

From Detection to Generation: Fine-tuning Large GPT2 Models for Paraphrasing and Poetry

Stanford CS224N Default Project

Anna Guo

Department of Computer Science
Stanford University
annaguo@stanford.edu

Abstract

As Large Language Models (LLMs) continue to grow in size, full-parameter fine-tuning becomes infeasible due to memory and compute constraints. Hence, Parameter-Efficient Fine-Tuning (PEFT) methods have gained popularity, building off of the idea of "intrinsic dimensionality," where the actual rank of a task-specific adaptation is significantly lower than the full parameter space of a given model. This report investigates the comparative performance of two such techniques, Low-Rank Adaptation (LoRA) and Low-Rank Representation Fine-Tuning (LoReFT), using a GPT-2 backbone across Paraphrase Detection and Sonnet Generation. The findings indicate that while both methods achieve performance close to those of full-parameter baselines, LoRA benefits significantly from expansion beyond attention layers into MLP blocks for stylistic tasks, whereas LoReFT boasts a greater reduction in parameters. Simultaneously, qualitative analysis reveals distinct tradeoffs, with LoRA being prone to thematic repetitio and LoReFT interventions prone to syntactic instability. Finally, bootstrap aggregating (bagging) the two via soft-voting provides a marginal gain in generation stability but risks output incoherence when underlying model representations diverge.

1 Key Information to include

- TA mentor: Nevin George
- External collaborators, External mentor, Sharing project: No
- Total late days remaining: 2; total late days applied for the report: 2

2 Introduction

The rising popularity of LLMs and large-scale pre-trained models has caused a shift from model training from scratch to fine-tuning pre-trained models for specific tasks. However, the large numbers of weights in architectures such as GPT-2 make full model fine-tuning inaccessible due to compute and memory constraints. Although linear probing is offers a low-cost alternative by only training the output head, it underperforms fully fine-tuned models on more nuanced test, such as detailed semantic understanding and creative text generation.

Thus, strategies such as Parameter-Efficient Fine-Tuning (PEFT) arose to address this problem. For example, Low-Rank Adaptation greatly reduces the number of trainable parameters while preserving performance and reducing latency [1]. Later works such as low-rank Representation Fine-Tuning (LoReFT) continue to reduce parameters while preserving performance, this time from the perspective of internal representations, rather than weights [2].

This project explores the trade-offs between weight-space adaptation (LoRA) and representation-space intervention (LoReFT). In particular, it investigate whether implementations of LoRA and

LoReFT may attain the same performance as that of fine-tuning all the parameters of the GPT-2 (small) model. This is evaluated across two complex Natural Language Processing tasks: Paraphrase Detection and Sonnet Generation. Finally, given that these approach the problem of parameter reduction from different perspectives, a soft-voting ensemble strategy is employed to test whether the outputs can be regularized through bootstrap aggregation.

3 Related Work

Large-scale pre-trained models, such as GPT-2, have demonstrated dramatic learning capabilities. Specifically, Radford et al. [3] demonstrates GPT-2’s success on zero-shot tasks and multitask capabilities without fine-tuning. Still, a significant performance gap remains between zero-shot inference and fully supervised models, thus requiring task-specific fine-tuning to achieve state-of-the-art accuracy [4]. However, with the increasing size of pre-trained models, full model fine-tuning becomes prohibitive due to compute and memory costs.

Kumar et al. [5] discusses the popular strategy of linear probing, a strategy of updating the "head" (the last linear layer) which requires tuning far fewer parameters, thus lowering the compute and memory requirements. Yet it too acknowledges the performance boost of strategies that tuned the entire model. Here, Parameter-Efficient Fine-Tuning (PEFT) presents the possibility of having few tunable parameters throughout the model, rather than concentrated at the head. For instance, "Adapters" proposed by Houlisby et al. [6], introduced task-specific layers between existing Transformer blocks. Despite reducing trainable parameters, "Adapters" still often increased inference latency because they added sequential layers.

