

Task-Specific GPT-2 Adaptation: Structured LoRA for Paraphrase Detection and DPO for Sonnet Generation

Stanford CS224N Default Project

Xijia Liu

xijialiu@stanford.edu

Puyang Du

puyangdu@stanford.edu

Abstract

We study GPT-2 adaptation under limited compute and data on two contrasting tasks: paraphrase detection and sonnet generation. Simple scaling is not reliably effective: on paraphrase detection, GPT-2 medium with uniform LoRA does not beat full fine-tuning of GPT-2 small, while on sonnet generation, supervised fine-tuning on only 131 sonnets is unstable and larger backbones do not help. We therefore test two task-aware strategies: sensitivity-guided LoRA, which reallocates a fixed rank budget across layers using Fisher-style importance scores, and DPO on top of the best supervised fine-tuned model. Sensitivity-guided LoRA improves dev accuracy from 0.874 to 0.882 and matches the 0.870 test accuracy of full fine-tuning; DPO consistently improves sonnet generation, reaching 42.28 chrF on the dev set and 42.04 on the test set in the best setting.

1 Key Information to include

- Mentor: Tristan Thrush
- Late Days:

Team member	Late days remaining
Xijia Liu	2
Puyang Du	2
Total pooled	4
Team size	2
Late days applied for the report	2

2 Introduction

Adapting large pretrained language models to downstream tasks is a central challenge in NLP: full fine-tuning is effective but expensive, while parameter-efficient methods such as LoRA [1] reduce cost but may sacrifice performance. This trade-off is especially pronounced under limited compute and data, where design choices—how to allocate adaptation capacity and how to construct training signal—can matter more than model scale alone.

We study this question on two contrasting tasks from the CS224N default project: paraphrase detection and sonnet generation. For paraphrase detection, we find that simply scaling from GPT-2 small to GPT-2 medium with uniform LoRA does not improve over full fine-tuning of the smaller model. We propose **sensitivity-guided LoRA**, which uses a Fisher-style score (the mean squared gradient norm of the answer-token loss on each layer’s query and value modules) to estimate layer importance, then reallocates a fixed rank budget accordingly, and show that it recovers the performance gap. For sonnet generation, supervised fine-tuning on only 131 sonnets is unstable and does not benefit from larger backbones. We apply **Direct Preference Optimization (DPO)** [2] on top of the best SFT model using rejection-sampled preference pairs, and show consistent chrF improvements across multiple seeds and data scales. Our results suggest that task-aware adaptation strategies—whether structural

(sensitivity-guided rank allocation) or training-signal-based (DPO)—can outperform naïve scaling under tight resource constraints.

3 Related Work

Efficient adaptation of pretrained language models. A central line of prior work studies how pretrained language models can be adapted efficiently to downstream tasks. GPT-2 showed that a single autoregressive pretrained model can transfer across a wide range of tasks, making it a natural backbone for downstream adaptation [3]. Building on this paradigm, LoRA introduced a simple parameter-efficient fine-tuning method that freezes the backbone and learns low-rank updates on selected modules [1]. Later work such as AdaLoRA argued that adaptation budget need not be distributed uniformly across layers, and that importance-aware allocation can improve efficiency and performance [4]. Together, these papers motivate our paraphrase experiments by suggesting that the design of LoRA itself, not just the choice between full fine-tuning and LoRA, can matter for downstream task performance.

Preference optimization for language generation. A separate line of work studies how to improve generation beyond standard supervised fine-tuning. Direct Preference Optimization (DPO) showed that preference learning can be performed with a simple objective over chosen and rejected outputs, avoiding the complexity of reinforcement-learning-based alignment methods [2]. RS-DPO further emphasized that the construction of preference pairs is itself an important design choice, and proposed using rejection sampling to obtain stronger pairs for DPO training [5]. These papers motivate our sonnet-generation experiments by suggesting that preference-based optimization can provide gains even after supervised fine-tuning.

Connection to our work. Our project is a useful next step in this literature because it examines whether these ideas remain effective in a smaller-scale GPT-2 setting on classic downstream tasks. Rather than directly reproducing prior methods, we test whether their main intuitions—importance-aware adaptation for LoRA and careful preference construction for DPO—can still lead to meaningful improvements under tighter computational and data constraints.

