Empirical Evaluation of DPZero for Private Fine-Tuning of RoBERTa-large on SST-2

Anushka Singh

CS839 — Final Report — May 1, 2026

Abstract

This project presents a focused empirical study of *DPZero*, a scalable differentially private fine-tuning method for language models that avoids backpropagation and instead uses zeroth-order, forward-only updates. I evaluate DPZero on SST-2 with `roberta-large` in a 512-shot setting and compare it to the non-private zeroth-order baseline MeZO. The study is organized around three practical questions: how utility changes with privacy budget and clipping threshold, how stable the method is across random seeds, and whether lightweight tuning variants look promising. The experiments show a clear pattern: moderate clipping works best, while large clipping thresholds 300–400 perform poorly across privacy budgets. A second consistent pattern is that stronger privacy shifts the better-performing clipping region toward smaller clipping thresholds. The strongest completed reproduced private configuration in the main comparison is $(\varepsilon = 6, C = 100)$, with mean test accuracy 0.7939 across seven seeds, while a lower-privacy-budget comparison point $(\varepsilon = 2, C = 50)$ reaches mean test accuracy 0.6541 across seven seeds. These reproduced results remain below the paper-reported DPZero accuracies at the same privacy levels, and the gap is especially large at smaller ε. Finally, a preliminary lightweight-variant sweep suggests that head-only tuning hurts utility, while top1 and top2 may preserve performance more effectively. Overall, the project identifies a plausible operating regime for DPZero, shows why plots and multi-seed summaries are necessary, and highlights the next experiments needed for a stronger final comparison.

1 Introduction

Fine-tuning large language models means starting from a pretrained model and adapting it to a downstream task using task-specific labeled data. This is one of the main reasons large pretrained models are practically useful: they already contain general language knowledge, and fine-tuning lets them specialize to one concrete task.

However, private fine-tuning is substantially harder than ordinary fine-tuning. In many realistic applications, the fine-tuning data may contain sensitive text, so the model update procedure must protect privacy. The standard private approach is usually gradient-based: compute per-example first-order gradients, clip them, add noise for privacy in every dimension of the gradients of datapoints, and then update the model. This becomes expensive in both memory and computation as the model grows, which makes private adaptation of large pretrained models difficult to scale in practice.

This motivates zeroth-order fine-tuning. Instead of relying on backpropagation to compute gradients, zeroth-order methods estimate a useful update direction using only forward evaluations of the loss. This is attractive because it can reduce the memory burden relative to gradient-based training, especially for large models.

DPZero is interesting because it combines these ideas. It starts from a zeroth-order, forward-only update and then privatizes that update through clipping and noise. In other words, it is not just a different optimizer; it is a different way to think about private fine-tuning. That makes it important to understand empirically how the method behaves under different privacy budgets and clipping thresholds.

The goal of this project is empirical rather than algorithmic. I study DPZero on SST-2 using `roberta-large`, compare it with the non-private zeroth-order baseline MeZO, and ask three concrete questions:

1. How sensitive is accuracy to the privacy budget ε and clipping threshold C?
2. How large is the seed-to-seed variation when hyperparameters are fixed?
3. Do lightweight tuning variants look promising enough to justify a fuller comparison?

This is therefore a replication-style empirical study of a recent DP method. The point is not to claim a new algorithm, but to build a stronger practical picture of where DPZero works, where it degrades sharply, and which follow-up directions look most worth pursuing.

2 Related Work

This project is centered on two zeroth-order fine-tuning methods. The first is **MeZO**, a memory-efficient zeroth-order optimizer for language-model fine-tuning using only forward passes. MeZO shows that carefully designed stochastic perturbations can make forward-only tuning competitive for downstream adaptation, and it serves here as a natural non-private baseline that isolates the optimization component without privacy.

The second is **DPZero**, which extends the zeroth-order viewpoint to the privacy setting. DPZero is appealing because it attempts to privately fine-tune language models without backpropagation, potentially making private adaptation more memory-efficient than conventional DP-SGD-style training. The original paper reports promising results on multiple language tasks, making it a strong candidate for a course project focused on empirical replication and practical evaluation.

This report does not attempt to improve either algorithm. Instead, it focuses on a narrower but practically important question: how sensitive is DPZero to privacy budget, clipping threshold, and random seed on a standard text-classification task? This is useful because deployment decisions depend on concrete operating points. A method can look strong overall while still being fragile to particular hyperparameters.