In response, Hu et al. [1] proposed Low-Rank Adaptation (LoRA), which both reduced trainable parameters and addressed the latency issue by injecting trainable low-rank matrices into the self-attention layers. This is in line with Aghajanyan et al.’s [7] notion of "intrinsic dimensionality", which asserts that the actual rank of task-specific adaptation is significantly lower than the full parameter space of the model. By freezing the pre-trained model weights and only updating the low-rank decomposition, LoRA maintains the model’s original capabilities while allowing for specialized performance on tasks like paraphrase detection and sentiment analysis.

Dettmers et al. [8] build upon Hu et al.’s work by demonstrating that applying LoRA to all linear layers, including intermediate and output dense blocks, improves performance on complex generative tasks. This contradicts Hu et al.’s [1] original assertion that it was sufficient to only adapt the query and value projects of the attention layers. Moreover, Zhang et al. [9] highlight that assigning a fixed rank to all layers may lead to sub-optimal parameter allocation. This sensitivity is reflected in the extensive hyperparameter tuning required for rank r and scaling factor α . Furthermore, some researchers argue that modifying weights is but an indirect method to shift the model’s internal representations. Thus, He et al. [10] provide a unified view of PEFT, suggesting that many of these methods are functionally equivalent to modifying the hidden states of the Transformer.

This perspective leads to the development of Representation Fine-Tuning (ReFT) and its low-rank variant, LoReFT [2]. Unlike LoRA, which modifies how a model "thinks" by altering its weights, ReFT intervenes directly on the "representation space" during the forward pass. By using a low-rank projection to intervene on specific hidden states, LoReFT even more aggressively reduces the number of trainable parameters than LoRA.

This project attempts to directly compare LoRA and LoReFT. It explores whether the precision of LoReFT remains superior to the weight-smearing approach of LoRA when applied to creative tasks such as sonnet generation, as compared to standard classification (sentiment analysis and paraphrase detection). In doing so, it also considers the principles of "Green AI" [11], evaluating whether the marginal gains in accuracy from larger PEFT configurations justify their increased wall-clock training time and corresponding compute resources.

4 Approach

The approach centers on extending the GPT-2 architecture through two primary Parameter-Efficient Fine-Tuning (PEFT) methodologies: Low-Rank Adaptation (LoRA) and Low-Rank Representation

Fine-Tuning (LoReFT). Top-performing models are bagged in an ensemble strategy to maximize generalization, with the secondary goal of minimizing parameters while maintaining performance.

4.1 Baseline

The baseline is a minimalist PyTorch implementation of the GPT-2 architecture, built upon provided skeleton code and the Transformer decoder blocks as described in Vaswani et al. [12]. The improvement upon the skeleton code features masked multi-head self attention, a 12-layer Transformer decoder (with hidden dimensions of $d_{model} = 768$ for the "small" configuration), and an AdamW Optimizer. Then, the model is loaded with official pre-trained weights from the gpt2 (small) model via the HuggingFace Transformers library. (Different dimensions support the medium and large models later on.)

Next, a task-specific linear head is added on top of the final hidden state of the last token for Sentiment Analysis (on SST-5 and CFIMDB), Paraphrase Detection (Quora Dataset), and Sonnet Generation.

- **Sentiment Analysis ($\mathbf{W}_{sent}$):** For the SST-5 and CFIMDB datasets, a cloze-style classification head is implemented. The contextualized embedding of the final token, $\mathbf{h}_{last} \in \mathbb{R}^d$, is extracted to compute sentiment logits:

$$\mathbf{z}_{sent} = \mathbf{W}_{sent} \cdot \text{dropout}(\mathbf{h}_{last}) + \mathbf{b}_{sent} \quad (1)$$

where $\mathbf{W}_{sent} \in \mathbb{R}^{d \times 5}$. This produces a distribution over the five sentiment labels. Two fine-tuning methods are evaluated: `last-linear-layer`, where the GPT-2 backbone is frozen and only $\mathbf{W}_{sent}$ is trained, and `full-model`, where all parameters are updated.

