

Where Does LoRA Actually Help? Probing Layer-Wise Adaptation in GPT-2 for Paraphrase Detection and Sonnet Generation

Stanford CS224N Default Project

Mona Anvari

Department of Computer Science
Stanford University
monaavr@stanford.edu

Abstract

Parameter-efficient fine-tuning has become essential as language models scale, yet a subtle but consequential question remains open: which weight matrices should be adapted, and does the answer depend on task structure? Low-Rank Adaptation (LoRA) defaults to adapting query and value attention projections based on experiments with large models on NLU tasks. This is a choice whose transferability to smaller regimes and structurally distinct tasks is untested. Mechanistic interpretability work suggests that attention layers should matter more for discriminative tasks relying on cross-sentence comparison, while MLP layers which are known to encode lexical and stylistic knowledge should dominate for generative tasks requiring distributional fluency. We test this directly via a controlled ablation on GPT-2 (small, 117M parameters) across two tasks: paraphrase detection (discriminative, 141K training pairs) and Shakespearean sonnet generation (generative, 143 examples). By matching trainable parameter budgets across four target configurations QV, QKVO, MLP, and ALL. At $\approx 600K$ parameters, we isolate the effect of where parameters are added from how many. We find that surprisingly MLP-only adaptation outperforms attention-only on paraphrase, while QV adaptation outperforms MLP on sonnet generation. Error analysis attributes the MLP advantage to more balanced class-recall, and QV's sonnet lead partly to high training variance under best-epoch selection. Combined adaptation (LoRA-ALL-r16) is reliably best on both tasks, recovering over 93% of full fine-tuning accuracy on paraphrase with only 2.3% of the parameters. While constrained by scale and dataset size discrepancies, these findings suggest that optimal LoRA target selection is task-dependent in ways current mechanistic accounts do not predict, motivating broader ablations across model scales and task families.

1 Key Information to include

- Mentor: Alisa Levin

2 Introduction

Fine-tuning pre-trained language models has become the dominant paradigm in NLP, but as models scale, updating every parameter grows prohibitively expensive. Low-Rank Adaptation (LoRA) [1] has emerged as a compelling parameter-efficient solution, freezing pretrained weights and injecting trainable low-rank residuals into selected weight matrices. While LoRA achieves near-full fine-tuning performance at a fraction of the parameter cost, a fundamental question remains unanswered: which weight matrices actually need to be adapted, and does the optimal target depend on the structural nature of the downstream task?

This question is practically important because LoRA’s default configuration of adapting only query and value (QV) attention projections was established empirically on massive models and Natural Language Understanding (NLU) benchmarks [1]. It remains unclear how this transfers to smaller model regimes or tasks with fundamentally different decoding objectives. To investigate this, we establish full fine-tuning baselines to contrast two structurally distinct tasks. Paraphrase detection is a discriminative task requiring cross-sentence comparison, an operation heavily reliant on self-attention routing. Conversely, sonnet generation is an autoregressive task requiring distributional fluency and stylistic consistency over long horizons, where feed-forward (MLP) layers, known to encode lexical and stylistic knowledge [2], might be the critical bottleneck. We hypothesize that the right adaptation target fundamentally differs between these task families.

Answering this empirically is challenging because varying the adaptation target typically varies the parameter count, conflating where parameters are added with how many. We address this through a controlled ablation on GPT-2 (small) as a base. We deliberately select four target configurations to test our hypothesis: **QV** (the standard LoRA default), **QKVO** (comprehensive attention adaptation to see if broader routing helps), **MLP** (strictly feed-forward adaptation to isolate stylistic/factual learning), and **ALL** (combined adaptation as a PEFT upper-bound).

At the 600K parameter budget, we find that MLP-only adaptation outperforms attention-only adaptation on paraphrase detection, while QV adaptation outperforms MLP on sonnet generation. Further analysis is performed to reveal more. Ultimately, combined adaptation (LoRA-ALL) proves the most reliable, recovering over 93% of full fine-tuning accuracy on paraphrase with only 2.3% of the parameters.

Summary of contributions. We provide: (1) A controlled matrix-wise LoRA ablation on GPT-2 (small) isolating the effect of adaptation target from parameter count (§4, §5); (2) Empirical evidence that the optimal LoRA target configuration reverses between discriminative and generative tasks at matched budgets (§5.4); and (3) Error analysis attributing the MLP advantage to class-recall calibration, and the QV advantage to training-variance under best-epoch selection (§6).

3 Related Work

Our work sits at the intersection of parameter-efficient fine-tuning, mechanistic interpretability of transformer components, and task-structure analysis for pre-trained language models.

Pre-trained language models and fine-tuning. The dominant adaptation paradigm traces to BERT [3], which introduced bidirectional masked language modeling and established strong baselines on discriminative sentence-pair benchmarks such as QQP [4], and to GPT-2 [5], a decoder-only model trained with a causal language modelling objective that transfers naturally to autoregressive generation tasks. Full fine-tuning of these models, updating all parameters jointly, remains a strong baseline but is computationally expensive at scale and prone to catastrophic forgetting [6] on small corpora, a concern directly relevant to our 143-sonnet corpus.