4 Approach

Structured LoRA for Paraphrase Detection. We study how to design LoRA for GPT-2 under a fixed parameter budget. Following Hu et al. [1], LoRA freezes the pretrained weight W_0 and learns a low-rank update

$$W = W_0 + \frac{\alpha}{r}BA,$$

where A and B are trainable low-rank matrices, r is the rank, and α is a scaling factor. Our uniform LoRA applies the same rank to the query and value projections in every transformer block.

Our main innovation is **sensitivity-guided LoRA**, which reallocates the same total rank budget across layers. Using the pretrained GPT-2 model, we estimate a Fisher-style importance score for each layer l on a random subset of N training examples:

$$I_l = \frac{1}{N} \sum_{n=1}^N \|\nabla_l^{(n)}\|_2^2$$

where $\nabla_l^{(n)}$ is the gradient of the answer-token loss with respect to the query and value modules in layer l . We then apply log smoothing, $\tilde{I}_l = \log(1 + I_l)$. Let R_{tot} be the total rank budget and r_{min} the minimum rank assigned to each layer, which prevents low-score layers from being dropped entirely. We allocate rank by

$$\hat{r}_l = r_{\text{min}} + (R_{\text{tot}} - Lr_{\text{min}}) \frac{\tilde{I}_l}{\sum_{j=1}^L \tilde{I}_j},$$

and then round with largest-remainder rounding so that $\sum_{l=1}^L r_l = R_{\text{tot}}$. This gives higher rank to more sensitive layers while keeping the total budget fixed.

DPO for Sonnet Generation. Our second main approach applies Direct Preference Optimization (DPO) [2] on top of a supervised fine-tuned (SFT) GPT-2 small model to improve sonnet generation. The pipeline has two stages. In the *SFT stage*, we fully fine-tune GPT-2 small on the training sonnets with cross-entropy loss and early stopping (patience 3), producing our best SFT model (B1). In the *DPO stage*, we freeze B1 as the reference model π_{ref} and inject fresh LoRA adapters (rank $r=4$, $\alpha=8$) into the query and value projections; only these adapters are trained. At initialization, the policy is identical to π_{ref} , so the initial DPO loss is $\approx \log 2 \approx 0.693$.

To construct preference pairs, we follow the rejection-sampling approach of RS-DPO [5]. For each training sonnet, we condition on the first 3 lines and sample k candidate continuations from the SFT model using nucleus sampling (temperature 0.8, top- p 0.9). Each candidate is scored by chrF against the gold continuation, and the highest- and lowest-scoring candidates form a chosen-rejected pair. We study $k \in \{4, 8, 16, 32\}$ to understand how the number of candidates affects both preference pair quality and downstream DPO performance. We additionally investigate whether filtering pairs by a minimum chrF gap threshold improves training, as proposed in RS-DPO.

Baselines and code base. Our baselines follow the provided CS224N default-project starter code¹, using full fine-tuning GPT-2-small as the main reference setting. All non-baseline components (e.g., LoRA and DPO) are implemented by us on top of this codebase.

5 Experiments

Data and Evaluation Method. We use the CS224N default-project datasets: Quora paraphrase detection (141,506 train / 20,215 dev / 40,431 test) and Shakespeare sonnet generation (131 train / 12 dev / 12 test sonnets). We evaluate paraphrase detection by **accuracy** and sonnet generation by **chrF** on dev set and test set.

Experimental details. For **paraphrase detection**, we train gpt2 and gpt2-medium on a single NVIDIA T4 GPU with AdamW, batch size 8, learning rate 1×10^{-5} , and 15 epochs, selecting the best dev checkpoint. Full fine-tuning of gpt2 takes about 1.5 hours per epoch. For LoRA-based gpt2-medium models, we inject adapters into the attention query and value projections. Uniform LoRA uses rank $r = 8$ and $\alpha = 2r$ in every block, with 786,432 trainable parameters (0.22%), and takes about 2 hours per epoch. Sensitivity-guided LoRA keeps the same modules and budget, computes layer importance from 200 sampled training examples after a 100-step rank-4 probe warm-up, and takes about 2.5 hours per epoch.

