

Parameter-Efficient Fine-Tuning of GPT-2: Comparing LoRA, LoReFT, and Prefix Tuning Across Classification and Generation Tasks

Stanford CS224N Default Project

Zijian Luo

Department of Symbolic Systems
Stanford University
zijianl@stanford.edu

Sally Wang

Department of Computer Science
Stanford University
sallyw02@stanford.edu

Abstract

Fine-tuning large language models on downstream tasks remains expensive when all parameters are updated. Parameter-Efficient Fine-Tuning (PEFT) methods address this by training only a small subset of parameters while keeping the pretrained backbone frozen. In this work, we implement GPT-2 from scratch—including causal self-attention, transformer layers, positional embeddings, and the AdamW optimizer—and evaluate three PEFT strategies against full fine-tuning on three diverse NLP tasks: sentiment classification (SST-2), cloze-style paraphrase detection (Quora Question Pairs), and conditional sonnet generation (Shakespeare). We compare Low-Rank Adaptation (LoRA), Low-Rank Representation Fine-Tuning (LoReFT), and Prefix Tuning. Our results show that LoReFT with rank 4 matches full fine-tuning on sentiment analysis (91.2% vs. 91.5% accuracy) while training only 75K parameters—a $1,652\times$ reduction. On paraphrase generation, LoReFT (rank 8) achieves comparable BLEU (7.04 vs. 7.86) and *superior* BERTScore F1 (0.572 vs. 0.528) with $835\times$ fewer parameters. For sonnet generation, LoReFT (rank 2) achieves the highest rhyme accuracy (48.6%) among all methods, surpassing full fine-tuning (45.7%). These findings demonstrate that representation-level interventions can be remarkably effective, achieving near-parity with full fine-tuning at a fraction of the parameter budget, though they require careful design of intervention positions for generative tasks.

1 Key Information

- Mentor: Chenglei Si
- External Collaborators: None
- Sharing project: No

2 Introduction

Large pretrained language models such as GPT-2 [1] have demonstrated strong generalization across diverse NLP tasks. However, adapting these models to specific downstream applications via full fine-tuning—updating all 124 million parameters—is computationally expensive, memory-intensive, and can lead to catastrophic forgetting when data is scarce. Parameter-Efficient Fine-Tuning (PEFT) methods address these challenges by training only a small fraction of model parameters, keeping the pretrained backbone frozen.

Three prominent PEFT paradigms have emerged: (1) **Low-Rank Adaptation (LoRA)** [2], which injects trainable low-rank matrices into weight projections; (2) **Prefix Tuning** [3], which prepends learnable continuous vectors to the key-value pairs in attention layers; and (3) **Representation**

Fine-Tuning (ReFT) [4], which learns lightweight interventions on hidden representations rather than modifying weights. While LoRA has become the dominant PEFT approach, the recently proposed LoReFT offers an intriguing alternative by operating directly in the representation space with dramatically fewer parameters.

In this work, we make the following contributions:

1. We implement GPT-2 from scratch, including causal self-attention, transformer layers, weight-tied embeddings, and the AdamW optimizer, achieving performance that matches the HuggingFace reference implementation.
2. We evaluate three PEFT methods (LoRA, LoReFT, Prefix Tuning) against full fine-tuning across three tasks spanning classification and generation: sentiment analysis (SST-2), paraphrase detection (Quora), and sonnet generation (Shakespeare).
3. We provide a systematic comparison along multiple dimensions—task accuracy, parameter efficiency, training time, and memory usage—revealing that LoReFT can match or exceed full fine-tuning quality with $845\text{--}1,656\times$ fewer trainable parameters.
4. We analyze intervention position strategies for LoReFT in generative tasks, identifying a train-test consistency issue that we resolve with a prefix-based intervention scheme.

3 Related Work

Transformer Language Models. The Transformer architecture [5] underpins modern language models. GPT-2 [1] demonstrated that autoregressive pretraining at scale yields strong zero-shot and few-shot capabilities. Our work builds on GPT-2 as the base architecture.

Parameter-Efficient Fine-Tuning. Adapter modules [6] pioneered PEFT by inserting small bottleneck layers between Transformer blocks. Prefix Tuning [3] and Prompt Tuning [7] instead prepend learnable continuous vectors to the input or attention layers. LoRA [2] injects trainable low-rank decomposition matrices into the attention projections, enabling weight merging at inference time with zero additional latency. LoRA has become the de facto standard for PEFT due to its simplicity and strong empirical performance.