Parameter-efficient fine-tuning. To reduce adaptation cost, a family of parameter-efficient fine-tuning (PEFT) methods has emerged. Adapter layers [7] insert small trainable bottleneck modules between frozen transformer sub layers. Prefix tuning [8] prepends trainable continuous vectors to each layer’s input while leaving model weights entirely frozen. These methods achieve near-full fine-tuning performance at a fraction of the parameter count, but adapters add inference latency [1] and prefix methods reduce the usable sequence length. Low-Rank Adaptation (LoRA) [1] avoids both limitations by injecting trainable low rank residuals directly into selected weight matrices, which can be merged with the frozen weights at inference time. Aghajanyan et al. [9] provide theoretical grounding, showing that fine-tuning loss landscapes have low intrinsic dimensionality, which supports the use of low-rank updates.

What LoRA does and does not study. The original LoRA paper evaluates on RoBERTa and DeBERTa for NLU and on GPT-2 medium and large for data-to-text generation [1]. Crucially, it includes one ablation of which weight matrices to adapt (Section 7.1 of [1]), varying across query, key, value, and output projections on GPT-3 for an NLU task. However, this ablation (i) is conducted only on a single large-scale model, (ii) covers only attention projections and not MLP feed-forward layers as an adaptation target, and (iii) does not compare adaptation targets across structurally different

task types. Subsequent work has reinforced the importance of target coverage: the default QV-only configuration has been shown to fail to match full fine tuning performance, with broader layer coverage required to close the gap [10], and LoRA in standard low-rank settings more generally has been found to under-perform full fine-tuning because fine-tuning learns weight perturbations of rank $10\text{--}100\times$ greater than typical LoRA configurations [11]. These findings, however, remain restricted to large models and a single task type at a time. Whether the optimal target configuration differs between a discriminative task requiring cross-sentence comparison and a generative task requiring distributional fluency, and at the scale of GPT-2 (small), remains an open question that our work directly addresses.

Mechanistic roles of attention and MLP layers. Further interpretability work motivates our hypothesis about which LoRA targets should matter. Geva et al. [12] showed that feed forward sub-layers function as key-value memories, associating input patterns with output vocabulary distributions. Meng et al. [2] extended this, localizing factual associations in GPT-style models primarily to MLP layers. Together, these accounts suggest that adapting MLP weights should be most consequential when a task requires accessing or modifying stored knowledge. This property plausibly is more relevant to stylistic generation than to cross-sentence comparison. Whether this mechanistic intuition actually predicts empirical LoRA behavior, however, has not been tested in a controlled ablation across structurally different tasks, nor at the scale of GPT-2 (small).

Our work tests this directly. We find directionally consistent but counterintuitive evidence: at a matched parameter budget of $\approx 600\text{K}$, the relative ordering of single-module strategies reverses across the two tasks, falsifying both directions of the mechanistic prediction. We report these findings as suggestive rather than conclusive, given the absence of formal significance testing and the small size of the sonnet evaluation set. Rather than settling the question, this reversal sharpens it into a testable hypothesis: that optimal LoRA target selection is task-dependent in ways the standard mechanistic account does not predict. Our error analysis motivates this hypothesis and larger scale replication could confirm.

4 Approach

Despite LoRA’s widespread adoption, its default QV-only target was established on large-scale NLU benchmarks [1], leaving open whether optimal adaptation targets differ between discriminative and generative tasks a gap we address via a controlled ablation on GPT-2 (small) across paraphrase detection and sonnet generation.

Baseline: Full Fine-Tuning of GPT-2

We implement GPT-2 (small, 117M parameters) from scratch following [5, 13], consisting of a byte-pair encoding tokenizer, a combined token-plus-positional embedding layer ($d_{\text{model}}=768$, context length 1024), and 12 masked decoder-only transformer layers each containing multi-head causal self-attention (12 heads, $d_k=64$) and a position-wise feed-forward network with hidden dimension 3072. The full Part 1 implementation, including the masked self attention module, GPT-2 layer, embedding function, and AdamW optimizer, was written by the primary contributor from scratch. Pretrained HuggingFace weights are loaded into this implementation and the full model is fine-tuned end-to-end, serving as the primary upper-bound baseline.

The two downstream tasks require meaningfully different output formulations. For paraphrase detection, we adopt the cloze-style formulation from [13]: given the prompt “Is sentence B a paraphrase of sentence A ?”, the model generates a single continuation token, and we classify by comparing the log-probabilities assigned to the numerical tokens v_{yes} and v_{no} in the GPT-2 vocabulary. Formally, given last hidden state $h_T \in \mathbb{R}^{d_{\text{model}}}$, we compute logits $W_{\text{unembed}}h_T$ and assign label 1 only if $\log p(\text{yes}) > \log p(\text{no})$. This formulation means the model makes its prediction by generating text naturally, rather than using a separate component trained specifically for classification.

For sonnet generation, we fine-tune on 143 Shakespearean sonnets using next-token cross-entropy loss. Given a sequence $x_1, \dots, x_N$, we minimize:

$$\mathcal{L}_{\text{LM}} = -\frac{1}{N} \sum_{t=1}^{N-1} \log p(x_{t+1} \mid x_1, \dots, x_t). \quad (1)$$

At inference time, we condition on the first three lines of each held-out sonnet and generate the remaining eleven lines autoregressively using nucleus (top- p) sampling [14] with temperature $\tau=1.2$ and $p=0.9$. These two fully fine-tuned models constitute the primary baselines against which all LoRA variants are evaluated for these two tasks.