For **sonnet generation**, all models are trained on a single NVIDIA T4 GPU. For SFT, we use AdamW, batch size 8, and learning rate 1×10^{-5} . B0 trains for 10 epochs without early stopping; B1–B3 use early stopping with patience 3, allowing up to 20 epochs for GPT-2 small and up to 100 epochs for GPT-2 medium with LoRA (B3), since LoRA’s small parameter count leads to slower convergence. Each SFT epoch takes about 3 seconds for gpt2 and about 5 seconds for gpt2-medium. For DPO (B4–B7), we freeze the B1 SFT checkpoint as the reference model and inject LoRA adapters with rank $r = 4$ and $\alpha = 2r$ into the query and value projections. We train DPO for 20 epochs with learning rate 1×10^{-6} , $\beta = 0.1$, and gradient clipping at norm 1.0; each DPO run takes about 5 minutes. To construct preference pairs, we sample $k \in \{4, 8, 16, 32\}$ candidate continuations per prompt using nucleus sampling (temperature 0.8, top- p 0.9), yielding 131 pairs per k value (one per training sonnet). Pair construction takes approximately 47 minutes for $k=16$ and 95 minutes for $k=32$ on a T4 GPU. To assess statistical significance, all sonnet models are evaluated across 5 random seeds, and we report mean chrF with 95% confidence intervals using the t -distribution.

Results. Table 1 summarizes our results. All sonnet results report the mean and 95% confidence interval over 5 random seeds. Paraphrase results are single-run due to limited computational budget.

For paraphrase detection, uniform LoRA on GPT-2 medium underperforms the GPT-2 small full fine-tuning baseline on both dev and test (0.874/0.823 vs. 0.899/0.870), suggesting that simply scaling the backbone with a uniform LoRA design is insufficient for this task. Our sensitivity-guided LoRA improves over uniform LoRA and recovers the test gap (0.882 dev, 0.870 test), indicating that layer-wise allocation of LoRA capacity may matter more than backbone scale alone. Figure 2

¹https://github.com/stanfordnlp/cs224n_gpt

Model	Configuration	Dev accuracy	Test accuracy
A0	GPT-2 small, full fine-tuning	0.899	0.870
A1	GPT-2 medium, LoRA-SFT (uniform, $r=8$)	0.874	0.823
A2	GPT-2 medium, LoRA-SFT (sensitivity-guided)	0.882	0.870

Model	Configuration	Dev chrF (mean $\pm$ CI)	Test chrF
B0	GPT-2 small, full FT, no early stop	41.28 ± 0.46	—
B1	GPT-2 small, full FT + early stop	41.72 ± 0.35	—
B2	GPT-2 medium, full FT + early stop	41.18 ± 0.50	—
B3	GPT-2 medium, LoRA-SFT ($r=16$) + early stop	41.39 ± 0.52	—
B4	B1 + LoRA-DPO ($r=4$, $k=16$)	42.03 ± 0.20	—
B5	B1 + LoRA-DPO ($r=4$, $k=4$)	41.82 ± 0.49	—
B6	B1 + LoRA-DPO ($r=4$, $k=8$)	41.83 ± 0.53	—
B7	B1 + LoRA-DPO ($r=4$, $k=32$)	42.28 ± 0.28	42.037

Table 1: Paraphrase detection (accuracy) and sonnet generation (chrF $\pm$ 95% CI over 5 seeds). B1–B3 use early stopping (patience 3). B4–B7 apply LoRA-DPO on top of frozen B1; k is the number of rejection-sampled candidates per prompt.

