USENIX Security '25 Artifact Appendix: Unlocking the Power of Differentially Private Zeroth-order Optimization for Fine-tuning LLMs

Ergute Bao[*], *Yangfan Jiang*[†], *Fei Wei*[*], *Xiaokui Xiao*[†], *Zitao Li*[*], *Yaliang Li*[*], *Bolin Ding*[*]
[*]*Alibaba Group,* [†]*National University of Singapore*

A Artifact Appendix

A.1 Abstract

The main idea of our proposed method, DP-AggZO, is to aggregate **multiple zeroth-order estimates** for the exact gradients, computed over independent perturbation vectors (random Gaussian vectors), before enforcing differential privacy (i.e., artificial clipping, taking the average, and then injecting random DP noises). Compared with the vanilla DPZO (or DPZero), which is effectively a degenerated version of DP-AggZO with only **one** zeroth-order estimate, our DP-AggZO achieves much better utility under the same privacy constraints. Our DP-AggZO also outperforms the state-of-the-art DP-AdamW in some cases. This artifact is used for validating the above claim.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Security, privacy, and ethical concerns are not applicable to this artifact as the models and datasets are publicly available.

A.2.2 How to access

Access via Zenodo in https://zenodo.org/records/15594622.

A.2.3 Hardware dependencies

A workstation or cloud computing node with a GPU (preferrably with GPU memory larger than 20 GB), e.g., RTX 4090 GPU 24GB, or above (larger GPU memory is needed if run larger models, e.g., OPT 6.7B).

A.2.4 Software dependencies

Linux system Ubuntu 22.04.4, installed with python 3.9.18, with torch==2.4.0+cu121, transformers==4.28.1, and opacus==1.4.0. More on enviroments can be found in the file named "environments.yml" provided.

A.2.5 Benchmarks

We evaluate on RoBERTa (355M) [1], which is a pretrained model on English language using a masked language modeling (MLM) objective, and OPT-1.3B and OPT-6.7B [2], which are parts of the a suite of decoder-only pre-trained transformers ranging from 125M to 175B parameters.

In the provided script, there are on six datasets for different classification tasks, including SST-2 and SST-5 [3] for sentiment analysis (i.e., determine if the given text is positive/negative), SNLI [4], MNLI [5], and RTE [6] for natural language inference (i.e., determine if the given premise and hypothesis are in the relationship of entailment/neutral/contradiction), and TREC [7] for topic assignment (i.e., determine which topic the given question falls under). For each dataset, we generate 512 samples for each class for training.

For larger models OPT-1.3B and OPT-6.7B that require more resources to fine-tune, we focus on one classification task SST-2 and one generation task SQuAD [8] (the fine-tuned model answers to a given question containing relevant contexts). For each dataset, we use 1000 samples for training.

A.3 Set-up

A.3.1 Installation

Please Use the file named "environments.yml" provided to install the environment using pip or conda. For testing RoBERTa (355M), go to folder roberta and install the dataset as specified in the readme file provided.

A.3.2 Basic Test

After installation of the environment and datasets, to test the functionality, run the following script in bash.

[1] https://huggingface.co/FacebookAI/roberta-large
[2] https://huggingface.co/facebook/opt-1.3b
[3] https://aclanthology.org/D13-1170/
[4] https://aclanthology.org/D15-1075/
[5] https://aclanthology.org/N18-1101/
[6] https://dl.acm.org/doi/10.1007/11736790_9
[7] https://aclanthology.org/L00-1018/
[8] https://aclanthology.org/D16-1264/

```
CUDA_VISIBLE_DEVICES=0 DPZERO_PRIVACY_EPS=2
    DP_SAMPLE_RATE=0.0208 STEP=500 \
SEED=42 NUM_DIRECTION=64
    RANDOM_DIRECTION_SEED=100 LR=6e-4 \
DPZERO_THRESHOLD=1 TASK="MNLI" bash examples
    /dpaggzo.sh
```

This script takes around 80 minutes to run on a RTX 4090 GPU and gives a test accuracy around 72%, which is much better than the utility of the original DPZO/DPZero under the same privacy constraint (around 65%, or refer to Table 2 on Page 9 of their original paper [9]).

Results on other datasets can be obtained by changing the "TASK" parameter. We also provided more examples in the "readme" file.

A.4 Evaluation workflow

A.4.1 Major Claims

We make two major claims.