Extension: LoRA with Targeted Layer Ablation

Full fine-tuning becomes problematic on our 143-sonnet corpus: with 117M trainable parameters and fewer than 2,000 training examples, the model risks overfitting to the training distribution and overwriting the pretrained representations that make GPT-2 a useful starting point. LoRA [1] addresses this by freezing the pretrained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ (where d and k denote the output and input dimensions of the adapted layer, respectively) and injecting a trainable low-rank residual into selected linear projections:

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

A is initialized from $\mathcal{N}(0, \sigma^2)$ (with σ a small constant as in [1]) and B to zero, so the adaptation contributes nothing at the start of training and the model begins from its pretrained state. The scaling factor α/r normalizes the magnitude of the update; we fix $\alpha = 2r$ throughout, keeping $\alpha/r \equiv 2$ across all rank settings so that changing rank does not implicitly change the update scale. The LoRA module, experiment sweep infrastructure, analysis plots, and reports were implemented with AI code assistance under the primary contributor’s direction.

Our central research question is whether the choice of which matrices to adapt differentially affects discriminative versus generative tasks. We define four target configurations. **QV** adapts only the query and value projections, following the default in [1]. **QKVO** extends this to all four attention projections (query, key, value, and output). **MLP** adapts exclusively the two feed-forward weight matrices $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ffn}}}$ and $W_2 \in \mathbb{R}^{d_{\text{ffn}} \times d_{\text{model}}}$ within each transformer layer’s position-wise network (where $d_{\text{ffn}} = 3072$ for GPT-2 small), leaving all attention weights frozen. **ALL** combines attention and MLP targets for maximum LoRA coverage. For each configuration we sweep ranks $r \in \{4, 8, 16\}$, yielding trainable parameter counts from approximately 150K (QV, $r=4$) to 2.7M (ALL, $r=16$). These are all well below the 117M of full fine-tuning. The trainable parameter comparisons at ≈ 600 K (QV- $r=16$, QKVO- $r=8$, and MLP- $r=8$) are of particular interest, as they isolate the effect of target choice from the effect of parameter count.

Our guiding hypothesis is that paraphrase detection, a discriminative task mediated by the final token’s contextual representation, should be most sensitive to attention-layer adaptation: self-attention controls which information from the two input sentences is aggregated into that representation, and adapting it directly shapes what the cloze completion attends to. Sonnet generation, by contrast, requires the model to produce coherent, stylistically consistent text across many tokens at once. Feed forward layers are understood to encode factual and stylistic knowledge in transformer models [2], suggesting that MLP adaptation may prove more beneficial for generation. Confirming or falsifying this hypothesis, and identifying whether any single LoRA configuration generalizes across both task structures is the central empirical contribution of this project.

5 Experiments

5.1 Data

We evaluate on two datasets that differ substantially in task structure, scale, and the nature of the expected output. For the discriminative task, we use the Quora Question Pairs (QQP) dataset [4], a large-scale binary classification benchmark consisting of question pairs labeled as paraphrases or non-paraphrases. The version provided for this project contains 141,506 training pairs, 20,215 development pairs, and 40,431 test pairs. Each example consists of two natural-language questions and a binary label indicating semantic equivalence. The model must determine whether the two questions are asking the same thing, despite surface-level lexical variation; a task that probes fine-grained semantic understanding rather than shallow string matching. Under our cloze-style formulation, the input to the model is a natural language prompt of the form “Is question B a paraphrase of question A ?”, and the output is the next generated token, which is interpreted as a yes/no prediction.

For the generative task, we use the Shakespeare Sonnets dataset provided by the course [13], consisting of 154 Shakespearean sonnets split into 143 training sonnets and 12 held-out test sonnets. Each sonnet follows a 14-line structure with the ABAB CDCD EFEF GG rhyme scheme. The small corpus size (roughly 2,000 lines of training text) makes this a challenging low-resource setting where full fine-tuning risks severe overfitting, which is part of the motivation for investigating LoRA. At test time, the model is conditioned on the first three lines of each held-out sonnet and asked to generate the remaining eleven lines autoregressively.

5.2 Evaluation method

The two tasks are evaluated using different metrics suited to their respective output types. For paraphrase detection, the primary metric is binary accuracy on the development and test splits: the fraction of question pairs for which the model’s predicted label (yes or no) matches the ground truth binary label. This metric is standard for QQP [4] and directly reflects the model’s ability to distinguish paraphrase from non-paraphrase pairs. All model checkpoints are evaluated on the development set after each training epoch, and the best-epoch checkpoint is used for final test predictions sent to the course leaderboard.

For sonnet generation, we use chrF [15], a character-level n -gram F-score that measures the overlap between the model’s generated continuation and Shakespeare’s original continuation. chrF is computed over character n -grams up to order 6, averaging precision (how much of the generated text appears in the reference) and recall (how much of the reference appears in the generated text), and is known to correlate well with human judgments of translation and generation quality. Because the sonnet dataset is small and chrF can peak early before degrading as the model overfits the training corpus, we save generated outputs at every epoch and select the best-epoch checkpoint per configuration based on development chrF. This per-epoch evaluation was designed by the primary contributor and implemented with Gemini assistance.

