PACZERO: PAC-Private Fine-Tuning of Language Models via Sign Quantization

Murat Bilgehan Ertan[*†]
CWI Amsterdam
Amsterdam, Netherlands
bilgehan@cwi.nl

Xiaochen Zhu[*]
MIT
Cambridge, MA, USA
xczhu@mit.edu

Phuong Ha Nguyen
eBay
San Jose, CA, USA
phuongha.ntu@gmail.com

Marten van Dijk[†]
CWI Amsterdam
Amsterdam, Netherlands
mevd@cwi.nl

Srinivas Devadas
MIT
Cambridge, MA, USA
devadas@mit.edu

Abstract

Differentially private (DP) zeroth-order (ZO) fine-tuning of large language models (LLMs) inherits DP's harsh utility-privacy trade-off. PAC Privacy, an instance-based notion, offers an alternative. It bounds the mutual information (MI) between the secret input and the output, and calibrates the noise to the stability of the output under the input distribution. We introduce PACZero, a family of ZO fine-tuning mechanisms under PAC Privacy. For stability, PACZero quantizes each update to a single bit, the sign of the ZO update aggregated over the secret subset of the training data. On many steps all candidate subsets agree on this sign, so the bit can be released at no cost in MI. On the remaining steps the bit passes through a noisy channel calibrated to a per-step MI budget. We study two channels, the standard additive Gaussian channel of PAC Privacy (PACZero-AGC) and a binary asymmetric channel with optimal bit flipping (PACZero-BAC), along with their zero-budget versions at $\mathrm{MI} \to 0$, PACZero-AGC$_0$ and PACZero-BAC$_0$. At zero MI, nothing released during training, including the fine-tuned model, carries any information about which secret subset was sampled, so no adversary inferring that subset can do better than its prior. In particular, the success rate of any membership inference attack (MIA) is bounded by 50% under a balanced prior. To our knowledge, this is the first such guarantee for LLM fine-tuning, and DP matches the MIA bound only at $\varepsilon = \delta = 0$ and infinite noise. We evaluate on SST-2 and SQuAD with models from 1.3B to 8B parameters, under LoRA and full fine-tuning. Both channels retain high utility under tight MI budgets, and even PACZero-BAC$_0$, which guarantees zero MI, stays within 2 points of non-private ZO fine-tuning with the same number of steps. To our knowledge no DP method achieves usable utility matching this MIA-resistance level.

1 Introduction

Zeroth-order (ZO) optimization has emerged as a memory-tractable alternative to first-order LLM supervised fine-tuning, replacing backpropagation with two forward passes per step [28]. When the training corpus is sensitive, trained weights leak individual records through membership inference attacks (MIA) and verbatim extraction [10, 11, 29, 31, 37]. The dominant defense is differential privacy (DP), which bounds the influence of any single training record on the released model through

[*]Equal contribution. [†]Affiliated with Vrije Universiteit Amsterdam.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

a parameter ε. Its standard instantiation, DP-SGD [1], clips per-record gradients at a fixed clipping threshold and adds Gaussian noise tracked across steps by the Rényi accountant [30], and it extends to fine-tuning [25, 52]. Applied to ZO, it yields DP zeroth-order methods [4, 26, 42, 56], which clip and noise only the per-step zeroth-order scalars, one per sampled direction. Every published DP zeroth-order method still inherits DP's harsh utility-privacy trade-off, because its underlying mechanism is unchanged. The noise is set by the worst-case sensitivity to a single record and composed over T steps. Utility loss at tight ε is a long-observed property of DP-SGD in iterative deep learning [2, 15, 16], so reaching a meaningful high-privacy regime for LLM fine-tuning remains a challenge.

PAC Privacy [46] bounds the mutual information (MI) between the secret and the release, and replaces DP's worst-case sensitivity-based noise with instance-based noise calibrated to the empirical covariance of the release. Under a known input distribution of the secret, it exploits the *stability* of the output under the specified distribution, adding less noise for more stable releases. Recent work enables adaptive composition and applies it to serve privacy-preserving *predictions* under adaptive and adversarial querying [60]. Sridhar et al. [41] give PAC-private gradient descent for smooth convex objectives, releasing a noisy gradient at every step. What remains open is private *iterative* fine-tuning of large models, such as LLMs, the setting in which DP-SGD operates, where an update is released to the adversary at every optimization step. At the scale of fine-tuning LLMs ($d \sim 10^9$, $T \sim 10^3$), however, calibrating the noise from the covariance of the output is prohibitively expensive, leaving large-scale private training out of reach of PAC Privacy.

The proposed PACZERO family addresses both problems, utility at high privacy and PAC Privacy for iterative LLM fine-tuning, by combining zeroth-order optimization [28] with sign quantization before the per-step release. Concretely, PACZERO follows the standard instantiation of PAC Privacy [6, 39, 41, 60] and adopts a subsampling-based input distribution used by PAC Privacy [46, 47] and assumed by MIAs [11, 37]. It constructs $M = 128$ random candidate subsets of the training data (i.e., the *universe*) such that each record in the universe is in exactly half of the candidate subsets and samples one of them uniformly at random as the secret S^*. At each step, it aggregates per-sample two-point zeroth-order loss differences over the M subsets, sign-quantizes each subset's mean response to $s_m \in \{-1, +1\}$, and privatizes and releases the sign produced by the secret subset (Figure 1).

Two structural consequences follow from quantizing to one bit. First, a sign is stable and therefore cheap to privatize under PAC Privacy. In particular, quantization creates frequent *unanimity*. On many steps all possible candidate subsets agree on the released bit. On these steps the bit reveals nothing about which candidate is the secret, so it can be released as is at zero conditional MI. On the remaining steps the bit is released with PAC Privacy through a noisy channel calibrated to a per-step MI budget. Because the channel's input is a single bit, its MI with the output under additive Gaussian noise is a one-dimensional integral that can be inverted numerically, allowing tight noise calibration. We consider an additive Gaussian channel on the bit, the standard choice in PAC Privacy (PACZERO-AGC), and an optimal binary asymmetric channel that flips the bit as rarely as the budget allows (PACZERO-BAC). Second, the MI budget can be pushed exactly to *zero*. We develop a zero-budget variant of each channel, a fair coin for the Gaussian channel (PACZERO-AGC_0) and the majority sign among candidate subsets for the binary channel (PACZERO-BAC_0). Under both variants, the release is independent of the secret on all steps, so $I(S^*; \tilde{Y}_{1:T}) = 0$ for every T, where $\tilde{Y}_{1:T}$ are the observations over T steps, and the fine-tuned model reveals *nothing* about the secret subset and achieves perfect resistance against MIAs inferring the membership of a record in the secret subset. That is, the MIA posterior success rate equals its prior success rate of $1/2$ for every record in the universe. In summary, we make three contributions:

- **The PACZERO family of PAC-private ZO mechanisms.** We introduce PACZERO[2], which brings PAC Privacy to iterative LLM fine-tuning. One accounting argument covers all four variants that differ in the MI budget and the noise channel (§3). We evaluate all four on SST-2 and SQuAD with OPT-1.3B, OPT-6.7B, Qwen2.5-7B and Llama-3.1-8B, under both LoRA and full fine-tuning (Table 1 and Appendix F).
- **Tighter and distortion-optimal calibration for a one-bit input.** For the additive Gaussian channel, we propose tighter noise calibration for the one-bit channel input, which sets the noise scale by numerically inverting the exact MI instead of a variance-based upper bound used in canonical PAC Privacy mechanisms (§3.2). The binary asymmetric channel goes

[2] `https://github.com/bilgehanertan/paczero-bac`

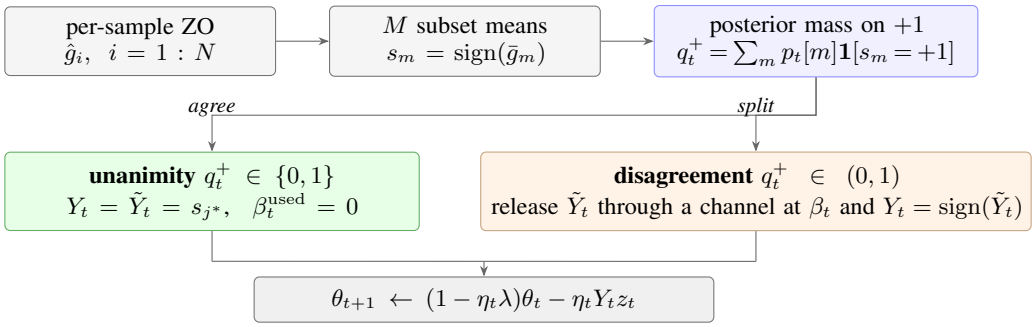

Figure 1: **The PACZERO per-step mechanism.** Per-sample finite-difference scalars are aggregated over $M = 128$ random subsets, sign-quantized to $s_m \in \{-1, +1\}$. On unanimity, the secret subset's sign is released as is at zero conditional MI. Otherwise it passes through a calibrated channel.

further and achieves the minimum distortion under a fixed MI budget (§3.3). Empirically, PACZERO-BAC matches or exceeds the utility of PACZERO-AGC, test accuracy on SST-2 and F1 on SQuAD, in almost every setting and budget.

- **Usable fine-tuning at zero MI.** To our knowledge, PACZERO-BAC_0 and PACZERO-AGC_0 give the first guarantee of zero MI between the secret subset and everything released during iterative LLM fine-tuning. Moreover, PACZERO-BAC_0 stays within 2 points of non-private ZO on all sixteen settings (Table 1 and Appendix F). DP reaches the same MIA bound only at $\varepsilon = \delta = 0$, which a noise-adding mechanism attains only with infinite noise (§3.4).

2 Background

Two-point zeroth-order gradient estimation. For a model with parameters $\theta \in \mathbb{R}^d$ and a per-sample loss $\ell_i(\theta)$, the two-point zeroth-order estimator [28, 32] samples a single direction $z \sim \mathcal{N}(0, I_d)$ and computes the per-sample finite-difference scalar

$$\hat{g}_i = \frac{\ell_i(\theta + \mu z) - \ell_i(\theta - \mu z)}{2\mu} = \langle \nabla \ell_i(\theta), z \rangle + O(\mu^2), \tag{1}$$

where $\mu > 0$ is the perturbation scale [28]. The parameter update is $\theta \leftarrow \theta - \eta \cdot \hat{g} \cdot z$, where $\hat{g}$ is obtained by clipping and averaging the $\hat{g}_i$ over the batch [4, 56]. The structural property that matters here, exploited by all DP zeroth-order methods [4, 42, 56], is that z is sampled *independently* of the data and is therefore public. The only data-dependent quantity per step is the scalar $\hat{g}$, so the per-step noise scale is dimension-independent.

Definition 1 (PAC Privacy [46])**.** Given a data generating distribution $\mathcal{D}$ over possible datasets in $\mathcal{X}$, and a binary-valued attack success criterion $\rho : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$, we say a mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is $(\delta, \rho, \mathcal{D})$-PAC private if for every adversary $\mathcal{A} : \mathcal{Y} \to \mathcal{X}$, who knows $\mathcal{D}$ and $\mathcal{M}$, the attack success rate, as measured by ρ, is at most $1 - \delta$ when the secret dataset X is drawn from $\mathcal{D}$ and the adversary observes $Y = \mathcal{M}(X)$, $1 - \delta_{\mathcal{A}} := \Pr_{X \sim \mathcal{D}, Y \leftarrow \mathcal{M}(X)}[\rho(\mathcal{A}(Y), X) = 1] \le 1 - \delta$.

PAC Privacy [46] provides an instance-based framework for measuring and bounding information leakage about a secret dataset from a mechanism's output. Definition 1 assumes a strong and informed adversary who knows the input distribution $\mathcal{D}$, the mechanism $\mathcal{M}$, and has unbounded compute resources. Privacy guarantees for $X \sim \mathcal{D}$ established under a strong adversary model naturally extend to realistic adversaries without such knowledge and/or computational power.

Given $\mathcal{D}$ and ρ, an informed adversary has a *prior success rate* even before observing any outputs, denoted as $1 - \delta_0 := \max_Q \Pr_{X \sim \mathcal{D}, \hat{X} \sim Q}[\rho(\hat{X}, X) = 1]$. A fundamental result of PAC Privacy [46] bounds the *posterior advantage*, $\Delta_{\text{KL}}\delta_{\mathcal{A}}$, of an adversary, the gain of its success rate over its prior, by mutual information (MI). For every criterion ρ and every adversary $\mathcal{A}$ of Definition 1 with $\delta_{\mathcal{A}} \le \delta_0$,

$$\Delta_{\text{KL}}\delta_{\mathcal{A}} := \text{KL}(1 - \delta_{\mathcal{A}} \,\|\, 1 - \delta_0) \le I(X; \mathcal{M}(X)), \tag{2}$$

where $\mathrm{KL}(p\|q) := p\ln(p/q) + (1-p)\ln((1-p)/(1-q))$ is the binary KL divergence. We note that the MI bound depends on neither ρ nor $\mathcal{A}$, so (2) is attack-agnostic. Given $\mathcal{D}$ and $\mathcal{M}$, MI bounds the posterior advantage of every attack. Hence, one can specify and enforce a total MI budget, and derive concrete bounds on success rate given the prior success rate of an informed adversary under ρ.

When the mechanism interacts with a persistent secret over T rounds, releasing outputs $Y_1, Y_2, \ldots, Y_T$, the chain rule of mutual information gives

$$I(X;\, Y_{1:T}) \;=\; \sum_{t=1}^{T} I(X;\, Y_t \mid Y_{1:t-1}). \tag{3}$$

To bound each term, Zhu et al. [60] maintain a posterior distribution $P_{X|Y_{1:t-1}=y_{1:t-1}}$ over the secret space, which represents the strong adversary's optimal belief given the observed transcript so far. At step t, the mechanism calibrates its randomness to ensure $I(X; Y_t \mid Y_{1:t-1} = y_{1:t-1}) \le \beta_t$ for every realization $y_{1:t-1}$. This reduces privacy accounting to per-step MI control, which we exploit in §3.

Instantiation and MIA guarantees. Following the standard instantiation of PAC Privacy [6, 39, 41, 60], which builds on the subsampling-based input distributions used by Xiao and Devadas [46], Xiao et al. [47], we start from a *universe* of N training examples $U = \{x_1, \ldots, x_N\}$ and construct M *candidate* subsets $S_1, \ldots, S_M \subset U$ such that every example lies in exactly $M/2$ of them, chosen uniformly at random and independently for each example as in Zhu et al. [60]. We fix the subsets before training and set $\mathcal{D} = \mathrm{Unif}\{S_1, \ldots, S_M\}$. Drawing the secret dataset from $\mathcal{D}$ then amounts to sampling a secret index $j^* \sim \mathrm{Unif}[M]$ and training on $S^* = S_{j^*}$ throughout the run. The finite support of $\mathcal{D}$ makes adaptive composition tractable, because the posterior over the secret index j^* is a discrete distribution over $[M]$.

In a membership inference attack (MIA) on a record $x_i \in U$, the adversary infers whether $x_i \in S^*$. Since x_i lies in half of the M equally likely candidates, the prior success rate is $1 - \delta_0 = 1/2$. A bound $I(S^*; \tilde{Y}_{1:T}) \le \mathrm{MI}_{\mathrm{tot}}$ on the total MI then caps the MIA success rate, through (2), at the largest $1 - \delta_{\mathcal{A}}$ with $\mathrm{KL}(1 - \delta_{\mathcal{A}} \,\|\, 1/2) \le \mathrm{MI}_{\mathrm{tot}}$, which we compute numerically. For example, $\mathrm{MI}_{\mathrm{tot}} = 1/4$ nats limits MIA success to $\approx 84\%$ and lowering it to $1/128$ tightens the guarantee to around 56%. $\mathrm{MI}_{\mathrm{tot}} = 0$ achieves perfect MIA resistance and bounds MIA success by its prior of 50%.

For comparison, in the standard membership game on neighboring datasets, (ε, δ)-DP bounds MIA success by $1-(1-\delta)/(1+e^{\varepsilon})$ [21, 39], which reaches the prior of 50% only at $\varepsilon = \delta = 0$. Throughout this paper we report the privacy of PACZERO in its native unit (MI in nats), and where useful we label an MI budget with the DP ε that yields a matching MIA upper bound at $\delta = 10^{-5}$. For example, $(2, 10^{-5})$-DP caps MIA success at 88.1%, matching the MIA bound of $\mathrm{MI}_{\mathrm{tot}} = 0.33$ nats, so we label 0.33 nats with $\varepsilon = 2$.

3 The PACZERO Mechanisms

The per-step procedure of PACZERO is summarized in Algorithm 1. At step t we compute per-sample finite-difference scalars $\hat{g}_i$ as in (1) for all N records of the universe U, aggregate them to subset means $\bar{g}_m$ over the M subsets of §2, and sign-quantize them to $s_m \in \{-1, +1\}$. The mechanism then privatizes the secret subset's sign s_{j^*} and releases a potentially noised value $\tilde{Y}_t$, whose sign Y_t updates parameters via $\theta_{t+1} = \theta_t - \eta_t Y_t z_t$. A step therefore reads the whole universe, at a cost of $2N$ forward passes. A step of PACZERO-AGC or PACZERO-AGC$_0$ takes about the time and memory of a non-private ZO step over the same universe, so the privacy layer itself adds little cost. The cost of a step grows linearly with N and shows no trend in the number of subsets M (Appendix V). The release is deterministic on the *unanimity* branch (when the posterior $p_t \in \Delta^{M-1}$ over the secret index j^* places all mass on subsets that agree on s_m) and passes through a noisy *channel* on the *disagreement* branch otherwise. We propose four variants under a unified privacy argument for any calibrated channel (§3.1), which differ in the channel (§3.2 and §3.3) and the MI budget (§3.4).

3.1 Per-step accounting and adaptive budget allocation

On a *unanimity* step, every candidate m with positive posterior mass, i.e., $p_t[m] > 0$, produces the same sign s_m. The bit s_{j^*} derived from the secret subset is then the same as every other candidate's, so releasing it reveals nothing about which candidate is the secret. Under PAC Privacy, the mechanism releases $\tilde{Y}_t = s_{j^*}$ and the posterior does not change, charging $\beta_t^{\mathrm{used}} = 0$ of the MI budget.

Algorithm 1 PACZERO per-step update at step t.

Require: parameters θ_t, posterior $p_t \in \Delta^{M-1}$ over the secret index j^*, per-step MI budget β_t

sample $z_t \sim \mathcal{N}(0, I_d)$ ▷ public ZO direction

for $i = 1, \ldots, N$ **do**

  $\hat{g}_i \leftarrow [\ell_i(\theta_t + \mu z_t) - \ell_i(\theta_t - \mu z_t)]/(2\mu)$ ▷ per-sample finite difference

  $\hat{g}_i \leftarrow \mathrm{sign}(\hat{g}_i) \cdot \min(|\hat{g}_i|, c)$ ▷ per-sample clipping, not privacy-relevant (Appendix C)

end for

$\bar{g}_m \leftarrow \frac{1}{|S_m|} \sum_{i \in S_m} \hat{g}_i, \quad s_m \leftarrow \mathrm{sign}(\bar{g}_m) \quad$ for $m = 1, \ldots, M$

$q_t^+ \leftarrow \sum_m p_t[m] \cdot \mathbf{1}[s_m = +1]$ ▷ posterior probability that $s_{j^*} = +1$

if $q_t^+ \in \{0, 1\}$ **then** ▷ unanimity branch

  $\tilde{Y}_t \leftarrow s_{j^*}, \quad Y_t \leftarrow \mathrm{sign}(\tilde{Y}_t), \quad \beta_t^{\mathrm{used}} \leftarrow 0, \quad p_{t+1} \leftarrow p_t$

else ▷ disagreement branch, release through a calibrated channel

  $\mathcal{C}_t \leftarrow \mathrm{Calibrate}(q_t^+, \beta_t)$ ▷ AGC (§3.2) or BAC (§3.3)

  $\tilde{Y}_t \leftarrow \mathcal{C}_t(s_{j^*}), \quad Y_t \leftarrow \mathrm{sign}(\tilde{Y}_t), \quad \beta_t^{\mathrm{used}} \leftarrow \min(\beta_t, h(q_t^+))$

  $p_{t+1}[m] \propto p_t[m] \cdot L_t(\tilde{Y}_t \mid s_m)$ ▷ Bayes' rule (4)

end if

$\theta_{t+1} \leftarrow (1 - \eta_t \lambda)\theta_t - \eta_t \cdot Y_t \cdot z_t$ ▷ $\lambda = 0$ in all experiments.

On a *disagreement* step the candidates do not all agree, so releasing s_{j^*} directly would reveal something about which secret subset S_{j^*} is used. The bit is therefore passed through a noisy channel $\mathcal{C}_t$, set so that, when its randomness is drawn independently, $\mathcal{C}_t(s_{j^*})$ carries at most β_t nats of information about s_{j^*}. This is the calibration step of PAC Privacy, which chooses the noise from algorithmic stability of the output, i.e., how much the noiseless output varies with the secret. Here, the noiseless output is the single bit s_{j^*}, whose distribution under the posterior p_t is specified by $q_t^+ = \sum_m p_t[m]\, \mathbf{1}[s_m = +1]$, the probability that s_{j^*} equals $+1$. Therefore, the mechanism chooses a channel $\mathcal{C}_t$ from q_t^+ and β_t alone. Formally, the channel must be calibrated in the following sense.

Definition 2 (Calibrated channel)**.** Let $q \in (0, 1)$ and $\beta \geq 0$. A channel $\mathcal{C}$ with input alphabet $\{-1, +1\}$ is *calibrated at* (q, β) if $I(\xi; \mathcal{C}(\xi)) \leq \beta$ for an input bit ξ with $P(\xi = +1) = q$.

