustness:** Based on scaling laws, we hypothesize that larger models (Large) possess greater parametric redundancy, making them more resilient to aggressive 4-bit quantization than Base models. **(H3) Optimization Sensitivity:** We hypothesize that the low-rank constraints of PEFT require significantly higher learning rates than full fine-tuning to escape local minima during adaptation.

This work addresses these gaps by presenting a systematic evaluation of PEFT and quantization techniques applied to Brazilian Portuguese models on the Portuguese version of the SQuAD v1 dataset to test these hypotheses. Focusing on BERTimbau, the most established Transformer-based model for Brazilian Portuguese, with well-defined baselines (F1 = 82.50 for Base and F1 = 84.43 for Large on Portuguese SQuAD v1) (Souza et al., 2020), we conduct a comparative evaluation of LoRA, QLoRA, DoRA, and QDoRA on both the Base (110M parameters) and Large (335M parameters) variants for the Question Answering task. We provide a comprehensive analysis of the trade-offs among performance (F1-score and Exact Match), temporal efficiency, and hardware accessibility, using full fine-tuning as the baseline.

2 Background

2.1 The Challenge of Adapting Large Models

The dominant paradigm in NLP consists of pre-training massive models on large corpora and adapting them via full fine-tuning for specific tasks (Devlin et al., 2019). However, as models scale according to established scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022), full fine-tuning becomes prohibitively expensive: it requires storing gradients and optimizer states for all parameters, demanding up to 12–18× more memory than inference. For example, fine-tuning GPT-3 175B with the Adam optimizer requires approximately 1.2TB of GPU memory (Hu et al., 2021), making the deployment of multiple specialized model instances impractical. This resource barrier motivated the development of Parameter-Efficient Fine-Tuning (PEFT) methods, which aim to update only a small fraction of parameters while maintaining competitive performance (Hu et al., 2021; Mangrulkar et al., 2022).

2.2 LoRA: Low-Rank Adaptation

LoRA (Hu et al., 2021) addresses this challenge by building on two key observations: (1) pre-trained models exhibit low intrinsic dimensionality (Aghajanyan et al., 2020), suggesting that the effective adaptation space is much smaller than the full parameter space, and (2) weight updates during fine-tuning exhibit low-rank structure. Motivated by these findings, LoRA keeps the pre-trained weights $W_0 \in \mathbb{R}^{d\times k}$ frozen (where d is the output dimension and k is the input dimension) and injects two trainable low-rank matrices: $A \in \mathbb{R}^{r\times k}$ and $B \in \mathbb{R}^{d\times r}$, where the rank $r \ll \min(d, k)$ is typically chosen from $r \in \{4, 8, 16\}$ in practice.[1] For an input $x \in \mathbb{R}^k$, the output $h \in \mathbb{R}^d$ is computed as:

$$h = W_0 x + \frac{\alpha}{r} BAx \tag{1}$$

where α is a scaling factor, typically set to $\alpha = 2r$ to stabilize training. This decomposition drastically reduces the number of trainable parameters: for a 768×768 matrix with $r = 16$, LoRA requires only $2 \times 16 \times 768 = 24{,}576$ parameters versus $768^2 = 589{,}824$ in full fine-tuning (a 96% reduction). Crucially, during inference, BA can be merged into W_0, eliminating any additional latency—an important advantage over adapter-based methods (Houlsby et al., 2019).

Other PEFT methods include Prefix-Tuning (Li and Liang, 2021), which optimizes task-specific continuous vectors prepended to input representations; Prompt Tuning (Lester et al., 2021), which learns soft prompts prepended to the input; and AdaLoRA (Zhang et al., 2023), which adaptively allocates the parameter budget based on the importance of each weight matrix. Recent studies propose a unified view of these methods (He et al.,

[1] We adopt the notation d, k, r to maintain consistency with the original LoRA formulation (Hu et al., 2021).

2022), identifying common patterns in how they modify the base model's representations.

2.3 Quantization and QLoRA

Complementary to PEFT methods, quantization approaches efficiency from an orthogonal perspective: reducing the numerical precision of model weights from floating-point representations (typically 32 or 16 bits) to lower-precision formats (8, 4, 3, or 2 bits) (Dettmers et al., 2023). For uniform n-bit quantization, a weight $w \in \mathbb{R}$ is mapped to a quantized integer $\tilde{w}$ as:

$$\tilde{w} = \text{round}\left(\text{clamp}\left(\frac{w - z}{s}, -2^{n-1}, 2^{n-1} - 1\right)\right) \quad (2)$$

where $s \in \mathbb{R}^+$ is the scale factor and $z \in \mathbb{R}$ is the zero-point. Prior work explores 8-bit optimizers (Dettmers et al., 2022) to reduce memory footprint during training, while GPTQ (Frantar et al., 2023) proposes post-training quantization based on layer-wise error minimization.