- **Paraphrase Detection ($\mathbf{W}_{para}$):** The input is a natural language query, while the output of the classification head is binary.

$$\mathbf{z}_{para} = \mathbf{W}_{para} \cdot \mathbf{h}_{last} + \mathbf{b}_{para} \quad (2)$$

Here, $\mathbf{W}_{para} \in \mathbb{R}^{d \times 2}$. The final hidden state consists of two logits, which are then mapped to "8505" (paraphrase) or "3919" (not a paraphrase) during evaluation. This simplicity both saves on compute and places the brunt of the semantic comparison on the transformer's attention layers rather than the linear output layer.

- **Sonnet Generation ($\mathbf{W}_{gen}$):** For autoregressive generation, the model produces a distribution over the entire vocabulary for every sequence position. The full sequence of hidden states $\mathbf{H} \in \mathbb{R}^{n \times d}$ is mapped through a generation head:

$$\mathbf{Z}_{gen} = \mathbf{H} \cdot \mathbf{W}_{gen}^T \quad (3)$$

where $\mathbf{W}_{gen} \in \mathbb{R}^{|V| \times d}$. Here, the output is generated one token at a time, and the representations are decoded back into English text.

4.2 LoRA: Weight-Space Adaptation

While the baseline GPT-2 architecture demonstrates admirable performance, full-parameter fine-tuning is comparatively computationally inefficient under standard hardware constraints, especially for larger models such as gpt2-medium and gpt2-large. To circumvent this, Low-Rank Adaptation (LoRA) is implemented to achieve weight-space modularity [1]. For any pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d \times k}$, the adaptation is constrained to a low-rank decomposition of the weight update:

$$\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \frac{\alpha}{r} \mathbf{B}\mathbf{A} \quad (4)$$

where $\mathbf{B} \in \mathbb{R}^{d \times r}$ and $\mathbf{A} \in \mathbb{R}^{r \times k}$ are trainable low-rank matrices with rank $r \ll \min(d, k)$, and α serves as a scaling factor to modulate the intervention's magnitude.

Sonnet-specific Implementation Standard PEFT implementations typically isolate updates within the attention mechanism ($\mathbf{W}_q, \mathbf{W}_v$). however, this study extends the LoRA bottleneck to the intermediate and output dense layers of the GPT-2 MLP blocks ($\mathbf{W}_{fc} \in \mathbb{R}^{d \times 4d}$ and $\mathbf{W}_{proj} \in \mathbb{R}^{4d \times d}$). This choice was made on the assumption that localized feed-forward "knowledge" neurons are the primary repositories for stylistic and phonetic constraints [8]. Hence, the model is forced to adapt its internal semantic knowledge base to the specific nuances of Shakespearean verse without degrading the foundational attention patterns established during pre-training.

4.3 LoReFT: Representation-Space Intervention

To explore aggressive parameter reduction, Low-Rank Representation Fine-Tuning (LoReFT) is implemented [2]. This method does not modify W_0 , but applies a learned linear subspace intervention to hidden representations h during the forward pass:

$$\Phi(h, L, \mathcal{I}) = h + R^T(Wh + b - Rh) \quad (5)$$

where $R \in \mathbb{R}^{r \times d}$ is an orthogonal projection matrix, and W, b are learned parameters applied at layer L and position $\mathcal{I}$.

Justification for Use: LoReFT typically reduces trainable parameters by $15\text{--}65\times$ compared to LoRA. This implementation tests whether these interventions, especially on larger models, can be successfully ported to the compressed representation space of GPT-2. Basically, it aims to maintain or even improve performance on evaluation metrics while using significantly fewer parameters.

4.4 Bagging and Soft-Voting Ensemble

Both past personal experience and established literature demonstrate that ensembling improves generalization beyond that of individual models. Here, bagging (Bootstrap Aggregating) was used to mitigate the high variance and risk of local minima inherent in PEFT, especially on small datasets (e.g., the 154-sonnet corpus). For each task, the $N = 3$ highest performing models across varied random seeds and hyperparameter configurations were used. The final prediction for a given input x is determined via soft-voting:

$$P(y|x) = \frac{1}{N} \sum_{i=1}^N \text{softmax}(\mathbf{z}_i) \quad (6)$$

where $\mathbf{z}_i$ represents the logit vector produced by the i -th model. This prevents overfitting on noise while being more compute-efficient than other ensemble strategies such as boosting.

