GRZO: Group-Relative Zeroth-Order Optimization for Large Language Model Fine-Tuning

Liyan Tan **Yequan Zhao** **Yifan Yang** **Ruijie Zhang** **Xinling Yu** **Zheng Zhang**
University of California, Santa Barbara
{liyan_tan, yequan_zhao, yifanyang, ruijiezhang, xyu644}@ucsb.edu
zhengzhang@ece.ucsb.edu

Abstract

Zeroth-order (ZO) optimization is a memory-efficient alternative to backpropagation for fine-tuning large language models, but its deployment is limited by the high variance of gradient estimation. We propose **GRZO**, a **G**roup-**R**elative **Z**eroth-**O**rder optimizer that draws one pseudo-independent perturbation per mini-batch example and aggregates the per-example losses through group-relative normalization, raising the effective gradient-direction count from one to the batch size at no additional forward cost while preserving inference-level memory. We prove that GRZO is directionally unbiased with variance shrinking proportionally to the batch size, yielding a tighter nonconvex convergence bound than MeZO. Across RoBERTa-large, Llama3-8B, and OPT-13B over multiple tasks, GRZO improves average accuracy on Llama3-8B by $+3.0$ over MeZO while staying within 0.5% of the forward-only inference memory floor; as a drop-in replacement for the MeZO core, it lifts sparse, low-rank, and quantized ZO variants by $+4.9$ on average.

1 Introduction

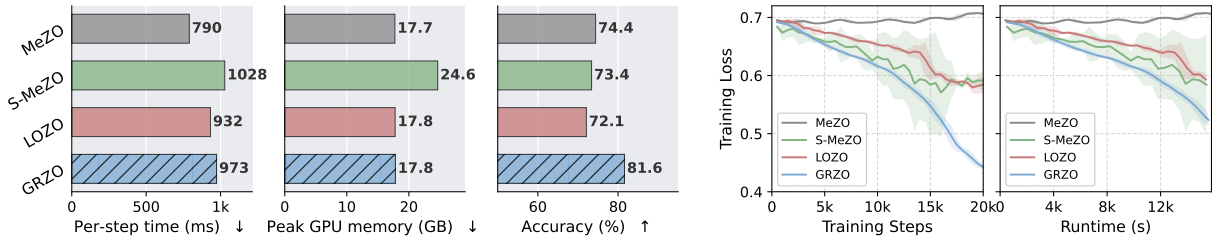
Fine-tuning large language models (LLMs) for downstream tasks remains essential, yet first-order fine-tuning is expensive: backpropagation requires storing activations, gradients, and optimizer states, and these costs grow linearly with model scale. Memory-efficient methods such as LoRA (Hu et al., 2022), Adapter (Houlsby et al., 2019), Prefix-Tuning (Li and Liang, 2021), Prompt-Tuning (Lester et al., 2021), GaLore (Zhao et al., 2024), CoLA (Liu et al., 2025b), and LaX (Zhang et al., 2025a) reduce certain memory footprints but still rely on backpropagation, inheriting most of its activation-storage cost. Moreover, many practical objectives—accuracy, F1, reward signals—are non-differentiable and lie outside the first-order pipeline; the same barrier arises beyond language modeling, for instance in analog/RF circuit design, where the objective is returned by a simulator that exposes no usable gradient (Tan et al., 2026). These considerations motivate zeroth-order (ZO) fine-tuning as a forward-only alternative.

The canonical ZO method for LLM fine-tuning is MeZO (Malladi et al., 2023a), a two-point estimator that approximates the gradient from the loss difference between two perturbed forward passes. Malladi et al. (2023a) report up to a $12\times$ memory reduction over SGD (Amari, 1993) and AdamW (Loshchilov and Hutter, 2019) fine-tuning, keeping training memory near inference levels while remaining compatible with non-differentiable objectives. The catch is that MeZO uses a single random perturbation direction per step; the variance of this estimator grows with model dimension, producing slow descent and brittle optimization once the backbone reaches the multi-billion-parameter range.