Representation Fine-Tuning. Wu et al. [4] introduced ReFT, a family of methods that fine-tune models by learning interventions on hidden representations rather than modifying weights. Drawing on causal abstraction [8], LoReFT learns a low-rank orthogonal projection that rotates and shifts representations at specific layer-position pairs. The authors showed that LoReFT matches or exceeds LoRA on commonsense reasoning and instruction following, using $10\text{--}50\times$ fewer parameters. Our work extends this comparison to paraphrase detection and constrained poetry generation, two tasks not explored in the original LoReFT paper.

4 Approach

4.1 GPT-2 Implementation

We implement GPT-2 (117M parameters) from scratch, including:

- **Causal Self-Attention:** Scaled dot-product attention with a causal mask:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V \quad (1)$$

where M is a lower-triangular mask that prevents attending to future tokens.

- **Transformer Layers:** Pre-LayerNorm architecture with residual connections and dropout applied after each sub-layer (attention and feed-forward).
- **Weight-Tied Embeddings:** The output projection reuses the input embedding matrix via $\text{logits} = h \cdot W_E^\top$, following GPT-2’s design.
- **AdamW Optimizer** [9]: We implement decoupled weight decay with bias-corrected moment estimates.

Pretrained weights from HuggingFace are loaded into our custom architecture to initialize the model before fine-tuning.

4.2 Task Formulations

Sentiment Analysis. We formulate sentiment classification on SST-2 [10] as extracting the last token’s hidden state and passing it through a learned linear classifier: $\hat{y} = \text{softmax}(W_c \cdot h_{\text{last}} + b_c)$, where h_{last} is the hidden state at the final non-padding position. We evaluate on both SST (5-class) and CFIMDB [11] (2-class) datasets.

Paraphrase Detection. Given two sentences, the model determines whether they are paraphrases using a cloze-style formulation. The input is formatted as “[sent1] [SEP] [sent2] [SEP] Is this a paraphrase? yes/no”, and the model predicts the next token. The logits for “yes” and “no” token IDs determine the prediction.

Sonnet Generation. The model is trained autoregressively on Shakespeare’s 154 sonnets. At test time, it receives the first three lines as a prompt and must generate the remaining 11 lines in the Shakespearean form (14 lines, iambic pentameter, ABAB CDCD EFEF GG rhyme scheme).

4.3 PEFT Methods

Full Fine-Tuning (Baseline). All 124M parameters are updated. This serves as the performance ceiling but requires the most memory and compute.

LoRA [2]. We freeze the base weights $W_0 \in \mathbb{R}^{d \times k}$ and add trainable low-rank matrices:

$$h = W_0 x + \frac{\alpha}{r} B A x, \quad B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k} \quad (2)$$

We apply LoRA to the combined QKV projection (c_attn) and output projection (c_proj) in all 12 layers, with rank $r = 8$ and $\alpha = 16$.

Prefix Tuning [3]. We prepend $L = 20$ learnable continuous vectors to the key and value matrices in each attention layer. A reparameterization MLP maps a smaller embedding to the prefix keys/values for training stability. Total trainable parameters: $\sim 370\text{K}$.

LoReFT [4]. We learn an orthogonal low-rank intervention at all 12 layers. For each intervention, a projection matrix $R \in \mathbb{R}^{d \times r}$ and bias $b \in \mathbb{R}^r$ modify the hidden representation h at selected positions:

$$\tilde{h} = h + R R^\top (W h + b - h) \quad (3)$$

where $W \in \mathbb{R}^{r \times d}$ is a learned linear map. For classification tasks, we intervene at the first and last token positions (“first+last”). For generation tasks, we use a prefix strategy (“f10”), intervening on the first 10 token positions to ensure consistency between training and inference. This design choice was critical: intervening on *all* positions during training but only on prompt tokens during generation caused a train-test mismatch that collapsed performance.