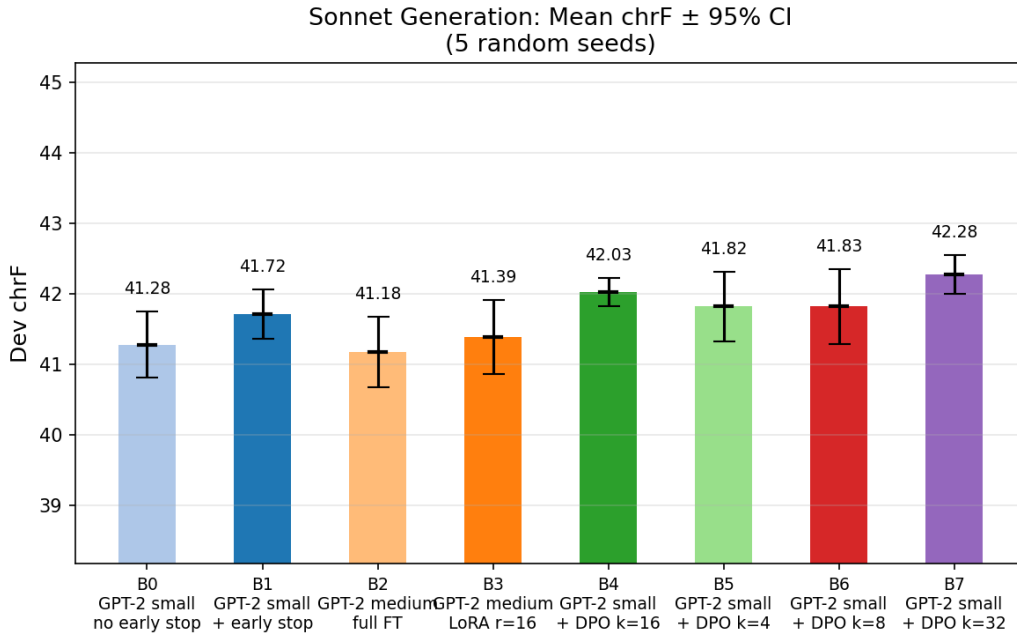
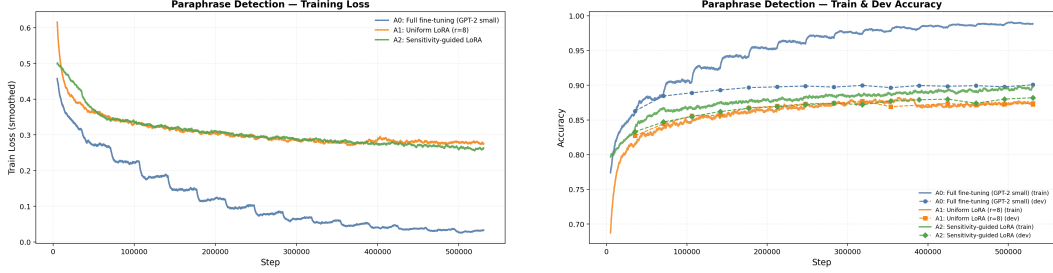


Figure 1: Mean dev chrF $\pm$ 95% CI for all sonnet models across 5 random seeds.

compares the training dynamics of all three approaches. The training loss curves (Figure 2a) show that both LoRA variants converge to a similar range, but the sensitivity-guided variant achieves a slightly lower final loss (~ 0.26 vs. ~ 0.28), consistent with its higher dev accuracy. The accuracy curves (Figure 2b) further illustrate this: while both LoRA methods plateau well below the full fine-tuning train accuracy (~ 0.99), the sensitivity-guided LoRA maintains a visible advantage over uniform LoRA in both train and dev accuracy throughout training, reaching a final dev accuracy of 0.882 compared to 0.874 for uniform LoRA.

For sonnet generation, early stopping gives a modest improvement in mean chrF (B1 vs. B0: $+0.44$), though the confidence intervals overlap. Scaling to GPT-2 medium with full fine-tuning (B2) actually hurts performance, likely due to overfitting on the small 131-sonnet training set. LoRA on GPT-2 medium (B3) partially recovers through implicit low-rank regularization but does not surpass B1.

All DPO variants (B4–B7) improve over the SFT baselines, if only looking at the mean of the Dev chrF of the 5 random seed generations. B7 ($k=32$) achieves the highest mean chrF of 42.28, followed



(a) Training loss (smoothed; first 5k steps omitted). (b) Train and dev accuracy (first 5k steps omitted).

Figure 2: Paraphrase detection training curves for full fine-tuning (A0), uniform LoRA (A1), and sensitivity-guided LoRA (A2). Left: training loss. Right: train accuracy (solid) and dev accuracy (dashed with markers).

