1. Dataset & Motivation

For this project, I selected the **AG News dataset**, a widely used benchmark for text classification. The dataset consists of 120,000 training samples and 7,600 test samples, evenly distributed across four balanced categories: *World, Sports, Business,* and *Science/Technology*. Each example contains a news title and description, which together provide sufficient context for topic classification. This dataset is a media domain dataset.

The motivation for choosing AG News is threefold. First, it is a **general benchmark dataset** that is frequently used in NLP research, making it suitable for evaluating fine-tuning strategies on pretrained Transformer models. Second, the dataset is **large, clean, and balanced**, which reduces the risk of bias and allows for fair performance comparison across classes. Finally, it is **computationally manageable** in the free-tier Google Colab environment: the text lengths are relatively short, and the dataset size enables efficient training while still being representative enough to draw meaningful conclusions.

Given the constraints of limited compute and the goal of comparing **parameter-efficient tuning (LoRA)** with **full fine-tuning**, AG News offers the ideal balance of practicality and rigor.

2. Models

For this project, I chose **DistilBERT** as the pretrained Transformer model. DistilBERT is a distilled version of BERT, designed to be **smaller, faster, and lighter**, while retaining approximately **95% of BERT's performance** on downstream NLP tasks. It has only 66 million parameters compared to BERT's 110 million, making it more suitable for training under **resource-constrained environments** such as the Google Colab free tier that I used.

The motivation for using DistilBERT is twofold. First, AG News is a **topic classification benchmark** with short news headlines and descriptions, which makes it a good match for DistilBERT's ability to capture semantic meaning in compact text. Larger models such as RoBERTa or GPT-2 may offer slight improvements in performance, but they require significantly more memory and computational power, which is impractical for this assignment setup. Second, DistilBERT is **widely adopted and well-supported** in HuggingFace's ecosystem, enabling easy integration with parameter-efficient tuning methods such as LoRA.

Overall, DistilBERT strikes a strong balance between **performance and efficiency**, making it an ideal base model for experimenting with fine-tuning strategies. By choosing this model, I could focus on evaluating the effectiveness of **different fine-tuning approaches** (LoRA vs full parameter tuning) without being limited by hardware constraints.

3. Fine-tuning strategies

To evaluate the effectiveness of different training approaches, I implemented and compared two fine-tuning strategies on DistilBERT, along with a baseline reference. These strategies highlight the trade-offs between efficiency and performance under limited computational resources.

3.1 Baseline (Frozen Model + Linear Classifier).
In the baseline setup, all DistilBERT layers were frozen, and only a linear classification head was trained on top. This provided a lightweight reference model with minimal training cost, establishing a lower bound for performance.

3.2 LoRA (Low-Rank Adaptation).
LoRA is a parameter-efficient fine-tuning (PEFT) method that inserts small low-rank trainable adapters into the Transformer layers while freezing the majority of model parameters. This allows the model to adapt to the AG News dataset while updating only ~1–2% of parameters, drastically reducing memory usage and training time. LoRA was chosen because it is particularly well-suited for resource-constrained environments like Colab, and it enables efficient experimentation with fine-tuning.

3.3 Full Fine-Tuning.
In contrast, full fine-tuning updates **all model parameters** of DistilBERT, including the Transformer layers and classification head. This strategy typically achieves the best possible performance but requires substantially more memory and computational resources. By comparing LoRA to full fine-tuning, I aimed to highlight the efficiency trade-offs between these two approaches.

Overall, these strategies were selected to provide a **comprehensive comparison**: a lightweight baseline, a resource-efficient fine-tuning approach (LoRA), and a resource-intensive full fine-tuning method. This design enables a meaningful analysis of **accuracy, efficiency, and error patterns** across different levels of parameter updating.

4. Experimental Setup

All experiments were conducted on **Google Colab (free tier)** with GPU acceleration, using HuggingFace's *transformers, datasets, evaluate,* and *peft* libraries. Training and evaluation were managed through the *Trainer* API, with custom metrics for **accuracy** and **macro-F1** (the latter ensuring balanced evaluation across all four AG News classes).

4.1 Baseline Setup (Frozen DistilBERT + Linear Head)

- **Model:** DistilBERT encoder frozen; only a linear classifier trained.
- **Learning rate: 5e-4**, higher than typical fine-tuning, since only the classifier head is updated and converges quickly.
- **Epochs: 2**, sufficient because training a small linear head requires fewer passes.
- **Batch sizes:** 32 (train), 64 (eval), chosen to balance speed and GPU memory.
- **FP16:** Enabled to reduce VRAM usage.