After calibrating the channel, the mechanism releases $\tilde{Y}_t = \mathcal{C}_t(s_{j^*})$ and updates the posterior by Bayes' rule to enable adaptive composition [60]. Each candidate m is weighted by how likely the channel is to produce the observed output when its input is s_m:

$$p_{t+1}[m] \;\propto\; p_t[m]\, L_t(\tilde{Y}_t \mid s_m), \qquad \text{where} \quad L_t(y \mid s) := \Pr[\mathcal{C}_t(s) = y], \tag{4}$$

is the probability (or density for continuous outputs), over the channel's own randomness, that $\mathcal{C}_t$ outputs y on input s. The optimizer uses only the sign $Y_t = \mathrm{sign}(\tilde{Y}_t)$ of the noisy release, updating the model by $\pm\eta_t z_t$. A calibrated disagreement step costs at most $\beta_t^{\mathrm{used}} = \min\{\beta_t, h(q_t^+)\}$ nats, since MI cannot exceed the input's binary entropy $h(q_t^+)$.

Adaptive budget allocation. Given a total MI budget $\mathrm{MI}_{\mathrm{tot}}$ and step budget T, the uniform budget allocation $\beta_t = \mathrm{MI}_{\mathrm{tot}}/T$ [60] is loose, because a step that takes the unanimity branch leaves β_t unspent and uniform allocation cannot recover it later. PACZERO uses

$$\beta_t \;=\; \max\Big(0,\, \mathrm{MI}_{\mathrm{tot}} - \mathrm{MI}_{\mathrm{used}}^{(t-1)}\Big) \;/\; (T - t + 1), \tag{5}$$

where $\mathrm{MI}_{\mathrm{used}}^{(t-1)} = \sum_{\tau < t} \beta_\tau^{\mathrm{used}}$ is the budget spent before step t. Dividing the remaining budget over the remaining number of steps keeps $\sum_t \beta_t^{\mathrm{used}} \leq \mathrm{MI}_{\mathrm{tot}}$ on every realization.

Privacy guarantee. Composing these per-step MI bounds over the run by the chain rule (3) and the adaptive composition of Zhu et al. [60] gives the end-to-end guarantee, which holds for any calibrated channel, and so for PACZERO-AGC and PACZERO-BAC of §3.2 and §3.3 (proof in Appendix B.1).

Theorem 3 (PACZERO privacy)**.** *Iteratively applying Algorithm 1 for T rounds with a total mutual information budget of $\mathrm{MI}_{\mathrm{tot}}$ under adaptive budget allocation* (5)*, with the disagreement branch releasing through a channel calibrated at (q_t^+, β_t), satisfies*

$$I(S^*;\, \tilde{Y}_{1:T}) \;\leq\; \mathrm{MI}_{\mathrm{tot}}. \tag{6}$$

The same bound holds for the update signs $Y_{1:T}$, which are computed from $\tilde{Y}_{1:T}$. Furthermore, end-to-end fine-tuning satisfies $I(S^*;\theta_{1:T+1}) \le \mathrm{MI}_{\mathrm{tot}}$.

Corollary 4 (MIA guarantee)**.** *For an MIA on any record $x_i \in U$, the informed adversary's prior success is $1-\delta_0 = 1/2$ (§2), and* (2) *yields $\mathrm{KL}\big(1-\delta_{\mathcal{A}} \,\big\|\, 1/2\big) \le \mathrm{MI}_{\mathrm{tot}}$, and the bound is tight (Proposition 10).*

What the adversary knows. As in Definition 1, we establish the guarantee assuming an informed adversary that knows everything except the sampled index j^* and the channels' random draws. This includes the universe U and the candidate subsets, the initial parameters, the hyperparameters, the directions z_t and the algorithm itself. The adversary observes every release $\tilde{Y}_t$, and therefore every update sign Y_t and every intermediate model. It can then simulate the run and compute, at every step, the sign s_m of every candidate, the posterior p_t, q_t^+, the budget β_t and the channel (Lemma 7). None of these quantities is secret to such a strong adversary, and PAC Privacy, via Theorem 3, protects $S^* = S_{j^*}$, the identity of the secret subset, from it.

3.2 The additive Gaussian channel (PACZERO-AGC)

We first follow the standard PAC Privacy approach [39, 41, 46, 47, 60] with an additive Gaussian channel (PACZERO-AGC). On a disagreement step, it releases $\tilde{Y}_t = s_{j^*} + \mathcal{N}(0,\sigma_t^2)$, where σ_t is set so that $\tilde{Y}_t$ carries at most β_t nats of information about s_{j^*}. These mechanisms invert a variance-based upper bound on MI, which is tight only when the noiseless output is itself Gaussian [58]. Because our channel input is a single bit with a known distribution, we can instead use the exact MI for tighter calibration. For $\xi \in \{-1,+1\}$ with $P(\xi=+1)=q$ and $\tilde{Y} = \xi + \nu$, $\nu \sim \mathcal{N}(0,\sigma^2)$, the MI is

$$I_q(\sigma) := I(\xi;\tilde{Y}) = \sum_{s\in\{-1,+1\}} P(\xi{=}s) \int \phi_\sigma(y-s)\, \log \frac{\phi_\sigma(y-s)}{q\,\phi_\sigma(y-1) + (1-q)\,\phi_\sigma(y+1)}\, dy, \quad (7)$$

where ϕ_σ is the Gaussian density with standard deviation σ. This is a one-dimensional integral, and decreases as σ grows, so we can invert it numerically to set $\sigma_t = I^{-1}_{q_t^+}(\beta_t)$ (Appendix C). The resulting channel is calibrated at (q_t^+, β_t) (Lemma 9), so Theorem 3 applies to PACZERO-AGC.

3.3 The distortion-optimal binary asymmetric channel (PACZERO-BAC)

Even when calibrated at the exact MI, the Gaussian channel is still loose for PACZERO: it spends the MI budget on the information that $\tilde{Y}_t \in \mathbb{R}$ carries about s_{j^*}, but the optimizer uses only its sign Y_t, which carries strictly less information. This motivates a binary channel that directly releases the bit by itself. We choose, among such channels, the one that achieves the minimal flip rate.

A general binary channel, which takes in $\xi \in \{-1,+1\}$ and emits $\tilde{Y} \in \{-1,+1\}$, can be specified by its flip probabilities $a = \Pr[\tilde{Y}{=}{+}1 \mid \xi{=}{-}1]$ and $b = \Pr[\tilde{Y}{=}{-}1 \mid \xi{=}{+}1]$. When ξ equals $+1$ with probability p, its distortion is $D = \Pr[\tilde{Y} \neq \xi] = (1-p)a + pb$. Choosing (a,b) to minimize D subject to $I(\xi;\tilde{Y}) \le \beta$ is the binary rate-distortion problem [13, §10.3.1], whose solution is

$$D^* = h^{-1}\big(\max\{h(p){-}\beta, 0\}\big), \quad \pi \;=\; \frac{p-D^*}{1-2D^*}, \quad a^* \;=\; \frac{\pi\, D^*}{1-p}, \quad b^* \;=\; \frac{(1-\pi)\, D^*}{p}. \quad (8)$$

Here h^{-1} is the inverse of the binary entropy function on $[0,1/2]$, and π is the probability that the channel emits $+1$. PACZERO-BAC sets its channel by evaluating (8) with $p = q_t^+$ and $\beta = \beta_t$. Its output is already a sign, so $Y_t = \tilde{Y}_t$, and none of the budget is spent on information that the optimizer discards. The identity $\pi - 1/2 = (p-1/2)/(1-2D^*)$ shows that the channel always leans toward the sign that holds more posterior mass, at least as strongly as p does. By (8), $I(\xi;\tilde{Y}) \le \beta_t$, so the channel is calibrated at (q_t^+, β_t) (Lemma 9) and Theorem 3 applies to PACZERO-BAC.

3.4 Releases at zero mutual information (PACZERO-AGC$_0$ and PACZERO-BAC$_0$)

Taking β_t to zero in either channel gives a release that ignores its input s_{j^*}. We use these limits as two further variants, PACZERO-AGC$_0$ and PACZERO-BAC$_0$, which operate at zero mutual information.

- For the additive Gaussian channel, as $\beta_t \to 0$ the noise grows without bound, and the sign of the release becomes a fair coin independent of s_{j^*}. PACZERO-AGC$_0$ releases that coin directly on every disagreement step. That is, $\tilde{Y}_t$ is independently drawn from $\mathrm{Unif}\{-1,+1\}$.
- For the binary asymmetric channel, with $\beta_t = 0$, (8) gives $D^* = \min(q_t^+, 1-q_t^+)$. The resulting channel ignores its input and always emits the sign that holds more posterior mass, $+1$ for $q_t^+ > 1/2$ and -1 for $q_t^+ < 1/2$, regardless of what s_{j^*} is. This is equivalent to releasing the majority sign $\tilde{Y}_t = \mathrm{sign}\big(\sum_m p_t[m]\, s_m\big)$, which also covers the case of unanimity. At $q_t^+ = 1/2$, (8) is undefined, and we emit $+1$, following the convention of $\mathrm{sign}(0) = +1$ (Appendix C). We call this variant PACZERO-BAC$_0$.

In both variants, no release depends on which candidate is the secret, on unanimity and disagreement steps alike. Hence, every candidate has the same likelihood in (4), and the posterior p_t stays uniform for the whole run. The two variants differ in what they release on a disagreement step. A disagreement step of PACZERO-AGC$_0$ moves the parameters by $+\eta_t z_t$ or $-\eta_t z_t$ with equal probability, so its progress comes only from unanimity steps, whereas a disagreement step of PACZERO-BAC$_0$ still moves them in the direction that most candidates favor.

Theorem 3 with $\mathrm{MI}_{\mathrm{tot}} = 0$ covers both variants, as stated in Theorem 5 (proof in Appendix B.3).

Theorem 5 (Zero MI)**.** PACZERO-AGC$_0$ *and* PACZERO-BAC$_0$ *satisfy* $I(S^*; \tilde{Y}_{1:T}) = 0$ *for every* T*. Hence no adversary infers whether a record of* U *is in* S^* *with probability greater than* $1/2$*.*

Remark 6 (Scope of the zero-MI guarantee)**.** Theorem 5 shows that the run reveals nothing about *which* candidate subset was sampled. It does not say that the release reveals nothing about the records of U, which the informed adversary already knows. What a release reveals about U is outside the guarantee, and we discuss it in §6. While our guarantee bounds MIA success by 50%, which DP matches at $\varepsilon = \delta = 0$, it is not equivalent to one. DP at $\varepsilon = \delta = 0$ is much stronger and would require the same output distribution on every pair of neighboring datasets, which a noise-adding mechanism achieves only with infinite noise.

4 Experiments

We evaluate PACZERO on the two LLM tasks reported by the strongest DP zeroth-order baseline, DP-AggZO [4], namely SST-2 (binary sentiment classification) [38] and SQuAD (extractive QA, F1) [35]. Each task is run on four models spanning three families and a factor of six in scale, OPT-1.3B and OPT-6.7B [57], Qwen2.5-7B [49] and Llama-3.1-8B [27]. Every model runs in two parameter tracks, LoRA r=8, α=16 [19] and full-parameter fine-tuning (FT). The main text reports OPT-6.7B and Llama-3.1-8B. OPT-1.3B and Qwen2.5-7B carry the same four mechanisms at the same budgets and are in Appendix F. Two tasks, four models and two tracks give sixteen settings in total.

The section supports three claims. (i) Utility barely depends on the MI budget, down to $I(S^*; \tilde{Y}_{1:T}) = 0$. On SST-2 FT OPT-1.3B, PACZERO-AGC test accuracy is flat across nearly four decades of MI budget (Table 7 in Appendix I), and at $\mathrm{MI} = 0$ PACZERO-BAC$_0$ stays within 2.0 points of the non-private ZO reference at the same step budget on all sixteen settings (Table 1 and Appendix F). (ii) The binary asymmetric channel gives higher utility than the Gaussian channel. PACZERO-BAC is at or above PACZERO-AGC on every setting at $\mathrm{MI} = 0.33$, and PACZERO-BAC$_0$ is above PACZERO-AGC$_0$ on fourteen of the sixteen (§4.2). (iii) Published DP zeroth-order accountants do not reach this MIA resistance at finite per-step noise. The DP MIA bound equals that of $\mathrm{MI} = 0$ only at $\varepsilon = \delta = 0$, and in our reproduction on SST-2 FT OPT-1.3B every DP run at $\varepsilon \le 1$, whose MIA bound is weaker, stays below both $\mathrm{MI} = 0$ variants (Table 9 in Appendix L).

4.1 Setup

We follow the DP-AggZO protocol [4] on the splits its released code inherits from MeZO [28], using a universe of $N = 1000$ records, a 500-record dev split, $\delta = 10^{-5}$, and weight decay $\lambda = 0$. Each run trains for $T = 1000$ steps (2000 on SST-2 LoRA). The dev split comes from the same seeded permutation as U and is disjoint from it, so no dev record belongs to any candidate subset. Dev-best checkpoint selection is a function of the released transcript and the dev split, so by the data-processing inequality it cannot increase the mutual information and the reported MI accounts for the full T-step

Table 1: **SST-2 and SQuAD main results (%).** Mean $\pm$ std over $n=3$ seeds (per-seed values in Appendix X). An entry shown without a spread is single-seed. OPT DP rows are as reported by Bao et al. [4]. † marks baseline and reference entries we ran ourselves, and every PACZERO entry is our own. Table 10 in Appendix M gives the hyperparameters and forward-pass count of every row. **Bold** marks the better of the two $\mathrm{MI}=0$ variants in each column and does not mark the column maximum. A DP ε and an MI budget in the same block approximately share the MIA upper bound (§2). Appendix F carries the four PACZERO variants on all four models. Appendix E describes the non-private ZO reference.

			SST-2 (acc.)		SQuAD (F1)	
Track	Privacy	Method	OPT-6.7B	Llama-3.1-8B	OPT-6.7B	Llama-3.1-8B
FT	DP $\varepsilon=2$ MI$=0.33$	DPZero $K=1$	92.7	92.89†	78.5	88.41†
		DP-AggZO $K=16$	93.8	93.81†	82.9	87.96†
		PACZERO-AGC	$91.09_{\pm 1.03}$	$91.44_{\pm 1.08}$	$75.65_{\pm 1.07}$	$87.74_{\pm 0.26}$
		PACZERO-BAC	$91.44_{\pm 0.24}$	$93.12_{\pm 0.34}$	$76.69_{\pm 1.01}$	$87.78_{\pm 1.33}$
	DP $\varepsilon=6$ MI$=0.68$	DPZero $K=1$	92.9	92.78†	80.1	88.50†
		DP-AggZO $K=16$	94.6	92.78†	83.3	87.53†
		PACZERO-AGC	$91.28_{\pm 1.09}$	$91.63_{\pm 0.75}$	$75.45_{\pm 1.61}$	$88.07_{\pm 0.96}$
		PACZERO-BAC	$92.16_{\pm 0.48}$	$93.12_{\pm 0.50}$	$75.84_{\pm 0.78}$	$87.74_{\pm 1.19}$
	MI$=0$	PACZERO-AGC$_0$	$89.87_{\pm 0.63}$	$92.20_{\pm 0.61}$	$74.83_{\pm 1.32}$	$87.39_{\pm 1.58}$
		PACZERO-BAC$_0$	$\mathbf{91.63_{\pm 0.30}}$	$\mathbf{92.78_{\pm 0.23}}$	$\mathbf{77.10_{\pm 1.39}}$	$\mathbf{87.80_{\pm 1.12}}$
	none	non-private ZO reference†	93.58	$92.43_{\pm 0.61}$	$75.20_{\pm 0.91}$	89.01
LoRA	MI$=0.33$	PACZERO-AGC	$92.62_{\pm 0.33}$	$88.80_{\pm 2.66}$	$73.93_{\pm 1.51}$	$88.19_{\pm 0.36}$
		PACZERO-BAC	$93.00_{\pm 0.91}$	$90.25_{\pm 0.75}$	$75.44_{\pm 1.39}$	$88.89_{\pm 1.69}$
	MI$=0.68$	PACZERO-AGC	$92.39_{\pm 0.83}$	$90.52_{\pm 2.36}$	$73.30_{\pm 1.49}$	$88.61_{\pm 1.41}$
		PACZERO-BAC	$93.43_{\pm 0.57}$	$90.79_{\pm 0.70}$	$76.10_{\pm 2.56}$	$88.93_{\pm 1.59}$
	MI$=0$	PACZERO-AGC$_0$	$\mathbf{93.62_{\pm 0.13}}$	$87.69_{\pm 0.63}$	$72.12_{\pm 1.81}$	$88.05_{\pm 1.22}$
		PACZERO-BAC$_0$	$93.31_{\pm 0.43}$	$\mathbf{89.37_{\pm 2.84}}$	$\mathbf{75.44_{\pm 0.77}}$	$\mathbf{88.20_{\pm 2.21}}$
	none	non-private ZO reference†	92.78	91.17	75.73	86.98

run. The budgets in Table 1 are total $\mathrm{MI} \in \{0.33, 0.68\}$ nats, approximately matching, through (2), the MIA upper bound of DP $\varepsilon \in \{2, 6\}$ at $\delta = 10^{-5}$ under the conversion of Zhu et al. [60]. Appendix F adds 0.01 and 0.11 nats. All four variants run at $M = 128$ subsets, held fixed throughout. Any even M gives the per-record prior of $1/2$, and M sets how many subsets must agree for a step to cost nothing. In each setting the four variants share one learning rate, one clipping threshold c and one step budget, so within a track the PACZERO rows of a column of Table 1 differ only in the mechanism and its budget. Appendix E summarizes the protocol, Table 11 in Appendix N gives the hyperparameters of every setting, and Appendix A lists the sweeps, ablations, cost measurements and compute.

We compare against two DP zeroth-order baselines. Our primary comparator is DPZero [56], which, like PACZERO, draws a single perturbation direction per step ($K = 1$). We also report DP-AggZO [4] at $K = 16$, whose per-example forward-pass count approximately matches that of PACZERO (Table 10), as a reference for methods that aggregate $K > 1$ directions per step. The OPT-6.7B DP rows of Table 1 are taken from Table 2 of Bao et al. [4], where each entry is the best over that paper's sampling-rate, learning-rate and clipping-threshold grids while every PACZERO entry is selected on dev (implementation parity checked at OPT-1.3B in Appendix K). Since Bao et al. [4] do not report DP-LoRA on OPT, all comparisons with DP are in the FT rows. On both tasks we run both DP baselines in-house on Llama-3.1-8B at $\varepsilon \in \{2, 6\}$, at a learning rate selected by a sweep at $\varepsilon = 2$ and used at both ε, and mark those entries †. The non-private ZO row is our own zeroth-order fine-tuning on all of U with no release mechanism, at the learning rate its own dev sweep selects (Table 14). It runs at $c = 1000$, the loosest clipping threshold of the grid. It measures what zeroth-order fine-tuning reaches at the same step budget without privacy.

4.2 Main results

Utility depends little on the MI budget. On SST-2 FT OPT-1.3B we hold the learning rate, clipping threshold and step budget fixed and sweep the total MI budget of PACZERO-AGC across nearly four decades, from 10^{-4} to 0.68 nats. Test accuracy stays between 88.30 and 90.48%, and the unanimity fraction, the share of steps that spend no budget, stays between 52.4 and 56.0% (Table 7 and Figure 2 in Appendix I). At $\mathrm{MI} = 0.33$, the Gaussian release flips its sign almost as often as a fair coin on disagreement steps (Appendix E). Most of the optimization signal at this budget therefore comes from unanimity steps. Across settings, the unanimity fraction ranges from 12.8 to 73.7% (Table 4 in Appendix G). Appendix J repeats the sweep on the LoRA track. In every setting, PACZERO-BAC moves by at most 1.76 points between $\mathrm{MI} = 0$ and $\mathrm{MI} = 0.68$ (Appendix F).

The binary asymmetric channel improves on the Gaussian channel. At $\mathrm{MI} = 0.33$, PACZERO-BAC is at or above PACZERO-AGC on every setting, by up to 3.44 points on SST-2 FT for Qwen2.5-7B. At $\mathrm{MI} = 0.68$ it is at or above PACZERO-AGC on thirteen of the sixteen settings and trails on the other three by at most 0.45 points (Appendix F).

The same holds at zero budget. At $I(S^*; \tilde{Y}_{1:T}) = 0$, PACZERO-BAC$_0$ improves on PACZERO-AGC$_0$ on fourteen of the sixteen settings, by up to 3.86 F1 on SQuAD LoRA for OPT-1.3B, and trails it by at most 0.31 points on the other two, both SST-2 LoRA on OPT. The two variants differ only in what the disagreement branch releases (§3.4) and both satisfy Theorem 5, so the gap measures what the majority of the subset signs buys over an unbiased coin at identical privacy. On six of the sixteen settings PACZERO-BAC$_0$ at $\mathrm{MI} = 0$ is above PACZERO-BAC at $\mathrm{MI} = 0.68$. At $\mathrm{MI} = 0$, PACZERO-BAC$_0$ is at most 1.95 points below the non-private ZO reference on all sixteen settings and above it on seven (Appendix F).

We also verify Theorem 5 empirically. Across all settings, nine black-box membership-inference attacks, among them RMIA and LiRA, separate the records of S^* from the rest of U no more often than chance. Their median AUC lies between 0.489 and 0.507 (Appendix H).

We now turn to DP. The comparisons in Table 1 set PACZERO-BAC$_0$ at $I(S^*; \tilde{Y}_{1:T}) = 0$ against DP at $\varepsilon = 2$, whose MIA bound is much weaker (§2). Against DPZero, the primary comparator, PACZERO-BAC$_0$ is at most 1.40 points below on all four FT settings. On Llama-3.1-8B FT, PACZERO-BAC$_0$ is within 1.03 points of the strongest DP baseline on both tasks. On OPT-6.7B FT, DP-AggZO $K = 16$ leads by 2.2 points on SST-2 and 5.8 F1 on SQuAD (Appendix M).