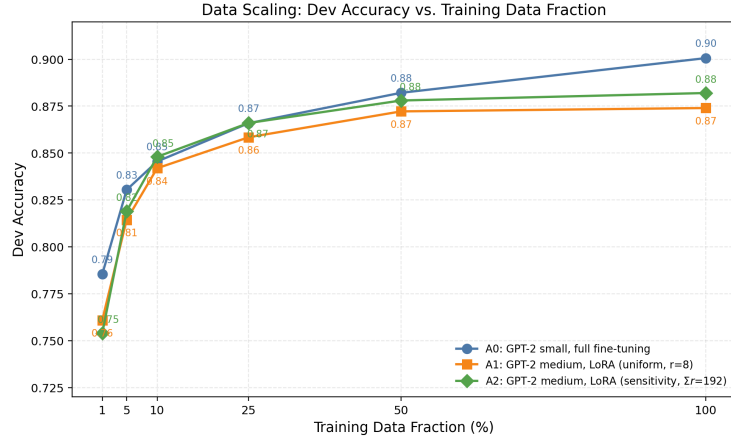


Figure 3: Data scaling for paraphrase detection. Dev accuracy vs. training data fraction for full fine-tuning (A0), uniform LoRA (A1), and sensitivity-guided LoRA (A2).

by B4 ($k=16$) at 42.03. However, the confidence intervals of B4–B7 largely overlap, indicating that the performance differences between DPO k variants are not statistically significant under our 5-seed evaluation. The DPO improvement is most visible for B7 and B4, whose CIs show the least overlap with the SFT baselines, though all differences remain within the margin of statistical uncertainty under 5 seeds. The DPO gains are smaller than we initially expected, which likely reflects the very limited training corpus (131 sonnets) constraining the quality of the rejection-sampled preference pairs.

6 Analysis

6.1 Paraphrase Detection Analysis

Data Scaling. Figure 3 shows dev accuracy as a function of training data fraction (1–100%) for all three approaches. All methods follow a log-linear scaling trend with diminishing returns beyond 25%. Full fine-tuning (A0) consistently leads, reaching 0.899 at 100%. Uniform LoRA (A1) saturates early—accuracy barely changes from 50% to 100% (0.872 $\rightarrow$ 0.874)—suggesting that a fixed rank-8 bottleneck limits how much the model can absorb from additional data. Sensitivity-guided LoRA (A2) scales more favorably than A1: it starts slightly lower at 1% (0.75 vs. 0.76), where the importance estimate is noisy, but overtakes A1 by 5% and maintains a consistent advantage thereafter, finishing at 0.882. The widening gap between A2 and A1 at higher data fractions indicates that concentrating LoRA capacity on task-critical layers improves not only final accuracy but also data efficiency.

Layer-wise Importance and Rank Allocation. Figure 4 visualizes the importance scores alongside the proportionally allocated ranks for our sensitivity-guided LoRA configuration. The importance

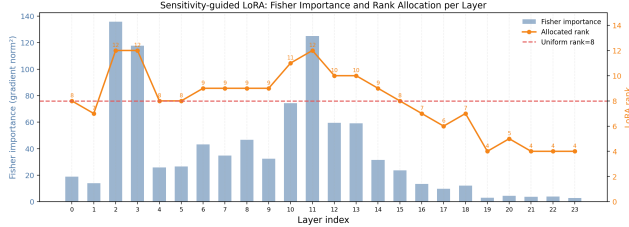


Figure 4: Layer-wise Fisher importance (bars, left axis) and allocated LoRA rank (line, right axis) for GPT-2 medium on the paraphrase detection task. The dashed red line indicates the uniform baseline rank of 8.

distribution is highly non-uniform and reveals two prominent peaks. Layers 2–3 exhibit the highest Fisher sensitivity (~ 136 and ~ 118), suggesting they are critical for early syntactic alignment between the two input sentences. A second peak occurs at layer 11 (~ 125), likely corresponding to mid-network semantic composition where the model integrates meaning across the sentence pair to form its paraphrase judgment. In contrast, the final six layers (18–23) receive near-zero gradient signal (all below 13), indicating that the output-side representations contribute minimally to task-specific adaptation for this classification task.

Under our proportional allocation scheme, the three peak layers receive rank 12—50% above the uniform baseline of $r=8$ —while the tail layers are assigned the minimum rank of 4. The total rank budget ($\sum r_i = 192 = 24 \times 8$) exactly matches that of uniform LoRA, ensuring a fair parameter-count comparison. This reallocation yields a dev accuracy of 0.882 compared to 0.874 for uniform LoRA, demonstrating that directing capacity toward the most task-sensitive layers is more effective.