QLoRA (Dettmers et al., 2023) represents a synergistic integration of quantization and LoRA, for which we show evidence here that 4-bit-quantized models can be fine-tuned without performance degradation. QLoRA introduces three key technical innovations: (1) **4-bit NormalFloat (NF4)**, a data type designed for normally distributed weights that uses quantile-based quantization and is theoretically optimal for deep neural networks; (2) **Double Quantization**, which also quantizes the scale and zero-point parameters, reducing the average footprint by approximately 0.37 bits per parameter; and (3) **Paged Optimizers**, which leverage unified memory to manage memory spikes. In QLoRA, the base weights are quantized to 4 bits and remain frozen, while the LoRA matrices are kept in bfloat16.

2.4 DoRA and QDoRA

DoRA (Liu et al., 2024) addresses the persistent accuracy gap of LoRA by decomposing weights into magnitude and direction components. Inspired by Weight Normalization (Salimans and Kingma, 2016), DoRA decomposes each weight matrix W as:

$$W = m\frac{V}{\|V\|_c} \quad (3)$$

where $m \in \mathbb{R}^d$ represents the magnitudes (column-wise norms) and $V \in \mathbb{R}^{d\times k}$ represents the normalized direction. During fine-tuning, DoRA applies the LoRA update only to the directional component and trains an additional vector Δm to adjust the magnitudes. QDoRA naturally extends DoRA to the quantized regime by combining this decomposition with QLoRA's quantization techniques.

2.5 Question Answering and Metrics

In extractive Question Answering (QA) tasks such as SQuAD v1 (Rajpurkar et al., 2016), given a context C and a question Q, the model must predict the start s and end e positions that delimit the answer span extracted from the context. The standard evaluation metrics are the **F1-score** (the harmonic mean of token-level precision and recall) and **Exact Match (EM)** (the percentage of predictions that exactly match the ground truth after normalization).

Beyond standalone applications, robust QA models serve as critical reasoning components in downstream pipelines, such as explainable automated fact-checking (Yang et al., 2022), where QA mechanisms act as proxies for claims against retrieved evidence and provide interpretability.

3 Related Work

Early efforts on pre-trained models for Brazilian Portuguese focused on BERT-style architectures, with particular emphasis on BERTimbau in its Base and Large variants (Souza et al., 2020). Subsequent studies explored autoregressive LLMs such as Sabiá (Pires et al., 2023) and Tucano (Corrêa et al., 2025), which were designed primarily for text generation. More recently, updated models such as Bertugues (Mazza Zago and Agnoletti dos Santos Pedotti, 2024) have further expanded the landscape of Portuguese representation learning. While multilingual models such as XLM-R (Conneau et al., 2020) demonstrate strong cross-lingual capabilities, BERTimbau's specific focus on Brazilian Portuguese makes it especially suitable for downstream tasks in this language.

In extractive QA, SQuAD-BR has become the standard benchmark for evaluating models in Portuguese, with BERTimbau Base and Large achieving reference results in F1 and Exact Match on the Portuguese version of SQuAD v1 (e.g., $F1 = 82.50\%$ and 84.43%, $EM = 70.49\%$ and 72.68%, respectively) (Souza et al., 2020; da Silva et al., 2022). While some recent studies

have explored self-supervised fine-tuning and layer-freezing strategies to adapt BERTimbau for specialized domains with limited labeled data (Nina and Vega-Oliveros, 2025; Condori-Luna et al., 2026), these works often prioritize classification tasks and do not explicitly analyze the computational costs of full fine-tuning in extractive QA.

In the context of English-language LLMs, Parameter-Efficient Fine-Tuning (PEFT) methods such as LoRA (Hu et al., 2021) and DoRA (Liu et al., 2024) have been proposed, introducing low-rank adaptations that drastically reduce the number of updated parameters while maintaining competitive performance. In parallel, quantization techniques such as QLoRA (Dettmers et al., 2023) enable storing model weights at low precision (4 bits) and combining this with PEFT to train LLMs on GPUs with limited memory. While benchmarks such as GLUE (Wang et al., 2019) and SuperGLUE (Wang et al., 2020) have established robust evaluation standards, they remain heavily English-centric, highlighting a significant representational gap for low-resource languages. For instance, techniques like progressive layer unfreezing have proven crucial for successfully optimizing large multilingual models in extremely low-resource settings, such as those required for indigenous languages (Nina and Vega-Oliveros, 2025).