A growing literature addresses this variance by reducing the dimension via low-rank (Chen et al., 2025) or sparse (Liu et al., 2025a; Zhang et al., 2025b) perturbations; in LLM fine-tuning, however, the loss landscape exhibits low effective rank (Aghajanyan et al., 2021; Malladi et al., 2023b), so the convergence rate can be independent of the parameter count. Another direction designs lower-variance ZO gradient estimators via control variates (Gautam et al., 2024), Hessian curvature (Zhao et al., 2025a), or minimum-variance two-point estimators (Ma and Huang, 2025).

Both families buy variance reduction at a cost—a narrowed update space, extra forward passes, or extra persistent memory—eroding the inference-level efficiency that motivated ZO in the first place. We identify a third, long-overlooked axis for variance reduction: *the mini-batch itself*. Existing ZO methods reuse a single perturbation direction across all B examples of a step, even though the loss is evaluated per example. Drawing B pseudo-independent

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 36551–36581
October 24-29, 2026 ©2026 Association for Computational Linguistics



(a) Efficiency and accuracy comparison.

(b) Loss convergence.

Figure 1: GRZO at a glance on RTE (Llama3-8B). **Left**: highest accuracy (81.6%) at inference-level peak memory (17.82 GB, +0.5% over the forward-only floor) for a 23% per-step time premium over MeZO. **Right**: fastest convergence in both training steps and wall-clock time.

directions instead—one per example—would, by standard Monte Carlo, reduce the SPSA (simultaneous perturbation stochastic approximation) (Spall, 2002) estimator's variance by a factor of $1/B$ at no additional forward cost, no parameter-space restriction, and no extra persistent memory. Realizing this axis efficiently, while preserving MeZO's two-forward-pass budget and inference-level memory footprint, is the central design problem of this paper.

We propose GRZO (Group-Relative Zeroth-Order Optimization), which realizes this axis by drawing B pseudo-independent perturbations via Flipout-style sign factorization (Wen et al., 2018) and aggregating the resulting per-example loss differences through GRPO-style group-relative normalization (Shao et al., 2024), all within a single two-forward-pass step. Figure 1 summarizes the resulting accuracy, memory, and convergence profile on RTE. Our contributions are:

- **Algorithm.** A ZO optimizer that turns the mini-batch into pseudo-independent perturbation directions while preserving MeZO's two-forward-pass budget and inference-level memory.
- **Theory.** We show the directional unbiasedness and batch-size-scaled variance reduction of GRZO, yielding a strictly tighter nonconvex convergence bound than single-direction ZO.
- **Experimental results.** We show that GRZO outperforms MeZO across models and tasks, and its sparse, low-rank, and quantized variants on most tasks. We further show that GRZO is complementary to sparse, low-rank, and quantized ZO variants and that they can be combined to achieve further performance benefit.

A mechanism-by-mechanism comparison of representative ZO methods is in Table 4 (Appendix C); these approaches are largely orthogonal to GRZO and compose with it.

2 Background and Related Work

2.1 Zeroth-Order (ZO) Optimization

Zeroth-order (ZO) optimization (Nesterov and Spokoiny, 2017; Ghadimi and Lan, 2013) adjusts model parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ using only forward queries of the loss $\mathcal{L}(\boldsymbol{\theta})$, avoiding the memory overhead caused by activation and gradient buffers in backpropagation. A ZO optimizer still performs gradient-descent update $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha\, \mathbf{g}$, but it approximates the gradient $\mathbf{g}$ via N forward passes

$$\mathbf{g} \approx \widehat{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{N\mu} \left[\mathcal{L}(\boldsymbol{\theta} + \mu\boldsymbol{\xi}_i) - \mathcal{L}(\boldsymbol{\theta})\right] \boldsymbol{\xi}_i, \tag{1}$$

with $\{\boldsymbol{\xi}_i\}_{i=1}^N$ drawn i.i.d. from an isotropic distribution $\rho(\boldsymbol{\xi})$ (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$ or Rademacher) and $\mu > 0$ a small sampling radius. The estimator $\widehat{\nabla}_{\boldsymbol{\theta}}\mathcal{L}$ is unbiased w.r.t. the gradient of the smoothed surrogate $f_\mu(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\xi}\sim\rho}[\mathcal{L}(\boldsymbol{\theta} + \mu\boldsymbol{\xi})]$, but biased w.r.t. the true gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ (Berahas et al., 2022), and its variance carries a dimension-dependent factor $O(D/N)$ at $\mu = O(1/\sqrt{N})$ (Liu et al., 2020; Duchi et al., 2015; Gao and Sener, 2022).