5.3 Experimental details

All experiments use GPT-2 (small, 117M parameters) as the backbone, with pretrained weights loaded from HuggingFace. Training is conducted on an NVIDIA H100 GPU accessed via Modal, a cloud GPU rental service. All configurations are run for 10 epochs with batch size 8 and fixed random seed 11711 for reproducibility.

The full fine-tuning baseline uses learning rate 1×10^{-5} with AdamW[16] and no weight decay for paraphrase detection. LoRA configurations use a higher learning rate of 1×10^{-4} , which is standard practice for PEFT methods: because the pretrained weights are frozen, the low-rank matrices can be trained more aggressively without destabilizing the base representations [1]. The LoRA scaling factor is fixed at $\alpha = 2r$ for all runs, maintaining a constant update scale of $\alpha/r = 2$ regardless of rank. Sonnet generation uses temperature $\tau = 1.2$ and nucleus sampling with $p = 0.9$ at inference time. A summary of all hyperparameter settings is given in Appendix Table 3.

The experiment sweep was orchestrated by a custom `run_experiments.py` pipeline (written with Gemini assistance under the primary contributor’s direction) that sequentially launches each configuration, streams training output in real time, parses dev metrics from stdout, and logs all results to a CSV for leaderboard generation. Per-epoch model checkpoints and prediction files are saved for both tasks, allowing post-hoc selection of the best epoch without re-training.

5.4 Results

The primary goal of this project is not to surpass full fine-tuning, but rather to understand *which* LoRA layer configuration is most effective, and whether this answer differs between a discriminative and a generative task. To establish an upper bound, we evaluated all configurations on the development set across all ten epochs, selecting the best-epoch checkpoint per model. For the course leaderboard, we use the full fine-tuning baseline, which achieved the highest dev performance in both tasks, as expected for a model with an order of magnitude more trainable parameters. To establish an upper bound, the full fine-tuning baseline achieved 41.50 Sonnet chrF and 0.876 paraphrase accuracy on the test set. Table 1 reports best-epoch development set results for all configurations. Full fine-tuning reaches 0.8979 accuracy on paraphrase detection and 42.11 chrF on sonnet generation. Among the

LoRA variants, the most parameter-heavy configurations perform best on both tasks: LoRA-ALL-r16 (2.7M trainable parameters, 2.3% of full fine-tuning) achieves 0.8919 paraphrase accuracy and 41.50 chrF, closing within 0.006 accuracy points and 0.61 chrF of the full fine-tuning baseline. LoRA-ALL-r8 follows closely at 0.8897 and 41.46 chrF. These results are largely consistent with expectations that as the number of trainable parameters grows, performance tends to improve, but the magnitude of LoRA’s competitiveness is strong: LoRA-ALL-r16 recovers over 93% of the full fine-tuning gain on paraphrase detection using just 2.3% of the parameters.

Table 1: Best-epoch development set results for all configurations. Full FT serves as the upper-bound baseline. Bold indicates the best result per metric. Params (M) refers to trainable parameters only.

Model	Params (M)	Para Acc	Sonnet chrF
Full FT	~117.0	0.8979	42.11
LoRA-QV-r4 (150K)	~150K	0.8703	40.49
LoRA-QV-r8 (300K)	~300K	0.8787	39.29
LoRA-QV-r16 (600K)	~600K	0.8824	40.79
LoRA-QKVO-r8 (600K)	~600K	0.8825	39.10
LoRA-MLP-r8 (600K)	~600K	0.8867	39.05
LoRA-ALL-r8 (1.2M)	~1.2M	0.8897	41.46
LoRA-ALL-r16 (2.7M)	~2.7M	0.8919	41.50

The most illuminating comparisons are the training parameter configurations at approximately 600K trainable parameters: LoRA-QV-r16, LoRA-QKVO-r8, and LoRA-MLP-r8. For paraphrase = detection, MLP-r8 (0.8867) outperforms both QKVO-r8 (0.8825) and QV-r16 (0.8824). For sonnet generation, the ordering reverses: QV-r16 (40.79) leads over QKVO-r8 (39.10) and MLP-r8 (39.05). Since all three configurations use an identical parameter budget, this difference is attributable entirely to the choice of which weight matrices are adapted, not to parameter count.

These results are visualized in Appendix Figures 1 and 2. The ablation heatmap (Appendix Figure 1) makes the task-specific reversal visible at a glance: MLP-r8 is the strongest single-module configuration for paraphrase, while QV-r16 is the strongest for sonnet generation at the same budget. The Pareto frontier plots (Appendix Figure 2) show that, for both tasks, LoRA-ALL-r16 traces the efficiency frontier most favorably, offering near-baseline performance at a fraction of the parameter cost.

Additional observations is that within the QV configurations specifically, rank scaling produces a monotonically increasing paraphrase accuracy (r4 $\rightarrow$ r8 $\rightarrow$ r16: 0.8703 $\rightarrow$ 0.8787 $\rightarrow$ 0.8824) but a non monotonic sonnet chrF (40.49 $\rightarrow$ 39.29 $\rightarrow$ 40.79), a divergence that anticipates the stability analysis in Section 6.