Overall, the existing literature provides: (i) robust Portuguese models with a focus on absolute performance, (ii) strong QA baselines on SQuAD-BR based on full fine-tuning, and (iii) a mature body of PEFT techniques evaluated primarily in English. However, systematic evidence that combines these three lines, i.e., evaluating PEFT and quantization on Brazilian Portuguese models for QA, remains scarce. This gap is precisely what the present work aims to address.

4 Methodology

4.1 Dataset and Model Configuration

We use SQuAD-BR (Rajpurkar et al., 2016), consisting of 87,599 question–answer pairs for training and 10,570 for evaluation. To evaluate scale robustness (H2), we utilize two variants of BERTimbau (Souza et al., 2020): (1) **BERTimbau-Base** with 12 layers, 768 hidden dimensions, 12 attention heads, and 110M parameters, and (2) **BERTimbau-Large** with 24 layers, 1024 hidden dimensions, 16 attention heads, and 335M parameters. Both models were pre-trained on the brWaC corpus with 2.68 billion tokens.

4.2 PEFT Configuration

For all PEFT methods, we use: LoRA rank $r = 16$, scaling factor $\alpha = 32$, target modules (the query, key, value, and output projection matrices of the attention mechanism), and a dropout rate of 0.1. For quantized variants, we apply 4-bit NF4 quantization to the base weights with double quantization enabled and bfloat16 as the compute dtype. All prompts used for generative model evaluation are publicly available to ensure replicability.[2]

To test our hypothesis on optimization sensitivity (H3), we systematically compare two learning rates: the standard BERT learning rate (4.25×10^{-5}) and a high learning rate optimized for PEFT (2×10^{-4}), training for 2 and 3 epochs. Common hyperparameters include the AdamW optimizer (Kingma and Ba, 2017; Loshchilov and Hutter, 2019), weight decay of 0.01, batch size of 16 (Base) and 8 (Large), maximum sequence length of 384, and gradient clipping with norm 1.0.

4.3 Computational Infrastructure

All experiments were conducted on a workstation with a single GPU: NVIDIA RTX A4500 with 20GB of VRAM. This setup simulates a constrained academic environment (relevant to H1), where full fine-tuning of large models is typically unfeasible without techniques like QLoRA. Software stack: CUDA 12.2, PyTorch 2.1.0, Transformers 4.36.0, PEFT 0.7.1, and bitsandbytes 0.41.0.

5 Experimental Results

5.1 Performance on BERTimbau-Base

Tables 1 and 2 present the complete results for BERTimbau-Base, considering variations in the *learning rate* (2×10^{-4} and 4.25×10^{-5}) and the number of epochs (2 and 3), using full fine-tuning (*Full FT*) solely as an upper reference for performance. The Full FT baseline values reported here are the result of our own re-execution under identical hardware and software conditions, enabling a fair comparison; they are close to, though not identical to, the original values reported by Souza et al. (2020), as we expect minor differences due to distinct software versions and random seeds.

With $lr = 2 \times 10^{-4}$, PEFT methods exhibit consistent and stable behavior, with **LoRA** and

[2] https://github.com/GPAM-ai/Efficient-FineTunning-QA-PEFT.git

DoRA standing out as the best-performing techniques (F1=78.01), closely approaching the *Full FT* baseline. From a practical perspective, LoRA is preferable because it achieves the same accuracy while reducing training time by 68.6% and peak GPU memory by 74.6% relative to full fine-tuning (3,687 MB vs. 14,493 MB; see Table 5).

Figure 1 summarizes these trends across all configurations, showing the F1 scores for all methods and learning rate combinations.

Table 1: BERTimbau-Base on SQuAD-BR (QA) with high learning rate (2×10^{-4}). Metrics: F1 and Exact Match (EM).

Method	Ep.	F1	EM	Time
Full FT	2	**79.74**	**67.15**	01:40:02
LoRA	2	**78.01**	**64.85**	00:31:37
QLoRA	2	73.23	60.26	00:30:03
DoRA	2	**78.01**	**64.89**	00:40:23
QDoRA	2	74.41	61.26	00:42:03
Full FT	3	**78.33**	**65.54**	02:29:04
LoRA	3	**78.01**	**65.03**	00:46:47
QLoRA	3	74.16	61.24	00:44:42
DoRA	3	**78.27**	**65.08**	00:59:59
QDoRA	3	74.46	61.32	01:02:44

Table 2: BERTimbau-Base on SQuAD-BR (QA) with standard learning rate (4.25×10^{-5}). Metrics: F1 and Exact Match (EM).