4.2 LoRA Setup (Parameter-Efficient Fine-Tuning)

- **LoRA config:** rank=8, α=16, dropout=0.1.
- **Trainable parameters:** ~**1.31%** (888k out of 67.8M).
- **Learning rate: 2e-4**, slightly higher than full FT since LoRA trains fewer parameters and can adapt quickly without destabilizing the frozen backbone.
- **Epochs: 4**, giving adapters more updates to capture domain-specific patterns.
- **Batch sizes:** 32 (train), 64 (eval).
- **FP16:** Enabled.

4.3 Full Fine-Tuning Setup

- **Model:** All DistilBERT parameters updated (100%).
- **Learning rate: 2e-5**, much lower to prevent catastrophic forgetting when fine-tuning all layers.
- **Epochs: 4**, to allow the entire network to adapt.
- **Batch sizes:** 32 (train), 64 (eval).
- **Warmup ratio: 0.06** for stable early training.
- **Weight decay: 0.01** to regularize the large number of trainable parameters.
- **FP16:** Enabled to fit training into ~2 GB VRAM.

This setup ensured a **controlled comparison**: all models were trained under consistent conditions, with hyperparameters adjusted only where necessary to reflect differences in fine-tuning strategy. The rationale behind learning rates, epochs, and regularization reflects best practices for each approach.e about e about

5. Results & Analysis

My experiments evaluated three strategies on the AG News dataset: a **baseline frozen model with a linear classification head**, **LoRA fine-tuning**, and **full parameter fine-tuning**. I compare them in terms of accuracy, macro-F1, computational efficiency, and error patterns.

5.1 Overall Performance

Method	Trainable Params (%)	Peak VRAM (MB)	Time (s)	Accuracy (Val)	Macro-F1 (Val)	Accuracy (Test)	Macro-F1 (Test)
Baseline	0.0046%	467	201	0.8722	0.8720	0.8737	0.8736
LoRA (r=8, α=16, drop=0.1)	1.31%	2522	1040	0.9423	0.9423	0.9396	0.9397
Full Fine-Tuning	100%	2075	1279	0.9459	0.9459	0.9438	0.9439

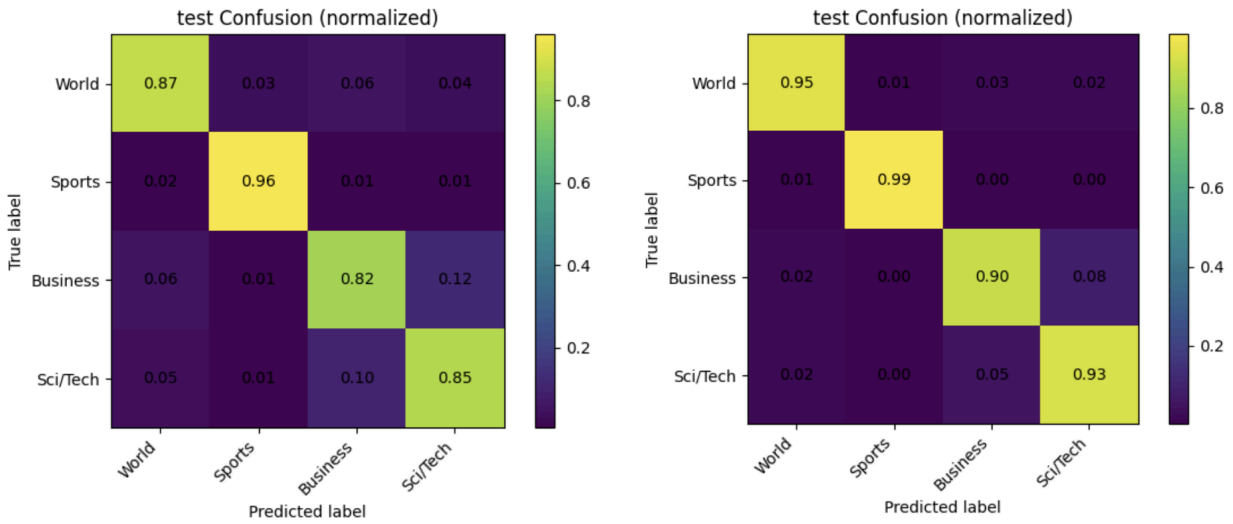
- **Baseline** achieved 87.4% accuracy and macro-F1 of 0.874 on the test set.
- **LoRA fine-tuning** substantially improved performance to 93.9% accuracy and macro-F1 of 0.940, while updating only ~1.3% of model parameters.
- **Full fine-tuning** achieved the best overall results, with 94.4% accuracy and macro-F1 of 0.944, though the improvement over LoRA was marginal (<1%).