4.5 Implementation Details

The foundational Transformer decoder architecture was provided via the project skeleton. However, I took responsibility for implementing the baseline models, integrating LoRA and LoReFT, and the development of the soft-voting ensemble logic.

5 Experiments

5.1 Data

Three primary datasets were provided by the CS224N project skeleton:

- **SST-5 / CFIMDB:** These test Sentiment Analysis, with the Stanford Sentiment Treebank (SST-5) involving fine-grained, five-class sentiment prediction, while the CFIMDB dataset involving binary classification.
- **Quora Question Pairs (QQP):** A corpus of roughly 400,000 question pairs. An instruction-based prompt: *'Is "{s1}" a paraphrase of "{s2}"? Answer "yes" or "no": '* makes this a binary classification task.
- **Shakespearean Sonnets:** A collection of 154 sonnets used for autoregressive generation. The task requires the model to emulate the Shakespearian vocabulary and style.

5.2 Evaluation Method

Accuracy is used to evaluate the discriminative or classification tasks (SST and Paraphrase). Meanwhile, the **chrF** (character n-gram F-score) metric is used for Sonnet Generation. Compared to BLEU, which operates on word-level tokens, chrF assess morphological nuances and archaic spellings with more granularity at the character level. Hence, it's better at capturing the dense, stylistic character distributions in Shakespearean English, especially given the constraints of a smaller dataset.

5.3 Experimental Details

Training was conducted on AWS **g6.xlarge** instances.

The baseline utilizes GPT-2 Small ($L = 12, d = 768$). The baseline sentiment analysis ran for 20 epochs with a learning rate of 0.001 for fine-tuning the last linear layer and 1E-5 for the full model. Similarly, paraphrase detection ran for 10 epochs with a 1E-5 learning rate. Finally, sentiment Analysis ran for 20 epochs with a 5E-4 learning rate.

Meanwhile, experiments with LoRA and LoREFT involved a more in-depth hyperparameter sweep, as shown in the tables in the next subsection.

5.4 Quantitative Results

Table 1: **Baseline Performance Across Tasks**

Model	SST-5 (Acc)		CFIMDB (Acc)		Paraphrase	Sonnet
	Last	Full	Last	Full	(Acc)	(chrF)
Baseline	0.470	0.508	0.873	0.980	0.897	40.774

Table 2: **Paraphrase (LoRA)** ($\eta = 5 \times 10^{-5}$, Epochs = 10).

Scale	Rank (r)	Alpha (α)	Time (Wall)	Dev Acc
Small	16	32	53m15s	0.846
Medium	16	32	122m26s	0.855
Medium	64	128	122m25s	0.867
Large	16	32	238m6s	0.868

Table 3: **Paraphrase (LoReFT)** ($\eta = 5 \times 10^{-5}$, Epochs = 10).

Scale	Rank (r)	Time (Wall)	Dev Acc
Small	4	34m45s	0.806
Small	8	43m23s	0.809
Small	16	44m14s	0.807
Large	4	208m24s	0.861
Large	8	209m29s	0.867
Large	16	219m17s	0.632

Table 4: **Sonnet (LoRA)**

Variant	Rank/Alpha	LR (η)	Epochs	Time (Wall)	chrF
Old (Attn-only)	8 / 16	5e-4	10	10m37s	39.200
Old (Attn-only)	16 / 32	5e-4	10	14m41s	39.497
Old (Attn-only)	16 / 32	1e-5	20	19m12s	28.928
Old (Attn-only)	32 / 64	5e-4	20	13m56s	37.960
New (MLP+Attn)	16 / 64	1e-5	20	25m14s	40.892
New (MLP+Attn)	16 / 64	5e-4	20	26m1s	41.489
New (MLP+Attn)	16 / 32	5e-4	20	24m10s	41.809
New (MLP+Attn)	32 / 64	5e-4	20	28m53s	41.753

Table 5: **Sonnet (LoReFT):** ($\eta = 5 \times 10^{-4}$, Epochs = 20).