Method	Ep.	F1	EM	Time
Full FT	2	**82.79**	**70.91**	01:40:04
LoRA	2	**71.81**	**58.07**	00:31:49
QLoRA	2	53.52	40.54	00:30:02
DoRA	2	71.36	57.68	00:40:13
QDoRA	2	54.10	41.15	00:42:10
Full FT	3	**82.18**	**70.40**	02:28:52
LoRA	3	**72.01**	**58.32**	00:41:30
QLoRA	3	53.19	39.81	00:40:20
DoRA	3	71.50	57.65	00:53:58
QDoRA	3	58.42	45.37	00:55:00

In contrast, with the standard *learning rate* ($lr = 4.25 \times 10^{-5}$), the performance of PEFT methods degrades significantly. LoRA reaches only F1=71.81 (86.7% of *Full FT*), while the quantized variants almost completely collapse (QLoRA: F1=53.52, QDoRA: F1=54.10), losing more than 20 F1 points relative to full fine-tuning. Using $lr = 2 \times 10^{-4}$, LoRA improves by **+6.20 F1 points** (F1=78.01, 94.2% of *Full FT*) and QLoRA recovers **+19.71 F1 points** (F1=73.23), while reducing peak memory to just 1,897 MB (86.9% reduction vs. full fine-tuning).

5.2 Performance on BERTimbau-Large

Tables 3 and 4 present the complete results for BERTimbau-Large. At a high learning rate ($lr = 2 \times 10^{-4}$), the Full FT baseline collapses critically (F1=3.02 and F1=5.14 at 2 and 3 epochs, respectively), while PEFT methods maintain robust performance. In particular, LoRA reaches F1=81.32 (95.8% of the optimal *Full FT* baseline) with a 73.5% reduction in training time and 50.2% reduction in peak memory (9,019 MB vs. 18,125 MB). QLoRA achieves F1=80.03 while requiring only 3,281 MB—an 81.9% memory reduction—demonstrating the feasibility of training large models under severe hardware constraints (Table 5).

Figure 1 further illustrates this behavior: while Full FT collapses at high learning rates (F1=3.02 and F1=5.14, shown in parentheses for 2 and 3 epochs), PEFT methods remain stable, suggesting that the low-rank structure of LoRA and DoRA acts as an implicit regularizer, preventing divergence during training.

Table 3: BERTimbau-Large on SQuAD-BR (QA) with high learning rate (2×10^{-4}). Metrics: F1 and Exact Match (EM).

Method	Ep.	F1	EM	Time
Full FT	2	**3.02**	**0.03**	05:15:30
LoRA	2	**81.32**	**68.67**	01:23:41
QLoRA	2	80.03	67.17	01:19:15
DoRA	2	80.61	68.09	01:47:37
QDoRA	2	77.96	65.05	01:57:30
Full FT	3	**5.14**	**0.11**	07:50:02
LoRA	3	**81.27**	**68.67**	02:05:20
QLoRA	3	80.28	67.63	01:57:39
DoRA	3	81.22	68.70	02:40:52
QDoRA	3	79.61	66.99	02:54:52

Table 4: BERTimbau-Large on SQuAD-BR (QA) with standard learning rate (4.25×10^{-5}). Metrics: F1 and Exact Match (EM).

Method	Ep.	F1	EM	Time
Full FT	2	**84.86**	**73.00**	05:15:39
LoRA	2	**75.65**	**62.21**	01:23:28
QLoRA	2	68.23	54.92	01:19:12
DoRA	2	74.93	62.02	01:47:46
QDoRA	2	70.32	56.88	01:57:30
Full FT	3	**83.74**	**72.04**	07:50:46
LoRA	3	**81.28**	**68.63**	02:05:08
QLoRA	3	71.03	57.66	01:58:54
DoRA	3	77.18	63.98	02:41:23
QDoRA	3	71.24	58.15	02:55:23