(C1): *Our DP-AggZO can outperform the vanilla DPZero/DPZO in terms of test accuracy under the same privacy constraints.*

(C2): *Our DP-AggZO can sometimes outperform DP-AdamW in terms of test accuracy under the same privacy constraints.*

A.4.2 Experiments

C1-1: Reproducing DP-AggZO results for MNLI with privacy level $\varepsilon = 2$. You can run the DP-AggZO experiments for the MNLI task directly using the following commands:

```
CUDA_VISIBLE_DEVICES=0 DPZERO_PRIVACY_EPS=2
    DP_SAMPLE_RATE=0.0416 STEP=1000 \
SEED=42 NUM_DIRECTION=64
    RANDOM_DIRECTION_SEED=100 LR=8e-5 \
DPZERO_THRESHOLD=5 TASK="MNLI" bash examples
    /dpaggzo.sh
```

or

```
CUDA_VISIBLE_DEVICES=0 DPZERO_PRIVACY_EPS=2
    DP_SAMPLE_RATE=0.0416 STEP=1000 \
SEED=42 NUM_DIRECTION=64
    RANDOM_DIRECTION_SEED=100 LR=5e-4 \
DPZERO_THRESHOLD=1 TASK="MNLI" bash examples
    /dpaggzo.sh
```

Expected outcome:

- Compute time: 5 hours on RTX 4090, 14 hours on RTX A5000.
- Test accuracy is ∼74% on RTX A5000 or H20 GPU; ∼71% on RTX 4090 GPU.
- Significantly better than vanilla DPZO/DPZero under the same privacy level (see below).

C1-2: Reproducing DP-AggZO results with $K = 1$ (equivalent to DPZero/DPZO baseline) with privacy level $\varepsilon = 2$.

```
CUDA_VISIBLE_DEVICES=0 DPZERO_PRIVACY_EPS=2
    DP_SAMPLE_RATE=0.0416 STEP=5000 \
SEED=42 NUM_DIRECTION=1
    RANDOM_DIRECTION_SEED=100 LR=2e-6 \
DPZERO_THRESHOLD=200 TASK="MNLI" bash
    examples/dpaggzo.sh
```

Expected outcome:

- Compute time: 40 minutes on RTX 4090
- Test accuracy: 65% on RTX 4090 GPU, or can refer to the original paper: arxiv.org/pdf/2310.09639

Combining the results of **C1-1** and **C1-2**, we can verify that DP-AggZO outperforms the vanilla DPZO/DPZero under the same privacy constraints.

C2-1: Reproducing DP-AggZO results for MNLI with privacy level $\varepsilon = 0.5$. You can run the DP-AggZO experiments for the MNLI task directly using the following command:

```
CUDA_VISIBLE_DEVICES=0 DPZERO_PRIVACY_EPS
    =0.5 DP_SAMPLE_RATE=0.0416 STEP=500 \
SEED=42 NUM_DIRECTION=64
    RANDOM_DIRECTION_SEED=100 LR=2e-4 \
DPZERO_THRESHOLD=1 TASK="MNLI" bash examples
    /dpaggzo.sh
```

Expected outcome:

- Compute time: 3 hours on RTX 4090, 7 hours on RTX A5000.
- Accuracy: ∼63.5% on H20 and A5000 GPUs
- Better than DP-AdamW under the same privacy level (∼ 62%) (see below).

C2-2: Reproducing the result on DP-AdamW using the following fommand:

```
CUDA_VISIBLE_DEVICES=0 DP_SAMPLE_RATE=0.0416
     STEP=1000 SEED=42 LR=1e-4 \
DPSGD_THRESHOLD=10 DPSGD_PRIVACY_EPS=0.5
    DPSGD_PRIVACY_DELTA=1e-5 \
TASK="MNLI" bash examples/dpsgd.sh
```

The results from **C2-1** and **C2-2** demonstrate that **DP-AggZO can outperform DP-AdamW** the same privacy constraints, but the improvement is not as significant as that for DPZO/DPZero. We refer to the "readme" file provided for more examples.

To use this artifact beyond the models presented in

[9] https://arxiv.org/pdf/2310.09639

A.5 Notes on Reusability

To use this artifact beyond the models presented in this paper, we would recommend refactoring the code based on the latest implementation and algorithms of the model of interest. Some functions/libraries may become obsolete in the future while the general algorithmic idea of using multiple independent zeroth-order estimates to reduce clipping error could still apply.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.