5 Experiments

5.1 Data

We evaluate on three datasets:

- **SST-2** [10]: 6,920 training / 872 development sentences for 5-class sentiment.
- **Quora Question Pairs** [12]: 283K training / 40K development pairs for binary paraphrase detection, formulated as cloze-style next-token prediction.
- **Shakespeare Sonnets**: 154 sonnets for training, with held-out sonnets for generation evaluation. The model receives the first 3 lines and generates the remaining 11. Our original proposal planned to augment this with ~ 500 public domain sonnets from Project Gutenberg; in practice, formatting inconsistencies made normalization unreliable, so we restricted training to Shakespeare’s 154 sonnets. This smaller training set may partly explain the difficulty all methods face in achieving high syllable accuracy.

5.2 Evaluation Metrics

Sentiment: Accuracy and macro-F1 on the SST-2 development set. **Paraphrase:** BLEU [13] and BERTScore F1 [14] on the development set. **Sonnet:** Validation loss (cross-entropy), plus structural metrics on generated sonnets: rhyme accuracy (percentage of line pairs following the ABAB CDCD EFEF GG scheme), syllable accuracy (percentage of lines with 10 syllables), and 14-line percentage.

5.3 Experimental Details

All experiments use our custom GPT-2 (117M parameters) with pretrained weights, trained on a single NVIDIA A100 (80GB) via Modal cloud infrastructure. We use mixed-precision (FP16) training with gradient clipping (max norm 1.0).

Full fine-tuning: Learning rate 2×10^{-5} , weight decay 0.01, linear warmup (6% of steps). 3 epochs for sentiment/paraphrase, 10 epochs for sonnets. Batch size 32/16/8 respectively.

LoRA: Learning rate 5×10^{-4} , rank 8, $\alpha = 16$, dropout 0.1. Applied to `c_attn` and `c_proj`. 5 epochs for sentiment, 3 for paraphrase, 10 for sonnets.

Prefix Tuning: Learning rate 5×10^{-4} , prefix length 20, weight decay 0.01. Same epoch counts as LoRA.

LoReFT: Learning rate 2×10^{-3} (sentiment) or 9×10^{-4} (generation tasks), no weight decay, warmup 10% of steps. Interventions at all 12 layers. 10 epochs for all tasks. We test ranks $r \in \{1, 2, 4, 8\}$ for sentiment and $r \in \{2, 4, 8\}$ for paraphrase/sonnet.

5.4 Results

5.4.1 Sentiment Analysis

Table 1: Sentiment analysis results on SST-2 development set. Best PEFT result **bolded**.

Method	Params	Train. %	Time (min)	Accuracy	F1
Full Fine-Tune	124.4M	100%	16.7	0.915	0.911
LoRA ($r=8$)	812K	0.65%	22.9	0.912	0.914
Prefix ($L=20$)	370K	0.30%	20.4	0.782	0.794
LoReFT ($r=1$)	20K	0.016%	44.7	0.908	0.908
LoReFT ($r=2$)	38K	0.031%	49.9	0.909	0.910
LoReFT ($r=4$)	75K	0.060%	56.5	0.912	0.913
LoReFT ($r=8$)	149K	0.120%	60.9	0.900	0.900

Table 1 presents our sentiment analysis results. Full fine-tuning achieves 91.5% accuracy. Remarkably, both LoRA ($r=8$) and LoReFT ($r=4$) match this at 91.2%, despite training vastly fewer parameters. LoReFT ($r=4$) uses only 75K parameters (0.06% of the model), a $1,656\times$ reduction over full fine-tuning. Prefix Tuning significantly underperforms at 78.2%, suggesting that simply prepending soft tokens is insufficient for this task. Interestingly, LoReFT ($r=8$) shows slight degradation compared to $r=4$, suggesting possible overfitting with higher-rank interventions.

5.4.2 Paraphrase Detection

Table 2 shows paraphrase detection results. Full fine-tuning achieves the highest BLEU (7.86), but LoReFT ($r=8$) reaches 7.04 BLEU—89.6% of full fine-tuning—with 835× fewer parameters. Notably, LoReFT achieves *higher* BERTScore F1 (0.572) than full fine-tuning (0.528), suggesting that its generations are more semantically aligned even when n-gram overlap is slightly lower. LoRA performs comparably on BLEU (6.98) but with lower BERTScore. Prefix Tuning again lags substantially.