MeZO, one of the most popular ZO optimizers for LLM fine-tuning, is the $N{=}1$ two-sided instantiation of Eq. (1): a seed-regenerated direction $\boldsymbol{\xi} \in \mathbb{R}^D$ drives two symmetric forward passes $\ell^{\pm} = \mathcal{L}(\boldsymbol{\theta} \pm \mu\boldsymbol{\xi}; \mathcal{B})$, and parameters are updated in place along $-\alpha(\ell^+ - \ell^-)\boldsymbol{\xi}/2\mu$ without materializing the perturbation tensor. Two knobs govern the estimator's quality: the parameter dimension D over which its variance scales, and the SPSA construction itself. The ZO fine-tuning literature addresses these knobs separately.

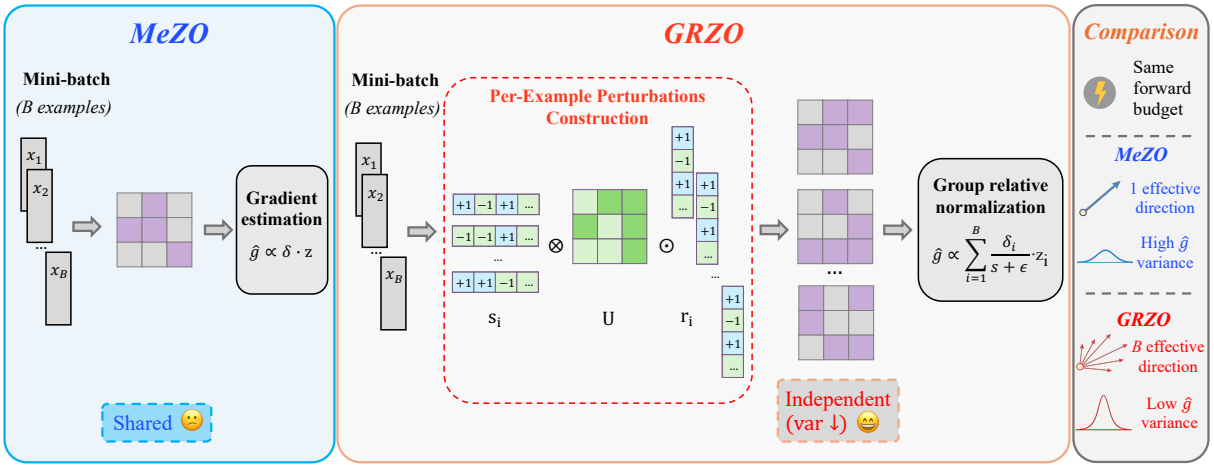


Figure 2: Side-by-side pipeline comparison of MeZO (left) and GRZO (right). By constructing pseudo-independent perturbations and group-relative normalization, GRZO achieves B effective perturbation directions and $1/B$ gradient variance under the same forward-pass budget compared to MeZO.

2.2 Reducing the Effective Dimension

One family of approaches shrinks the update space to mitigate the $O(D)$ variance scaling. DeepZero (Chen et al., 2024) and Sparse-MeZO (Liu et al., 2025a) restrict updates to a sparsity mask, and MaZO (Zhang et al., 2025b) extends masking to multi-task fine-tuning; Low-rank methods reparameterize the perturbation through low-rank matrices (Chen et al., 2025) or tensors (Zhao et al., 2023; Yang et al., 2024). These methods exploit the low effective rank of LLM fine-tuning at the cost of full-parameter expressivity. Along an orthogonal axis, QuZO (Zhou et al., 2025) and Poor-Man's Training (Zhao et al., 2025b) reduce memory via low-bit forward passes without altering the SPSA estimator.