SST-2 is used here as a first testbed because it is standard, binary, and easy to interpret. The limitation, of course, is that conclusions from one task should not be overgeneralized. Still, a careful SST-2 study is a reasonable starting point for a project whose purpose is to build a strong empirical foundation.

3 Experimental Setup

I use `roberta-large` on SST-2 within the DPZero codebase. The experiments use the same zeroth-order training pipeline for both MeZO and DPZero. The main runs use a 512-shot setting ($K = 512$), meaning only 512 labeled training examples are used for training. The remaining main hyperparameters are batch size 64, learning rate 10^{-6}, zeroth-order perturbation scale 10^{-3}, and 3000 training steps.

The main DPZero sweep varies two privacy-related hyperparameters:

- privacy budget $\varepsilon \in \{6, 3, 2\}$,
- clipping threshold $C \in \{50, 100, 200, 300, 400\}$.

The full hyperparameter grid was run on 7 seeds. Separately, I also ran a preliminary lightweight-variant comparison at $(\varepsilon = 6, C = 100)$ with four variants: full DPZero, head-only tuning, last-one-block tuning (top1), and last-two-block tuning (top2). Those runs were checked from the logs to verify that the intended freezing configuration was actually applied.

This distinction matters for interpretation. The seed-42 sweep is best viewed as an exploratory diagnostic that reveals trends across clipping thresholds and privacy budgets. The seven-seed comparison is more appropriate for the main conclusion, because average performance over random seeds is the standard way to summarize an algorithm.

4 Results

4.1 Non-private baseline

Table 1 shows the non-private MeZO baseline. This is the strongest result obtained so far in the study and serves as the main utility reference point.

Method	Seed	Dev acc	Test acc
MeZO	42	0.7881	0.8796

Table 1: Non-private zeroth-order baseline.

4.2 Exploratory clipping-threshold sweep

Figure 1 is the most useful summary of the initial hyperparameter sweep. For each fixed ε, it plots dev and test *error* rather than accuracy, because error makes the degradation pattern easier to interpret visually.

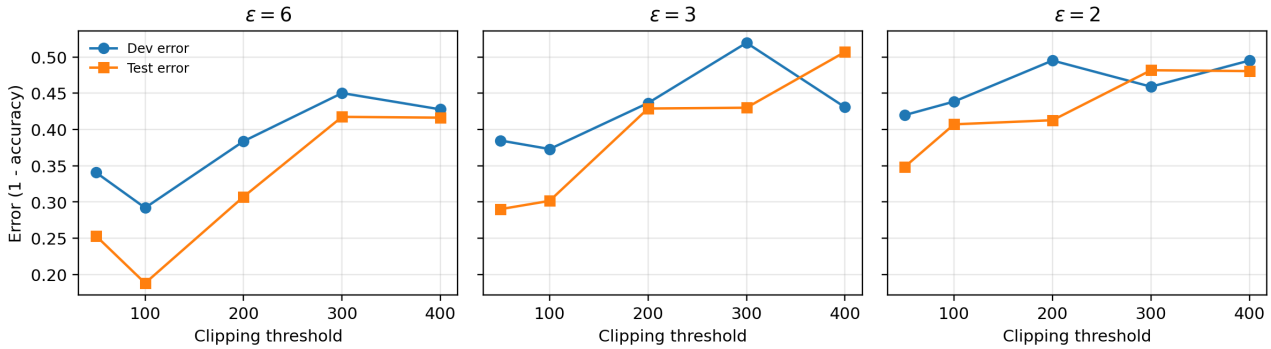

Figure 1: Seed-42 dev and test error as a function of clipping threshold for three privacy budgets. Moderate clipping works best, while large clipping thresholds consistently hurt.

The clearest pattern is that clipping matters sharply. The best seed-42 point overall occurs at $(\varepsilon = 6, C = 100)$, where test accuracy reaches 0.8119. For $\varepsilon = 6$, the relationship between clipping and performance is clearly non-monotone: moving from $C = 50$ to $C = 100$ helps, but moving to $C = 200$ and beyond hurts substantially. The large-clipping regime is consistently poor. At $\varepsilon = 6$, test error rises from about 0.19 at $C = 100$ to about 0.42 at $C = 300$–400. Similar deterioration appears for $\varepsilon = 3$ and $\varepsilon = 2$.