In the high-privacy regime the order reverses. We reproduce both DP baselines on SST-2 FT OPT-1.3B at $\varepsilon \in \{0.2, 0.3, 0.5, 1.0, 2.0\}$, with DPZero at DP-AggZO's learning rate, clipping threshold and sampling rate (Table 9 in Appendix L). DPZero $K = 1$ stays at or near chance for every $\varepsilon < 1$. At $\varepsilon \leq 0.5$, DP-AggZO $K = 64$ improves when its run is shortened to about the per-example forward passes of PACZERO, since a shorter run composes fewer times and needs less noise per step. Its best run at $\varepsilon \leq 1$ reaches 86.93%. In the same setting PACZERO-BAC$_0$ and PACZERO-AGC$_0$ at $\mathrm{MI} = 0$ reach 90.10% and 89.83%, above every DP run at $\varepsilon \leq 1$ and within half a point of the strongest in-house DP run at $\varepsilon = 2$. PACZERO thus produces usable accuracy at $\mathrm{MI} = 0$, a level that published DP zeroth-order accountants [4, 42, 56] reach only at $\varepsilon = \delta = 0$ with infinite per-step noise.

5 Related Work

A mature line of work adapts DP-SGD [1] to LLM fine-tuning, with refinements to clipping and adapter integration [8, 9, 18, 23, 25, 52, 59]. DP zeroth-order methods privatize only the per-step finite-difference scalars, one per sampled direction, which removes explicit dimension dependence from the per-step noise. Zhang et al. [56] (DPZero) prove nearly dimension-independent rates with a Gaussian scalar mechanism. DP-ZO [42] studies the same scalar privatization empirically, with Gaussian or Laplace noise, the latter giving pure ε-DP. Bao et al. [4] (DP-AggZO) aggregate K finite-difference scalars into a vector before clipping and report the strongest results among DP zeroth-order methods on OPT-1.3B/6.7B. Liu et al. [26] schedule the ZO hyperparameters in stages and prune parameters without data. PAC Privacy [46] measures and controls what a data-processing algorithm reveals about a secret drawn from a known distribution. Later work gives PAC Privacy guarantees for classical algorithms [39, 41], quantifies the leakage of learnable obfuscation [47], and brings PAC Privacy to SQL queries [6, 40] and private prediction [60]. Additionally, Xiao et al. [48] study one-sided noise for PAC Privacy and Zhang and Vorobeychik [58] use f-divergence for tighter

noise calibration. PACZERO applies PAC Privacy to LLM fine-tuning. No per-record sensitivity enters its accounting, and per-sample clipping serves only as a utility stabilizer. The sign-quantized update shares structure with signSGD [7, 24], and PACZERO calibrates its release for privacy while signSGD quantizes for compression.

6 Discussion and Limitations

PAC Privacy and DP. DP gives a worst-case guarantee that does not depend on the data. Under DP, the output distributions on every pair of neighboring datasets must be close, which makes it the stronger guarantee. PAC Privacy gives an instance-based guarantee with respect to a distribution $\mathcal{D}$ over the input data, and calibrates its noise to how much the output varies when the input is drawn from $\mathcal{D}$ (§3.1). After sign quantization and subset aggregation this variation can be far smaller than the largest change a single record can make, and on unanimity steps it is zero. DP relaxations that use instance-specific information still charge for any single record's influence (Appendix D).

The guarantee and the MIA game. For most training sets no distribution over the secret data can be specified, since the data were drawn once from an unknown source and cannot be drawn again. The MIA game supplies one [11, 37, 50]: a challenger samples a training set from a universe of records and trains a model on it, and an adversary, who may know the universe, tries to decide whether a given record was in the training set. Our construction of §2 is an instance of this game, with the training set drawn uniformly from the candidate subsets of U. PACZERO bounds what an adversary can infer about the secret subset S^* even when it knows U, and therefore also when it does not. Furthermore, the bound covers any other inference whose target is S^*, such as reconstructing it [47]. At $I(S^*; \tilde{Y}_{1:T}) = 0$ the release tells the adversary *nothing* about S^* beyond what it knew before training.

Distributional leakage. Since the guarantee is stated with respect to $\mathcal{D}$, the distribution itself lies outside it. We call what a release reveals about $\mathcal{D}$ *distributional leakage*. PAC Privacy states its guarantee for an adversary that already knows $\mathcal{D}$, that is, U and the candidate subsets for PACZERO, so the mechanism may use U without charge to the budget, and all four variants do. Every step evaluates each record of U, and the release depends on how the posterior mass over the M candidate subsets splits between the two signs, which is q_t^+. The variants use this split to different extents. The additive Gaussian channel of PACZERO-AGC uses it only through how evenly the candidates are divided, which fixes the variance of the secret subset's sign. The binary asymmetric channel of PACZERO-BAC additionally uses which sign holds the larger share and tilts its output toward that sign. At zero budget, PACZERO-AGC$_0$ uses only whether a step is unanimous and, when it is, the common sign, while PACZERO-BAC$_0$ releases the majority sign on every step. This use of U is the stability that PAC Privacy exploits, and it is what makes zero-cost steps possible. A realistic adversary that does not know U may therefore learn something about U from the release, and the guarantee does not bound this. Whether such an adversary, which observes only the releases $\tilde{Y}_{1:T}$, can recover q_t^+ from them, let alone reconstruct properties of U, is an open question for future work.

7 Conclusion

We introduced PACZERO, a family of PAC-private zeroth-order mechanisms for fine-tuning large language models, built around one primitive, which is to sign-quantize subset-aggregated ZO updates and release the sign of the secret subset, through a calibrated channel when the candidates disagree. Quantizing to one bit admits exact mutual-information calibration, creates frequent unanimity, on which a step costs zero conditional MI, and lets one accounting argument cover both channels at every budget. At $I(S^*; \tilde{Y}_{1:T}) = 0$, PACZERO-BAC$_0$ stays within 2.0 points of the non-private ZO reference at the same step budget on all sixteen settings of Table 1 and Appendix F, reaching $\mathbf{91.63 \pm 0.30}\%$ on SST-2 OPT-6.7B FT and $\mathbf{87.80 \pm 1.12}$ F1 on SQuAD Llama-3.1-8B FT. Quantizing before calibrating noise opens an MIA-resistance level, $I(S^*; \tilde{Y}_{1:T}) = 0$, that no published DP zeroth-order accountant reaches at finite per-step noise. Natural next steps are a convergence analysis of PACZERO, larger corpora, which need minibatched subset estimation with a new privacy analysis, and settings beyond supervised fine-tuning.

Acknowledgments

The contribution of Marten van Dijk and Murat Bilgehan Ertan to this publication is part of the project CiCS of the research program Gravitation which is (partly) financed by the Dutch Research Council (NWO) under the grant 024.006.037. Xiaochen Zhu is supported by the MIT Schwarzman College of Computing PhD Fellowship. We acknowledge the use of the DAS-6 High-Performance Computing cluster at Vrije Universiteit Amsterdam for GPU-based experiments [3]. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant nos. EINF-20085 and EINF-18721. The authors declare no competing interests.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

[2] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15453–15462, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fc0de4e0396fff257ea362983c2dda5a-Abstract.html.

[3] Henri E. Bal, Dick H. J. Epema, Cees de Laat, Rob van Nieuwpoort, John W. Romein, Frank J. Seinstra, Cees Snoek, and Harry A. G. Wijshoff. A medium-scale distributed system for computer science research: Infrastructure for the long term. *Computer*, 49(5):54–63, 2016. doi: 10.1109/MC.2016.127. URL https://doi.org/10.1109/MC.2016.127.

[4] Ergute Bao, Yangfan Jiang, Fei Wei, Xiaokui Xiao, Zitao Li, Yaliang Li, and Bolin Ding. Unlocking the power of differentially private zeroth-order optimization for fine-tuning LLMs. In Lujo Bauer and Giancarlo Pellegrino, editors, *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pages 1569–1588. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/bao-ergute.

[5] Ergute Bao, Yangfan Jiang, Fei Wei, Xiaokui Xiao, Zitao Li, Yaliang Li, and Bolin Ding. Unlocking the power of differentially private zeroth-order optimization for fine-tuning LLMs, 2025. URL https://github.com/erguteb/dp-aggzo/blob/main/tr.pdf. Technical report.

[6] Ilaria Battiston, Dandan Yuan, Xiaochen Zhu, and Peter Boncz. SIMD-PAC-DB: Pretty performant PAC privacy. *CoRR*, abs/2603.15023, 2026. doi: 10.48550/ARXIV.2603.15023. URL https://doi.org/10.48550/ARXIV.2603.15023.

[7] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 560–569. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/bernstein18a.html.

[8] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Differentially private optimization on large model at small cost. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 3192–3218. PMLR, 2023. URL https://proceedings.mlr.press/v202/bu23a.html.

[9] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/8249b30d877c91611fd8c7aa6ac2b5fe-Abstract-Conference.html.

[10] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In Michael D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association, 2021. ISBN 9781939133243. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[11] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 1897–1914. IEEE, 2022. ISBN 9781665413169. doi: 10.1109/SP46214.2022.9833649. URL https://doi.org/10.1109/SP46214.2022.9833649.

[12] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1964–1974. PMLR, 2021. URL http://proceedings.mlr.press/v139/choquette-choo21a.html.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, April 2005. ISBN 9780471748823. doi: 10.1002/047174882x. URL http://dx.doi.org/10.1002/047174882X.

[14] Paul Cuff and Lanqing Yu. Differential privacy as a mutual information constraint. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 43–54. ACM, 2016. ISBN 9781450341394. doi: 10.1145/2976749.2978308. URL https://doi.org/10.1145/2976749.2978308.

[15] Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *CoRR*, abs/2204.13650, 2022. doi: 10.48550/ARXIV.2204.13650. URL https://doi.org/10.48550/ARXIV.2204.13650.

[16] Murat Bilgehan Ertan and Marten van Dijk. Fundamental limitations of favorable privacy-utility guarantees for DP-SGD. *CoRR*, abs/2601.10237, 2026. doi: 10.48550/ARXIV.2601.10237. URL https://doi.org/10.48550/ARXIV.2601.10237. Accepted at ACM CCS 2026.

[17] Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a Rényi filter. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28080–28091, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ec7f346604f518906d35ef0492709f78-Abstract.html.

[18] Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. Exploring the limits of differentially private deep learning with group-wise clipping. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ozeOclVGPeX.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth*

International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[20] David Ilić, David Stanojević, and Kostadin Cvejoski. Powerful training-free membership inference against fine-tuned autoregressive language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14077–14093, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.640. URL https://aclanthology.org/2026.acl-long.640/.

[21] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1376–1385. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/kairouz15.html.

[22] Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Trans. Database Syst.*, 39(1):3:1–3:36, 2014. doi: 10.1145/2514689. URL https://doi.org/10.1145/2514689.

[23] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=bVuP3ltATMz.

[24] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. signSGD via zeroth-order oracle. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BJe-DsC5Fm.

[25] Xiao-Yang Liu, Rongyi Zhu, Daochen Zha, Jiechao Gao, Shan Zhong, Matt White, and Meikang Qiu. Differentially private low-rank adaptation of large language model using federated learning. *ACM Trans. Manag. Inf. Syst.*, 16(2):1–24, 2025. doi: 10.1145/3682068. URL https://doi.org/10.1145/3682068.

[26] Zhihao Liu, Jian Lou, Wenjie Bao, Yuke Hu, Bo Li, Zhan Qin, and Kui Ren. Differentially private zeroth-order methods for scalable large language model finetuning. *CoRR*, abs/2402.07818, 2024. doi: 10.48550/ARXIV.2402.07818. URL https://doi.org/10.48550/ARXIV.2402.07818.

[27] Llama Team. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/ARXIV.2407.21783.

[28] 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 Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a627810151be4d13f907ac898ff7e948-Abstract-Conference.html.

[29] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8332–8347. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.570. URL https://doi.org/10.18653/V1/2022.EMNLP-MAIN.570.

[30] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 263–275. IEEE Computer Society, 2017. ISBN 9781538632178. doi: 10.1109/CSF.2017.11. URL https://doi.org/10.1109/CSF.2017.11.

[31] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=vjel3nWP2a.

[32] Yurii E. Nesterov and Vladimir G. Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, 2017. doi: 10.1007/S10208-015-9296-2. URL https://doi.org/10.1007/S10208-015-9296-2.

[33] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84. ACM, 2007. ISBN 9781595936318. doi: 10.1145/1250790.1250803. URL https://doi.org/10.1145/1250790.1250803.

[34] Theshani Nuradha and Ziv Goldfeld. Pufferfish privacy: An information-theoretic study. *IEEE Trans. Inf. Theory*, 69(11):7336–7356, 2023. doi: 10.1109/TIT.2023.3296288. URL https://doi.org/10.1109/TIT.2023.3296288.

[35] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

[36] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

[37] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society, 2017. ISBN 9781509055333. doi: 10.1109/SP.2017.41. URL https://doi.org/10.1109/SP.2017.41.

[38] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL, 2013. ISBN 9781937284978. doi: 10.18653/V1/D13-1170. URL https://doi.org/10.18653/V1/D13-1170.

[39] Mayuri Sridhar, Hanshen Xiao, and Srinivas Devadas. PAC-Private algorithms. In Marina Blanton, William Enck, and Cristina Nita-Rotaru, editors, *IEEE Symposium on Security and Privacy, SP 2025, San Francisco, CA, USA, May 12-15, 2025*, pages 3839–3857. IEEE, 2025. ISBN 9798331522360. doi: 10.1109/SP61157.2025.00034. URL https://doi.org/10.1109/SP61157.2025.00034.

[40] Mayuri Sridhar, Michael A. Noguera, Chaitanyasuma Jain, Kevin Kristensen, Srinivas Devadas, Hanshen Xiao, and Xiangyao Yu. PAC-Private databases. *IACR Cryptol. ePrint Arch.*, 2026: 238, 2026. URL https://eprint.iacr.org/2026/238.

[41] Mayuri Sridhar, Xiaochen Zhu, and Srinivas Devadas. Privacy-conscious algorithm design via PAC privacy. In *IEEE Symposium on Security and Privacy, SP 2026, San Francisco, CA, USA, May 18-21, 2026*, pages 2171–2190. IEEE, 2026. ISBN 9798331560652. doi: 10.1109/SP63933.2026.00147. URL https://doi.org/10.1109/SP63933.2026.00147.

[42] Xinyu Tang, Ashwinee Panda, Milad Nasr, Saeed Mahloujifar, and Prateek Mittal. Private fine-tuning of large language models with zeroth-order optimization. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=3Y3o0yFZfu.

[43] Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9583–9592. PMLR, 2020. URL http://proceedings.mlr.press/v119/triastcyn20a.html.

[44] Yu-Xiang Wang. Per-instance differential privacy. *J. Priv. Confidentiality*, 9(1), 2019. doi: 10.29012/JPC.662. URL https://doi.org/10.29012/JPC.662.

[45] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=3eIrli0TwQ.

[46] Hanshen Xiao and Srinivas Devadas. PAC Privacy: Automatic privacy measurement and control of data processing. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II*, volume 14082 of *Lecture Notes in Computer Science*, pages 611–644. Springer, 2023. ISBN 9783031385445. doi: 10.1007/978-3-031-38545-2_20. URL https://doi.org/10.1007/978-3-031-38545-2_20.

[47] Hanshen Xiao, G. Edward Suh, and Srinivas Devadas. Formal privacy proof of data encoding: The possibility and impossibility of learnable encryption. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 1834–1848. ACM, 2024. ISBN 9798400706363. doi: 10.1145/3658644.3670277. URL https://doi.org/10.1145/3658644.3670277.

[48] Hanshen Xiao, Jun Wan, Elaine Shi, and Srinivas Devadas. One-sided bounded noise: Theory, optimization algorithms and applications. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS 2025, Taipei, Taiwan, October 13-17, 2025*, pages 4214–4228. ACM, 2025. doi: 10.1145/3719027.3765110. URL https://doi.org/10.1145/3719027.3765110.

[49] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

[50] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 268–282. IEEE Computer Society, 2018. ISBN 9781538666807. doi: 10.1109/CSF.2018.00027. URL https://doi.org/10.1109/CSF.2018.00027.

[51] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *CoRR*, abs/2109.12298, 2021. doi: 10.48550/ARXIV.2109.12298. URL https://doi.org/10.48550/ARXIV.2109.12298.

[52] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=Q42f0dfjECO.

[53] Da Yu, Gautam Kamath, Janardhan Kulkarni, Tie-Yan Liu, Jian Yin, and Huishuai Zhang. Individual privacy accounting for differentially private stochastic gradient descent. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=l4Jcxs0fpC.

[54] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 58244–58282. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zarifzadeh24a.html.

[55] Jingyang Zhang, Jingwei Sun, Eric C. Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao (Frank) Yang, and Hai Li. Min-K%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=ZGkfoufDaU.

[56] Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. DPZero: Private fine-tuning of language models without backpropagation. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 59210–59246. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zhang24af.html.

[57] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. doi: 10.48550/ARXIV.2205.01068. URL https://doi.org/10.48550/ARXIV.2205.01068.

[58] Tao Zhang and Yevgeniy Vorobeychik. Residual-PAC privacy: Automatic privacy control beyond the Gaussian barrier. In *35th USENIX Security Symposium (USENIX Security 26)*, pages 1327–1346, Baltimore, MD, August 2026. USENIX Association. ISBN 978-1-939133-58-8. URL https://www.usenix.org/conference/usenixsecurity26/presentation/zhang-tao-residual.

[59] Xinwei Zhang, Zhiqi Bu, Steven Wu, and Mingyi Hong. Differentially private SGD without clipping bias: An error-feedback approach. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=uFbWHyTlPn.

[60] Xiaochen Zhu, Mayuri Sridhar, and Srinivas Devadas. Private prediction via PAC privacy. *CoRR*, abs/2601.14033, 2026. doi: 10.48550/ARXIV.2601.14033. URL https://doi.org/10.48550/arXiv.2601.14033.

A Appendix Organization

Appendix is organized as follows.

- Appendix B proves Theorem 3 (Appendix B.1), through Lemma 7, which shows that the state of the mechanism is determined by the earlier channel outputs, and Lemma 8, which bounds the information released in one step. It then shows that the channels of all four variants are calibrated (Appendix B.2), proves Theorem 5 (Appendix B.3), and proves Proposition 10 (Appendix B.4), that the MIA bound cannot be improved from the budget alone.
- Appendix C collects the implementation details of PACZERO, namely the role of per-sample clipping, numerical evaluation of the Gaussian-channel MI integral, the edge cases of the PACZERO-BAC calibration, the sign convention at zero, the unanimity tolerance, and the entropy-ceiling cap on β_t.

- Appendix D compares PACZERO with DP relaxations.
- Appendix E carries the protocol and baseline detail behind §4.1.
- Appendix F gives all four variants on all sixteen settings (Tables 2 and 3).
- Appendix G reports the unanimity fraction f per setting.
- Appendix H tests the $\mathrm{MI} = 0$ guarantee empirically with nine black-box membership-inference attacks.
- Appendix I gives the MI-budget sweep summarized in §4.2 (Table 7).
- Appendix J reports the LoRA counterpart of the same sweep (Table 8).
- Appendix K reports the parity check at OPT-1.3B between our in-house DP baselines and Table 2 of Bao et al. [4].
- Appendix L reports the in-house DP baseline reproduction (Table 9).
- Appendix M sets PACZERO against the DP comparators.
- Appendix N reports the hyperparameters of every setting (Table 11).
- Appendix O gives the learning-rate sweep that selects the learning rate for every setting and for the non-private reference (Tables 13 and 14).
- Appendix P reports the clipping-threshold ablations, namely the LoRA grid on SST-2 across all four models, the FT grid on both tasks across all four models, and the SQuAD LoRA grid.
- Appendix Q reports the 6.7B LoRA rank ablation.
- Appendix R decomposes the PACZERO per-step release into deterministic surrogates and reports the random-sign negative control.
- Appendix S reports the PACZERO-AGC_0 step-budget sweep on SST-2 OPT-1.3B FT (Table 20).
- Appendix T extends the per-step release to $K \in \{4, 16, 64\}$ following the DP-AggZO-style aggregation pattern on SST-2.
- Appendix U reports the universe-size sweep (Table 22).
- Appendix V profiles the mechanism's cost against the non-private ZO reference at matched settings and reports how cost scales in the universe size N and the subset count M (Tables 23 and 24).
- Appendix W reports the hardware and the GPU-hours by kind of experiment.
- Appendix X gives per-seed values for every multi-seed PACZERO entry of Table 1 and of Appendix F.

B Deferred full proofs

B.1 Proof of Theorem 3

Throughout this subsection we fix the directions $z_1, \ldots, z_T$. The universe, the candidate subsets, the initial parameters and the hyperparameters are fixed as well, so the only randomness is the secret index j^*, which is uniform on $[M]$, and the channels' randomness, which each step draws independently of j^* and of the earlier steps. We first show that the state of the mechanism is determined by the earlier channel outputs and that p_t is the exact posterior. We then bound what one step releases and sum over the steps. The last part of the proof removes the assumption that the directions are fixed.

Lemma 7. *For every step t, the parameters θ_t, the subset signs $s_1, \ldots, s_M$, the posterior p_t, the probability q_t^+, the budget β_t and the channel $\mathcal{C}_t$ are functions of the earlier channel outputs $\tilde{Y}_{1:t-1}$. Moreover, $p_t[m] = \Pr[j^* = m \mid \tilde{Y}_{1:t-1}]$ for every m.*

Proof. We argue by induction on t. At the first step the parameters are the initial ones, the posterior is uniform, and $\beta_1 = \mathrm{MI}_{\mathrm{tot}}/T$. Suppose the first claim holds at every step up to t. The parameters θ_t and the direction z_t fix every per-sample scalar $\hat{g}_i$, hence every subset mean and every sign s_m, and with p_t they fix q_t^+. The budget β_t follows from (5) and the charges of the earlier steps, each of

which is 0 or $\min\{\beta_\tau, h(q_\tau^+)\}$ and so is already fixed. The channel is a function of q_t^+ and β_t. After the output $\tilde{Y}_t$, the update rule gives θ_{t+1} from θ_t, z_t and $\mathrm{sign}(\tilde{Y}_t)$, and the update (4) gives p_{t+1} from p_t, the signs and $\tilde{Y}_t$. This proves the first claim at step $t+1$.