6.2 Sonnet Generation Analysis

Effect of k on preference pair quality. To understand why higher k may benefit DPO, we analyze the chrF gap distributions of the preference pairs constructed with $k \in \{4, 8, 16, 32\}$.

k	Mean gap	Min gap	Max gap	Mean chosen chrF	Mean rejected chrF
4	5.08	1.33	14.45	25.86	20.78
8	7.24	1.89	17.99	26.83	19.59
16	9.73	4.10	21.97	27.70	17.97
32	10.97	5.92	23.99	28.15	17.18

Table 2: Preference pair chrF gap statistics by k . Higher k produces larger gaps between chosen and rejected candidates.

As shown in Table 2 and Figure 5, increasing k consistently produces larger chrF gaps between chosen and rejected candidates. This improvement is driven by both better chosen samples (mean chrF rising from 25.86 to 28.15) and worse rejected samples (mean chrF falling from 20.78 to 17.18), since sampling more candidates makes it more likely to find both strong and weak continuations. The gains are subject to diminishing returns: the mean gap increases by +2.16 from $k=4$ to $k=8$, but only by +1.24 from $k=16$ to $k=32$.

Although the performance differences between k variants are not statistically significant under our 5-seed evaluation (overlapping 95% CIs in Figure 1), we observe a consistent ordering: B7 ($k=32$, 42.28) > B4 ($k=16$, 42.03) > B6 ($k=8$, 41.83) $\approx$ B5 ($k=4$, 41.82). This ordering mirrors the mean gap ordering ($10.97 > 9.73 > 7.24 > 5.08$), suggesting that better-separated preference pairs may contribute to improved DPO performance. A larger-scale evaluation with more seeds would be needed to confirm this trend.

Threshold ablation: does filtering low-gap pairs help? RS-DPO [5] proposes filtering preference pairs by reward gap to remove noisy pairs where the chosen and rejected candidates are insufficiently separated. We investigate whether applying a minimum chrF gap threshold improves DPO on our sonnet generation task.

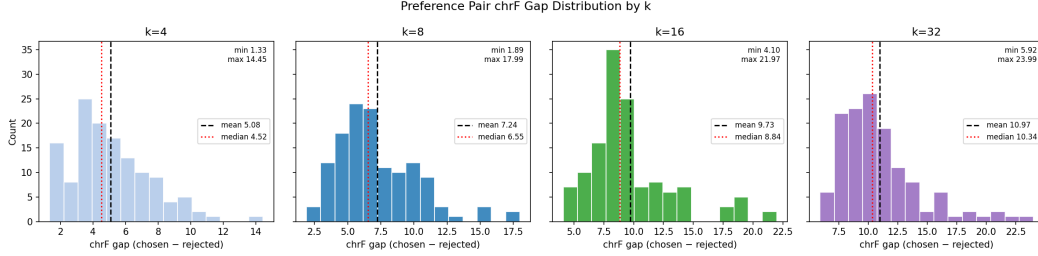


Figure 5: Distribution of chrF gaps (chosen – rejected) for preference pairs constructed with $k = 4, 8, 16, 32$. Dashed and dotted lines indicate the mean and median gap, respectively.

We conduct this ablation with $k=16$ (B4 configuration), which offers a practical balance between pair quality and compute cost. Pair construction with $k=32$ requires approximately twice the time (~ 95 minutes vs. ~ 47 minutes on a T4 GPU), making multi-threshold sweeps impractical under our compute budget. Based on the $k=16$ gap distribution (Figure 5, third panel), which ranges from 4.10 to 21.97 with a mean of 9.73 and median of 8.84, we select thresholds at 5, 7, 10, and 12 to span the distribution from near the minimum to the upper tail.

Min gap threshold	Pairs retained	Dev chrF
0 (no filter)	131 / 131 (100%)	42.03
5.0	131 / 131 (100%)	42.01
7.0	98 / 131 (75%)	41.73
10.0	35 / 131 (27%)	41.55
12.0	17 / 131 (13%)	41.57

Table 3: Threshold ablation for DPO with $k=16$ (single seed). Filtering low-gap preference pairs does not improve chrF; aggressive filtering reduces dataset size and hurts performance.