The pattern is more informative than a statement like "the best threshold is 100." What matters is the trend: moderate clipping works better than either extreme, and the poor large-C regime appears repeatedly across privacy budgets.

4.3 Why both very low and very high clipping can hurt

A useful way to interpret the clipping curves is to explain why both extremes hurt. When C is too low, the estimated update is clipped too aggressively, so the model cannot move enough in a genuinely useful direction. This leads to under-powered learning and higher error. In this regime, privacy is being enforced strongly, but the useful optimization signal is also being weakened.

When C is too high, the update is less restricted, but the privacy-noise scale also grows with the clipping threshold. As a result, the private update becomes noisier and less stable. This helps explain the observed U-shaped pattern: error is high when clipping is too small, lower in a moderate middle region, and high again when clipping is too large.

Therefore, the main practical lesson from the clipping sweep is not just that one threshold happened to perform best. The more transferable lesson is that DPZero seems to require a moderate clipping regime in which useful update signal is preserved, but noisy and unstable updates are still controlled.

4.4 Why the best clipping region shifts left under stricter privacy

Another important pattern in the clipping curves is that the better-performing clipping region shifts left as the privacy budget becomes stricter. In other words, when ε is smaller, lower values of C tend to become more reasonable than very large clipping thresholds.

This happens because the amount of privacy noise depends on both the clipping threshold and the privacy budget. A larger clipping threshold allows larger updates, but it also increases the sensitivity of the update. To maintain differential privacy, more noise must be added when the sensitivity is larger. Similarly, when ε is smaller, the privacy requirement is stricter, so the algorithm must add more noise. Therefore, the harmful effect of large C becomes stronger at lower ε.

This explains why a setting like $C = 100$ can work well when $\varepsilon = 6$, but the best region moves closer to $C = 50$–100 when ε decreases. At stricter privacy levels, large clipping thresholds make the private update too

noisy, so smaller clipping thresholds become more favorable. In practical terms, stronger privacy requires more conservative clipping.

4.5 Effect of privacy budget at fixed clipping thresholds

The previous subsection fixed the privacy budget and varied the clipping threshold. It is also useful to look at the complementary view: fix the clipping threshold C and vary the privacy budget ε. This makes the privacy–utility tradeoff easier to interpret.