5.4.3 Sonnet Generation

Table 3 presents sonnet generation results. Full fine-tuning achieves the lowest validation loss (4.16) and highest syllable accuracy (85.7%). However, LoReFT ($r=2$ and $r=4$) achieve the *highest* rhyme

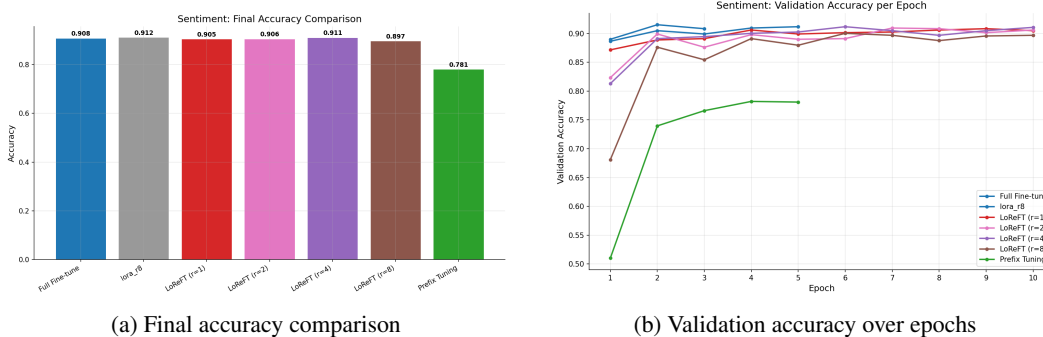


Figure 1: Sentiment analysis: (a) final development accuracy across methods, and (b) validation accuracy curves during training. LoReFT variants converge more slowly but ultimately reach competitive performance.

Table 2: Paraphrase detection results (generation-based, cloze-style). Best PEFT result **bolded**.

Method	Params	Train. %	Time (min)	BLEU	BERTScore F1
Full Fine-Tune	124.4M	100%	36.3	7.86	0.528
LoRA ($r=8$)	811K	0.65%	30.0	6.98	0.518
Prefix ($L=20$)	369K	0.30%	27.0	3.93	0.455
LoReFT ($r=2$)	37K	0.030%	99.3	6.32	0.566
LoReFT ($r=4$)	74K	0.059%	101.1	6.81	0.569
LoReFT ($r=8$)	148K	0.119%	97.1	7.04	0.572

accuracy (48.6%), surpassing full fine-tuning (45.7%). This is a remarkable finding: intervening on only 37K parameters can produce sonnets with better rhyming structure than updating all 124M parameters. LoReFT ($r=8$) balances multiple objectives well, achieving 72.4% syllable accuracy and 45.7% rhyme accuracy.

LoRA performs surprisingly poorly on structural sonnet metrics despite achieving a competitive validation loss (4.25), suggesting that low-rank weight adaptations may not effectively capture the structural constraints of poetry. Prefix Tuning, while consistently producing 14-line outputs, generates very short lines (3.1 syllables on average), indicating that it learns a degenerate generation mode.

5.4.4 Parameter Efficiency Summary

Figure 4 visualizes the efficiency-accuracy trade-off for sentiment analysis. LoReFT variants cluster in the upper-left corner (high accuracy, few parameters), representing the best Pareto frontier. LoRA achieves strong accuracy but at $\sim 10\times$ more parameters than LoReFT. Prefix Tuning occupies an unfavorable position with moderate parameter counts but significantly lower accuracy.

6 Analysis

6.1 Why LoReFT Excels with Minimal Parameters

The most striking finding is that LoReFT, with as few as 20K parameters (rank 1), achieves 90.8% accuracy on sentiment—within 0.7% of full fine-tuning. This success is attributable to the method’s inductive bias: by intervening on representations at the first and last token positions across all 12 layers, LoReFT effectively creates a 24-point intervention network that can steer the model’s behavior through small rotations in representation space. The pretrained GPT-2 representations already encode rich linguistic structure; LoReFT merely redirects this structure toward the target task.