The finding that MLP-only adaptation outperforms attention-only adaptation on paraphrase detection is unexpected under our original hypothesis, which predicted the reverse. Equally, the advantage of QV adaptation over MLP for sonnet generation runs contrary to the intuition that feed-forward layers encode more lexical and stylistic knowledge. The fact that both findings contradict the hypothesis is informative: rather than one module type dominating cleanly, the results suggest a more nuanced picture in which combined (ALL) adaptation is reliably best, and the relative ordering of single-module strategies is sensitive to task structure in ways not predicted by the standard mechanistic account. These patterns are examined in greater depth in the Analysis section.

6 Analysis

6.1 Hypothesis Assessment

Our central hypothesis predicted a clean task-module alignment: attention adaptation should benefit the discriminative paraphrase task more, while MLP adaptation should benefit generative sonnet production more. The training parameter comparison at $\approx$ 600K directly falsifies both directions. Appendix Table 2 summarizes the result.

We note that the paraphrase gap (0.0043 accuracy over $\sim$ 20K dev pairs) is modest and should be interpreted cautiously; the sonnet gap of 1.74 chrF is larger but evaluated on only 12 held-out sonnets,

Table 2: Hypothesis assessment at the 600K training parameter point.

Task	Predicted	Observed	Verdict
Para Acc	QV-r16 wins	MLP-r8 (0.8867 vs. 0.8824)	×
Sonnet chrF	MLP-r8 wins	QV-r16 (40.79 vs. 39.05)	×

so both findings are suggestive rather than conclusive without significance testing. Nevertheless, both reversals are directionally consistent and visible in the heatmap (Appendix Figure 1), where the MLP-r8 row is the darkest single-module cell in the Para Acc column, while the QV-r16 row is the darkest in the Sonnet chrF column.

6.2 Why MLP Adaptation Helps Paraphrase

The MLP-r8 advantage is counterintuitive because paraphrase detection depends on comparing two sentences, which is supposedly an attention mediated operation. We offer two explanations. First, GPT-2’s pretrained attention heads already encode strong co-reference and similarity signals from large-scale language modeling; adapting Q and V projections may disturb these useful patterns without providing commensurate gain. Adapting MLP layers instead steers the final token representation toward the yes/no decision boundary without perturbing cross-sentence routing.

Second, and more concretely, the error analysis reveals a striking class balance divergence across adaptation targets (see Appendix Table 4). LoRA-QKVO-r8 achieves a paraphrase-class recall of only 0.8233 while its non-paraphrase recall rises to 0.9170. This is a ten-point gap indicating a systematic bias toward “no.” LoRA-QV-r16 shows the same skew. LoRA-MLP-r8 is substantially more balanced, suggesting that attention-only adaptation amplifies surface sensitivity at the cost of paraphrase recall, while MLP adaptation produces a more calibrated decision boundary. A concrete example: “*How do I write an Android application?*” vs. “*How do I create an Android application?*” (True=1) is correctly labeled by Full FT but rejected by LoRA-QV-r16, illustrating the tendency of attention-only configurations to over-reject near-synonymous pairs. The paraphrase learning curves (Appendix Figure 4) confirm that MLP-r8 gradually separates above QV-r16 and QKVO-r8 from epoch five onward.

6.3 Why Attention Adaptation Helps Sonnet Generation

QV-r16’s advantage on sonnet generation is similarly surprising, since MLP layers are understood to encode lexical and stylistic knowledge in transformer models [2]. We suggest that sonnet generation places an unusually high emphasis on long-range structural coherence: rhyme scheme and thematic development require the model to attend to specific prior tokens across fourteen lines. Adapting Q and V projections directly shapes this cross-token routing, while the pretrained MLP weights may already encode sufficient Shakespearean vocabulary to serve the task without explicit fine-tuning.

Critically, QV-r16’s headline score of 40.79 chrF comes at a significant cost in stability. Appendix figure 3 shows that it is the most volatile configuration ($\sigma=3.40$ [17], minimum chrF = 28.48), exhibiting the sharp epoch-2 crash visible in Appendix Figure 5. LoRA-QKVO-r8 is the most stable LoRA variant ($\sigma=0.96$) yet peaks at only 39.10, suggesting a stability–performance trade-off in attention-targeting. Because our evaluation uses best-epoch selection, QV-r16’s advantage is partly a result of high variance: a deployment scenario requiring consistent performance would favor QKVO-r8 or a combined configuration over QV-r16 (full per-epoch stability statistics are reported in Appendix Table 5). This instability is further reflected in the rank-scaling behavior of the QV configurations on sonnet generation: rather than the monotonically increasing trend one would expect if higher rank provided consistent benefit, chrF follows the non-monotonic ordering QV-r8 (39.29) < QV-r4 (40.49) < QV-r16 (40.79). By contrast, the same QV rank sweep on paraphrase detection is strictly monotonic (0.8703 $\rightarrow$ 0.8787 $\rightarrow$ 0.8824), confirming that rank scaling is predictable for the discriminative task but erratic for sonnet generation. Together, the epoch-level variance ($\sigma=3.40$) and the rank-level non-monotonicity support the same interpretation: QV-r16’s leading chrF reflects a favorable draw under high variance best-epoch selection rather than a principled rank-scaling advantage.