As shown in Table 3, no threshold improves over using all pairs. Mild filtering (threshold 5.0) retains all 131 pairs and performs comparably. As the threshold increases, the number of retained pairs drops sharply, and performance degrades. This suggests that on our small dataset (131 preference pairs), **dataset size matters more than pair quality**: the DPO loss is already robust to noisy pairs, consistent with the Bradley-Terry formulation that implicitly downweights pairs with small preference margins. Practitioners working with small corpora need not invest in threshold tuning for DPO.

Data scaling. To understand whether DPO’s improvement over SFT is consistent across different training set sizes, we train B1 (SFT only) and B4 (SFT + DPO) on subsets of the training sonnets ranging from 5% to 100% of the full corpus.

Figure 6 shows single-seed results: B4 outperforms B1 across all dataset sizes from approximately 35% of the training set onward. At 45 sonnets (35%), B4 leads by $+0.75$ chrF; at 131 sonnets (100%), B4 leads by $+0.66$. Both models exhibit roughly log-linear scaling, with B4’s curve shifted upward. At very small sizes (6–13 sonnets), B4 shows an advantage that may stem from the implicit regularization of LoRA adapters used during DPO. The DPO benefit does not diminish with more data, suggesting that preference optimization provides a complementary training signal beyond what cross-entropy SFT captures.

Qualitative inspection Since chrF is an automatic metric that measures surface-level n-gram overlap, we perform a brief qualitative inspection to check whether the DPO improvement reflects genuine quality gains. We compare B1 (SFT) and B4 (SFT + DPO, $k=16$) generations on the 12 held-out dev sonnets (selected examples in Appendix A.1); this is not an exhaustive analysis across all models but a quick sanity check on our best SFT and DPO configurations. Both models produce largely incoherent continuations, unsurprising given only 131 training sonnets. B4 uses slightly more Shakespearean vocabulary, which may inflate chrF via n-gram overlap without improving coherence. B4 also occasionally degenerates (e.g., producing numeric tokens absent from verse), consistent with reward hacking [6] on the chrF proxy reward. The $+0.56$ mean chrF gain (B7 vs. B1) should be interpreted with this caveat.

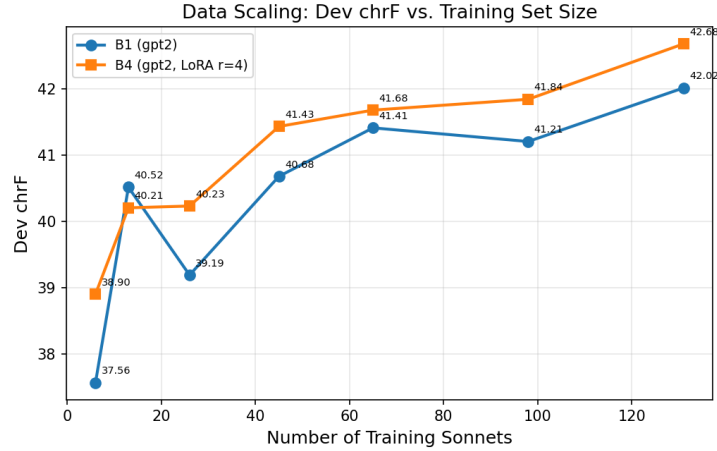


Figure 6: Dev chrF as a function of training set size for B1 (SFT) and B4 (SFT + DPO, $k=16$). DPO provides a consistent improvement over SFT across all dataset sizes from $\sim 35\%$ onward.

7 Conclusion

We studied GPT-2 adaptation on paraphrase detection and sonnet generation under limited compute and data. On paraphrase detection, sensitivity-guided LoRA reallocates rank budget to task-critical layers and improves over uniform LoRA, recovering the full fine-tuning test accuracy of 0.870 while using only 0.22% trainable parameters. On sonnet generation, DPO with rejection-sampled preference pairs consistently improves over SFT baselines, with our best model (B7, $k=32$) achieving 42.28 mean dev chrF and 42.037 test chrF.