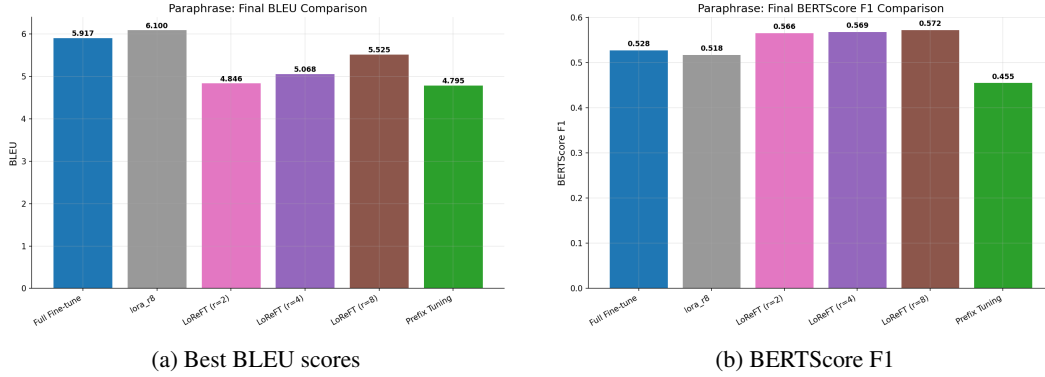


Figure 2: Paraphrase detection: (a) BLEU scores and (b) BERTScore F1 across methods. LoReFT variants achieve the highest semantic similarity (BERTScore) despite lower n-gram overlap (BLEU).

Table 3: Sonnet generation results. Best PEFT result **bolded**. Structural metrics are evaluated on 5 generated sonnets at the final epoch.

Method	Params	Time	Val Loss	Rhyme %	Syllable %	14-Line %
Full Fine-Tune	124.4M	3.7m	4.16	45.7	85.7	100
LoRA ($r=8$)	811K	3.4m	4.25	14.3	44.2	40
Prefix ($L=20$)	369K	3.6m	5.76	8.6	7.1	100
LoReFT ($r=2$)	37K	3.0m	4.35	48.6	42.9	100
LoReFT ($r=4$)	74K	3.0m	4.32	48.6	48.6	80
LoReFT ($r=8$)	148K	3.8m	4.29	45.7	72.4	80

6.2 Intervention Position Strategy for Generation

A critical implementation challenge arose when applying LoReFT to generation tasks. Initially, we intervened on *all* token positions during training, which worked well for loss minimization. However, during autoregressive generation, interventions could only be applied to prompt tokens (via `intervene_on_prompt=True`), creating a train-test mismatch. Generated tokens received no interventions at inference time, causing the model to revert to base GPT-2 behavior and produce degenerate outputs (repeated separator tokens, near-zero BLEU).

We resolved this by switching to a prefix-based intervention strategy (“f10”), intervening on only the first 10 token positions. This ensures that the same set of positions is intervened on during both training and generation, since the prompt always covers the first 10 positions. This design insight—that intervention positions must be consistent between training and inference—has not been previously documented in the LoReFT literature for generative tasks.

6.3 LoReFT vs. LoRA on Structured Generation

An unexpected result is LoReFT’s superiority over LoRA for rhyme accuracy in sonnet generation. We hypothesize that representation-level interventions are particularly effective for structural constraints because they operate at the point where the model “decides” what to generate, rather than modifying how it processes inputs. The intervention at early layers may help establish a structural template (rhyme scheme, line lengths) that persists through the remaining forward pass, while LoRA’s distributed weight modifications may conflict across layers when trying to impose global structure.

6.4 Prefix Tuning Limitations

Prefix Tuning consistently underperforms across all tasks. For sentiment, it achieves only 78.2% accuracy, 13 points below full fine-tuning. For sonnets, it generates extremely short lines. We attribute this to the fundamental limitation of soft prefixes: they can only influence the model through attention interactions, and with only 20 prefix tokens competing against entire input sequences, their

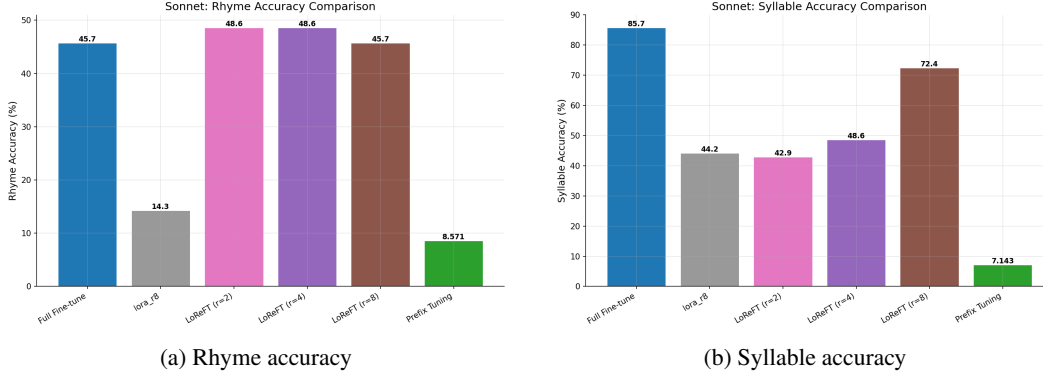


Figure 3: Sonnet generation: (a) rhyme accuracy and (b) syllable accuracy. LoReFT achieves the highest rhyme accuracy, while full fine-tuning leads on syllable accuracy.