For the second claim, p_1 is uniform, which is the distribution of j^*. Suppose that $p_t[m] = \Pr[j^* = m \mid \tilde{Y}_{1:t-1}]$. By the first claim, $\tilde{Y}_{1:t-1}$ fixes the channel $\mathcal{C}_t$, and its randomness at step t is independent of j^* and of the earlier steps, so given $\tilde{Y}_{1:t-1}$ and $j^* = m$ the output $\tilde{Y}_t$ is distributed as $\mathcal{C}_t(s_m)$, with probability (or density) $L_t(\cdot \mid s_m)$. Bayes' rule then gives

$$\Pr[j^* = m \mid \tilde{Y}_{1:t}] \;\propto\; \Pr[j^* = m \mid \tilde{Y}_{1:t-1}]\, L_t(\tilde{Y}_t \mid s_m) \;=\; p_t[m]\, L_t(\tilde{Y}_t \mid s_m),$$

which is the update (4). On a unanimity step $\tilde{Y}_t = s_{j^*}$ and every candidate m with $p_t[m] > 0$ has $s_m = \tilde{Y}_t$, so the likelihood is the same for all of them and $p_{t+1} = p_t$, as in Algorithm 1. In each case $p_{t+1}[m] = \Pr[j^* = m \mid \tilde{Y}_{1:t}]$. □

Lemma 8 (One step)**.** *Fix a step t and a realization $y_{1:t-1}$ of $\tilde{Y}_{1:t-1}$, and let q_t^+ and β_t be the values it determines. Then $I(j^*; \tilde{Y}_t \mid \tilde{Y}_{1:t-1} = y_{1:t-1}) \le \beta_t^{\mathrm{used}}$, where $\beta_t^{\mathrm{used}} = 0$ on a unanimity step and $\beta_t^{\mathrm{used}} = \min\{\beta_t, h(q_t^+)\}$ on a disagreement step.*

Proof. By Lemma 7, the realization fixes the signs, q_t^+, β_t and the channel, and given it j^* has distribution p_t. In the unanimity case, $q_t^+ \in \{0, 1\}$. If $q_t^+ = 1$, then $s_m = +1$ for all m with $p_t[m] > 0$, so $s_{j^*} = +1$ with probability 1 and $\tilde{Y}_t = s_{j^*} = +1$ is a constant. Symmetrically if $q_t^+ = 0$. A constant carries zero MI.

In the disagreement case, $q_t^+ \in (0, 1)$. Define $\xi := s_{j^*} \in \{-1, +1\}$, a deterministic function of j^* given the fixed signs, with $P(\xi = +1) = q_t^+$. The channel's randomness at step t is independent of j^* and of the earlier steps, so $\tilde{Y}_t = \mathcal{C}_t(\xi)$ depends on j^* only through ξ, and $j^* \to \xi \to \tilde{Y}_t$ is a Markov chain. By the chain rule,

$$I(j^*;\, \tilde{Y}_t) \;=\; I(\xi;\, \tilde{Y}_t) \;+\; \underbrace{I(j^*;\, \tilde{Y}_t \mid \xi)}_{=\,0},$$

where the second term vanishes because $\tilde{Y}_t$ is conditionally independent of j^* given ξ. The channel is calibrated at (q_t^+, β_t), so $I(\xi; \tilde{Y}_t) \le \beta_t$. A binary input also carries at most its entropy, $I(\xi; \tilde{Y}_t) \le H(\xi) = h(q_t^+)$, so $I(\xi; \tilde{Y}_t) \le \min\{\beta_t, h(q_t^+)\} = \beta_t^{\mathrm{used}}$. □

Proof of Theorem 3. Since S^* is a function of j^*, the data-processing inequality and the chain rule give

$$I(S^*; \tilde{Y}_{1:T}) \;\le\; I(j^*; \tilde{Y}_{1:T}) \;=\; \sum_{t=1}^{T} I(j^*; \tilde{Y}_t \mid \tilde{Y}_{1:t-1}) \;=\; \sum_{t=1}^{T} \mathbb{E}\big[I(j^*; \tilde{Y}_t \mid \tilde{Y}_{1:t-1} = y_{1:t-1})\big], \quad (9)$$

where the expectation is over the realization $y_{1:t-1}$ of $\tilde{Y}_{1:t-1}$. By Lemma 8, the t-th term is at most $\mathbb{E}[\beta_t^{\mathrm{used}}]$, so

$$I(S^*;\, \tilde{Y}_{1:T}) \;\le\; \mathbb{E}\left[\sum_{t=1}^{T} \beta_t^{\mathrm{used}}\right] \;\le\; \mathrm{MI}_{\mathrm{tot}}, \quad (10)$$

where the second inequality holds because $\sum_t \beta_t^{\mathrm{used}} \le \mathrm{MI}_{\mathrm{tot}}$ for every realization (by the adaptive budget allocation (5)).

The update signs $Y_t = \mathrm{sign}(\tilde{Y}_t)$ are a function of $\tilde{Y}_{1:T}$, and so are the parameters, which the update rule of Algorithm 1 computes from them with the directions fixed. By the data-processing inequality, $I(S^*; Y_{1:T})$ and $I(S^*; \theta_{1:T+1})$ are at most $I(S^*; \tilde{Y}_{1:T}) \le \mathrm{MI}_{\mathrm{tot}}$.

It remains to let the directions $Z = z_{1:T}$ be random. They are drawn independently of j^* and of the channels' randomness, so $I(S^*; Z) = 0$, and the bound above holds for every value of Z. Hence $I(S^*; \tilde{Y}_{1:T}, Z) = I(S^*; Z) + I(S^*; \tilde{Y}_{1:T} \mid Z) \le \mathrm{MI}_{\mathrm{tot}}$. The update signs $Y_{1:T}$ and the parameters $\theta_{1:T+1}$ are functions of $(\tilde{Y}_{1:T}, Z)$, so the bound holds for them even when the adversary also sees the directions. □

B.2 The channels are calibrated

Lemma 9. *Let $q \in (0,1)$. For every $\beta > 0$, the Gaussian channel with $\sigma = I_q^{-1}(\beta)$ and the binary asymmetric channel* (8) *are calibrated at (q, β). The releases of* PACZERO-AGC$_0$ *and* PACZERO-BAC$_0$ *on a disagreement step are calibrated at $(q, 0)$.*

Proof. Let ξ be a bit with $P(\xi = +1) = q$. We bound $I(\xi; \mathcal{C}(\xi))$ for each channel.

For the Gaussian channel, $I_q(\sigma)$ decreases continuously from $h(q)$ at $\sigma = 0$, where the channel adds no noise, to 0 as $\sigma \to \infty$. We take $I_q^{-1}(\beta)$ to be the smallest $\sigma \geq 0$ with $I_q(\sigma) \leq \beta$, so $I_q(I_q^{-1}(\beta)) \leq \beta$. When $\beta < h(q)$ this holds with equality, and when $\beta \geq h(q)$ we have $I_q^{-1}(\beta) = 0$.

For the binary asymmetric channel, suppose first that $\beta \geq h(q)$. Then $D^* = 0$, $\pi = q$ and $a^* = b^* = 0$, so the channel is the identity and $I(\xi; \tilde{Y}) = h(q) \leq \beta$. Suppose next that $\beta < h(q)$. Then $h(D^*) = h(q) - \beta > 0$. Since h increases on $[0, 1/2]$ and $h(D^*) < h(q) = h(\min\{q, 1-q\})$, we have $0 < D^* < \min\{q, 1-q\}$, and therefore $0 < \pi < 1$. Consider the joint distribution in which $\tilde{Y} = +1$ with probability π and, given $\tilde{Y}$, ξ differs from $\tilde{Y}$ with probability D^*. Its marginal is $P(\xi = +1) = \pi(1 - D^*) + (1 - \pi)D^* = q$ by the definition of π, and its conditional probabilities are $\Pr[\tilde{Y} = +1 \mid \xi = -1] = \pi D^*/(1-q) = a^*$ and $\Pr[\tilde{Y} = -1 \mid \xi = +1] = (1-\pi)D^*/q = b^*$. So it is the joint distribution of the input and the output of the channel (8). Given either output, ξ differs from it with probability D^*, so $H(\xi \mid \tilde{Y}) = h(D^*)$ and $I(\xi; \tilde{Y}) = h(q) - h(D^*) = \beta$.

At zero budget, the coin of PACZERO-AGC$_0$ and the release of PACZERO-BAC$_0$ do not depend on ξ, so $I(\xi; \mathcal{C}(\xi)) = 0$. □

B.3 Proof of Theorem 5

Proof of Theorem 5. With $\mathrm{MI}_{\mathrm{tot}} = 0$, the allocation (5) sets $\beta_t = 0$ on every step, so Theorem 3 applies to any variant whose disagreement steps release through a channel calibrated at $(q_t^+, 0)$. By Lemma 9, the coin of PACZERO-AGC$_0$ and the release of PACZERO-BAC$_0$ are such channels. Theorem 3 therefore gives $I(S^*; \tilde{Y}_{1:T}) \leq 0$, so $I(S^*; \tilde{Y}_{1:T}) = 0$, and the same holds for the parameters $\theta_{1:T+1}$.

For the membership claim, $I(S^*; \tilde{Y}_{1:T}) = 0$ makes the release independent of S^*, and hence of the uniform bit $\mathbf{1}[x_i \in S^*]$, so every guess computed from the release is correct with probability exactly $1/2$. □

B.4 The MIA bound cannot be improved from the budget alone

Proposition 10. *Fix a record x_i, and let A_i be the $M/2$ candidates that contain it. For any release W with $I(j^*; W) \leq \mathrm{MI}_{\mathrm{tot}}$, the success probability s of the best guess of whether $x_i \in S^*$ satisfies $\mathrm{KL}(s \,\|\, 1/2) \leq \mathrm{MI}_{\mathrm{tot}}$. For every $s \in [1/2, 1]$ some release attains success s with $I(j^*; W) = \mathrm{KL}(s \,\|\, 1/2)$.*

Proof. Let p be the posterior of j^* given $W = w$, and let $r = p(A_i)$ be its mass on A_i. The best guess succeeds with probability $\max\{r, 1-r\}$, and both A_i and its complement contain $M/2$ candidates. Grouping the entropy of p by the two halves gives $H(p) \leq h(r) + \ln(M/2)$, with equality when p is uniform within each half. Hence $\ln M - H(p) \geq \ln 2 - h(r) = \mathrm{KL}(\max\{r, 1-r\} \,\|\, 1/2)$. The function $\phi(s) = \ln 2 - h(s)$ is convex, and the average of $\ln M - H(p)$ over w is $I(j^*; W)$, so Jensen's inequality gives $\mathrm{KL}(s \,\|\, 1/2) \leq I(j^*; W) \leq \mathrm{MI}_{\mathrm{tot}}$ for the average success s. For attainment, let $V = \mathbf{1}[x_i \in S^*]$, which is uniform, and release $W = V$ with probability s and $W = 1 - V$ otherwise, independently of everything else. The posterior given W is uniform within each half, the best guess succeeds with probability s, and $I(j^*; W) = I(V; W) = \ln 2 - h(s)$. □

The proposition reproves (2) for this construction and shows that no bound depending only on $\mathrm{MI}_{\mathrm{tot}}$ can improve on it.

C PACZERO implementation details

The finite-difference error. The $O(\mu^2)$ term in (1) holds when the Hessian of ℓ_i is Lipschitz, by a second-order Taylor expansion [32].

The role of per-sample clipping. Per-sample clipping at the threshold c before subset aggregation is a *utility* stabilizer. It bounds any single record's leverage on the subset mean. A sign change from one record stays possible when the remaining subset sum is close enough to zero, so clipping bounds a record's influence without ruling out a flip. Because the release depends on the secret only through the sign s_{j^*}, and the channel accounts for all leakage of s_{j^*}, the magnitude of $\hat{g}_i$ does not enter the privacy proof. Clipping only affects the empirical sign distribution. Appendix P shows it is utility-relevant, with dev accuracy spanning 1.0 to 2.2 points across the clipping-threshold grid on SST-2 LoRA, but it needs no calibration to a record-level sensitivity.

Numerical evaluation of the Gaussian-channel MI integral. This paragraph and the entropy-ceiling cap below concern the Gaussian channel of §3.2 only. The binary asymmetric channel of §3.3 is calibrated in closed form by (8).

We evaluate (7) by 60-node Gauss-Hermite quadrature and find σ_t by bisection in $\log\sigma$, since $I_q(\sigma)$ decreases in σ. The accountant charges β_t whatever the solver returns, so the bisection targets $0.999\,\beta_t$, a margin that exceeds the quadrature's measured error over the operating range, and returns the larger-σ endpoint. Hence $\sigma_t \geq I_q^{-1}(\beta_t)$ and $I_q(\sigma_t) \leq \beta_t$. Below $\beta_t = 10^{-10}$ the quadrature loses its significant digits, so the solver bisects the analytic bound $I_q(\sigma) \leq 2q(1-q)/\sigma^2$ there. An independent 60-digit evaluation of (7), over 7 values of q and 12 budgets from 10^{-15} to $\ln 2$, confirms $I_q(\sigma_t) \leq \beta_t$ at a worst-case relative margin of 10^{-3}.

Edge cases of the binary asymmetric channel. The calibration (8) needs $0 < p < 1$, and at $p = 1/2$ also $\beta_t > 0$. At $p \in \{0, 1\}$ the step is a unanimity step. There $h(p) = 0$, so $D^* = 0$ and $\pi = p$, the formula leaves the sign that occurs unchanged, and we take the channel to be the identity. More generally, when $\beta_t \geq h(p)$ the budget constraint $I(\xi; \tilde{Y}) \leq \beta_t$ does not bind, $D^* = 0$, and the channel is the identity, so the unanimity branch of Algorithm 1 is the non-binding case of the same calibration. At $p = 1/2$ and $\beta_t = 0$ the formula is undefined, and PACZERO-BAC$_0$ emits $+1$ by the sign convention below.

Sign convention at zero. In Algorithm 1 we adopt the convention $\mathrm{sign}(0) = +1$, so exact zeros in either $\bar{g}_m$ or the disagreement-branch channel output $\tilde{Y}_t$ are mapped to $+1$. An empty candidate subset is given mean 0. This preserves the binary support $\{-1, +1\}$ assumed by Eq. (7) and Lemma 8.

Unanimity tolerance. Algorithm 1 writes the unanimity test as $q_t^+ \in \{0, 1\}$. Our implementation evaluates it with a tolerance, $q_t^+ \leq \tau$ or $q_t^+ \geq 1 - \tau$ with $\tau = 10^{-12}$, so floating-point roundoff in the posterior weights does not push a near-unanimity step onto the disagreement branch. Theorem 3 is stated for the exact test. The two zero-budget variants test unanimity on the exact subset-sign counts, so no tolerance term enters Theorem 5. On either branch, PACZERO-BAC$_0$ releases a sign that does not depend on which candidate is the secret, and PACZERO-AGC$_0$ releases the common sign or a coin drawn independently of j^*.

Entropy-ceiling cap. A single bit cannot leak more MI than the binary entropy of the sign s_{j^*} under p_t, $h(q_t^+) = -q_t^+ \log q_t^+ - (1 - q_t^+)\log(1 - q_t^+)$. When the posterior has concentrated, $h(q_t^+)$ shrinks below β_t and the Gaussian bisection invoked at line 11 of Algorithm 1 would target an infeasible MI. Before inverting I_q we therefore cap $\beta_t \leftarrow \min(\beta_t, 0.999 \cdot h(q_t^+))$. The Gaussian is calibrated at this capped value, so Lemma 8 applies with it in place of β_t, and the step is charged the capped value. The cap is fixed by the state before the release (Lemma 7), so admissibility under adaptive composition is preserved.

D Comparison with DP relaxations

Several relaxations of DP lower the noise or tighten the accounting below the global-sensitivity worst case by using instance-specific or distributional information. Smooth sensitivity calibrates noise

to a smooth upper bound on the local sensitivity of the query [33]. Per-instance DP reports an ε for a fixed dataset together with a fixed target record [44]. Individual privacy accounting tracks a separate privacy loss for each participant, either post hoc for an unchanged DP-SGD run [53] or through a filter that stops using a participant once that loss reaches its budget [17]. Bayesian DP takes the privacy loss in probability over a record drawn from the data distribution [43]. Each of these is a quantity between a dataset and the same dataset with one record changed, so each charges a positive cost whenever that record's inclusion changes the output distribution, and the smooth upper bound charges a positive cost even at instances where the local sensitivity is zero. Pufferfish builds a privacy definition from a set of potential secrets, a set of discriminative pairs of secrets that an attacker should be unable to tell apart, and a class of plausible data-generating distributions [22]. A mutual-information form of that framework came later [34], following the mutual-information form of DP [14]. PAC Privacy bounds an adversary's posterior advantage by the mutual information between a secret drawn from a declared distribution and the release [46]. Our contribution is the instantiation of that criterion on a sign-quantized update, where the secret is the identity of the secret *subset* as in the standard construction [39, 60] and the cost is evaluated exactly per step. When every candidate subset with posterior mass agrees on the released sign, the cost is zero while the released bit still depends on the training records, and PACZERO-AGC$_0$ takes its optimization direction from those steps. PACZERO-BAC$_0$ releases the majority of the subset signs at zero cost on every step (§3.4).

E Protocol and baseline details

Test F1 on SQuAD is measured on a 1000-example subset. On SST-2, test accuracy is measured on the whole 872-example GLUE validation split. The zeroth-order perturbation scale of (1) is $\mu = 10^{-3}$ in every setting. Models load in float16. The learning rate decays polynomially over each run's own step budget, so two runs of the same setting at different T follow different schedules. Each PACZERO and reference run evaluates the dev loss every 100 steps, keeps the checkpoint with the lowest one and reports its test score. LoRA uses $r = 8$ and $\alpha = 16$ on the query and value projections, with the up-projection initialized at zero so the adapter is the identity at step 0. SQuAD answers are greedy decodes of at most 50 new tokens, and the reported F1 is computed on those decodes. The in-house DP baselines run the released DP-AggZO code with a constant learning rate and dev evaluation every 100 steps and at the last step. The three seeds resample the universe U, and on SQuAD the 1000-example evaluation subset, while the public perturbation directions keep a fixed seed.

Secret index and dev split. Every run, sweeps included, draws its own secret index from a private random source, so hyperparameters chosen on earlier runs do not depend on it, and the proof of Theorem 3 treats them as fixed. The guarantee is defined over U, and the 500-example dev split that selects the learning rate, the clipping threshold and the checkpoint lies outside it. Our in-house DP baselines [4, 56] run the same protocol, selecting checkpoints on the 500-example dev split of MeZO [28] at no charge to ε.

The flip statistic of §4.2 is measured over the 48 PACZERO-AGC runs at $\mathrm{MI}_{\mathrm{tot}} = 0.33$, the sixteen settings at three seeds. They cover $30{,}589$ steps that spend budget, and the flip probability of a step is $\Phi(-1/\sigma_t)$. Per-setting medians lie between 0.4717 and 0.4869.

Baselines. The reference updates with the clipped mean of all N records of U, with no subsets and no channel, at each setting's step budget, at its own dev-selected learning rate (Table 14) and at $c = 1000$, so it measures what zeroth-order fine-tuning reaches at that budget. At that matched budget it falls below PACZERO-BAC$_0$ at $\mathrm{MI} = 0$ in two of the four FT settings of Table 1, by 0.35 and 1.90 points. On OPT-6.7B FT, training the reference four times as long takes it past PACZERO-BAC$_0$ on SQuAD and leaves SST-2 where it was. At $T = 4000$, at a learning rate selected on dev at that budget, the reference reaches 93.69% on SST-2 against 93.58 at $T = 1000$, and 83.59 F1 on SQuAD against $75.20_{\pm 0.91}$. The SQuAD reference at the matched budget is therefore 8.39 F1 short of the longer run, and PACZERO-BAC$_0$ at $\mathrm{MI} = 0$ is 6.49 F1 below it. Trained for the same $T = 4000$, PACZERO-BAC$_0$ reaches 79.00 F1 (seed 0), 4.59 below the reference. The DP rows run at the step budgets of Table 10.

Table 2: **Full SST-2 grid (%), all four variants on all four models.** Mean $\pm$ std over $n=3$ seeds, accuracy. The non-private reference is single-seed where no spread is shown. Within each track and model all four variants share one learning rate, one clipping threshold and one step budget (Appendix N), and all use $M=128$. The OPT-6.7B and Llama-3.1-8B entries at $\mathrm{MI} \in \{0.33, 0.68, 0\}$ repeat Table 1.