6.4 Short vs. Long Pairs and the Case for Combined Adaptation

The subset breakdown (Appendix Figure 6) shows that the short/long accuracy gap (≤ 10 words vs. > 10 words) is consistent across every LoRA configuration. No single adaptation target preferentially recovers performance on short pairs, which are the harder category for all models. This universality suggests that the deficit is systemic: short, surface-similar pairs (e.g., “*Are floppy disks still used today?*” vs. “*Are floppy disks still useful?*”; True=1, missed by MLP-r8, QV r16, and others) require pragmatic knowledge that no low-rank subspace with $\leq 600K$ parameters appears to capture.

Taken together, these analysis support a consistent conclusion: at GPT-2 (small) scale, neither attention nor MLP modules alone are sufficient, and combined adaptation (LoRA-ALL) is the most reliable choice across both tasks, length subsets, and stability criteria. The Pareto frontier (Appendix Figure 2) shows LoRA-ALL-r16 dominating all single-module strategies, recovering over 93% of full fine-tuning accuracy on paraphrase with 2.3% of the parameters. Rather than a clean task-module specialization, our results suggest that at this scale both module types provide complementary adaptations. Attention routing re-weights contextual signals while MLP nonlinearities calibrate output distributions. Cleanly disentangling their contributions may require larger models or more targeted interventions.

7 Conclusion

This work investigated whether the choice of LoRA adaptation target (attention projections, MLP feed forward layers, or both) differentially affects performance on a discriminative task (paraphrase detection) and a generative task (sonnet generation). Our central hypothesis, that attention adaptation would benefit paraphrase detection and MLP adaptation would benefit sonnet generation, was falsified in both directions in a micro-environment. At an training parameter budget of 600K, MLP-r8 outperforms QV-r16 on paraphrase, while QV-r16 outperforms MLP-r8 on sonnet generation. Error analysis suggests the MLP advantage on paraphrase stems from more balanced class recall, while QV-r16’s sonnet advantage is partly the result of high training variance rather than consistent superiority. Combined adaptation (LoRA-ALL) is reliably best across both tasks, recovering over 93% of full fine-tuning accuracy on paraphrase with only 2.3% of the parameters.

A primary limitation of this work is scale and statistical power. All findings are specific to GPT-2 (small, 117M parameters), and the generative evaluation rests on a test set of only 12 held-out sonnets. Consequently, while the observed task-based reversals are directionally consistent, they fall short of statistical significance without formal testing. Future work should replicate this ablation on larger models, apply significance testing across dev-set folds, and investigate layer-wise targeting to more precisely localize where adaptation is beneficial.

A second critical limitation is the conflation of task structure with data availability. Our comparative analysis contrasts a high-resource discriminative task (over 141K training paraphrase pairs) with an extremely low-resource generative task (143 training sonnets). It is highly plausible that the observed preference for QV-r16, and the factually-backed high training variance used to explain it, are results of this low-resource regime rather than the generative objective itself. To successfully disentangle these variables, future work must cross-evaluate on high-resource generative tasks and low-resource discriminative tasks.

Contributions, Disclosure, and Scope Note

Single-author project. The GPT-2 baseline and fine-tuning code were written independently from scratch. The LoRA extension, experiment pipeline, and analysis suite are AI-assisted (Gemini) infrastructure under the supervision of the author; the core baseline architecture remained completely intact. All experimental design, analysis, and interpretation are exclusively the author’s.

Note that some metrics reported in the analysis section are not elaborated on as they are not evaluation metrics but rather diagnostic analysis metrics. For these metrics, appropriate citations are provided for the reader to read further.

Note that this report is optimized for the thoroughness of information per section guidelines.

References

- [1] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.
- [2] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [4] Quora. Quora question pairs - kaggle, 2017.
- [5] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. Accessed: 2024-11-15.
- [6] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.
- [7] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019.
- [8] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190, 2021.
- [9] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *CoRR*, abs/2012.13255, 2020.
- [10] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.
- [11] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. Lora learns less and forgets less, 2024.
- [12] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *EMNLP*, 2021.
- [13] Stanford CS224N Staff. CS 224N default final project: Build GPT-2. https://web.stanford.edu/class/cs224n/project/DFP_Instructions.pdf, 2025. Course project instructions, Stanford University.
- [14] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *CoRR*, abs/1904.09751, 2019.
- [15] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.
- [17] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. *CoRR*, abs/2006.04884, 2020.

A Appendix

A.1 Experimental setup

Table 3: Hyperparameters shared across all configurations unless otherwise noted.

Hyperparameter	Full FT	LoRA variants
Learning rate	1×10^{-5}	1×10^{-4}
Batch size	8	8
Epochs	10	10
LoRA scaling α/r	—	2 (fixed)
Ranks r swept	—	$\{4, 8, 16\}$
Dropout p	0.1	0.1
Random seed	11711	11711
Inference temperature	1.2 (sonnet)	1.2 (sonnet)
Nucleus p	0.9 (sonnet)	0.9 (sonnet)