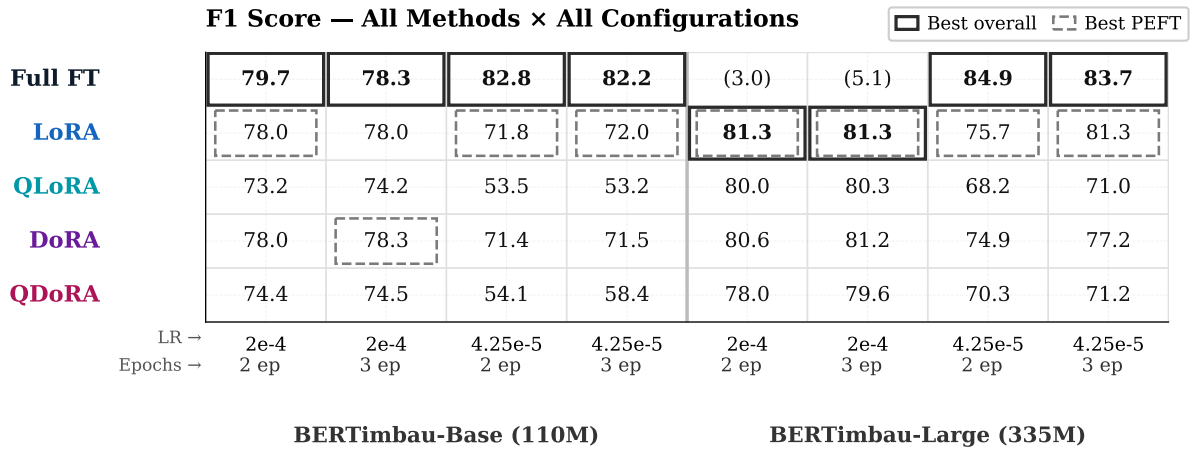

Figure 1: F1 score for all PEFT methods across all configurations (learning rate × epochs × architecture). A bold border remarks the single best overall result per column; gray dashed borders show the best PEFT method per column. When both criteria coincide (i.e., a PEFT method is also the best overall), they are shown together. Values in parentheses indicate training collapse of full fine-tuning (*Full FT*) on BERTimbau-Large under high learning rate (lr=2×10^{-4}), observed across both 2 and 3 epoch settings.

With $lr = 2 \times 10^{-4}$, LoRA achieves F1=81.32 (95.8% of *Full FT* baseline), whereas with $lr = 4.25 \times 10^{-5}$ its performance drops to F1=75.65 (89.1%), a difference of **+5.67 F1 points**. QLoRA drops from F1=80.03 to 68.23, losing 11.80 F1 points under the standard *learning rate*. Crucially, QLoRA on Large shows a degradation of only −4.83 F1 points (vs. −9.56 on Base), an approximate 2× difference that confirms greater quantization resilience in larger models **(H2)**.

An additional finding is the critical collapse of full fine-tuning with $lr = 2 \times 10^{-4}$ (F1=3.02). This optimization divergence occurs because the high learning rate destabilizes the pre-trained representations across the full parameter space, whereas. In contrast, the structures of LoRA and DoRA act as implicit regularizers, preventing the model from drifting too far from the pre-trained manifold.

Table 5: Peak GPU memory (MB) during training on SQuAD-BR.

Method / Setting	Peak Memory (MB)
BERTimbau (Extractive QA, Training)	
Full Fine-tuning (Base)	14,493
LoRA (Base)	3,687
DoRA (Base)	4,295
QLoRA (4-bit, Base)	1,897
QDoRA (4-bit, Base)	2,163
Full Fine-tuning (Large)	18,125
LoRA (Large)	9,019
DoRA (Large)	10,659
QLoRA (4-bit, Large)	3,281
QDoRA (4-bit, Large)	3,331
Generative LLMs (LoRA Fine-tuning)	
Tucano-1B (Zero-shot)	2,682
Tucano-1B + LoRA	4,301
Sabiá-7B (Zero-shot)	17,737
Sabiá-7B + LoRA	15,642

5.3 Peak GPU Memory Consumption

Table 5 consolidates peak GPU memory usage for both BERTimbau variants and the generative models evaluated in Section 5.5, enabling a direct comparison of hardware requirements across all architectures.

Among BERTimbau variants, QLoRA achieves the largest reductions: 86.9% for Base (1,897 MB vs. 14,493 MB) and 81.9% for Large (3,281 MB vs. 18,125 MB), both well within the range of consumer-grade GPUs. Notably, Sabiá-7B shows higher memory consumption at zero-shot (17,737 MB) than with LoRA training (15,642 MB), since zero-shot corresponds to full-precision inference while LoRA fine-tuning uses mixed-precision (4-bit base + bfloat16 adapters). Comparing across architectures, BERTimbau-Base with LoRA (3,687 MB) achieves a similar F1 score to Sabiá-7B with LoRA (15,642 MB) while requiring **4.2× less memory** and **3× less training time**; a compelling efficiency advantage for resource-constrained environments.