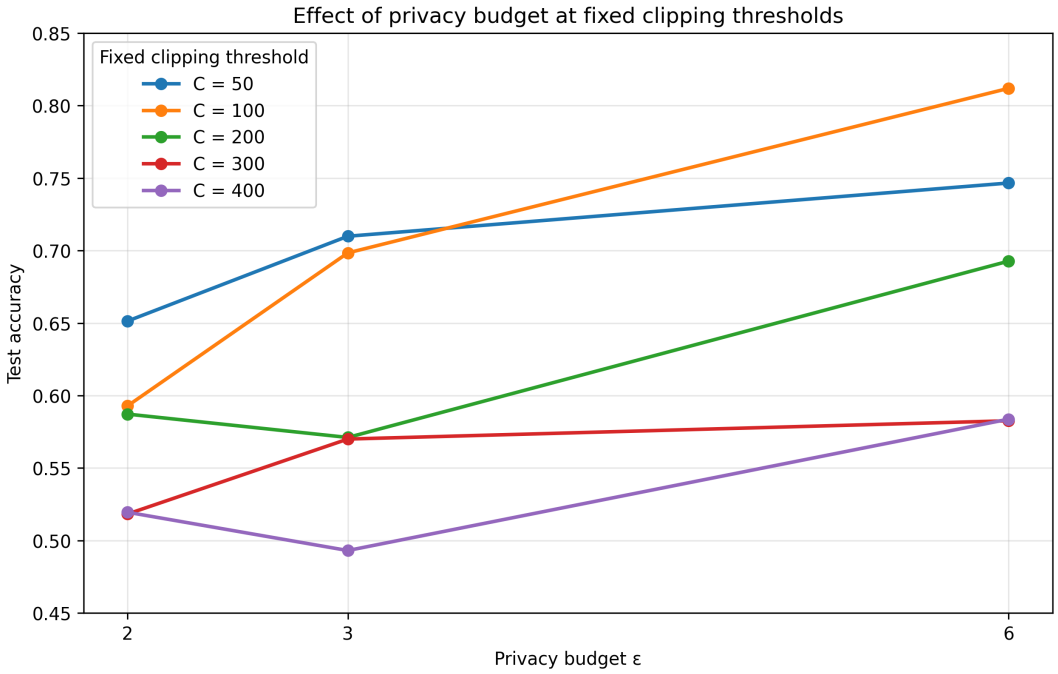

Figure 2: Seed-42 test accuracy as a function of privacy budget for several fixed clipping thresholds. In general, accuracy improves as ε increases, although the trend becomes noisier for large clipping thresholds.

Across most clipping thresholds, the same broad pattern appears: larger ε gives better accuracy. This is expected, since a larger privacy budget corresponds to less privacy noise and therefore less distortion in the update. For example, when $C = 50$, the test accuracy decreases from 0.7466 at $\varepsilon = 6$ to 0.7099 at $\varepsilon = 3$ and then to 0.6514 at $\varepsilon = 2$. A similar trend appears even more clearly at $C = 100$, where test accuracy drops from 0.8119 to 0.6984 to 0.5929 as ε decreases from 6 to 3 to 2.

At the same time, the trend becomes less smooth when the clipping threshold is large. For instance, at $C = 200$, $C = 300$, and $C = 400$, the lower-ε settings do not always decrease monotonically. Instead, the accuracies fluctuate more across ε. This is still consistent with the overall picture: once clipping is already in a poor regime, the optimization becomes noisy enough that the privacy trend is less clean. In other words, the expected privacy–utility tradeoff is most visible when clipping is in a reasonable range, and it becomes harder to see when clipping itself is already hurting performance.

4.6 Comparison with the paper-reported DPZero accuracy

For the main empirical comparison, I focus on the same privacy levels emphasized in the presentation: $\varepsilon = 6$ and $\varepsilon = 2$. The strongest reproduced configuration in my runs is $(\varepsilon = 6, C = 100)$, while the lower-privacy-budget comparison point is $(\varepsilon = 2, C = 50)$. Table 2 compares my reproduced mean test accuracies against the DPZero accuracies reported in the original paper at the same ε values.

Setting	Reported in paper	My reproduced mean	Gap
DPZero, $\varepsilon = 6$	0.9220	0.7939 ($C = 100$)	-0.1281
DPZero, $\varepsilon = 2$	0.9180	0.6541 ($C = 50$)	-0.2639

Table 2: Comparison between the DPZero accuracy reported in the original paper and my reproduced mean test accuracy from the presentation. The reproduced means use the best completed settings I found for those privacy levels.

The main point is not just that my reproduced numbers are lower, but that the gap grows under stronger privacy. At $\varepsilon = 6$, the reproduced mean is lower than the paper by about 0.13, while at $\varepsilon = 2$ the gap is much larger, about 0.26. This suggests that the lower-ε regime is harder to reproduce reliably in the current setup, and that the practical performance of DPZero may be quite sensitive to tuning, implementation details, or stability across seeds.

This table should therefore be read together with the trend plots. The paper-comparison table tells us that my implementation does not fully recover the paper's utility, while the trend plots explain why: clipping and privacy interact strongly, and the useful operating region is narrow enough that poor hyperparameter choices can quickly degrade performance.

4.7 Seed variability remains an important practical issue

Even though the main table above emphasizes the gap with the paper rather than a standalone seed-average comparison table, seed variability remains one of the most important empirical findings in this project. Table 3 summarizes the reproduced seven-seed comparison shown in the presentation for two contrasting DPZero settings: a stronger setting, $(\varepsilon = 6, C = 50)$, and a weaker setting with stricter privacy and larger clipping, $(\varepsilon = 2, C = 100)$.

Setting	Seeds	Mean test acc	Std. dev.	Min test acc	Max test acc	Range
$\varepsilon = 6, C = 50$	7	0.7414	0.083	0.6400	0.8500	0.2100
$\varepsilon = 2, C = 100$	7	0.5671	0.133	0.3900	0.7000	0.3100

Table 3: Seed variability in the reproduced seven-seed DPZero runs. The range is computed as maximum test accuracy minus minimum test accuracy across seeds.