Track	Method	MI (nats)	OPT-1.3B	OPT-6.7B	Qwen2.5-7B	Llama-3.1-8B
FT	PACZERO-AGC	0.01	$89.11_{\pm 0.61}$	$91.97_{\pm 0.41}$	$88.72_{\pm 0.63}$	$92.55_{\pm 0.75}$
	PACZERO-AGC	0.11	$89.53_{\pm 0.70}$	$91.32_{\pm 0.46}$	$88.76_{\pm 0.87}$	$91.55_{\pm 0.58}$
	PACZERO-AGC	0.33	$89.45_{\pm 0.69}$	$91.09_{\pm 1.03}$	$89.03_{\pm 0.93}$	$91.44_{\pm 1.08}$
	PACZERO-AGC	0.68	$89.45_{\pm 0.90}$	$91.28_{\pm 1.09}$	$89.72_{\pm 0.96}$	$91.63_{\pm 0.75}$
	PACZERO-BAC	0.01	$89.41_{\pm 0.29}$	$91.86_{\pm 0.11}$	$92.28_{\pm 0.46}$	$93.00_{\pm 0.41}$
	PACZERO-BAC	0.11	$89.56_{\pm 0.11}$	$91.59_{\pm 0.29}$	$92.51_{\pm 0.24}$	$92.70_{\pm 0.58}$
	PACZERO-BAC	0.33	$89.45_{\pm 0.40}$	$91.44_{\pm 0.24}$	$92.47_{\pm 0.53}$	$93.12_{\pm 0.34}$
	PACZERO-BAC	0.68	$89.41_{\pm 0.37}$	$92.16_{\pm 0.48}$	$92.32_{\pm 0.91}$	$93.12_{\pm 0.50}$
	PACZERO-AGC$_0$	0	$89.83_{\pm 0.54}$	$89.87_{\pm 0.63}$	$89.37_{\pm 0.33}$	$92.20_{\pm 0.61}$
	PACZERO-BAC$_0$	0	$90.10_{\pm 0.35}$	$91.63_{\pm 0.30}$	$92.24_{\pm 0.40}$	$92.78_{\pm 0.23}$
	non-private ZO reference	none	89.22	93.58	92.78	$92.43_{\pm 0.61}$
LoRA	PACZERO-AGC	0.01	$89.83_{\pm 1.03}$	$92.28_{\pm 0.35}$	$92.70_{\pm 0.18}$	$89.30_{\pm 3.15}$
	PACZERO-AGC	0.11	$90.10_{\pm 0.67}$	$92.70_{\pm 0.54}$	$93.35_{\pm 0.40}$	$88.91_{\pm 1.79}$
	PACZERO-AGC	0.33	$89.98_{\pm 0.18}$	$92.62_{\pm 0.33}$	$92.58_{\pm 0.57}$	$88.80_{\pm 2.66}$
	PACZERO-AGC	0.68	$89.95_{\pm 0.13}$	$92.39_{\pm 0.83}$	$92.81_{\pm 0.18}$	$90.52_{\pm 2.36}$
	PACZERO-BAC	0.01	$90.06_{\pm 0.54}$	$93.12_{\pm 0.53}$	$94.19_{\pm 0.18}$	$90.25_{\pm 1.03}$
	PACZERO-BAC	0.11	$90.10_{\pm 0.33}$	$92.93_{\pm 0.58}$	$94.04_{\pm 0.50}$	$89.03_{\pm 1.33}$
	PACZERO-BAC	0.33	$90.02_{\pm 0.50}$	$93.00_{\pm 0.91}$	$93.92_{\pm 0.72}$	$90.25_{\pm 0.75}$
	PACZERO-BAC	0.68	$90.18_{\pm 0.83}$	$93.43_{\pm 0.57}$	$93.92_{\pm 0.91}$	$90.79_{\pm 0.70}$
	PACZERO-AGC$_0$	0	$90.02_{\pm 1.09}$	$93.62_{\pm 0.13}$	$92.47_{\pm 0.37}$	$87.69_{\pm 0.63}$
	PACZERO-BAC$_0$	0	$89.91_{\pm 0.11}$	$93.31_{\pm 0.43}$	$94.23_{\pm 0.52}$	$89.37_{\pm 2.84}$
	non-private ZO reference	none	90.37	92.78	94.38	91.17
Zero-shot			53.67	61.24	63.65	59.75

F All sixteen settings

Table 1 reports a subset of the results of this appendix. Table 2 and Table 3 give every entry on all four models, at the four positive MI budgets and at $\mathrm{MI}=0$, so the two main-text models can be read against OPT-1.3B and Qwen2.5-7B under the same protocol. Each track closes with the non-private ZO reference of every setting, at the setting's step budget and its own dev-selected learning rate (Table 14), and the last row is zero-shot, the pretrained model evaluated with no training. PACZERO-BAC$_0$ at $\mathrm{MI}=0$ is at most 1.95 points below the reference on the sixteen settings and above it on seven.

G Unanimity fraction per setting

Table 4 reports the unanimity fraction f, the share of steps that take the unanimity branch, of the seed-0 runs of every setting. For every model and track, each SQuAD run takes the unanimity branch more often than each SST-2 run. The ten runs of a setting differ by at most 20.3 points, with a median of 9.25, so the rate varies mainly with the setting.

Within a run the rate declines. Split into tenths, each of the 32 seed-0 runs of the two zero-budget variants has a lower rate in its last tenth than in its first. PACZERO-AGC$_0$ takes its optimization direction only from unanimity steps, so it needs the rate to stay positive, and it does. Over all 48 PACZERO-AGC$_0$ runs no tenth falls below 9%, and the longest gap between unanimity steps is 32 steps. PACZERO-BAC$_0$ carries an optimization direction on every step and needs no such property.

Table 3: **Full SQuAD grid (%), all four variants on all four models.** Mean $\pm$ std over $n=3$ seeds, F1. The non-private reference is single-seed where no spread is shown. Within each track and model all four variants share one learning rate, one clipping threshold and one step budget (Appendix N), and all use $M=128$. The OPT-6.7B and Llama-3.1-8B entries at $\mathrm{MI} \in \{0.33, 0.68, 0\}$ repeat Table 1.

Track	Method	MI (nats)	OPT-1.3B	OPT-6.7B	Qwen2.5-7B	Llama-3.1-8B
FT	PACZERO-AGC	0.01	$67.70_{\pm 1.63}$	$74.65_{\pm 2.37}$	$87.69_{\pm 0.33}$	$87.69_{\pm 0.19}$
	PACZERO-AGC	0.11	$67.90_{\pm 1.78}$	$75.48_{\pm 1.88}$	$87.71_{\pm 0.49}$	$87.20_{\pm 0.71}$
	PACZERO-AGC	0.33	$68.01_{\pm 1.75}$	$75.65_{\pm 1.07}$	$87.86_{\pm 0.33}$	$87.74_{\pm 0.26}$
	PACZERO-AGC	0.68	$67.98_{\pm 1.13}$	$75.45_{\pm 1.61}$	$88.00_{\pm 0.46}$	$88.07_{\pm 0.96}$
	PACZERO-BAC	0.01	$69.47_{\pm 0.45}$	$77.07_{\pm 1.29}$	$88.33_{\pm 0.13}$	$87.90_{\pm 1.13}$
	PACZERO-BAC	0.11	$69.36_{\pm 0.41}$	$77.07_{\pm 1.47}$	$88.51_{\pm 0.24}$	$87.84_{\pm 1.21}$
	PACZERO-BAC	0.33	$69.56_{\pm 1.05}$	$76.69_{\pm 1.01}$	$88.48_{\pm 0.23}$	$87.78_{\pm 1.33}$
	PACZERO-BAC	0.68	$69.77_{\pm 1.24}$	$75.84_{\pm 0.78}$	$88.39_{\pm 0.06}$	$87.74_{\pm 1.19}$
	PACZERO-AGC$_0$	0	$67.71_{\pm 0.40}$	$74.83_{\pm 1.32}$	$87.70_{\pm 0.66}$	$87.39_{\pm 1.58}$
	PACZERO-BAC$_0$	0	$69.56_{\pm 0.25}$	$77.10_{\pm 1.39}$	$88.39_{\pm 0.23}$	$87.80_{\pm 1.12}$
	non-private ZO reference	none	69.83	$75.20_{\pm 0.91}$	87.51	89.01
LoRA	PACZERO-AGC	0.01	$67.94_{\pm 1.20}$	$73.66_{\pm 0.54}$	$87.58_{\pm 0.88}$	$86.82_{\pm 1.46}$
	PACZERO-AGC	0.11	$67.87_{\pm 1.02}$	$71.70_{\pm 2.23}$	$88.06_{\pm 0.64}$	$87.41_{\pm 1.65}$
	PACZERO-AGC	0.33	$68.61_{\pm 0.76}$	$73.93_{\pm 1.51}$	$87.97_{\pm 0.87}$	$88.19_{\pm 0.36}$
	PACZERO-AGC	0.68	$67.77_{\pm 2.31}$	$73.30_{\pm 1.49}$	$88.52_{\pm 0.49}$	$88.61_{\pm 1.41}$
	PACZERO-BAC	0.01	$68.82_{\pm 0.37}$	$74.86_{\pm 1.48}$	$88.19_{\pm 0.31}$	$88.80_{\pm 1.29}$
	PACZERO-BAC	0.11	$69.25_{\pm 1.28}$	$74.56_{\pm 1.33}$	$88.28_{\pm 0.35}$	$89.07_{\pm 1.57}$
	PACZERO-BAC	0.33	$68.95_{\pm 0.24}$	$75.44_{\pm 1.39}$	$88.31_{\pm 0.51}$	$88.89_{\pm 1.69}$
	PACZERO-BAC	0.68	$68.95_{\pm 0.46}$	$76.10_{\pm 2.56}$	$88.07_{\pm 0.68}$	$88.93_{\pm 1.59}$
	PACZERO-AGC$_0$	0	$65.41_{\pm 1.26}$	$72.12_{\pm 1.81}$	$88.20_{\pm 0.40}$	$88.05_{\pm 1.22}$
	PACZERO-BAC$_0$	0	$69.27_{\pm 1.39}$	$75.44_{\pm 0.77}$	$88.34_{\pm 0.31}$	$88.20_{\pm 2.21}$
	non-private ZO reference	none	69.33	75.73	88.00	86.98
Zero-shot			26.91	36.55	85.18	64.79

H Empirical membership inference at $I(S^*; \tilde{Y}_{1:T}) = 0$

Setup. We attack the seed-2 PACZERO-AGC$_0$ and PACZERO-BAC$_0$ checkpoint of every setting (Appendix F), two tasks, four models, FT and LoRA, each the dev-selected checkpoint whose test score the paper reports. The game is the one Theorem 5 covers. Given the public universe U ($N=1000$), the adversary decides for each $x_i \in U$ whether $x_i \in S^*$ ($|S^*|=525$). Every attack scores the fine-tuned model and its pretrained base on identical records.

Black-box attacks. We run the LOSS attack [50], the difficulty-calibrated loss [45] with the pretrained model as reference, offline RMIA [54], Min-K% [36] and Min-K%++ [55] with $k=20\%$, the zlib ratio [10], the label-only gap attack [12, 50] (greedy-decoding F1 on SQuAD, correctness on SST-2), EZ-MIA [20] and offline LiRA [11]. RMIA uses the pretrained model as its reference and $\gamma=2$, and its population is 1000 records of the training split outside U and the dev split. Its offline parameter a is chosen for each attacked checkpoint by the AUC with which it separates the universe U of another seed's model of the same variant from 1000 further such records, so the attacked checkpoint plays no part in choosing it. LiRA uses the models of the other seeds of the same setting, which were trained on other universes, as OUT references for the records they did not train on. Each attack gives an AUC with a 95% bootstrap interval. A checkpoint's AUC is compared with the AUC of the same attack on the other seeds' models with the same labels, on the records that none of those models used for training or dev, which removes any separation that the records themselves carry. LiRA, which already uses those models as references, is compared with an AUC of 0.5, and the label-only attack with the pretrained model. Of the checkpoint and attack pairs, 0.7% have a lower confidence bound above their reference, against 2.5% expected by chance. The headline attack, RMIA against the pretrained model, gives AUCs from 0.481 to 0.537 (mean 0.505). Table 5 reports,

Table 4: **Unanimity fraction f per setting.** Range over the ten seed-0 runs of all four variants in each setting, read from the final step of each run.

Task	Track	Model	f range (%)
SQuAD	FT	OPT-1.3B	63.1–68.1
SQuAD	FT	OPT-6.7B	58.8–65.4
SQuAD	FT	Qwen2.5-7B	48.6–56.7
SQuAD	FT	Llama-3.1-8B	70.4–73.7
SQuAD	LoRA	OPT-1.3B	56.4–62.4
SQuAD	LoRA	OPT-6.7B	49.4–58.6
SQuAD	LoRA	Qwen2.5-7B	62.8–70.4
SQuAD	LoRA	Llama-3.1-8B	43.1–52.4
SST-2	FT	OPT-1.3B	44.7–56.3
SST-2	FT	OPT-6.7B	38.8–48.8
SST-2	FT	Qwen2.5-7B	12.8–22.8
SST-2	FT	Llama-3.1-8B	35.5–40.9
SST-2	LoRA	OPT-1.3B	32.8–46.6
SST-2	LoRA	OPT-6.7B	33.6–46.9
SST-2	LoRA	Qwen2.5-7B	29.6–39.1
SST-2	LoRA	Llama-3.1-8B	16.9–37.2

per setting, its AUC minus the AUC of the same attack on the other seeds' models, and Table 6 summarizes every attack.

Table 5: **Black-box membership inference on S^* at $\mathrm{MI} = 0$.** RMIA with the pretrained model as reference: AUC of the seed-2 checkpoint minus the AUC of the same attack on the other seeds' models, with its 95% confidence interval. Zero means no separation beyond what the records carry.

Task	Track	Model	PACZERO-AGC$_0$	PACZERO-BAC$_0$
SST-2	FT	OPT-1.3B	-0.007 $[-0.020, +0.007]$	-0.005 $[-0.017, +0.008]$
SST-2	FT	OPT-6.7B	$+0.003$ $[-0.011, +0.016]$	-0.003 $[-0.020, +0.013]$
SST-2	FT	Qwen2.5-7B	-0.005 $[-0.016, +0.006]$	-0.003 $[-0.015, +0.010]$
SST-2	FT	Llama-3.1-8B	0.000 $[-0.017, +0.017]$	$+0.006$ $[-0.006, +0.018]$
SST-2	LoRA	OPT-1.3B	-0.013 $[-0.024, -0.002]$	0.000 $[-0.014, +0.015]$
SST-2	LoRA	OPT-6.7B	-0.002 $[-0.016, +0.012]$	$+0.001$ $[-0.013, +0.015]$
SST-2	LoRA	Qwen2.5-7B	-0.006 $[-0.020, +0.008]$	-0.016 $[-0.031, -0.001]$
SST-2	LoRA	Llama-3.1-8B	$+0.013$ $[-0.007, +0.031]$	$+0.004$ $[-0.018, +0.026]$
SQuAD	FT	OPT-1.3B	$+0.001$ $[-0.005, +0.007]$	$+0.005$ $[-0.001, +0.011]$
SQuAD	FT	OPT-6.7B	-0.004 $[-0.013, +0.005]$	-0.002 $[-0.013, +0.009]$
SQuAD	FT	Qwen2.5-7B	0.000 $[-0.007, +0.007]$	-0.006 $[-0.013, 0.000]$
SQuAD	FT	Llama-3.1-8B	0.000 $[-0.008, +0.008]$	$+0.008$ $[-0.003, +0.021]$
SQuAD	LoRA	OPT-1.3B	-0.004 $[-0.016, +0.009]$	$+0.002$ $[-0.010, +0.013]$
SQuAD	LoRA	OPT-6.7B	-0.007 $[-0.020, +0.007]$	$+0.004$ $[-0.012, +0.021]$
SQuAD	LoRA	Qwen2.5-7B	$+0.002$ $[-0.007, +0.011]$	-0.003 $[-0.012, +0.006]$
SQuAD	LoRA	Llama-3.1-8B	$+0.010$ $[-0.009, +0.029]$	$+0.002$ $[-0.019, +0.023]$

I MI-budget sweep on SST-2 FT OPT-1.3B

This is the PACZERO-AGC budget sweep summarized in §4.2, and Figure 2 plots it with the LoRA sweep of Appendix J. Over nearly four decades of total MI budget, test accuracy stays between 88.30 and 90.48%. Dev accuracy stays between 89.60 and 90.20% and the unanimity fraction between 52.4 and 56.0%. In contrast, DP-AggZO at $K = 64$ and $T = 1000$ gains 18.47 test-accuracy points between $\varepsilon = 0.5$ and $\varepsilon = 1.0$ on the same setting (Table 9 in Appendix L).

J MI-budget sweep on SST-2 LoRA OPT-1.3B

Table 8 and Figure 2 give the LoRA counterpart of the FT sweep of Table 7. Across the same nearly four decades of total MI budget, test accuracy stays between 88.65 and 90.94% and dev accuracy

Table 6: **Every black-box attack on S^* at** $\mathrm{MI}=0$, over the seed-2 PACZERO-AGC$_0$ and PACZERO-BAC$_0$ checkpoints of every setting: AUC median and range, and the true-positive rate at 1% false-positive rate (mean and maximum, in %). A dash marks an attack whose score takes few distinct values, so that no threshold reaches a 1% false-positive rate.

Attack	AUC median [min, max]	TPR at 1% FPR, mean [max]
LOSS	0.498 $[0.476, 0.518]$	1.1 $[2.7]$
Loss calibrated by the pretrained model	0.499 $[0.478, 0.531]$	1.9 $[3.8]$
RMIA	0.502 $[0.481, 0.537]$	1.7 $[3.6]$
Min-K%	0.493 $[0.478, 0.519]$	0.9 $[2.7]$
Min-K%++	0.493 $[0.475, 0.521]$	1.2 $[3.0]$
zlib	0.489 $[0.481, 0.496]$	0.7 $[1.5]$
Label-only	0.496 $[0.477, 0.513]$	–
EZ-MIA	0.507 $[0.484, 0.536]$	1.4 $[2.7]$
Offline LiRA	0.500 $[0.467, 0.529]$	1.2 $[4.1]$

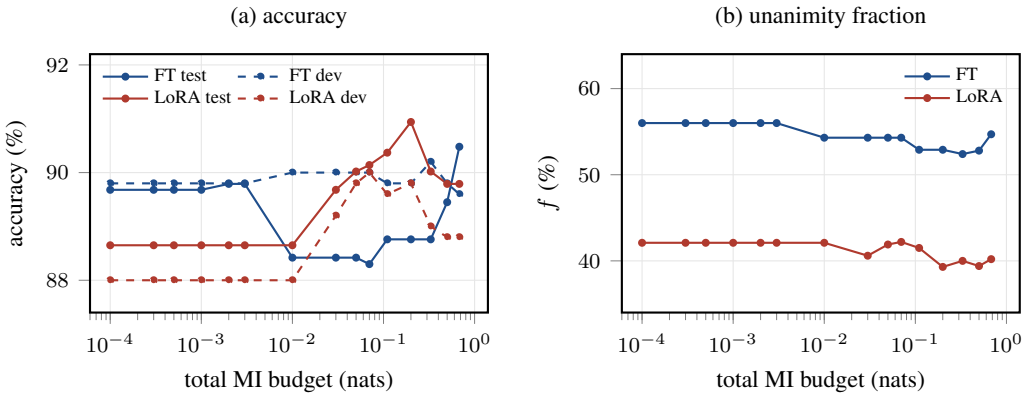

Figure 2: **The MI-budget sweep on SST-2 OPT-1.3B.** PACZERO-AGC with fixed hyperparameters on each track, run at 15 total MI budgets spanning 10^{-4} to 0.68 nats, each divided over the run by (5), FT at $T=1000$ and LoRA at $T=2000$, seed 0. Panel (a) plots test and dev accuracy, panel (b) the unanimity fraction f. The point values are in Tables 7 and 8.

between 88.00 and 90.00%. The unanimity fraction stays between 39.3 and 42.2% with no monotone dependence on the budget, because unanimity depends on whether the subset means agree in sign, which the noise level affects only through earlier updates.

K DP-AggZO parity check

Our in-house DP baselines run the released DP-AggZO code, and on SST-2 OPT-1.3B FT at $\varepsilon=2$ they land within a point of Table 2 of Bao et al. [4]. DPZero reaches 87.39% against a published 86.6%, and DP-AggZO at $K=64$ reaches 90.25% against 91.2% (Table 9), single-seed on both sides. We therefore use the published OPT-6.7B results of Bao et al. [4] as the comparators in Table 1.

L In-house DP baseline reproduction

This appendix reports the ε sweep behind the DP comparison of §4.

Setup. Table 9 is at OPT-1.3B, the scale at which both Zhang et al. [56] and Bao et al. [4] release configurations and one that Table 1 omits. The PACZERO results at this scale are in Appendix F. DPZero $K=1$ runs at DPZero's published step count ($T=20{,}000$) [56] and at DP-AggZO's learning rate and clipping threshold ($5\cdot10^{-6}$ and $c=25$), so the two DP rows are matched on both. Both DP rows use DP-AggZO's sample rate of 0.064, so the two are matched on subsampling. DPZero's published rate is 0.008. DP-AggZO $K=64$ follows the released SST-2 example configuration of Bao et al. [4] ($T=1000$, lr$=5\cdot10^{-6}$, $c=25$, sample-rate 0.064). The matched-forwards row holds the

Table 7: **MI-budget sweep on SST-2 FT OPT-1.3B.** PACZERO-AGC with adaptive β_t at the hyperparameters of Table 11, $T = 1000$, lr$= 10^{-4}$, clip 100, $M = 128$, seed 0. **Bold** marks the dev-best run.

MI (nats)	10^{-4}	$3{\cdot}10^{-4}$	$5{\cdot}10^{-4}$	10^{-3}	$2{\cdot}10^{-3}$	$3{\cdot}10^{-3}$	10^{-2}	$3{\cdot}10^{-2}$	$5{\cdot}10^{-2}$	0.07	0.11	0.20	0.33	0.50	0.68
dev %	89.80	89.80	89.80	89.80	89.80	89.80	90.00	90.00	90.00	90.00	89.80	89.80	**90.20**	89.80	89.60
test %	89.68	89.68	89.68	89.68	89.79	89.79	88.42	88.42	88.42	88.30	88.76	88.76	88.76	89.45	90.48
f %	56.0	56.0	56.0	56.0	56.0	56.0	54.3	54.3	54.3	54.3	52.9	52.9	52.4	52.8	54.7

Table 8: **MI-budget sweep on SST-2 LoRA OPT-1.3B.** PACZERO-AGC, $T = 2000$, lr $5 \cdot 10^{-4}$, $c = 25$, $M = 128$, seed 0. **Bold** marks the dev-best run.