A.2 Main results visualizations

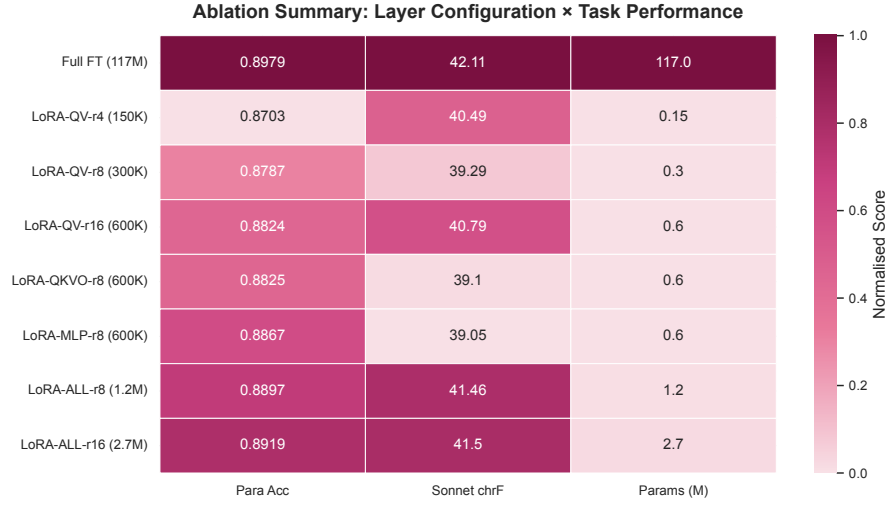


Figure 1: Ablation heatmap of normalized scores across all configurations and metrics. Darker cells indicate higher normalized performance. The inversion between the MLP and QV columns across the two task columns illustrates the task-dependent nature of optimal layer targeting.

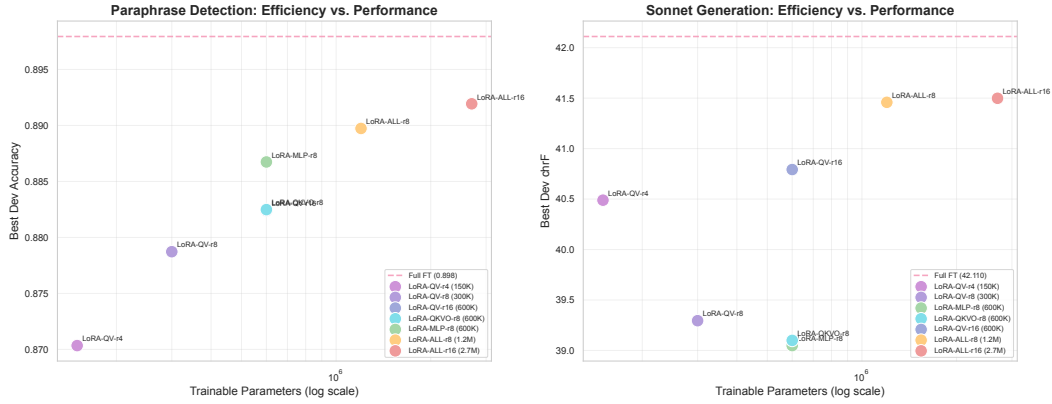


Figure 2: Pareto frontier of trainable parameter count (log scale) versus best development performance for paraphrase detection (left) and sonnet generation (right). The dashed horizontal line marks the full fine-tuning baseline. LoRA-ALL-r16 consistently traces the efficiency frontier closest to the baseline.

A.3 Training stability across tasks

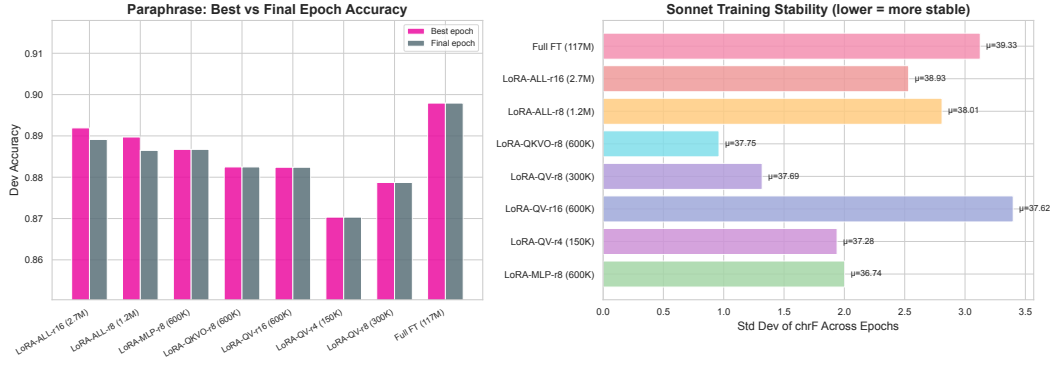


Figure 3: Left: paraphrase best-epoch vs. final-epoch accuracy (all configs converge stably). Right: chrF standard deviation across 10 epochs for sonnet generation. LoRA-QV-r16 is the most volatile ($\sigma=3.40$) [17]; LoRA-QKVO-r8 is most stable ($\sigma=0.96$). LoRA-ALL variants achieve the best mean chrF while maintaining moderate variance.