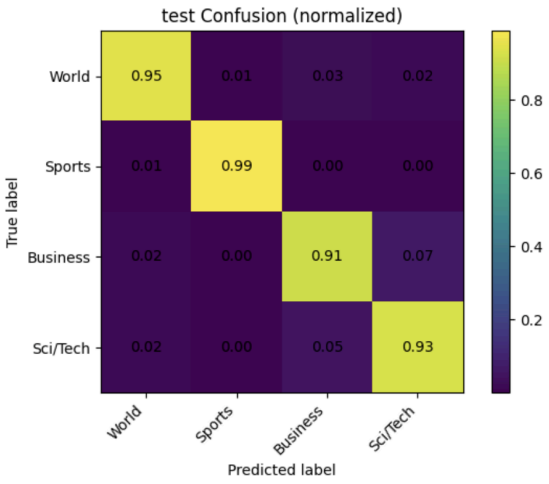
5.2 Efficiency Trade-Offs

- The baseline was fastest to train (~200 seconds) and used the least GPU memory (~467MB), but its accuracy lagged behind.
- LoRA required ~1040 seconds and ~2.5GB VRAM but reached near state-of-the-art performance while being parameter-efficient.
- Full fine-tuning consumed the most resources (~1280 seconds, ~2.0GB VRAM, 100% of parameters updated), delivering only slightly better results than LoRA.

→ This highlights the *efficiency–accuracy trade-off*: LoRA provides a nearly optimal balance, achieving almost full FT performance at a fraction of the cost.

5.3 Confusion Matrix Insights





1st image: Baseline, 2nd image: LoRA, 3rd image: Full Fine-Tuning

The confusion matrices provide a class-level view of where models struggle:

- **Baseline**: While "Sports" was classified almost perfectly (96% accuracy), the model struggled to distinguish between **Business** and **Sci/Tech**, with *~12%* of Business articles misclassified as Sci/Tech.
- **LoRA**: Reduced this confusion significantly, boosting Business accuracy to ~90% and Sci/Tech to ~93%. The World and Sports categories also improved, reaching ~95% and ~99% accuracy respectively.
- **Full Fine-Tuning**: Further improved Business (~91%) and Sci/Tech (~93%) but only marginally compared to LoRA.

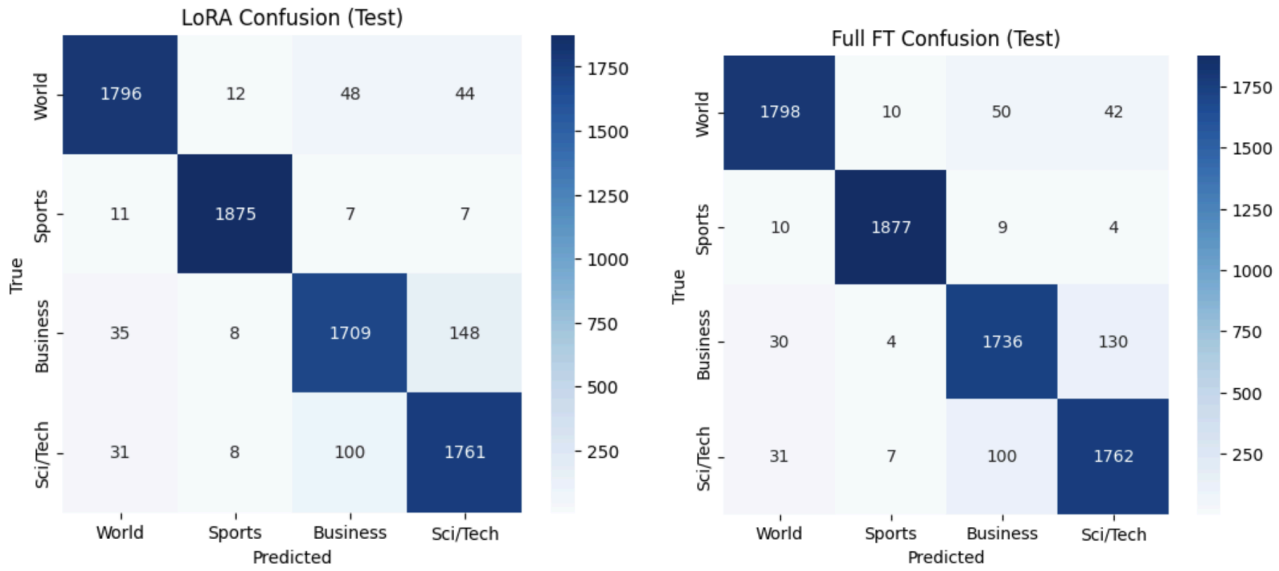
5.4 Overall Analysis

The results demonstrate that **parameter-efficient tuning (LoRA)** is highly competitive with full fine-tuning for topic classification. It resolves most of the baseline's weaknesses, particularly the Business–Sci/Tech confusion, and delivers nearly the same accuracy while requiring far fewer trainable parameters and compute resources. Full fine-tuning provides slightly higher performance but at significantly greater cost, which may not be justified for resource-constrained environments.