MI (nats)	10^{-4}	$3{\cdot}10^{-4}$	$5{\cdot}10^{-4}$	10^{-3}	$2{\cdot}10^{-3}$	$3{\cdot}10^{-3}$	10^{-2}	$3{\cdot}10^{-2}$	$5{\cdot}10^{-2}$	0.07	0.11	0.20	0.33	0.50	0.68
dev %	88.00	88.00	88.00	88.00	88.00	88.00	88.00	89.20	89.80	**90.00**	89.60	89.80	89.00	88.80	88.80
test %	88.65	88.65	88.65	88.65	88.65	88.65	88.65	89.68	90.02	90.14	90.37	90.94	90.02	89.79	89.79
f %	42.1	42.1	42.1	42.1	42.1	42.1	42.1	40.6	41.9	42.2	41.5	39.3	40.0	39.4	40.2

forward passes at PACZERO's. PACZERO spends $2NT$ forward passes per run, two million at $N = 1000$ and $T = 1000$, and DP-AggZO spends $2KBT$ at an expected batch of $B = 64$, which approximately matches that budget at $T = 244$ when $K = 64$ (1,998,848). Learning rate, clipping threshold, K and sampling rate are identical across the two rows, so T is the only thing that moves. The noise multiplier is solved from the Opacus [51] RDP accountant for the run's own step count, so at a fixed ε a shorter run composes fewer times and needs less noise per step, roughly in proportion to the square root of the step count. Shortening the run therefore helps where the per-step noise dominates, taking $\varepsilon = 0.3$ from 62.96 to 78.33 and $\varepsilon = 0.5$ from 68.46 to 83.49, and costs accuracy once $\varepsilon \geq 1$, where the noise is no longer the binding constraint, so fewer steps is less training. At $\varepsilon = 0.3$ and $\varepsilon = 0.5$ most of the drop in the $T = 1000$ row is therefore a step-budget effect. Doubling the run to $T = 2000$ at $\varepsilon = 0.2$ leaves it at chance (49.66%). Every DP row stays below both MI $= 0$ PACZERO entries at every $\varepsilon \leq 1$, where the best DP entry reaches 86.93%. At $\varepsilon = 2$ the strongest in-house DP run is $K = 64$ FT at 90.25%, against $90.10 \pm 0.35\%$ for PACZERO-BAC$_0$ and $89.83 \pm 0.54\%$ for PACZERO-AGC$_0$ at MI $= 0$.

M PACZERO against the DP comparators

Against the DP comparators, PACZERO-BAC$_0$ at MI $= 0$ is 2.2 points below DP-AggZO $K = 16$ at $\varepsilon = 2$ on SST-2 FT OPT-6.7B (91.63 against 93.8) and 5.8 F1 below it on SQuAD FT (77.10 against 82.9). On Llama-3.1-8B FT, where both DP baselines run in-house, PACZERO-BAC$_0$ at MI $= 0$ is 1.03 points below the strongest DP entry at $\varepsilon = 2$ on SST-2 (92.78 against 93.81 for DP-AggZO $K = 16$) and 0.61 F1 below it on SQuAD (87.80 against 88.41 for DPZero $K = 1$).

Table 10 gives the hyperparameters and the forward-pass count of every row of Table 1 and of Appendix F. PACZERO and the non-private ZO reference spend $2.0 \cdot 10^6$ forward passes per FT run, the DP-AggZO $K = 16$ runs of Bao et al. [4] at most $2.05 \cdot 10^6$, and the in-house Llama-3.1-8B DP runs between $1.28 \cdot 10^5$ (DPZero on SQuAD) and $2.56 \cdot 10^6$ (DPZero on SST-2).

We also ran DP-AggZO on SQuAD at matched forward passes, on OPT-1.3B FT. At an expected DP batch of 64, the $2.0 \cdot 10^6$ forward passes of PACZERO give $T = 977$ at $K = 16$ and $T = 244$ at $K = 64$, each at the released SQuAD configuration of Bao et al. [4] (lr 10^{-5} and $c = 7.5$ at $K = 16$, lr $6 \cdot 10^{-5}$ and $c = 2$ at $K = 64$, sample rate 0.064). Those two runs reach 75.74 and 73.29 F1 at $\varepsilon = 2.0$, against $69.56_{\pm 0.25}$ for PACZERO-BAC$_0$ at MI $= 0$ in the same setting (Table 3). At matched compute and $\varepsilon = 2.0$, DP-AggZO is ahead of PACZERO-BAC$_0$ on SQuAD OPT-1.3B by 3.7 to 6.2 F1, while on SST-2 at matched compute DP-AggZO trails PACZERO-BAC$_0$ at every ε (Table 9).

N Hyperparameters of every setting

Table 11 gives the learning rate, clipping threshold and step budget of every setting of Table 1 and of Appendix F, with $M = 128$ throughout. After a first pass that fixes a provisional clipping threshold, the learning rate is the dev-best value of the setting's sweep (Table 13) at the setting's clipping threshold, and the clipping threshold is the dev-best value of the setting's clipping-threshold

Table 9: **In-house DP baseline reproduction on SST-2 OPT-1.3B FT** (test accuracy %, single-seed). The matched-forwards row runs at the step count whose $2KBT$ forward passes approximately match PACZERO's $2NT$ at $T=1000$, and ε is at $\delta=10^{-5}$. **Bold** marks the best accuracy at each ε.

Method	T	$\varepsilon=0.2$	$\varepsilon=0.3$	$\varepsilon=0.5$	$\varepsilon=1.0$	$\varepsilon=2.0$
DPZero $K=1$ FT	20,000	49.31	50.34	51.72	67.78	87.39
DP-AggZO $K=64$ FT	1000	48.85	62.96	68.46	**86.93**	**90.25**
DP-AggZO $K=64$ FT, doubled steps	2000	49.66	—	—	—	—
DP-AggZO $K=64$ FT, matched forwards	244	**64.11**	**78.33**	**83.49**	83.14	83.37

ablation (Appendix P) at the selected learning rate. The one exception is SST-2 OPT-6.7B LoRA (Appendix P).

Learning rate at other clipping thresholds. The learning rate of every setting is selected by the sweep in Table 13, on dev accuracy or dev F1, with the clipping threshold and the step budget held at the setting's values. Table 12 compares learning rates on SST-2 OPT-1.3B at other clipping thresholds. On FT at $c=1000$ the dev-best rate is again 10^{-4}, and on LoRA without clipping $5\cdot10^{-4}$ beats $2\cdot10^{-4}$. Both comparisons favor the learning rates selected at the settings' own clipping thresholds.

Table 10: **Hyperparameters and forward passes per run for the rows of Table 1.** A forward pass here is one loss evaluation on one training example, so a run costs $2KBT$ for the DP methods at expected batch B and $2NT$ for PACZERO and the reference, which evaluate all $N=1000$ training examples at $\theta\pm\mu z$ on every step. Evaluation passes are excluded. Bao et al. [4] report each DP entry as the best over the grids of Tables 6 and 7 of their technical report [5] without naming the selected configuration, so those rows give the range. The in-house Llama-3.1-8B rates are the dev-best grid value at $\varepsilon=2$, applied at both ε, over $\{2\cdot10^{-7}, 5\cdot10^{-7}, 10^{-6}, 5\cdot10^{-6}, 2\cdot10^{-5}\}$ for DPZero and $\{10^{-6}, 5\cdot10^{-6}, 2\cdot10^{-5}\}$ for DP-AggZO on SST-2, and $\{2\cdot10^{-6}, 10^{-5}, 5\cdot10^{-5}\}$ for both on SQuAD. PACZERO rows use the hyperparameters of Table 11.

Row	T	K	B	clip	lr	forward passes
OPT-6.7B, SST-2 and SQuAD, as reported by Bao et al. [4]						
DPZero $K=1$	$\{1, 2, 5, 10, 20, 50, 100\}\cdot10^3$	1	$\{8, 32, 64\}$	grid	grid	$1.6\cdot10^4$ to $1.28\cdot10^7$
DP-AggZO $K=16$	1000	16	$\{8, 32, 64\}$	grid	grid	$2.56\cdot10^5$ to $2.05\cdot10^6$
Llama-3.1-8B, in-house (†), seed 0*, sample rate* 0.064						
DPZero $K=1$, SST-2	20,000	1	64	25	10^{-6}	$2.56\cdot10^6$
DP-AggZO $K=16$, SST-2	1000	16	64	25	$5\cdot10^{-6}$	$2.05\cdot10^6$
DPZero $K=1$, SQuAD	1000	1	64	7.5	10^{-5}	$1.28\cdot10^5$
DP-AggZO $K=16$, SQuAD	1000	16	64	7.5	10^{-5}	$2.05\cdot10^6$
All models						
PACZERO, four variants, FT and SQuAD LoRA	1000	1	N	Table 11	Table 11	$2.0\cdot10^6$
PACZERO, four variants, SST-2 LoRA	2000	1	N	Table 11	Table 11	$4.0\cdot10^6$
Non-private ZO reference, FT	1000	1	N	1000	Table 14	$2.0\cdot10^6$
Non-private ZO reference, LoRA	2000 (SST-2) / 1000 (SQuAD)	1	N	1000	Table 14	$4.0\cdot10^6$ / $2.0\cdot10^6$

Table 11: **Hyperparameters of every setting.** All four variants of a setting run these values, with $M = 128$ throughout. The sweep that selects each learning rate is Table 13.

Task	Track	Model	lr	clip	T
SQuAD	FT	OPT-1.3B	10^{-4}	1000	1000
SQuAD	FT	OPT-6.7B	10^{-4}	250	1000
SQuAD	FT	Qwen2.5-7B	$5 \cdot 10^{-5}$	250	1000
SQuAD	FT	Llama-3.1-8B	$5 \cdot 10^{-5}$	250	1000
SQuAD	LoRA	OPT-1.3B	10^{-3}	10	1000
SQuAD	LoRA	OPT-6.7B	10^{-3}	10	1000
SQuAD	LoRA	Qwen2.5-7B	10^{-3}	25	1000
SQuAD	LoRA	Llama-3.1-8B	10^{-3}	25	1000
SST-2	FT	OPT-1.3B	10^{-4}	100	1000
SST-2	FT	OPT-6.7B	10^{-4}	100	1000
SST-2	FT	Qwen2.5-7B	10^{-4}	1000	1000
SST-2	FT	Llama-3.1-8B	10^{-4}	250	1000
SST-2	LoRA	OPT-1.3B	$5 \cdot 10^{-4}$	25	2000
SST-2	LoRA	OPT-6.7B	$5 \cdot 10^{-4}$	25	2000
SST-2	LoRA	Qwen2.5-7B	10^{-3}	10	2000
SST-2	LoRA	Llama-3.1-8B	10^{-3}	10	2000

Table 12: **PACZERO-AGC$_0$ learning-rate sweep at clipping thresholds the setting does not run, SST-2 OPT-1.3B.** Single seed, $M = 128$, LoRA at $T = 2000$ and FT at $T = 1000$. The settings run $c = 25$ on LoRA and $c = 100$ on FT (Table 11), and those runs are in Table 13. **Bold** marks the dev-best rate per track and clipping threshold. FT at $c = 10^9$ runs only the selected rate.

Track	clip c	lr	dev %	test %
LoRA	10^9	$2 \cdot 10^{-4}$	75.60	71.10
LoRA	$\mathbf{10^9}$	$\mathbf{5 \cdot 10^{-4}}$	**89.80**	**90.14**
FT	1000	$5 \cdot 10^{-5}$	80.40	79.59
FT	**1000**	$\mathbf{10^{-4}}$	**86.80**	**88.42**
FT	1000	$5 \cdot 10^{-4}$	60.60	56.42
FT	$\mathbf{10^9}$	$\mathbf{10^{-4}}$	**86.40**	**88.42**

O Learning-rate sweeps

Table 13 gives the sweep that selected the learning rate of each setting. Every sweep run uses PACZERO-AGC$_0$, so each setting's rate is tuned on the $\mathrm{MI}=0$ coin release and inherited unchanged by the other three variants in Table 1 and Appendix F.

Table 13: **Learning-rate sweeps, one per setting.** Single seed, each sweep at its setting's clipping threshold and step budget (Table 11), varying the learning rate alone. The dev-best grid value is the selected value and is shown in bold.

Task	Track	Model	lr	dev %
SQuAD	FT	OPT-1.3B	10^{-5}	33.78
			$5 \cdot 10^{-5}$	52.02
			$\mathbf{10^{-4}}$	**59.07**
			$5 \cdot 10^{-4}$	0.00
			10^{-3}	0.00
SQuAD	FT	OPT-6.7B	10^{-5}	46.03
			$5 \cdot 10^{-5}$	61.86
			$\mathbf{10^{-4}}$	**66.13**
			$5 \cdot 10^{-4}$	0.00
			10^{-3}	0.00
SQuAD	FT	Qwen2.5-7B	10^{-5}	77.68
			$\mathbf{5 \cdot 10^{-5}}$	**81.08**
			10^{-4}	79.71
			$5 \cdot 10^{-4}$	58.80
			10^{-3}	0.34
SQuAD	FT	Llama-3.1-8B	10^{-5}	66.82
			$\mathbf{5 \cdot 10^{-5}}$	**79.41**
			10^{-4}	79.27
			$5 \cdot 10^{-4}$	0.00
			10^{-3}	0.00
SQuAD	LoRA	OPT-1.3B	10^{-4}	32.05
			$5 \cdot 10^{-4}$	51.03
			$\mathbf{10^{-3}}$	**55.79**
			$2 \cdot 10^{-3}$	42.40
			$4 \cdot 10^{-3}$	0.00
			$8 \cdot 10^{-3}$	0.00
SQuAD	LoRA	OPT-6.7B	10^{-4}	42.14
			$5 \cdot 10^{-4}$	62.64
			$\mathbf{10^{-3}}$	**64.62**
			$2 \cdot 10^{-3}$	50.94
			$4 \cdot 10^{-3}$	0.00
			$8 \cdot 10^{-3}$	0.00
SQuAD	LoRA	Qwen2.5-7B	10^{-4}	78.33
			$5 \cdot 10^{-4}$	73.75
			$\mathbf{10^{-3}}$	**81.35**
			$2 \cdot 10^{-3}$	79.38
			$4 \cdot 10^{-3}$	74.81
			$8 \cdot 10^{-3}$	0.81
SQuAD	LoRA	Llama-3.1-8B	10^{-4}	62.43
			$5 \cdot 10^{-4}$	70.35
			$\mathbf{10^{-3}}$	**77.90**

Task	Track	Model	lr	dev %
			$2 \cdot 10^{-3}$	67.62
			$4 \cdot 10^{-3}$	2.41
			$8 \cdot 10^{-3}$	0.00
SST-2	FT	OPT-1.3B	$5 \cdot 10^{-5}$	81.40
			$\mathbf{10^{-4}}$	**88.00**
			$5 \cdot 10^{-4}$	58.60
SST-2	FT	OPT-6.7B	$5 \cdot 10^{-6}$	62.80
			10^{-5}	65.60
			$5 \cdot 10^{-5}$	83.00
			$\mathbf{10^{-4}}$	**88.40**
			$5 \cdot 10^{-4}$	58.00
SST-2	FT	Qwen2.5-7B	$5 \cdot 10^{-5}$	79.80
			$\mathbf{10^{-4}}$	**84.80**
			$5 \cdot 10^{-4}$	73.20
SST-2	FT	Llama-3.1-8B	$5 \cdot 10^{-6}$	61.80
			10^{-5}	64.40
			$5 \cdot 10^{-5}$	83.60
			$\mathbf{10^{-4}}$	**90.00**
			$5 \cdot 10^{-4}$	58.00
SST-2	LoRA	OPT-1.3B	10^{-4}	63.40
			$\mathbf{5 \cdot 10^{-4}}$	**88.80**
			10^{-3}	88.20
			$2 \cdot 10^{-3}$	82.60
SST-2	LoRA	OPT-6.7B	10^{-4}	65.20
			$2 \cdot 10^{-4}$	74.80
			$\mathbf{5 \cdot 10^{-4}}$	**90.80**
			10^{-3}	84.00
			$2 \cdot 10^{-3}$	82.00
			$4 \cdot 10^{-3}$	58.20
			$8 \cdot 10^{-3}$	57.40
SST-2	LoRA	Qwen2.5-7B	$5 \cdot 10^{-4}$	80.40
			$\mathbf{10^{-3}}$	**86.40**
			$2 \cdot 10^{-3}$	85.80
			$4 \cdot 10^{-3}$	80.00
			$8 \cdot 10^{-3}$	50.80
SST-2	LoRA	Llama-3.1-8B	10^{-4}	61.60
			$2 \cdot 10^{-4}$	61.80
			$5 \cdot 10^{-4}$	68.00
			$\mathbf{10^{-3}}$	**87.80**
			$2 \cdot 10^{-3}$	69.60
			$4 \cdot 10^{-3}$	58.00
			$8 \cdot 10^{-3}$	58.60

Table 14 gives the sweep that selected the learning rate of the non-private ZO reference in each setting.

Table 14: **Learning-rate sweeps of the non-private ZO reference, one per setting.** Single seed, privacy disabled, $c = 1000$, each sweep at its setting's step budget, plus OPT-6.7B FT at $T = 4000$. The dev-best grid value is the selected value and is shown in bold. n.f. marks a run stopped by the trainer's non-finite guard. At the matched step budget, where a separate reference run exists at the selected rate, Table 1 and Appendix F report that run, over three seeds for SST-2 Llama-3.1-8B FT and SQuAD OPT-6.7B FT. The $T = 4000$ values in Appendix E are those of the selected sweep runs.

Task	Track	Model	T	lr	dev %
SQuAD	FT	OPT-1.3B	1000	10^{-7}	36.30
				$5 \cdot 10^{-7}$	51.62
				10^{-6}	55.83
				$2 \cdot 10^{-6}$	58.44
				$\mathbf{3 \cdot 10^{-6}}$	**63.76**
				$5 \cdot 10^{-6}$	52.12
				10^{-5}	0.00
SQuAD	FT	OPT-6.7B	1000	$5 \cdot 10^{-7}$	61.51
				10^{-6}	65.20
				$\mathbf{2 \cdot 10^{-6}}$	**66.49**
				$3 \cdot 10^{-6}$	53.93
				$5 \cdot 10^{-6}$	0.00
				$8 \cdot 10^{-6}$	0.00
SQuAD	FT	OPT-6.7B	4000	10^{-6}	71.95
				$\mathbf{2 \cdot 10^{-6}}$	**73.96**
				$3 \cdot 10^{-6}$	54.85
				$5 \cdot 10^{-6}$	3.11
SQuAD	FT	Qwen2.5-7B	1000	$2 \cdot 10^{-7}$	79.08
				$5 \cdot 10^{-7}$	81.36
				$\mathbf{10^{-6}}$	**81.76**
				$2 \cdot 10^{-6}$	80.92
				$3 \cdot 10^{-6}$	80.71
				$5 \cdot 10^{-6}$	78.76
				10^{-5}	70.65
				$2 \cdot 10^{-5}$	0.03
SQuAD	FT	Llama-3.1-8B	1000	$5 \cdot 10^{-7}$	74.45
				10^{-6}	76.45
				$\mathbf{2 \cdot 10^{-6}}$	**80.40**
				$3 \cdot 10^{-6}$	80.38
				$5 \cdot 10^{-6}$	0.00
SQuAD	LoRA	OPT-1.3B	1000	10^{-7}	26.15
				$5 \cdot 10^{-7}$	26.09
				10^{-6}	26.33
				$5 \cdot 10^{-6}$	28.99
				10^{-5}	31.76
				$5 \cdot 10^{-5}$	48.25
				10^{-4}	56.75
				$\mathbf{2 \cdot 10^{-4}}$	**61.75**
				$5 \cdot 10^{-4}$	0.00
SQuAD	LoRA	OPT-6.7B	1000	10^{-5}	39.87
				$5 \cdot 10^{-5}$	57.37
				10^{-4}	62.66
				$\mathbf{2 \cdot 10^{-4}}$	**69.34**

Task	Track	Model	T	lr	dev %
				$5 \cdot 10^{-4}$	0.00
				10^{-3}	0.00
				$2 \cdot 10^{-3}$	0.05
SQuAD	LoRA	Qwen2.5-7B	1000	10^{-5}	78.56
				$5 \cdot 10^{-5}$	78.49
				10^{-4}	77.19
				$2 \cdot 10^{-4}$	76.59
				$\mathbf{5 \cdot 10^{-4}}$	**81.50**
				10^{-3}	81.01
				$2 \cdot 10^{-3}$	0.00
SQuAD	LoRA	Llama-3.1-8B	1000	10^{-5}	61.98
				$5 \cdot 10^{-5}$	62.71
				10^{-4}	65.00
				$\mathbf{2 \cdot 10^{-4}}$	**76.88**
				$5 \cdot 10^{-4}$	66.87
				10^{-3}	0.00
				$2 \cdot 10^{-3}$	0.01
SST-2	FT	OPT-1.3B	1000	10^{-7}	60.40
				$5 \cdot 10^{-7}$	70.00
				10^{-6}	79.40
				$2 \cdot 10^{-6}$	82.20
				$\mathbf{5 \cdot 10^{-6}}$	**90.20**
				10^{-5}	89.80
				$2 \cdot 10^{-5}$	81.20
				$5 \cdot 10^{-5}$	57.40
SST-2	FT	OPT-6.7B	1000	10^{-6}	79.00
				$2 \cdot 10^{-6}$	82.80
				$\mathbf{5 \cdot 10^{-6}}$	**88.80**
				10^{-5}	87.40
				$2 \cdot 10^{-5}$	74.20
SST-2	FT	OPT-6.7B	4000	$2 \cdot 10^{-6}$	92.00
				$\mathbf{5 \cdot 10^{-6}}$	**92.20**
				10^{-5}	82.20
SST-2	FT	Qwen2.5-7B	1000	10^{-6}	78.60
				$2 \cdot 10^{-6}$	81.40
				$5 \cdot 10^{-6}$	85.00
				$\mathbf{10^{-5}}$	**85.60**
				$2 \cdot 10^{-5}$	80.20
				$5 \cdot 10^{-5}$	53.80
SST-2	FT	Llama-3.1-8B	1000	10^{-6}	81.60
				$2 \cdot 10^{-6}$	85.20
				$\mathbf{5 \cdot 10^{-6}}$	**90.00**
				10^{-5}	86.00
				$2 \cdot 10^{-5}$	57.20
SST-2	LoRA	OPT-1.3B	2000	10^{-8}	60.00
				10^{-7}	60.00
				10^{-6}	60.00
				10^{-5}	59.80
				10^{-4}	81.60
				$2 \cdot 10^{-4}$	90.00
				$\mathbf{5 \cdot 10^{-4}}$	**91.00**