2.3 Improving the Estimator Construction

A second family of research keeps the full-parameter update space and injects additional information into the estimator. MeZO-SVRG (Gautam et al., 2024) pairs each probe with a periodic full-batch reference for an SVRG-style control variate, at the cost of doubled persistent memory; HiZOO (Zhao et al., 2025a) adds a diagonal-Hessian preconditioner estimated from one extra forward pass per step; FZOO (Dang et al., 2026) samples N parallel directions for $1/N$-scale variance reduction at $N+1$ forwards per step; subspace-orthogonalization (Lang et al., 2026) decorrelates the direction sequence across steps; and Ma and Huang (2025) revisit minimum-variance two-point estimator design; SharpZO (Yang et al., 2025) extends the forward-only paradigm to sharpness-aware VLM prompt tuning. In every case, the variance reduction is achieved with extra forwards, extra persistent memory, or both.

3 Method

Both ZO families reviewed in Section 2 reduce variance at a cost: shrinking the effective parameter dimension sacrifices full-parameter expressivity, while enriching the SPSA estimator adds forward passes or persistent memory. GRZO instead turns the mini-batch dimension itself into the variance-reduction lever: a single two-forward-pass step yields B pseudo-independent gradient directions, preserving the inference-level memory and two-forward-pass envelope of MeZO. Section 3.1 constructs the per-example perturbation directions via a sign factorization of a shared base perturbation; Section 3.2 then converts the resulting B per-example loss signals into group-relative weights inspired by GRPO advantages. Figure 2 contrasts the resulting pipeline with MeZO.

3.1 Per-Example Perturbations via Structured Injection

GRZO constructs per-example perturbations through a sign factorization originally proposed for Bayesian weight sampling (Wen et al., 2018). For a linear layer with weight $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we first generate a shared base perturbation matrix $\mathbf{U} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ in each step from a symmetric isotropic distribution (e.g., $\mathcal{N}(0,1)$ or Rademacher entries). Then for each data sample in $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{B}$, we modulate $\mathbf{U}$ with per-example sign vectors $\mathbf{r}_i \in \{\pm 1\}^{d_{\text{out}}}$ and $\mathbf{s}_i \in \{\pm 1\}^{d_{\text{in}}}$ (independent

Rademacher pairs) to obtain

$$\Delta \mathbf{W}_i \;=\; \mathbf{U} \odot (\mathbf{r}_i \mathbf{s}_i^\top), \tag{2}$$

where $\odot$ denotes the element-wise (Hadamard) product. This construction yields B pseudo-independent per-example perturbations from a single shared base $\mathbf{U}$, without materializing B separate weight copies; see Figure 2 for the pipeline and Appendix B for the vectorized form. The overhead is a small constant per linear layer (one extra matrix multiplication plus elementwise sign modulations).

Lemma 1 (Isotropy and conditional decorrelation)**.** *Let* $\mathbf{z}_i := \mathrm{vec}(\Delta \mathbf{W}_i) = \mathrm{vec}(\mathbf{U} \odot (\mathbf{r}_i \mathbf{s}_i^\top))$ *with* $\mathbf{U}$ *and* $\{(\mathbf{r}_i, \mathbf{s}_i)\}_{i=1}^B$ *as defined above. Then* $\mathbb{E}[\mathbf{z}_i \mathbf{z}_i^\top] = \mathbf{I}_{d_{out} d_{in}}$ *marginally and* $\mathbb{E}[\mathbf{z}_i \mathbf{z}_j^\top \mid \mathbf{U}] = 0$ *for* $i \neq j$*. Drawing an independent base* $\mathbf{U}^{(\ell)}$ *and independent sign pairs per layer, the concatenation* $\mathbf{z}_i = (\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(L)}) \in \mathbb{R}^D$ *over all* L *perturbed blocks inherits both properties on* $\mathbb{R}^D$*, where* D *is the total number of perturbed parameters (Appendix J.6.1).*