5.4 Critical Impact of the Learning Rate

The experiments confirm that the learning rate is the most decisive factor for the success of PEFT

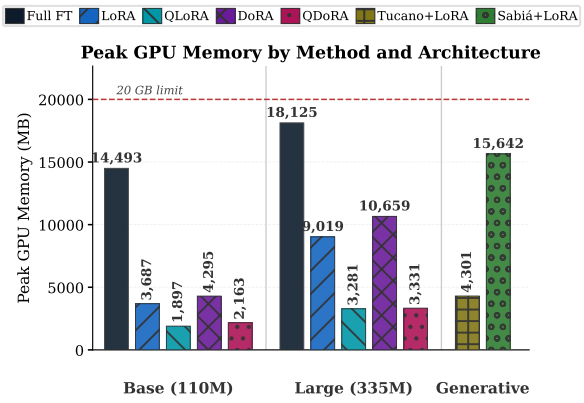

Figure 2: Peak GPU memory consumption per method and architecture. Quantized methods (QLoRA, QDoRA) reduce memory by up to 86.9% for Base and 81.9% for Large relative to Full FT. Memory values for generative models (Tucano-1B, Sabiá-7B) are included for cross-architecture comparison; results for those models are discussed in Section 5.5. The dashed red line marks the 20 GB GPU limit of the hardware used.

methods, consistent with our hypothesis on optimization sensitivity (H3), surpassing the influence of the specific method, the number of epochs, or the application of quantization. Figure 1 makes this effect directly visible: the left columns (lr=2×10^{-4}) show consistently higher F1 than the right columns (lr=4.25×10^{-5}) for all PEFT methods.

For both model sizes, higher learning rates ($lr = 2 \times 10^{-4}$) are critical for maximizing performance. In BERTimbau-Base, LoRA improves performance by **+6.20 F1 points** relative to the standard *learning rate*, whereas in BERTimbau-Large the gain is **+5.67 F1 points**. This pattern is further amplified under quantization: QLoRA exhibits gains of **+19.71 F1 points** in Base and **+11.80 F1 points** in Large when higher *learning rates* are employed. Standard learning rates prove inadequate for PEFT schemes in our experiments, leading to severe degradation and, for Large models, complete *Full FT* collapse.

5.5 Exploratory Evaluation with Generative Models: Tucano and Sabiá

To analyze the feasibility of autoregressive generative models for extractive Question Answering, we conducted an exploratory evaluation of Tucano (Corrêa et al., 2025) and Sabiá (Pires et al., 2023) on the same SQuAD-BR evaluation set used for BERTimbau. Unlike BERTimbau, these models are designed for free-text generation, which introduces a structural mismatch with SQuAD's exact-span extraction evaluation protocol. Consequently, we interpret these results as indicative of task-architecture compatibility rather than a direct performance comparison. Therefore, they should not be interpreted as a direct benchmark against encoder-based models. We evaluated two configurations: *zero-shot* inference and fine-tuning with LoRA.

Table 6: Exploratory results with generative models for extractive QA.

Model	Config.	EM (%)	F1 (%)	Time
Tucano	Zero-shot	4.02	14.68	00:16:59
Sabiá	Zero-shot	25.37	39.82	00:55:26
Tucano	LoRA	49.30	63.86	00:25:28
Sabiá	LoRA	64.11	78.10	01:31:15
For reference (same evaluation set):				
B-Base	LoRA	64.85	78.01	00:31:37
B-Large	LoRA	68.67	81.32	01:23:41

Table 6 reveals that LoRA fine-tuning substantially improves both generative models: Tucano improves from F1=14.68% to F1=63.86% (+49.18 points), while Sabiá improves from F1=39.82% to F1=78.10% (+38.28 points). Notably, Sabiá with LoRA (F1=78.10%) reaches a performance level comparable to BERTimbau-Base with LoRA (F1=78.01%), and BERTimbau-Large (F1=81.32%) still achieves the best overall result. However, this competitive F1 comes at a substantial computational cost: Sabiá-7B with LoRA requires 15,642 MB of GPU memory and 01:31:15 of training time, compared to 3,687 MB and 00:31:37 for BERTimbau-Base—a **4.2× memory overhead** and **3× longer training** for an equivalent F1 score. These results confirm that, for extractive QA in Brazilian Portuguese, encoder-based architectures offer a superior efficiency-performance trade-off, particularly in resource-constrained environments.