Table 15: **Clipping-threshold ablation on SST-2 LoRA at $\mathrm{MI}=0.33$, dev accuracy.** Single seed, PACZERO-AGC, $T=2000$, $M=128$, each column at its setting's learning rate (Table 11). **Bold** marks the dev-winner. A dash marks a grid value outside the sweep.

clip	OPT-1.3B	OPT-6.7B	Qwen2.5-7B	Llama-3.1-8B
5	—	90.00	—	—
10	88.20	**90.20**	**85.60**	**86.80**
25	**89.00**	89.60	84.20	86.40
50	88.00	88.60	83.40	85.80

Task	Track	Model	T	lr	dev %
				10^{-3}	57.40
				$2\cdot10^{-3}$	55.80
SST-2	LoRA	OPT-6.7B	2000	10^{-5}	62.80
				$5\cdot10^{-5}$	67.20
				10^{-4}	81.20
				$2\cdot10^{-4}$	88.40
				$\mathbf{5\cdot10^{-4}}$	**90.60**
				10^{-3}	60.00
				$2\cdot10^{-3}$	59.20
SST-2	LoRA	Qwen2.5-7B	2000	10^{-5}	66.60
				$5\cdot10^{-5}$	66.80
				10^{-4}	67.00
				$5\cdot10^{-4}$	79.60
				10^{-3}	86.40
				$\mathbf{2\cdot10^{-3}}$	**89.20**
				$4\cdot10^{-3}$	n.f.
				$8\cdot10^{-3}$	n.f.
SST-2	LoRA	Llama-3.1-8B	2000	10^{-5}	60.80
				$5\cdot10^{-5}$	61.20
				10^{-4}	61.80
				$2\cdot10^{-4}$	65.00
				$\mathbf{5\cdot10^{-4}}$	**88.40**
				10^{-3}	85.40
				$2\cdot10^{-3}$	59.20

P Clipping-threshold ablations

LoRA clipping-threshold ablation. Table 15 reports the clipping-threshold ablation for SST-2 LoRA across all four models. The FT grid is Table 16. Each clipping threshold is a seed-0 run at the setting's learning rate and step budget, and the table reports dev because the clipping threshold is selected on dev. Every setting runs its dev-best clipping threshold except OPT-6.7B, which runs $c=25$ at 89.60 against 90.20 at $c=10$.

Full fine-tuning clipping-threshold ablation. Table 16 reports the FT clipping-threshold ablation on both tasks and all four models, three clipping thresholds each, at the setting's learning rate and step budget. Every setting runs its dev-best clipping threshold, and the largest dev margin over the runner-up is 2.8 points (SST-2 Llama-3.1-8B). Two of the eight settings select $c=1000$. Across the eight runs at $c=1000$ and these step budgets, clipping changes at most 0.25% of the per-sample finite-difference scalars on average over steps. At least one scalar is clipped on 18.7% to 80.8% of steps.

Table 17 reports the same ablation on the SQuAD LoRA track, where every setting runs its dev-best clipping threshold.

Table 16: **Clipping-threshold ablation on the FT track, dev metric (%).** Dev accuracy (SST-2) and dev F1 (SQuAD), PACZERO-AGC at MI$=0.33$, $M=128$, seed 0, at each setting's step budget. **Bold** marks the dev-best clipping threshold, which is the clipping threshold the setting runs.

Task	Model	$c=100$	$c=250$	$c=1000$	selected
SQuAD	OPT-1.3B	60.80	59.82	**61.64**	1000
SQuAD	OPT-6.7B	68.56	**69.23**	67.39	250
SQuAD	Qwen2.5-7B	81.33	**82.22**	81.52	250
SQuAD	Llama-3.1-8B	79.42	**79.52**	78.59	250
SST-2	OPT-1.3B	**90.20**	88.00	89.20	100
SST-2	OPT-6.7B	**90.80**	90.20	88.00	100
SST-2	Qwen2.5-7B	81.40	81.60	**84.20**	1000
SST-2	Llama-3.1-8B	86.20	**89.60**	86.80	250

Table 17: **Clipping-threshold ablation on the SQuAD LoRA track, dev F1 (%).** Dev F1, PACZERO-AGC at MI $= 0.33$, $M = 128$, $T = 1000$, lr 10^{-3}, seed 0. Runs differ only in the clipping threshold. **Bold** marks the dev-best clipping threshold, which is the clipping threshold the setting runs.

Model	$c=10$	$c=25$	$c=50$	selected
OPT-1.3B	**60.46**	57.98	59.02	10
OPT-6.7B	**68.78**	66.43	65.30	10
Qwen2.5-7B	80.33	**80.81**	80.65	25
Llama-3.1-8B	76.78	**80.08**	78.78	25

Q LoRA rank ablation

Table 18 reports the LoRA rank ablation. The dev-best rank is $r = 4$, and every setting runs the standard $r=8$.

Table 18: **LoRA rank ablation on SST-2 OPT-6.7B.** PACZERO-AGC, $T=2000$, lr $5 \cdot 10^{-4}$, $c=25$, $M=128$, MI$=0.33$, seed 0. **Bold** marks the dev-winner.

rank r	dev %	test %
4	**90.40**	**92.20**
8	89.60	92.43
16	85.80	91.51

R Mechanism decomposition

Table 19 removes the channel from the PACZERO per-step release and replaces the released bit with one quantity computed from the step's clipped per-sample finite-difference scalars. The independent coin is the negative control. It stays at 54.24%, so the released bit carries the optimization signal. Sign quantization is T-sensitive. At $T = 1000$ the secret-subset sign lags the secret-subset mean by 6.4 points, and at $T=2000$ that gap closes to 0.7. The universe mean with magnitude was run only at $T=2000$, where its sign lags it by 1.15 points. Sign quantization therefore costs steps, and doubling T closes most of the gap. Restricting the aggregate to the secret subset costs 0.8 points against the universe mean at $T=2000$. With the channel in place PACZERO-AGC reaches 88.65% at MI$=0.001$ and $89.98 \pm 0.18\%$ at MI$=0.33$, both at $T=2000$.

S PACZERO-AGC$_0$ step-budget sweep

PACZERO-AGC$_0$ releases an unbiased coin on every disagreement step. Table 20 varies the step budget on SST-2 OPT-1.3B FT with no clipping (lr 10^{-4}, $M=128$, seed 0) to test whether accuracy degrades as coin-flip releases accumulate over a long run. Dev accuracy moves from 86.40 at $T=1000$ to 90.40 at $T=5000$ and test from 88.42 to 89.68, with no downward drift over the sweep.

Table 19: **Mechanism decomposition on SST-2 LoRA OPT-1.3B.** The rows above the rule use seed 0 at the setting's hyperparameters (lr $5 \cdot 10^{-4}$, $c=25$, $M=128$), with the channel removed and no mutual-information accounting. Each row above the rule replaces the released bit with one quantity computed from the step's clipped per-sample finite-difference scalars, and the last of them replaces it with an independent coin. Each T is a separate run whose learning-rate decay spans its own step budget. The two rows below the rule are PACZERO-AGC with the channel in place, at $\mathrm{MI}=0.001$ under the uniform budget allocation $\beta_t=\mathrm{MI}_{\mathrm{tot}}/T$ and at $\mathrm{MI}=0.33$ under (5).

Released quantity	T	Test %
Universe mean, with magnitude	2000	90.60
Sign of the universe mean	2000	89.45
Sign of the universe mean	1000	81.77
Secret-subset mean, with magnitude	2000	89.79
Secret-subset mean, with magnitude	1000	88.65
Secret-subset sign, no channel	2000	89.11
Secret-subset sign, no channel	1000	82.22
Independent ± 1 coin, every step	2000	54.24
PACZERO-AGC, $\mathrm{MI}=0.001$, uniform β_t, seed 0	2000	88.65
PACZERO-AGC, $\mathrm{MI}=0.33$, adaptive β_t, $n=3$	2000	$89.98_{\pm 0.18}$

Table 20: **PACZERO-AGC$_0$ step-budget sweep on SST-2 OPT-1.3B FT.** No clipping, lr 10^{-4}, $M=128$, seed 0.

T	1000	2000	3000	4000	5000
dev %	86.40	89.00	90.00	88.80	90.40
test %	88.42	89.79	89.11	89.68	89.68

T K-aggregation ablation (DP-AggZO-style)

The DP-AggZO mechanism of Bao et al. [4] aggregates K independent zeroth-order direction estimates per training step before the privacy release. PACZERO as defined in §3 makes one release per step ($K=1$). We extend the per-step release to $K \in \{4, 16, 64\}$ in the DP-AggZO aggregation pattern. Each step samples K independent ZO directions, each direction is released through its own channel at budget β_t/K, the K bits $Y_{t,1}, \ldots, Y_{t,K} \in \{-1, +1\}$ are averaged, and the parameter update is $\theta_{t+1}=\theta_t - \eta_t(1/K)\sum_k Y_{t,k}\cdot z_k$. The posterior is updated after each of the K sub-releases, so every sub-channel is calibrated at the posterior conditioned on the sub-releases before it. Lemma 8 applies to each of the K releases given the ones before it, so by the chain rule the step is charged at most β_t and Theorem 3 applies unchanged. At $K=1$ the extension reproduces Algorithm 1, and the two produce identical transcripts.

Table 21 reports the K-ablation on SST-2 at $\mathrm{MI}=0.33$. The $K=1$ column is the three-seed entry of Table 2, and the $K>1$ entries are seed-0 runs at the same hyperparameters. The upper block runs every K at the setting's step budget, so a step costs K times the forward passes of $K=1$. The largest gains at that step budget are about one point on OPT-1.3B, 1.5 on OPT-6.7B LoRA and three on OPT-6.7B FT. The lower block holds the forward passes at the $K=1$ budget, so a $K>1$ run takes T/K steps at the learning rate selected for $K=1$. There every $K>1$ entry falls below $K=1$, to at most 76.95%. The gain of the upper block is bought with extra forward passes. Appendix M gives the DP-AggZO comparison at matched forward passes.

U Unanimity against the universe size

This section tests whether the unanimity the mechanism relies on survives as the universe grows.

Unanimity rises with the universe size. Table 22 sweeps the universe size N over an $8\times$ range on SST-2 OPT-1.3B LoRA under PACZERO-AGC$_0$, three seeds per point, and includes the $N=1000$ used by every setting. Unanimity rises monotonically across all four points, from 37.8% at $N=500$

Table 21: K**-aggregation ablation on SST-2, test accuracy %.** PACZERO-AGC at $\mathrm{MI}=0.33$, at the hyperparameters of Table 11, with $K>1$ runs at seed 0. The upper block runs every K at the setting's T. The lower block runs T/K steps (250, 63, 16 on FT and 500, 125, 32 on LoRA), so every run spends the forward passes of $K=1$ to within 2.4%. A dash marks a grid value outside the sweep.

Model	Track	$K=1$ ($n=3$)	$K=4$	$K=16$	$K=64$
Same step budget T					
OPT-1.3B	FT	$89.45_{\pm 0.69}$	89.68	90.25	89.45
OPT-1.3B	LoRA	$89.98_{\pm 0.18}$	90.94	90.71	—
OPT-6.7B	FT	$91.09_{\pm 1.03}$	94.38	94.15	—
OPT-6.7B	LoRA	$92.62_{\pm 0.33}$	94.15	93.46	—
Same forward passes, T/K steps					
OPT-1.3B	FT	$89.45_{\pm 0.69}$	67.43	56.42	54.24
OPT-1.3B	LoRA	$89.98_{\pm 0.18}$	61.01	54.59	53.78
OPT-6.7B	FT	$91.09_{\pm 1.03}$	76.95	63.53	61.24
OPT-6.7B	LoRA	$92.62_{\pm 0.33}$	66.74	62.50	61.47

to 55.0% at $N=4000$. Mean test accuracy does not follow it and spans 1.1 points, while unanimity moves 17.2 points.

Each candidate subset holds $N/2$ records in expectation, so a larger universe gives larger subsets. On this sweep the M subset means agree in sign more often as N grows, so PACZERO-AGC$_0$ releases the common sign more often. Appendix V reports what a larger universe costs per step.

Table 22: **Unanimity against universe size** N, SST-2 OPT-1.3B LoRA, PACZERO-AGC$_0$. This sweep is a separate study from the main settings and runs one set of hyperparameters, lr 10^{-3}, $c=25$, $T=1000$, $M=128$, so only N varies across its rows. Each point averages three seeds, which differ in the training and dev sample, in the public perturbation seed and in the private draw.

N	unanimity	mean test %
500	0.378	88.3
1000	0.429	88.0
2000	0.477	88.6
4000	0.550	89.1

V Matched per-step timing and the N and M sweeps

This section reports the cost of the mechanism directly. Appendix W gives the GPU-hours by kind of experiment. The matched comparison of Table 23 runs 60 steps per method, and the N and M sweeps of Table 24 run 10 steps per configuration. Every run is on a single exclusive H100 with no other job on the node, so the methods are compared under identical conditions.

Table 23: **Matched per-step cost on SST-2 OPT-1.3B**, $N=1000$, $M=128$, 60 steps, one exclusive H100 per run. "Job" is total job wall clock divided by 60 and therefore includes model load and final evaluation, and "loop" is the training-loop rate excluding them. Peak memory is the maximum resident GPU allocation sampled at 1 Hz over the whole job. Appendix M gives the per-step and per-run forward counts of the DP baselines.

	FT (s/step)		LoRA (s/step)		peak mem (MiB)	
Method	job	loop	job	loop	FT	LoRA
PACZERO-AGC	4.65	3.63	4.90	3.75	5679	5625
PACZERO-AGC$_0$	4.37	3.46	4.98	3.86	5669	5625
Non-private ZO reference	4.45	3.51	4.88	3.80	5661	5625

Cost. The three methods span 6.4% on FT and 2.0% on LoRA on the job metric, and 4.9% and 2.9% on the loop metric, and the ordering is not stable between the two tracks. Peak memory is within 54 MiB, 1.0%, across all six runs. The per-step cost of PACZERO-AGC and PACZERO-AGC$_0$ is that of the non-private ZO reference, which runs the MeZO update [28] over the full universe in the same trainer.

Cost grows with the universe size N. Table 24 sweeps N over an $8\times$ range at fixed $M = 128$. Per-step cost grows $5.26\times$ across that range. Fitting $t(N) = \kappa_0 + \kappa_1 N$ by least squares gives a marginal $\kappa_1 = 4.02$ ms per record and a fixed $\kappa_0 = 1.05$ s, and reproduces all four points to within 10.3%.

No monotone cost trend in the subset count M. The same table sweeps M over a $32\times$ range at fixed $N = 1000$. Per-step cost varies by 20.4% with no monotone trend in M.

Table 24: **Scaling in the universe size N and the subset count M**, SST-2 OPT-1.3B LoRA, 10 steps per configuration on one exclusive H100. The left block sweeps N at $M = 128$ and the right block sweeps M at $N = 1000$. "Fit" is $t(N) = 1.05 + 0.00402N$ seconds, fitted to the four N points.

N sweep ($M = 128$)			M sweep ($N = 1000$)	
N	s/step	fit	M	s/step
500	3.24	3.06	32	3.92
1000	4.60	5.07	128	4.60
2000	9.49	9.09	512	3.82
4000	17.03	17.13	1024	4.43

The full-universe batch bounds the corpus size. Because every step evaluates the loss on all N records, the mechanism as instantiated here does not scale to corpora far larger than $N = 1000$ without change. At $N = 4000$ a 1000-step run already costs about 4.7 GPU-hours on an H100, at the 17.03 s per step of Table 24. The natural next step is minibatched subset estimation, in which each step draws a minibatch and forms the M candidate subsets inside it. That changes the accounting of the guarantee, since the secret is then observed through a sampled view of the universe, and we leave the analysis to future work.

W Compute

Every reported run uses one NVIDIA H100 GPU. Reproducing every number in this paper takes about $3{,}600$ GPU-hours: about $1{,}930$ for the PACZERO runs of the sixteen settings, 920 for the learning-rate sweeps, 480 for the clipping-threshold, K, rank and mechanism ablations, 100 for the MI-budget sweeps and step-budget runs, 90 for the in-house DP baselines, 50 for the non-private references and 20 for the universe-size sweep. Scoring the checkpoints for the membership-inference attacks of Appendix H takes about 10 more. Including failed, superseded and exploratory runs, the project used about $8{,}000$ GPU-hours.

X Per-seed values for multi-seed entries

Table 25 gives the three per-seed test values behind each of the 160 PACZERO means of Table 1 and Appendix F, 80 per task, accuracy on SST-2 and F1 on SQuAD. The reported spread is the sample standard deviation ($n-1$ denominator).

Table 25: **Per-seed test values for every multi-seed PACZERO entry of Table 1 and Appendix F.** Accuracy on SST-2, F1 on SQuAD. Seeds 0, 1 and 2 in order.

Task	Track	Method	MI	Model	per-seed
SQuAD	FT	PACZERO-BAC$_0$	0	OPT-1.3B	69.67, 69.74, 69.27
SQuAD	FT	PACZERO-BAC$_0$	0	OPT-6.7B	78.23, 75.55, 77.51
SQuAD	FT	PACZERO-BAC$_0$	0	Qwen2.5-7B	88.20, 88.33, 88.64
SQuAD	FT	PACZERO-BAC$_0$	0	Llama-3.1-8B	87.50, 89.05, 86.86
SQuAD	FT	PACZERO-BAC	0.01	OPT-1.3B	69.71, 69.74, 68.95
SQuAD	FT	PACZERO-BAC	0.01	OPT-6.7B	78.23, 75.69, 77.29
SQuAD	FT	PACZERO-BAC	0.01	Qwen2.5-7B	88.20, 88.33, 88.47
SQuAD	FT	PACZERO-BAC	0.01	Llama-3.1-8B	87.71, 89.11, 86.87
SQuAD	FT	PACZERO-BAC	0.11	OPT-1.3B	69.81, 69.24, 69.03
SQuAD	FT	PACZERO-BAC	0.11	OPT-6.7B	78.27, 75.43, 77.50
SQuAD	FT	PACZERO-BAC	0.11	Qwen2.5-7B	88.31, 88.78, 88.45
SQuAD	FT	PACZERO-BAC	0.11	Llama-3.1-8B	87.53, 89.17, 86.82
SQuAD	FT	PACZERO-BAC	0.33	OPT-1.3B	70.77, 68.99, 68.93
SQuAD	FT	PACZERO-BAC	0.33	OPT-6.7B	77.01, 75.57, 77.50
SQuAD	FT	PACZERO-BAC	0.33	Qwen2.5-7B	88.26, 88.72, 88.45
SQuAD	FT	PACZERO-BAC	0.33	Llama-3.1-8B	87.64, 89.17, 86.53
SQuAD	FT	PACZERO-BAC	0.68	OPT-1.3B	71.20, 69.08, 69.03
SQuAD	FT	PACZERO-BAC	0.68	OPT-6.7B	75.87, 75.04, 76.60
SQuAD	FT	PACZERO-BAC	0.68	Qwen2.5-7B	88.41, 88.45, 88.33
SQuAD	FT	PACZERO-BAC	0.68	Llama-3.1-8B	87.57, 89.01, 86.64
SQuAD	FT	PACZERO-AGC	0.01	OPT-1.3B	68.91, 68.34, 65.84
SQuAD	FT	PACZERO-AGC	0.01	OPT-6.7B	76.70, 75.21, 72.05
SQuAD	FT	PACZERO-AGC	0.01	Qwen2.5-7B	87.82, 87.31, 87.93
SQuAD	FT	PACZERO-AGC	0.01	Llama-3.1-8B	87.48, 87.70, 87.87
SQuAD	FT	PACZERO-AGC	0.11	OPT-1.3B	69.05, 68.80, 65.85
SQuAD	FT	PACZERO-AGC	0.11	OPT-6.7B	77.43, 75.35, 73.67
SQuAD	FT	PACZERO-AGC	0.11	Qwen2.5-7B	88.23, 87.24, 87.65
SQuAD	FT	PACZERO-AGC	0.11	Llama-3.1-8B	86.94, 88.00, 86.65
SQuAD	FT	PACZERO-AGC	0.33	OPT-1.3B	69.05, 69.00, 65.99
SQuAD	FT	PACZERO-AGC	0.33	OPT-6.7B	76.86, 74.84, 75.25
SQuAD	FT	PACZERO-AGC	0.33	Qwen2.5-7B	88.10, 87.49, 88.01
SQuAD	FT	PACZERO-AGC	0.33	Llama-3.1-8B	87.46, 87.97, 87.79
SQuAD	FT	PACZERO-AGC	0.68	OPT-1.3B	68.66, 68.60, 66.67
SQuAD	FT	PACZERO-AGC	0.68	OPT-6.7B	76.02, 76.69, 73.62
SQuAD	FT	PACZERO-AGC	0.68	Qwen2.5-7B	88.27, 87.47, 88.26
SQuAD	FT	PACZERO-AGC	0.68	Llama-3.1-8B	87.78, 89.14, 87.30
SQuAD	FT	PACZERO-AGC$_0$	0	OPT-1.3B	67.58, 68.16, 67.39
SQuAD	FT	PACZERO-AGC$_0$	0	OPT-6.7B	75.22, 73.36, 75.91
SQuAD	FT	PACZERO-AGC$_0$	0	Qwen2.5-7B	88.43, 87.15, 87.52
SQuAD	FT	PACZERO-AGC$_0$	0	Llama-3.1-8B	87.10, 89.10, 85.97
SQuAD	LoRA	PACZERO-BAC$_0$	0	OPT-1.3B	70.14, 67.67, 69.99
SQuAD	LoRA	PACZERO-BAC$_0$	0	OPT-6.7B	76.14, 75.57, 74.60
SQuAD	LoRA	PACZERO-BAC$_0$	0	Qwen2.5-7B	87.99, 88.43, 88.59
SQuAD	LoRA	PACZERO-BAC$_0$	0	Llama-3.1-8B	86.47, 90.69, 87.44
SQuAD	LoRA	PACZERO-BAC	0.01	OPT-1.3B	69.10, 68.40, 68.96
SQuAD	LoRA	PACZERO-BAC	0.01	OPT-6.7B	75.53, 75.87, 73.17
SQuAD	LoRA	PACZERO-BAC	0.01	Qwen2.5-7B	88.27, 88.45, 87.85