Scale	Rank (r)	Time (Wall)	chrF
Small	4	10m44s	37.832
Small	8	11m0s	40.500
Medium	4	17m25s	38.037
Medium	8	18m7s	39.803
Large	4	27m33s	40.260
Large	8	31m10s	41.438
Large	16	31m28s	40.452

An ensemble of the 3 Sonnet models yielded a dev score of 40.891 and a test score of 41.635.

Table 6: **Best Dev and Test Performance:**

	Dev	Test
Paraphrase Detection	0.897	0.843
Sonnet Generation	41.809	41.635

6 Analysis

The results demonstrate the comparatively minor decrease in performance on evaluation metrics despite the significant decrease in parameters, thanks to LoRA and LoReFT. Unfortunately, none of the implementations were able to significantly outperform the baselines purely on performance metrics.

6.1 Baseline Overview

As shown in Table 1, there is a significant gap in gpt2-small’s performance on last-linear-layer tuning and full-model fine-tuning. This is most noticeable in the CFIMDB sentiment task (0.873 vs. 0.980), meaning that simple linear mapping isn’t sufficient for the task, despite the robustness of the pre-trained features. Meanwhile, the baseline Paraphrase Detection (0.897) and Sonnet Generation (40.774) serve as strong competitive marks that the PEFT methods aim to replicate or exceed while using only a fraction of the trainable parameters.

6.2 Performance Issues in Paraphrase Detection

Neither LoRA (0.868) nor LoReFT (0.867) was able to even approach the Baseline Paraphrase accuracy (0.897). This persistent difference of around 3% suggests that the structural constraints of low-rank adaptation may be too restrictive for comparing the semantic meanings between two sets of phrases.

Specifically, using a causal decoder for Paraphrase Detection requires the model to hold the context of the first sentence while evaluating the second within the same forward pass. Full fine-tuning likely allows for global shifts in the attention heads that facilitate this comparison, whereas PEFT methods provide a more "localized" adjustments that may struggle to optimize for inter-token dependencies across longer prompt sequences.

Alternatively, Paraphrase Detection may have benefited from the addition of LoRA to dense layers as applied in Sonnet Generation. However, the long runtimes for the Paraphrase Detection task made it difficult to experiment with. It is nonetheless worth noting that Paraphrase Detection achieved within around 3% of the baseline with only around 6,000,000 ($36 \times 4 \times 2 \times 16 \times 1280$) parameters in LoRA and 700,000 ($36 \times 2 \times 8 \times 1280$) parameters in LoReFT for the GPT-2 Large models, compared to the 124M parameters required for the GPT-2 (small) baseline.

6.3 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) performs differently based on the layers that LoRA is applied to. The move from an Attention-only configuration (query and value) to a joint MLP and Attention injection (includes intermediate dense and output dense layers) resulted in a measurable increase in chrF scores, peaking at 41.809. This suggests that stylistic adaptation is better captured in feed-forward layers, supporting Dettmers et al.'s [8] claim that applying LoRA to all linear layers is necessary to match full fine-tuning performance.

Hence, by also fine-tuning the feed-forward networks, the model is better able to store and retrieve specific Shakespearean vocabulary and thematic markers. For instance, in this excerpt from a generated Sonnet 0, the model correctly adopts archaic pronouns "thy," "thou," and "thee," along with other period-specific vocabulary such as "doth," "shouldst," and "shalt." However, it over-emphasizes the concept of "love", perhaps demonstrating LoRA's tendency to apply a stylistic smear. This is even more dramatic in an excerpt from a generated Sonnet 1 (which does not contain mentions of love within the initial three line input).

Excerpt from Dev Sonnet 0

For those eyes which thy **love** in thy heart doth stand,
Then shouldst thou say my **love** is far holds thee.
But if thou were as thy name, mine eye of thee doth grow,
Since I'll but say thy **love** is thy fair part,
And thou shalt by thy **love** shouldst see the rich dead.

Excerpt from Dev Sonnet 1

And to my **love** to **love** is by thee I then.
For not my **love** to me do I lay
And mine is still my **love**, and my **love**.