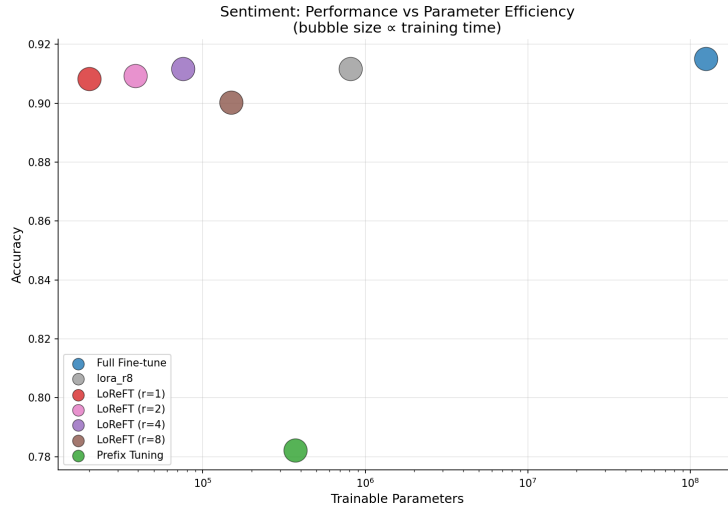


Figure 4: Accuracy vs. trainable parameters (log scale) for sentiment analysis. LoReFT ($r=4$) achieves near-identical accuracy to full fine-tuning with $\sim 1,652\times$ fewer parameters.

signal is diluted. Additionally, the reparameterization MLP adds optimization complexity without commensurate representational benefit.

6.5 Qualitative Sonnet Analysis

Below is an excerpt from a sonnet generated by our full fine-tuning model, conditioned on the first three lines of Shakespeare’s Sonnet 147:

*My love is as a fever, longing still
For that which longer nurseth the disease,
Feeding on that which doth preserve the ill,
And would be locked, or at least spare, in his pores;
Hath yet yet if his breath should be quench,
Safe by nature, renewing that doth to idle his fury,
Our love at world domination in opulence lingered...*

While the model captures Shakespearean vocabulary and cadence, it struggles with coherent semantics beyond the first few generated lines. Rhyming is partially successful (“pores”/indeterminate, “quench”/“fury”), illustrating the challenge of maintaining both form and meaning in constrained generation.

7 Conclusion

We implemented GPT-2 from scratch and systematically compared four fine-tuning strategies across sentiment classification, paraphrase detection, and sonnet generation. Our key findings are:

1. **LoReFT is remarkably parameter-efficient.** With only 75K trainable parameters (0.06% of the model), LoReFT ($r=4$) matches full fine-tuning accuracy on sentiment analysis. On paraphrase detection, LoReFT ($r=8$) achieves the highest BERTScore F1 among all methods.
2. **Representation interventions suit structured generation.** LoReFT achieves the highest rhyme accuracy on sonnet generation, suggesting that intervening on representations is particularly effective for imposing structural constraints.
3. **Intervention position consistency is critical.** For generative tasks, the positions where LoReFT intervenes must match between training and inference. Our prefix-based strategy (“f10”) resolves the train-test mismatch inherent in all-position intervention.
4. **Prefix Tuning underperforms.** Despite its conceptual simplicity, Prefix Tuning consistently lags behind LoRA and LoReFT, suggesting that attention-only signal injection is insufficient for these tasks.

Our original proposal targeted $\geq 94\%$ sentiment accuracy; our best result of 91.5% reflects the inherent limitation of GPT-2 as a decoder-only model for classification, where a BERT-style encoder would likely close this gap. Future work could explore combining LoReFT with constrained decoding algorithms for poetry, investigating layer-selective intervention strategies, and extending the comparison to larger models where the parameter efficiency advantages of LoReFT become even more pronounced.