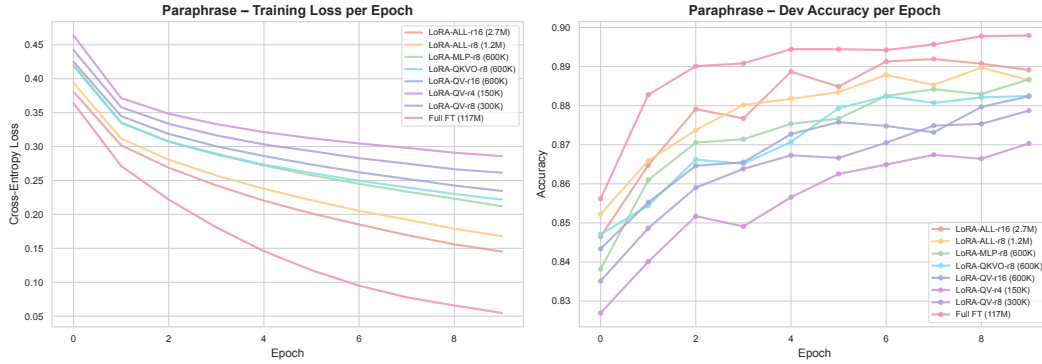


Figure 4: Paraphrase training loss (left) and dev accuracy per epoch (right). LoRA-MLP-r8 separates above QV and QKVO configurations from epoch 5 onward.

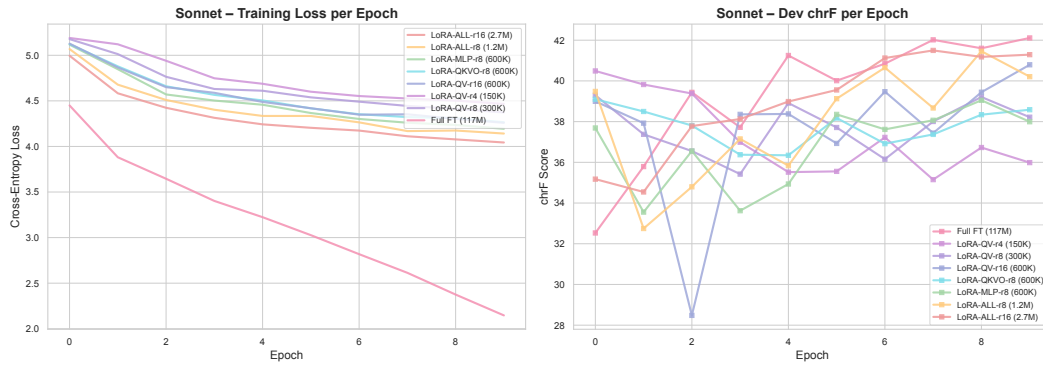


Figure 5: Sonnet training loss (left) and dev chrF per epoch (right). LoRA-QV-r16 exhibits a characteristic crash to $\text{chrF} \approx 28.5$ at epoch 2 before recovering, consistent with its high variance ($\sigma=3.40$).

A.4 Paraphrase analysis

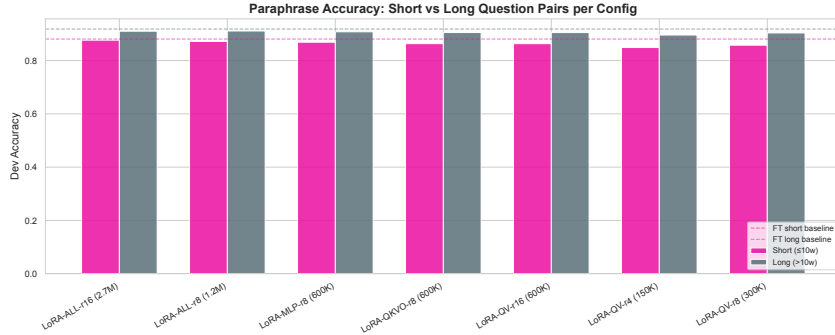


Figure 6: Paraphrase dev accuracy on short (≤ 10 words) vs. long (> 10 words) question pairs. The gap is consistent across all configurations; no LoRA variant differentially recovers performance on short pairs.

Table 4: Per-class accuracy on the paraphrase dev set. Para = paraphrase class (label=1); Non-para = non-paraphrase class (label=0). Attention-only configurations (QV-r16, QKVO-r8) show a systematic skew toward non-paraphrase prediction; MLP-r8 is notably more balanced.

Model	Para Recall	Non-Para Recall
Full FT	0.8846	0.9057
LoRA-QV-r16	0.8525	0.8999
LoRA-QKVO-r8	0.8233	0.9170
LoRA-MLP-r8	0.8808	0.8902
LoRA-ALL-r8	0.8636	0.9050
LoRA-ALL-r16	0.8920	0.8919

A.5 Sonnet analysis

Table 5: Sonnet chrF training stability across 10 epochs per configuration. LoRA-QV-r16 achieves the highest best-epoch score among 600K configurations but at the cost of the highest variance; LoRA-QKVO-r8 is the most stable LoRA variant overall.

Model	Mean chrF	Std Dev	Min	Max
Full FT (117M)	39.33	3.13	32.53	42.11
LoRA-QV-r4 (150K)	37.28	1.94	35.15	40.49
LoRA-QV-r8 (300K)	37.69	1.32	35.42	39.29
LoRA-QV-r16 (600K)	37.62	3.40	28.48	40.79
LoRA-QKVO-r8 (600K)	37.75	0.96	36.35	39.10
LoRA-MLP-r8 (600K)	36.74	2.00	33.56	39.05
LoRA-ALL-r8 (1.2M)	38.01	2.81	32.75	41.46
LoRA-ALL-r16 (2.7M)	38.93	2.53	34.54	41.50