6 Discussion

The results show that PEFT techniques preserve most of the performance of full fine-tuning while achieving substantial cost reductions. BERTimbau-Large achieves 95.8% of baseline performance (F1=81.32 vs 84.86) while reducing training time by 73.5% and peak memory by 50.2%. In BERTimbau-Base, LoRA retains 94.2% of the baseline performance while achieving a 68.6% reduction in training-time, reinforcing the viability of PEFT in resource-constrained scenarios. These results are close to the classical BERTimbau base-

lines reported for SQuAD-BR (Souza et al., 2020; da Silva et al., 2022), validating our experimental configuration and confirming the hypothesis on low-resource efficiency (H1).

The learning rate emerges as the most decisive factor for PEFT success, consistent with the hypothesis on optimization sensitivity (H3). With $lr = 2 \times 10^{-4}$, LoRA and DoRA achieve their best performance, whereas with the standard learning rate (4.25×10^{-5}), the performance degrades consistently, with losses of up to 6.20 F1 points in Base and 5.67 points in Large. This effect becomes critical under quantization: QLoRA improves by +19.71 F1 points in Base when the learning rate is increased, indicating that quantized PEFT schemes require different optimization dynamics than full fine-tuning.

A key finding is the opposite behavior of full fine-tuning under high learning rates. In BERTimbau-Large, full fine-tuning collapses almost completely (F1=3.02 with $lr = 2 \times 10^{-4}$), while PEFT methods remain stable. This contrast suggests that the low-rank updates in LoRA and DoRA serve as implicit regularizers, limiting the magnitude of parameter updates even with aggressive learning rates, thereby preventing optimization divergence.

Four-bit quantization introduces a size-dependent degradation. In BERTimbau-Base, QLoRA loses 9.56 F1 points relative to the optimal baseline, whereas in BERTimbau-Large this loss is reduced to 4.83 points, retaining 94.3% of baseline performance. This approximate $2\times$ difference confirms **(H2)**. While QLoRA underperforms standard LoRA in F1-score, this slight degradation is heavily outweighed by its memory reduction: 86.9% for Base (1,897 MB) and 81.9% for Large (3,281 MB), well within the range of consumer-grade GPUs and effectively eliminating out-of-memory errors in resource-constrained environments.

The comparison between LoRA and DoRA shows that DoRA introduces a consistent temporal overhead of approximately 28% in BERTimbau-Large without clear improvements in F1 or Exact Match. LoRA thus emerges as the more efficient and preferred option: it closely approaches the classical BERTimbau baselines (Souza et al., 2020; da Silva et al., 2022), significantly reduces training time and energy consumption, and, when combined with controlled quantization on Large models, enables near state-of-the-art performance under strict computational constraints.

The exploratory evaluation with Tucano and Sabiá highlights a critical efficiency gap between encoder and decoder architectures. While Sabiá-7B with LoRA achieves a competitive F1 score (78.10%) comparable to BERTimbau-Base (78.01%), this comes at the cost of $4.2\times$ more GPU memory (15,642 MB vs. 3,687 MB) and $3\times$ longer training time. BERTimbau-Large with LoRA (F1=81.32%) surpasses all generative models while requiring significantly fewer computational resources. These findings confirm that for extractive QA in Brazilian Portuguese, smaller encoder-based architectures offer a clearly superior efficiency-performance trade-off. In fact, recent studies have demonstrated that Portuguese-native generative models excel when applied to well-aligned generative tasks, such as translating natural language to SQL (Freitas et al., 2025), reinforcing that task-architecture alignment is a key factor in model selection.

7 Conclusions

This work presented a systematic evaluation of PEFT and quantization techniques applied to BERTimbau for Question Answering on SQuAD-BR. We demonstrated that it is possible to achieve competitive performance while substantially reducing computational cost. LoRA on BERTimbau-Large reaches 95.8% of baseline performance while reducing training time by 73%. In contrast, QLoRAoRA preserves 94.3% of baseline performance while cutting peak GPU memory by 81.9% (3,281 MB), well within the range of consumer-grade GPUs.