3.2 Group-Relative Aggregation for Zeroth-Order Updates

Given the per-example perturbations $\Delta \mathbf{W}_i$ in Eq. (2) and a perturbation scale $\sigma > 0$, example i contributes the two-sided perturbed losses $\ell_i^\pm = L(\boldsymbol{\theta} \pm \sigma \Delta \mathbf{W}_i; \mathbf{x}_i, \mathbf{y}_i)$ and the perturbation-induced loss difference

$$\delta_i := \ell_i^+ - \ell_i^-. \tag{3}$$

GRZO converts $\{\delta_i\}_{i=1}^B$ into advantage-like weights via group-relative normalization. We compute the within-batch standard deviation

$$s = \sqrt{\frac{1}{B} \sum_{i=1}^{B} \big(\delta_i - \bar{\delta}\big)^2} \text{ with } \bar{\delta} = \frac{1}{B} \sum_{i=1}^{B} \delta_i, \tag{4}$$

and define the group-relative weights

$$a_i = \frac{\delta_i}{s + \epsilon}. \tag{5}$$

Here $\epsilon > 0$ is a small constant to ensure numerical stability. These weights are scale-invariant under rescaling of $\{\delta_i\}$, decoupling the update from loss-magnitude drift; two-sided differences also give $\mathbb{E}[\delta_i] = 0$ by the $\mathbf{z}_i \leftrightarrow -\mathbf{z}_i$ symmetry, so the numerator needs no explicit mean-centering. The update direction is

$$\widehat{\mathbf{g}} \;=\; \frac{1}{2\sigma B} \sum_{i=1}^{B} a_i \, \mathbf{z}_i, \tag{6}$$

where $\mathbf{z}_i$ is the per-example perturbation direction from the preceding Section 3.1; full pseudocode is in Algorithm 1 (Appendix A). The conditional decorrelation in Lemma 1 eliminates the cross-example covariance terms in $\mathrm{Var}(\hat{\mathbf{g}})$ *given* the shared base $\mathbf{U}$; the residual fluctuation of $\mathbf{U}$ itself is not removed by per-example averaging and is accounted for separately in Theorem 2 (Appendix J.6).

4 Theoretical Analysis

The construction in Section 3 promises B pseudo-independent gradient directions per step at no extra forward cost. We now make this precise. Let D denote the total number of perturbed parameters and let $\mathbf{z}_i \in \mathbb{R}^D$ be the concatenation of example i's per-block Flipout perturbations (Lemma 1). All statements below are for this *joint* perturbation: a single loss difference δ_i probes every block at once, so the estimator restricted to one block still carries the perturbations and gradients of all other blocks and cannot be analyzed in isolation (Appendix J.8). We first analyze the canonical estimator

$$\hat{\mathbf{g}}_{\text{can}} \;=\; \tfrac{1}{2\sigma B} \textstyle\sum_{i=1}^{B} \delta_i \, \mathbf{z}_i, \tag{7}$$

and then quantify the finite-B effect of the group-relative normalization that turns (7) into the GRZO estimator (6). Proofs are in Appendix J.

Theorem 1 (Unbiasedness)**.** *Under standard smoothness assumptions,* $\mathbb{E}[\hat{\mathbf{g}}_{\text{can}} \mid \boldsymbol{\theta}] = \nabla F_\sigma(\boldsymbol{\theta}) + O(\sigma^2)$*, where* $F_\sigma(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}}[F(\boldsymbol{\theta} + \sigma \mathbf{z})]$ *and the* $O(\sigma^2)$ *term is the standard ZO smoothing bias, exactly zero for Gaussian base noise.*

Theorem 2 (Variance Bound)**.** *Let* $m_4 = \mathbb{E}[\mathbf{U}_{jk}^4]$*. Under the assumptions of Appendix J.6,*

$$\mathrm{Var}(\hat{\mathbf{g}}_{\text{can}}) \leq \frac{D-1}{B} \big(\|\nabla F\|^2 + \nu^2\big) + (m_4 - 1)\Big(\|\nabla F\|^2 + \frac{\nu^2}{B}\Big) + O(\rho^2 \sigma^4 D^4).$$