Team Contributions

Zijian Luo (100%): Implemented GPT-2 from scratch (attention, layers, embeddings, optimizer). Implemented all three downstream tasks (sentiment, paraphrase, sonnet). Designed and implemented LoRA, LoReFT, and Prefix Tuning modules for the custom GPT-2 architecture. Built the Modal cloud training pipeline. Diagnosed and fixed the LoReFT train-test mismatch for generation. Ran all experiments and wrote the report.

References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.
- [2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [3] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.
- [4] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Representation fine-tuning for language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [6] Neil Houlsby, Andrei Giber, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019.

- [7] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.
- [8] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstraction for faithful model interpretation. *arXiv preprint arXiv:2301.04709*, 2024.
- [9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.
- [10] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.
- [11] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [12] Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. First quora dataset release: Question pairs. 2017. <https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>.
- [13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.
- [14] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations (ICLR)*, 2020.

A Additional Training Curves

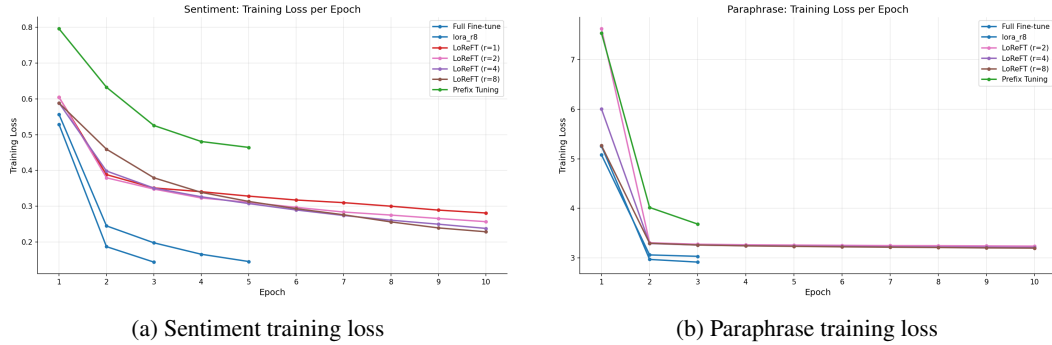


Figure 5: Training loss curves for sentiment and paraphrase tasks. Full fine-tuning converges fastest due to its larger gradient signal, while LoReFT converges slowly but steadily.

B Parameter Count Breakdown

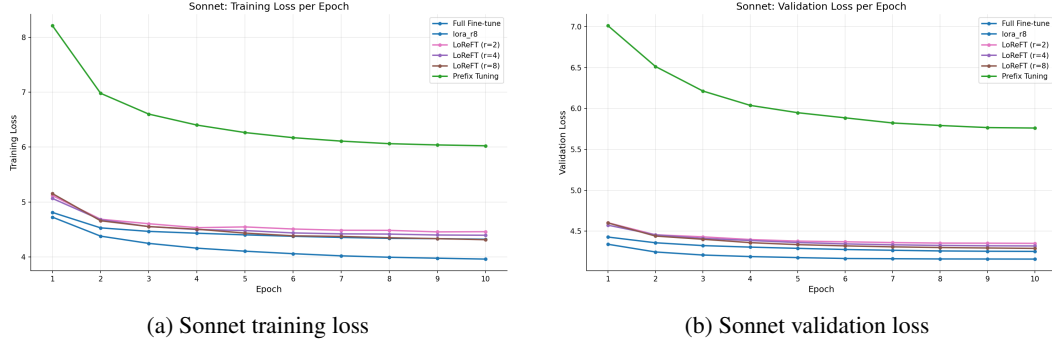


Figure 6: Sonnet training and validation loss curves. All methods converge within 10 epochs, with full fine-tuning achieving the lowest final validation loss.

Table 4: Trainable parameter counts and reduction factors relative to full fine-tuning.

Method	Trainable Params	% of Total	Reduction Factor
Full Fine-Tune	124,441,344	100%	1×
LoRA ($r=8$)	812,544	0.65%	153×
Prefix ($L=20$)	370,176	0.30%	336×
LoReFT ($r=8$)	149,088	0.12%	835×
LoReFT ($r=4$)	75,312	0.06%	1,652×
LoReFT ($r=2$)	38,424	0.03%	3,238×
LoReFT ($r=1$)	19,980	0.016%	6,228×