6.4 LoREFT Intervention

Rather than changing the model layers, LoREFT (Low-rank Representation Editing Fine-Tuning) modifies the data passed between layers. The qualitative evaluation reveals that LoREFT is remarkably efficient, achieving a competitive chrF of 41.438 on the Large backbone with a fraction of the trainable parameters used by LoRA. However, the system does not continuously increase with rank. In fact, for the Large model, increasing the rank from 8 to 16 resulted in a performance decrease to 40.452.

It's possible that high-rank interventions in LoREFT may introduce excessive noise into the hidden states, leading to syntactic instability. To illustrate, this excerpt from a sonnet generated through LoREFT covers more diverse subjects than LoRA, it seems to devolve into a smattering of poetic words that has more obvious grammatical clashes than the LoRA implementation.

Excerpt from Dev Sonnet 7

In one of the present happy spring's waste
Hath the painted banquet bids her be tomb.
But since the spring's breath was not so strong
Whereon my books did my worth so much hold.

6.5 Bootstrap Aggregation

The ensemble approach, utilizing bootstrap aggregating by using soft voting across three models to generate one token at a time, yielded the highest overall test score of 41.635. This improvement is likely due to noise reduction. In averaging the predictions of multiple models, the system smooths out individual errors/noise to stabilize the generated output.

Yet the regularization isn't necessarily beneficial. Although the italicized lines below are an impressive demonstration of iambic pentameter and rhyme, further examination reveals that they were lifted directly from an actual Shakespeare sonnet. Furthermore, the end of the sonnet devolves into more dramatic gibberish than either the LoRA and LoREFT models did, likely because of the competing

representations. It may be the case that bagging boosts model capabilities when the underlying models are in semi-alignment, but greatly deteriorates performance otherwise.

Dev Sonnet 11

Lo, as a careful huswife runs to catch
One of her feathered creatures broke away,
Sets down her babe, and makes all swift dispatch.
Yet seemed it not so with me,
That, when my lover's shape hath cover,
The worst of wrongs should not seem so great.
*The bloody spur cannot provoke him on
That sometimes anger thrusts into his hide,
Which heavily he answers with a groan,
More sharp to me than spurring to his side;*
And, for my sake being remembered,
I hold not my- looks Save for my love's and love's right.
For I must ne'er love him whom he best endowed
To leave for myself such a make- defense
As he

7 Conclusion

Weight-space adaptation (LoRA) and representation-space intervention (LoReFT) are valuable alternatives to full-parameter fine-tuning for GPT-2 models. Both methods provide an incredible reduction in trainable parameters while maintaining a performance close to the baseline. Specifically, the Paraphrase Detection LoReFT models with GPT2-large that trained less than 1% of the total parameters within the full GPT2 (small) model were able to achieve within 3% of the baseline. Meanwhile, LoRA and LoReFT approaches showed a marginal increase over the baseline on the Sonnet Generation task.

Simultaneously, qualitative investigation of errors revealed tradeoffs between LoRA and LoReFT, as well as the viability of bootstrap aggregation. For instance, LoRA is generally stable, but it is susceptible to overemphasizing certain themes, such as with the excessive recurrence of the word "love." Meanwhile, LoReFT is much more parameter-efficient, but it also displayed greater sensitivity to hyperparameter tuning. At higher ranks, it degraded syntactic coherence, resulting in poetic gibberish. Finally, while bagging provided the highest chrF scores by a tiny margin, the regularization resulted in both directly regurgitating training samples and diverging into unintelligible phrases.

This project was limited by the computational resources inherent to the many parameters and the complexity of the task, hence the focus on more parameter efficient fine-tuning methods such as LoRA and LoReFT. Future work could involve a deeper dive into either methods (layer-wise rank variation) or explore more integrated methods of hybridizing LoRA and LoReFT beyond bootstrap aggregation.

References

- [1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.
- [2] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024.
- [3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [4] Dani Yogatama, Cyprien de Masson d’Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019.

- [5] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.
- [6] Neil Houlsby, Andrei Giurgiu, 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*, pages 2790–2799. PMLR, 2019.
- [7] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328, 2021.
- [8] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.
- [9] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.
- [10] Junxian He, Chunting Zhou, Xuezhe Ma, and Taylor Berg-Kirkpatrick. Graham neubigtowards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations (ICLR)*, 2022.
- [11] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.