At the same time, our results have important limitations. For paraphrase detection, all results are based on single runs due to compute constraints, so the observed gains of sensitivity-guided LoRA over uniform LoRA are not supported by significance testing and may partly reflect run-to-run variance. More broadly, the paraphrase findings are limited to one dataset, one model family, and one fixed parameter budget, so we cannot yet conclude that the same rank-allocation pattern will generalize across tasks or scales. Our Fisher-style importance estimate is also computed from a relatively small subset of training examples, which may introduce noise in low-data settings. For sonnet generation, qualitative inspection shows that the chrF gains partly reflect surface-level n-gram overlap rather than coherent poetry, and occasional degeneration suggests overoptimization of the chrF proxy.

There are several promising directions for future work. For paraphrase detection, a natural next step is multi-seed evaluation, significance testing, and broader ablations to verify the robustness of the sensitivity-guided LoRA gains. It would also be useful to test whether the same rank allocation strategy generalizes beyond Quora and GPT-2. For sonnet generation, future work could explore learned reward models, larger sonnet corpora, or constrained decoding to enforce meter and rhyme.

Team Contributions

- **Xijia Liu:** Led baseline implementation and conducted extension research for sonnet generation: SFT, DPO pipeline (rejection-sampled preference pairs, LoRA-DPO training), k -value sweeps, threshold ablation)
- **Puyang Du:** Led baseline training and conducted extension research for paraphrase detection (uniform LoRA and sensitivity-guided LoRA).

References

- [1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

- [2] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- [3] Alec Radford and Jeffrey Wu. Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners*. *OpenAI blog*, 1(8):9, 2019.
- [4] 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.
- [5] Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. Rs-dpo: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1665–1680, 2024.
- [6] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc., that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.

A.1 Qualitative examples

Below we show three held-out sonnet completions from B1 (SFT) and B4 (SFT + DPO), alongside the gold reference. The **prompt** (first 3 lines) is shared; the **continuation** is model-generated. These examples illustrate that while B4's chrF is higher, both models produce incoherent continuations, and B4 occasionally degenerates.

A.1.1 Sonnet 8 (B4 degeneration example)

Prompt: Be wise as thou art cruel; do not press / My tongue-tied patience with too much disdain, / Lest sorrow lend me words, and words express

B1: "...but purest speech, / Instead of speaking cruel accusations; / Therefore, having thine own temptation put / Behind both complaints for yet another, / I think up impious daily quarrels in calumny..."

B4: "...fiercer songs' opinion. / The blest memory wherever perceived may be; / ...Being thence says **2,207** fair era times white. / Envious, then, why report what she forens that night"

Gold: "...The manner of my pity-wanting pain. / If I might teach thee wit, better it were, / Though not to love, yet, love, to tell me so..."

B4 produces a numeric token ("2,207") that never appears in Shakespearean verse—a sign the policy has drifted from the SFT distribution.

A.1.2 Sonnet 5 (both models incoherent)

Prompt: Thou blind fool, Love, what dost thou to mine eyes / That they behold and see not what they see? / They know what beauty is, see where it lies,

B1: "...Yet go free not to yourself, nor look into their creases. / They see not gay, nor transgressive love for your auguries / Within thy heart thou sake how coldest no wind doth dwell."

B4: "...hear tays true thy cry / That love lies hidden, home on the common sleep / Until the heavens are filled up with it; / If this womankind / Seems very late to entertain thee..."

Gold: "...Yet what the best is take the worst to be. / If eyes, corrupt by overpartial looks, / Be anchored in the bay where all men ride..."

Neither model captures the gold's argumentative structure or iambic pentameter.

A.1.3 Sonnet 11 (B4 vocabulary shift)

Prompt: Lo, as a careful huswife runs to catch / One of her feathered creatures broke away, / Sets down her babe, and makes all swift dispatch

B1: "...From the ancient stream fled / Or with that grace returned straddled to your home / ...All these marvellous names lie. / Which wonderful voices cheer thee, standing new"

B4: "...with it. / Or comes up with impossible limbs and turns him wide with bird, / Then steps by that mad, aching breast up sparing outstretched! / ...And compounded all our fury out of shame"

Gold: "...In pursuit of the thing she would have stay, / Whilst her neglected child holds her in chase, / Cries to catch her whose busy care is bent..."

B4 produces more dramatic vocabulary ("fury," "shame," "aching breast") that may overlap with Shakespearean n-grams, but the narrative coherence of the gold (a mother-child metaphor) is absent from both.