6. Further experiment (Error analysis)

To complement the overall performance metrics, I conducted a series of error analyses to better understand where the models struggled and how their behaviors differed. **I will specifically focus on the misclassification between Business and Sci/Tech.**

6.1 Misclassification Patterns



Examining misclassified examples revealed a consistent weakness across both LoRA and full fine-tuning: **confusion between the Business and Sci/Tech categories**. Multiple test samples that were truly Business articles were incorrectly predicted as Sci/Tech. This suggests overlap in vocabulary and topical content — for example, technology-driven financial news (e.g., "tech IPOs" or "digital banking") may blur the boundary between these categories.

However, from two confusion matrices, I can see that full fine-tuning showed slightly fewer such confusions, aligning with its marginally higher class-level F1 scores for Business.

6.2 Effect of Text Length

```
LoRA length-based analysis
Short texts (≤20 words): Acc=0.883 | F1=0.784 | n=94
Long texts  (>20 words): Acc=0.940 | F1=0.940 | n=7506

Full FT length-based analysis
Short texts (≤20 words): Acc=0.894 | F1=0.789 | n=94
Long texts  (>20 words): Acc=0.944 | F1=0.944 | n=7506
```

I next compared performance across **short texts (≤20 words)** and **longer texts (>20 words)**.

- **LoRA:** Short texts achieved only **Acc=0.883, F1=0.784**, whereas longer texts were classified far more reliably (**Acc=0.940, F1=0.940**).
- **Full Fine-Tuning:** Similarly, short texts underperformed (**Acc=0.894, F1=0.789**) compared to longer texts (**Acc=0.944, F1=0.944**).

This indicates that **brevity reduces context**, making classification harder. Both models leverage contextual embeddings, so insufficient word coverage limits their discriminative ability. Encouragingly, both LoRA and full fine-tuning reached near-identical performance on longer texts, reinforcing that parameter-efficient tuning can capture rich contextual signals when the input is sufficiently informative.

6.3 Confidence Calibration

```
LoRA confidence bins
Conf 0.5–0.6: n=6 | Acc=0.667
Conf 0.6–0.7: n=3 | Acc=0.333
Conf 0.7–0.8: n=11 | Acc=0.636
Conf 0.8–0.9: n=5 | Acc=0.200
Conf 0.9–1.0: n=6 | Acc=0.500

Full FT confidence bins
Conf 0.6–0.7: n=1 | Acc=0.000
Conf 0.7–0.8: n=1 | Acc=1.000
Conf 0.8–0.9: n=1 | Acc=0.000
Conf 0.9–1.0: n=2 | Acc=1.000
```

I also analyzed **confidence bins** (model prediction probabilities).

- **LoRA:** Showed mixed calibration, with accuracies ranging from **66.7% in the 0.5–0.6 bin** to just **20% in the 0.8–0.9 bin**, suggesting occasional overconfidence in incorrect predictions.
- **Full Fine-Tuning:** Displayed similar issues in low-confidence bins but generally improved in high-confidence ranges (e.g., 100% accuracy for some samples in 0.9–1.0).

Interpretation: Both models can assign high probabilities, but these scores are not always trustworthy indicators of correctness. This means a "confident" prediction is not guaranteed to be right — a weakness that could be critical in domains like healthcare or finance. In future work, calibration methods (e.g., **temperature scaling or isotonic regression**) could be applied to make probability estimates more reliable.

6.4 Key Takeaways

- Both LoRA and full fine-tuning perform well overall, but **Business vs. Sci/Tech remains the hardest distinction**.
- **Short texts** are a weak spot for both approaches, with substantially lower F1 scores.
- Confidence analysis shows **overconfidence in misclassifications**, suggesting opportunities for better calibration.

Overall, these insights highlight that **LoRA achieves competitive results despite updating only ~1.3% of parameters**, while full fine-tuning slightly improves robustness and calibration. Error analysis therefore strengthens the comparative evaluation by uncovering nuanced model behaviors that aggregate metrics alone might obscure.