Task	Track	Method	MI	Model	per-seed
SQuAD	LoRA	PACZERO-BAC	0.01	Llama-3.1-8B	88.94, 90.00, 87.44
SQuAD	LoRA	PACZERO-BAC	0.11	OPT-1.3B	68.02, 69.15, 70.57
SQuAD	LoRA	PACZERO-BAC	0.11	OPT-6.7B	74.70, 75.81, 73.17
SQuAD	LoRA	PACZERO-BAC	0.11	Qwen2.5-7B	88.27, 88.63, 87.93
SQuAD	LoRA	PACZERO-BAC	0.11	Llama-3.1-8B	88.94, 90.70, 87.56
SQuAD	LoRA	PACZERO-BAC	0.33	OPT-1.3B	68.79, 68.83, 69.22
SQuAD	LoRA	PACZERO-BAC	0.33	OPT-6.7B	76.61, 75.81, 73.91
SQuAD	LoRA	PACZERO-BAC	0.33	Qwen2.5-7B	88.27, 88.84, 87.83
SQuAD	LoRA	PACZERO-BAC	0.33	Llama-3.1-8B	89.16, 90.43, 87.09
SQuAD	LoRA	PACZERO-BAC	0.68	OPT-1.3B	69.19, 68.42, 69.24
SQuAD	LoRA	PACZERO-BAC	0.68	OPT-6.7B	76.30, 78.56, 73.45
SQuAD	LoRA	PACZERO-BAC	0.68	Qwen2.5-7B	87.55, 88.84, 87.83
SQuAD	LoRA	PACZERO-BAC	0.68	Llama-3.1-8B	89.16, 90.40, 87.24
SQuAD	LoRA	PACZERO-AGC	0.01	OPT-1.3B	68.90, 68.33, 66.60
SQuAD	LoRA	PACZERO-AGC	0.01	OPT-6.7B	74.28, 73.36, 73.33
SQuAD	LoRA	PACZERO-AGC	0.01	Qwen2.5-7B	87.76, 88.37, 86.63
SQuAD	LoRA	PACZERO-AGC	0.01	Llama-3.1-8B	87.71, 87.62, 85.13
SQuAD	LoRA	PACZERO-AGC	0.11	OPT-1.3B	68.47, 68.44, 66.69
SQuAD	LoRA	PACZERO-AGC	0.11	OPT-6.7B	73.98, 71.57, 69.54
SQuAD	LoRA	PACZERO-AGC	0.11	Qwen2.5-7B	88.47, 88.39, 87.31
SQuAD	LoRA	PACZERO-AGC	0.11	Llama-3.1-8B	87.72, 88.89, 85.62
SQuAD	LoRA	PACZERO-AGC	0.33	OPT-1.3B	68.41, 67.97, 69.45
SQuAD	LoRA	PACZERO-AGC	0.33	OPT-6.7B	75.67, 73.16, 72.97
SQuAD	LoRA	PACZERO-AGC	0.33	Qwen2.5-7B	88.56, 88.38, 86.98
SQuAD	LoRA	PACZERO-AGC	0.33	Llama-3.1-8B	88.12, 88.59, 87.87
SQuAD	LoRA	PACZERO-AGC	0.68	OPT-1.3B	70.31, 65.80, 67.20
SQuAD	LoRA	PACZERO-AGC	0.68	OPT-6.7B	73.02, 74.91, 71.96
SQuAD	LoRA	PACZERO-AGC	0.68	Qwen2.5-7B	88.77, 88.84, 87.95
SQuAD	LoRA	PACZERO-AGC	0.68	Llama-3.1-8B	88.33, 90.14, 87.35
SQuAD	LoRA	PACZERO-AGC$_0$	0	OPT-1.3B	64.55, 64.82, 66.86
SQuAD	LoRA	PACZERO-AGC$_0$	0	OPT-6.7B	74.18, 71.41, 70.77
SQuAD	LoRA	PACZERO-AGC$_0$	0	Qwen2.5-7B	88.45, 88.41, 87.74
SQuAD	LoRA	PACZERO-AGC$_0$	0	Llama-3.1-8B	86.65, 88.58, 88.92
SST-2	FT	PACZERO-BAC$_0$	0	OPT-1.3B	89.79, 90.02, 90.48
SST-2	FT	PACZERO-BAC$_0$	0	OPT-6.7B	91.28, 91.86, 91.74
SST-2	FT	PACZERO-BAC$_0$	0	Qwen2.5-7B	91.86, 92.20, 92.66
SST-2	FT	PACZERO-BAC$_0$	0	Llama-3.1-8B	92.78, 92.55, 93.00
SST-2	FT	PACZERO-BAC	0.01	OPT-1.3B	89.11, 89.68, 89.45
SST-2	FT	PACZERO-BAC	0.01	OPT-6.7B	91.97, 91.86, 91.74
SST-2	FT	PACZERO-BAC	0.01	Qwen2.5-7B	91.74, 92.55, 92.55
SST-2	FT	PACZERO-BAC	0.01	Llama-3.1-8B	93.35, 92.55, 93.12
SST-2	FT	PACZERO-BAC	0.11	OPT-1.3B	89.56, 89.68, 89.45
SST-2	FT	PACZERO-BAC	0.11	OPT-6.7B	91.28, 91.86, 91.63
SST-2	FT	PACZERO-BAC	0.11	Qwen2.5-7B	92.32, 92.43, 92.78
SST-2	FT	PACZERO-BAC	0.11	Llama-3.1-8B	93.23, 92.09, 92.78
SST-2	FT	PACZERO-BAC	0.33	OPT-1.3B	88.99, 89.68, 89.68
SST-2	FT	PACZERO-BAC	0.33	OPT-6.7B	91.51, 91.17, 91.63
SST-2	FT	PACZERO-BAC	0.33	Qwen2.5-7B	91.86, 92.78, 92.78
SST-2	FT	PACZERO-BAC	0.33	Llama-3.1-8B	93.46, 92.78, 93.12
SST-2	FT	PACZERO-BAC	0.68	OPT-1.3B	88.99, 89.56, 89.68
SST-2	FT	PACZERO-BAC	0.68	OPT-6.7B	92.55, 92.32, 91.63
SST-2	FT	PACZERO-BAC	0.68	Qwen2.5-7B	91.28, 93.00, 92.66
SST-2	FT	PACZERO-BAC	0.68	Llama-3.1-8B	92.55, 93.46, 93.35
SST-2	FT	PACZERO-AGC	0.01	OPT-1.3B	88.42, 89.56, 89.33
SST-2	FT	PACZERO-AGC	0.01	OPT-6.7B	92.09, 91.51, 92.32
SST-2	FT	PACZERO-AGC	0.01	Qwen2.5-7B	89.45, 88.30, 88.42
SST-2	FT	PACZERO-AGC	0.01	Llama-3.1-8B	92.43, 91.86, 93.35

Task	Track	Method	MI	Model	per-seed
SST-2	FT	PACZERO-AGC	0.11	OPT-1.3B	88.76, 89.68, 90.14
SST-2	FT	PACZERO-AGC	0.11	OPT-6.7B	91.86, 91.06, 91.06
SST-2	FT	PACZERO-AGC	0.11	Qwen2.5-7B	89.56, 88.88, 87.84
SST-2	FT	PACZERO-AGC	0.11	Llama-3.1-8B	90.94, 91.63, 92.09
SST-2	FT	PACZERO-AGC	0.33	OPT-1.3B	88.76, 89.45, 90.14
SST-2	FT	PACZERO-AGC	0.33	OPT-6.7B	92.09, 90.02, 91.17
SST-2	FT	PACZERO-AGC	0.33	Qwen2.5-7B	89.68, 89.45, 87.96
SST-2	FT	PACZERO-AGC	0.33	Llama-3.1-8B	91.06, 90.60, 92.66
SST-2	FT	PACZERO-AGC	0.68	OPT-1.3B	90.48, 88.88, 88.99
SST-2	FT	PACZERO-AGC	0.68	OPT-6.7B	91.86, 90.02, 91.97
SST-2	FT	PACZERO-AGC	0.68	Qwen2.5-7B	90.83, 89.11, 89.22
SST-2	FT	PACZERO-AGC	0.68	Llama-3.1-8B	92.32, 90.83, 91.74
SST-2	FT	PACZERO-AGC$_0$	0	OPT-1.3B	90.25, 89.22, 90.02
SST-2	FT	PACZERO-AGC$_0$	0	OPT-6.7B	89.91, 90.48, 89.22
SST-2	FT	PACZERO-AGC$_0$	0	Qwen2.5-7B	89.56, 89.56, 88.99
SST-2	FT	PACZERO-AGC$_0$	0	Llama-3.1-8B	91.51, 92.43, 92.66
SST-2	LoRA	PACZERO-BAC$_0$	0	OPT-1.3B	89.91, 90.02, 89.79
SST-2	LoRA	PACZERO-BAC$_0$	0	OPT-6.7B	93.12, 93.81, 93.00
SST-2	LoRA	PACZERO-BAC$_0$	0	Qwen2.5-7B	94.27, 93.69, 94.72
SST-2	LoRA	PACZERO-BAC$_0$	0	Llama-3.1-8B	91.40, 86.12, 90.60
SST-2	LoRA	PACZERO-BAC	0.01	OPT-1.3B	89.45, 90.25, 90.48
SST-2	LoRA	PACZERO-BAC	0.01	OPT-6.7B	92.66, 93.69, 93.00
SST-2	LoRA	PACZERO-BAC	0.01	Qwen2.5-7B	94.15, 94.04, 94.38
SST-2	LoRA	PACZERO-BAC	0.01	Llama-3.1-8B	89.22, 90.25, 91.28
SST-2	LoRA	PACZERO-BAC	0.11	OPT-1.3B	89.91, 89.91, 90.48
SST-2	LoRA	PACZERO-BAC	0.11	OPT-6.7B	92.32, 93.46, 93.00
SST-2	LoRA	PACZERO-BAC	0.11	Qwen2.5-7B	93.46, 94.38, 94.27
SST-2	LoRA	PACZERO-BAC	0.11	Llama-3.1-8B	89.22, 90.25, 87.61
SST-2	LoRA	PACZERO-BAC	0.33	OPT-1.3B	89.68, 90.60, 89.79
SST-2	LoRA	PACZERO-BAC	0.33	OPT-6.7B	92.32, 94.04, 92.66
SST-2	LoRA	PACZERO-BAC	0.33	Qwen2.5-7B	94.15, 93.12, 94.50
SST-2	LoRA	PACZERO-BAC	0.33	Llama-3.1-8B	89.45, 90.94, 90.37
SST-2	LoRA	PACZERO-BAC	0.68	OPT-1.3B	89.22, 90.60, 90.71
SST-2	LoRA	PACZERO-BAC	0.68	OPT-6.7B	92.78, 93.81, 93.69
SST-2	LoRA	PACZERO-BAC	0.68	Qwen2.5-7B	93.58, 93.23, 94.95
SST-2	LoRA	PACZERO-BAC	0.68	Llama-3.1-8B	91.40, 90.02, 90.94
SST-2	LoRA	PACZERO-AGC	0.01	OPT-1.3B	88.65, 90.37, 90.48
SST-2	LoRA	PACZERO-AGC	0.01	OPT-6.7B	92.20, 92.66, 91.97
SST-2	LoRA	PACZERO-AGC	0.01	Qwen2.5-7B	92.66, 92.89, 92.55
SST-2	LoRA	PACZERO-AGC	0.01	Llama-3.1-8B	91.17, 85.67, 91.06
SST-2	LoRA	PACZERO-AGC	0.11	OPT-1.3B	90.37, 89.33, 90.60
SST-2	LoRA	PACZERO-AGC	0.11	OPT-6.7B	93.12, 92.89, 92.09
SST-2	LoRA	PACZERO-AGC	0.11	Qwen2.5-7B	92.89, 93.58, 93.58
SST-2	LoRA	PACZERO-AGC	0.11	Llama-3.1-8B	90.60, 87.04, 89.11
SST-2	LoRA	PACZERO-AGC	0.33	OPT-1.3B	90.02, 89.79, 90.14
SST-2	LoRA	PACZERO-AGC	0.33	OPT-6.7B	92.43, 92.43, 93.00
SST-2	LoRA	PACZERO-AGC	0.33	Qwen2.5-7B	92.32, 92.20, 93.23
SST-2	LoRA	PACZERO-AGC	0.33	Llama-3.1-8B	91.63, 88.42, 86.35
SST-2	LoRA	PACZERO-AGC	0.68	OPT-1.3B	89.79, 90.02, 90.02
SST-2	LoRA	PACZERO-AGC	0.68	OPT-6.7B	91.86, 91.97, 93.35
SST-2	LoRA	PACZERO-AGC	0.68	Qwen2.5-7B	92.78, 92.66, 93.00
SST-2	LoRA	PACZERO-AGC	0.68	Llama-3.1-8B	92.78, 88.07, 90.71
SST-2	LoRA	PACZERO-AGC$_0$	0	OPT-1.3B	91.06, 88.88, 90.14
SST-2	LoRA	PACZERO-AGC$_0$	0	OPT-6.7B	93.46, 93.69, 93.69
SST-2	LoRA	PACZERO-AGC$_0$	0	Qwen2.5-7B	92.20, 92.32, 92.89
SST-2	LoRA	PACZERO-AGC$_0$	0	Llama-3.1-8B	87.73, 87.04, 88.30

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the Introduction state three contributions that map directly to formal results and experiments: the PACZERO family with four variants PACZERO-AGC, PACZERO-AGC$_0$, PACZERO-BAC and PACZERO-BAC$_0$ (defined in §3 and evaluated in Tables 1 to 9), exact MI calibration of the additive Gaussian channel in place of the variance-based upper bound of canonical PAC Privacy (§3.2, with the MI-budget sweep in Table 7) and the distortion-optimal binary asymmetric channel (§3.3), compared with the Gaussian channel in Table 1, and the $I(S^*; \tilde{Y}_{1:T}) = 0$ guarantee of Theorem 5 (§3.4, proof in Appendix B.3) with the $\mathrm{MI} = 0$ entries of Table 1. Privacy semantics are stated in their native units throughout, and the paper states that PAC Privacy and DP are not equivalent (§3.4, §6).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: §6 (Discussion and Limitations) discusses (i) the semantic scope of the PAC Privacy guarantee, which is instance-based and not equivalent to worst-case DP, and (ii) distributional leakage, since the guarantee covers which candidate subset was sampled and does not bound what a release reveals about U. §7 names as open directions (iii) a convergence analysis of PACZERO, (iv) larger corpora, since every step reads all N records and minibatched subset estimation needs a new privacy analysis (Appendix V), and (v) settings beyond supervised fine-tuning. §3.4 states the scope of the zero-MI guarantee.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical objects are numbered, cross-referenced, and proved. Definition 1 (PAC Privacy) and the chain rule (3) are stated in §2 with the data-generating distribution $\mathcal{D} = \mathrm{Unif}\{S_1, \ldots, S_M\}$ explicitly constructed. Theorem 3 (PACZERO privacy, for any calibrated channel and therefore for PACZERO-AGC and PACZERO-BAC) and Corollary 4 (the MIA bound via the binary-KL inequality) are stated in §3.1, and Theorem 5 (zero MI of PACZERO-AGC$_0$ and PACZERO-BAC$_0$) in §3.4. Appendix B gives the full proofs together with the lemmas they rely on, and Appendix C gives the entropy-cap argument used in adaptive composition.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Algorithm 1 fully specifies the per-step procedure of PACZERO, including the unanimity branch, the disagreement branch, and the posterior update, and §3.2 to §3.4 specify the channels and the zero-budget variants. The numerical calibration is in Appendix C. The data splits and protocol follow Bao et al. [4] and are stated in §4.1, with the test sets and the remaining protocol details in Appendix E. Per-setting hyperparameters (learning rate, clipping threshold c, T, M) are tabulated in Appendix N (Table 11), with the learning-rate sweeps in Table 13 and the clipping-threshold ablations in Appendix P. The DP-AggZO parity check (Appendix K) documents the in-house DP baseline implementation against published entries. The code release at `https://github.com/bilgehanertan/paczero-bac` gives a configuration and instructions for every reported experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and the configuration of every reported experiment are released at `https://github.com/bilgehanertan/paczero-bac`, with pinned environments and instructions for setup and data loading. SST-2 and SQuAD load from Hugging Face Datasets. No model weights are released. All runs start from the public OPT-1.3B, OPT-6.7B, Qwen2.5-7B and Llama-3.1-8B base models.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: §4.1 and Appendix E give the training and evaluation protocol, and §3 gives the update rule. Table 11 (Appendix N) gives the hyperparameters of every setting and how they were selected. The learning-rate sweeps are in Table 13, and the clipping-threshold, LoRA-rank, K-aggregation and PACZERO-AGC$_0$ step-budget ablations are in Appendices P, Q, T and S.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All multi-seed entries in Table 1 report mean $\pm$ std with $n=3$ stated in the table caption. The error bars are sample standard deviations across random seeds with $n-1$ denominator, stated explicitly at the start of Appendix X. The seeds resample the universe U and, on SQuAD, the evaluation subset (Appendix E). Per-seed values for every multi-seed PACZERO entry of Table 1 and Appendix F are in Table 25, so a reader can recompute those means and sample standard deviations. Single-seed entries are marked as single-seed or seed 0, or shown without a spread. Every PACZERO entry of Table 1, on both tasks, is multi-seed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix W gives the hardware, one NVIDIA H100 GPU per reported run, and the GPU-hours by kind of experiment: about $3{,}600$ to reproduce every number in the paper and about $8{,}000$ for the whole project, including failed, superseded and exploratory runs. Appendix V reports per-step time and peak memory.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conforms with the NeurIPS Code of Ethics. The work is a privacy-mechanism contribution evaluated on standard public benchmarks (SST-2, SQuAD) using publicly released base models (OPT-1.3B, OPT-6.7B, Qwen2.5-7B and Llama-3.1-8B). No human subjects, no crowdsourcing, no scraped data, and no new dataset of personally identifiable information are involved.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper's positive societal impact is methodological: it expands the operating regime in which LLM fine-tuning on sensitive corpora is defensible, by producing usable utility at $I(S^*;\tilde{Y}_{1:T})=0$, an MIA-resistance level prior DP zeroth-order methods cannot reach at finite per-step noise (Table 9, §7). The principal misuse is to read the guarantee as worst-case DP, or as a bound on what a release reveals about the universe U. §3.4 and §6 state that it is neither. The paper does not introduce a new generative model or scraped corpus.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any pre-trained model, image generator, or scraped dataset. The released artifact is a training-time mechanism (algorithm and reference implementation) that operates on publicly available base models (OPT-1.3B, OPT-6.7B, Qwen2.5-7B and Llama-3.1-8B) and public benchmarks (SST-2, SQuAD). No fine-tuned checkpoints are distributed.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The models, datasets and baseline code are credited by citation. OPT-1.3B and OPT-6.7B [57] are released by Meta under their custom OPT model license, Qwen2.5-7B [49] under Apache-2.0 and Llama-3.1-8B [27] under the Llama 3.1 Community License, all loaded from Hugging Face. SST-2 [38], distributed without a stated license, and SQuAD v1.1 [35], under CC BY-SA 4.0, are used with the splits and sample counts published by Bao et al. [4]. Our `src/` task scaffolding is adapted from the DP-AggZO reference codebase [4], which builds on MeZO [28], and our LoRA layers from loralib [19], all three under the MIT license. We use Hugging Face Transformers `4.45.2` (Apache-2.0) and PyTorch (BSD-3-Clause). Privacy-bound conversions referencing prior PAC Privacy work [39, 46, 60] are cited at every use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new asset is the PACZERO reference implementation, released under the MIT license at `https://github.com/bilgehanertan/paczero-bac`. Its README documents the setup, the code layout and how to run each variant and reproduce each experiment. No new dataset or model is released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are automated fine-tuning runs on the public SST-2 and SQuAD benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects or any data collected from human participants, so no IRB review applies. The training data is the public SST-2 and SQuAD benchmarks, used at the splits specified in §4.1.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodological contributions are the sign-quantized PAC-private mechanism, the exact MI calibration of the Gaussian channel (§3.2), the binary asymmetric channel (§3.3) and the zero-MI variants PACZERO-AGC$_0$ and PACZERO-BAC$_0$ (§3.4). None of these involves an LLM as an original or non-standard component. LLMs are used for implementing standard methods, editing (e.g., grammar, spelling, word choice), facilitating or running experiments, drafting sections of the paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.