Experimental results validate the three core hypotheses driving this study. First, regarding low-resource efficiency **(H1)**, PEFT methods drastically reduce computational and memory overhead while maintaining near-baseline performance. Second, concerning scale robustness **(H2)**, larger models show approximately $2\times$ better resilience to aggressive 4-bit quantization (4.83 vs 9.56 F1 loss), positioning BERTimbau-Large with QLoRA as the optimal choice for GPU memory-constrained scenarios. Third, validating optimization sensitivity **(H3)**, higher learning rates (2×10^{-4}) appear critical for PEFT success, substantially improving F1 by up to +19.71 points over standard rates in our experiments. Additionally, DoRA offers no practical advantages over LoRA, matching its performance at the cost of a 28% increase in training time.

The exploratory evaluation confirms that while generative models such as Sabiá-7B can reach competitive F1 scores with LoRA fine-tuning (78.10% vs. 78.01% for BERTimbau-Base), they require 4.2× more GPU memory and 3× more training time. BERTimbau-Large with LoRA (F1=81.32%) achieves the best overall result at a fraction of the computational cost, reinforcing that smaller encoder-based architectures offer a superior efficiency-performance trade-off for extractive QA in resource-constrained environments.

The findings provide concrete guidelines for sustainable QA research: prioritize BERTimbau-Large with LoRA or QLoRA under high learning rates when GPUs with limited VRAM are available; use full fine-tuning only as a reference baseline; and avoid aggressive quantization on Base models unless memory constraints are extreme. The 73.5% reduction in training time makes PEFT techniques particularly valuable for resource-constrained environments, promoting more accessible NLP practices aligned with *Green AI* principles (Strubell et al., 2019).

As future work, we propose extending these models to broader Portuguese datasets and downstream applications such as explainable fact-checking (Yang et al., 2022), while exploring automated hyperparameter searches that jointly optimize performance and computational cost. By integrating PEFT with adaptive strategies, like progressive layer unfreezing (Nina and Vega-Oliveros, 2025), we also aim to extend these results to indigenous and underrepresented languages, thereby driving broader digital inclusion.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001 and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

The authors also acknowledge the use of Anthropic's Claude Sonnet 4.6 and Grammarly for language editing and correction of English errors throughout the manuscript. All scientific content, experimental design, and interpretations are the sole responsibility of the authors.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *Preprint*, arXiv:2012.13255.

Gian Franco Condori-Luna, Didier Vega-Oliveros, and Marcelo da Silva Reis. 2026. Reducing dependence on labeled data: A self-supervised fine-tuning approach for low-resource language models. In *Intelligent Systems*, pages 395–409, Cham. Springer Nature Switzerland.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Nicholas Kluge Corrêa, Aniket Sen, Sophia Falk, and Shiza Fatimah. 2025. Tucano: Advancing neural text generation for portuguese. *Patterns*, 6(11):101325.

E. da Silva, J. Laterza, and T. Faleiros. 2022. New state-of-the-art for question answering on portuguese squad v1.1. In *Anais do X Symposium on Knowledge Discovery, Mining and Learning*, pages 98–105, Porto Alegre, RS, Brasil. SBC.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *Preprint*, arXiv:2110.02861.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Christian Freitas, Didier A. Vega-Oliveros, and Lilian Berton. 2025. Enhancing industrial data access with text-to-sql using portuguese llms and langgraph. In *2025 12th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 278–282.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. *Preprint*, arXiv:2110.04366.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training

compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *Preprint*, arXiv:2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Preprint*, arXiv:2101.00190.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. *Preprint*, arXiv:2402.09353.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Ricardo Mazza Zago and Luciane Agnoletti dos Santos Pedotti. 2024. Bertugues: A novel bert transformer model pre-trained for brazilian portuguese. *Semina: Ciências Exatas e Tecnológicas*, 45:e50630.

Mariela M. Nina and Didier A. Vega-Oliveros. 2025. Maximizing model adaptation for low-resource languages: A progressive unfreezing strategy for spanish-aymara translation. In *2025 12th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 259–263.

Ramon Pires, Hugo Abonizio, Thales Sales Almeida, and Rodrigo Nogueira. 2023. *Sabiá: Portuguese Large Language Models*, page 226–240. Springer Nature Switzerland.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Preprint*, arXiv:1606.05250.

Tim Salimans and Diederik P. Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Preprint*, arXiv:1602.07868.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. Bertimbau: Pretrained bert models for brazilian portuguese. In *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *Preprint*, arXiv:1906.02243.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Jing Yang, Didier Vega-Oliveros, Taís Seibt, and Anderson Rocha. 2022. Explainable fact-checking through question answering. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8952–8956.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*.