The swings across seeds are large, especially relative to the total scale of the accuracy values. For $(\varepsilon = 6, C = 50)$, test accuracy ranges from 0.6400 to 0.8500, giving a range of 0.2100. For $(\varepsilon = 2, C = 100)$, test accuracy ranges from 0.3900 to 0.7000, giving a larger range of 0.3100. This makes it clear that reporting only one seed would be misleading. A single strong seed can overstate the method's stability, while a single weak seed can understate its potential.

The comparison also shows that the weaker setting has both lower average utility and higher variability. In particular, moving from $(\varepsilon = 6, C = 50)$ to $(\varepsilon = 2, C = 100)$ decreases the mean test accuracy from 0.7414 to 0.5671, while increasing the standard deviation from 0.083 to 0.133 and increasing the range from 0.2100 to 0.3100. This is consistent with the earlier clipping analysis. A smaller privacy budget means stronger privacy and therefore more noise, while a larger clipping threshold increases the sensitivity of the private update and can further increase the effective noise scale. Together, lower ε and higher C make the private update noisier and less stable across random seeds.

Therefore, seed variability should be treated as a real empirical object of study, not just as nuisance noise around a point estimate. For DPZero, the mean accuracy alone is not enough; the standard deviation and range are also important because they show how reliable the method is across different random seeds.

4.8 Preliminary lightweight variants

I also ran a preliminary lightweight-variant sweep at $(\varepsilon = 6, C = 100)$. The verified seed-42 sanity-check runs show that head-only tuning is faster but weaker, while top1 and top2 match the full-model result on that seed. In addition, the presentation results summarize preliminary multi-run means for the same variants, shown in Table 4. I treat these as promising but still preliminary.

Variant	Mean test acc	Mean runtime (sec)
full	0.5879	645
head only	0.4992	480
top1	0.5879	641
top2	0.5879	641

Table 4: Preliminary variant comparison reported in the presentation: mean test accuracy and mean runtime under the same DPZero setting. Head-only is faster but weaker; top1 and top2 currently match full.

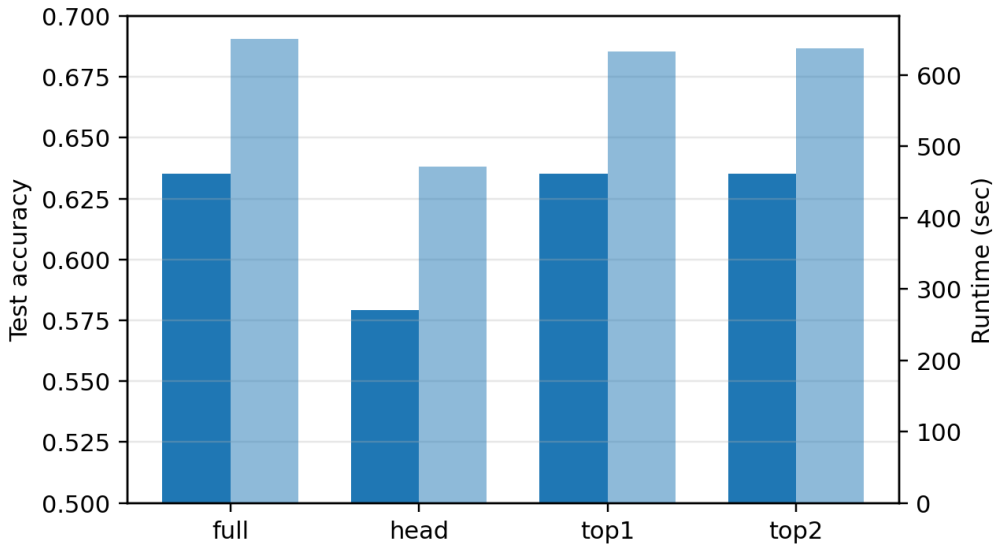

Figure 3: Verified seed-42 sanity-check comparison for the lightweight DPZero variants at ($\varepsilon = 6, C = 100$). Head-only is faster but noticeably weaker, while top1 and top2 match the full variant on this verified run.

Two observations are worth keeping. First, head-only tuning underperforms the other variants, which is consistent with the intuition that the trainable parameter set may become too restrictive. Second, top1 and top2 do not look worse than full in the current preliminary comparison. That is interesting, but it should not yet be overinterpreted. More seeds are necessary before claiming that the lightweight variants truly preserve utility.

4.9 Runtime observations

The runtime logs show that DPZero is feasible to iterate on in this environment. Earlier sweep runs were often in the 34–36 minute range, while the later verified variant runs on seed 42 were closer to 8–11 minutes depending on the variant. This matters practically: the experiments are cheap enough to rerun, which makes it realistic to expand the seed count or add one more axis of comparison before extending the project further.

5 Discussion

The strongest conclusion from the current experiments is that DPZero has a workable but fragile operating regime. The practical lesson is not simply that one exact clipping threshold is best. The more transferable pattern is that moderate clipping beats both extremes: very small clipping thresholds over-restrict useful updates, while very large clipping thresholds allow noisier and less stable privatized updates. This is why the plots are more informative than isolated tables; they reveal a clear non-monotone trend.

The second major lesson is that stronger privacy changes where the useful clipping region lies. As ε becomes smaller, the private update becomes noisier, and the better-performing clipping region shifts lower. In the current runs, the best region is around $C = 100$ for $\varepsilon = 6$, but moves closer to $C = 50$–100 at stricter privacy levels. This kind of trend is much more useful than simply saying that one single threshold is best for one exact setup.

The paper-vs-reproduction comparison adds another important dimension. My implementation does not fully recover the paper-reported accuracy, and the gap grows substantially at stronger privacy. This does not mean DPZero fails. Instead, it suggests that the low-ε regime is harder to reproduce robustly and may be especially sensitive to tuning, implementation details, or seed variation.

At the same time, the seed variability is itself interesting. The current data suggest that instability is not a minor nuisance but a real part of DPZero's behavior in this regime. The lightweight variants add one more lesson: a head-only version looks weaker in the current runs, while top1 and top2 remain plausible candidates for a fuller future comparison. Together, these results support a balanced conclusion: DPZero is promising, but its practical story is mainly about tuning and stability rather than about one isolated strong number.

6 Future Directions

The next steps are clearer now than they were in the draft stage.

1. Promote the lightweight-variant sweep from diagnostic to main comparison. The right follow-up is to run the verified variant pipeline across at least three to five seeds and then compare full, head, top1, and top2 using mean accuracy and variance. That would directly answer whether lighter tuning improves stability or preserves utility.

2. Strengthen the paper-vs-reproduction comparison. The current comparison is already informative, but it would become much more convincing with additional multi-seed runs at the strongest privacy levels and with even tighter verification of implementation settings.

3. Explain why moderate clipping works best. The current evidence is descriptive. A stronger follow-up would inspect whether the scale of the zeroth-order update interacts with privacy noise in a way that makes very large clipping thresholds ineffective.

4. Compare equal-step and equal-time budgets. The present study mostly fixes the number of optimization steps. A more refined evaluation could ask whether lower-ε configurations recover some of the gap with more steps or equalized runtime.

5. Extend beyond SST-2. A second text-classification task would test whether the clipping trends observed here are specific to SST-2 or reflect a broader pattern.

7 Conclusion

This project provides a focused empirical study of DPZero on SST-2 using RoBERTa-large in a 512-shot setting. The main empirical result is that my reproduced DPZero accuracy remains below the paper's reported accuracy at the same privacy levels, with a smaller gap at $\varepsilon = 6$ and a much larger gap at $\varepsilon = 2$. At the same time, the experiments reveal a clear and interpretable practical pattern: moderate clipping works best, very low and very high clipping thresholds both hurt, and the better-performing clipping region shifts lower as privacy becomes stronger. The results also show that seed variability is large enough that average-over-seeds reporting is essential. A preliminary lightweight-variant sweep suggests that head-only tuning is weaker and that top1/top2 deserve a fuller comparison. As a course project, this is a strong empirical foundation because it goes beyond isolated numbers and identifies the main trends that govern when DPZero works better or worse in practice.

References

[1] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *International Conference on Learning Representations*, 2019.

[2] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-Tuning Language Models with Just Forward Passes. In *Advances in Neural Information Processing Systems*, 2023.

[3] Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. DPZero: Private Fine-Tuning of Language Models without Backpropagation. In *International Conference on Machine Learning*, 2024.