Which term shrinks with B. The first term is the per-example estimation noise: the B conditionally decorrelated directions of Lemma 1 average it down by $1/B$. The second term is the fluctuation of the *shared* base $\mathbf{U}$, which every example sees identically and which per-example averaging therefore cannot remove. For Rademacher base noise $m_4{=}1$ and this term vanishes identically, leaving the pure $1/B$ law; for Gaussian noise $m_4{=}3$ and it is a batch-independent floor. Its share of the bound is $O(B/D)$—below 10^{-8} for an 8B backbone at

B=16—so it is numerically irrelevant at LLM scale, but it is a second reason, beyond the memory argument of Section 5.4, to take Rademacher as the default.

Proposition 1 (Effect of group-relative normalization)**.** *Let $\kappa_\delta = \mathbb{E}[\delta_i^4]/(\mathbb{E}[\delta_i^2])^2$, $s_\star^2 = \mathbb{E}[s^2]$, $c_\star = 1/(s_\star + \epsilon)$, and let $d_{\text{eff}} = \|\nabla F\|^4/\sum_k(\partial_k F)^4$ be the participation ratio of the gradient. Then*

$$\frac{\text{std}(s)}{s_\star} \le \sqrt{\frac{\kappa_\delta - 1}{4B}} + O(d_{\text{eff}}^{-1/2}),$$

and consequently, with $\Delta = O(B^{-1} + d_{\text{eff}}^{-1})$, $\mathbb{E}[\langle \nabla F_\sigma, \hat{\mathbf{g}}_{\text{GRZO}}\rangle] = c_\star(1+\Delta)\|\nabla F_\sigma\|^2 + O(\sigma^2)$ and $\text{Var}(\hat{\mathbf{g}}_{\text{GRZO}}) \le c_\star^2(1+\Delta)\,\text{Var}(\hat{\mathbf{g}}_{\text{can}})$.

The scalar $1/(s+\epsilon)$ is random and correlated with $\hat{\mathbf{g}}_{\text{can}}$, so it cannot simply be absorbed into the learning rate. Proposition 1 states what it does instead: it concentrates around $c_\star$ at rate $O(B^{-1/2})$ and perturbs both the descent direction and the variance by a relative $O(1/B)$, changing constants but neither the $1/B$ scaling of Theorem 2 nor the rate below. This correction grows as B shrinks, which is the mechanism behind the small-batch instability of Appendix G; the constant $\epsilon > 0$ additionally caps $1/(s+\epsilon) \le 1/\epsilon$ and rules out tail blow-up.

Assumption 1. *F_σ is $\mathcal{L}$-smooth and lower bounded by $F_\sigma^\star$; per-example gradients satisfy $\mathbb{E}\|\nabla\ell(\boldsymbol{\theta};\xi) - \nabla F(\boldsymbol{\theta})\|^2 \le \nu^2$; and $\eta \le \min\{\frac{1}{2\mathcal{L}}, \frac{1}{8\mathcal{L}\kappa}\}$ with $\kappa = \frac{D-1}{B} + m_4 - 1$.*

Theorem 3 (Nonconvex Convergence)**.** *Under Assumption 1, GRZO iterations satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla F_\sigma(\boldsymbol{\theta}_t)\big\|^2 \le \frac{8\big(F_\sigma(\boldsymbol{\theta}_0) - F_\sigma^\star\big)}{\eta T} + 4\mathcal{L}\eta\,\beta, \tag{8}$$

where $\beta = \frac{D+m_4-2}{B}\nu^2 + 2\kappa c_{\text{bias}}^2\sigma^4 D^2 + O(\rho^2\sigma^4 D^4)$. (Proof: Theorem 4, Appendix J.8.)

Comparison with MeZO. At the same batch size, MeZO draws a single direction $\mathbf{z}$ and evaluates the batch-mean loss, so its estimator is $\langle \bar{\mathbf{g}}, \mathbf{z}\rangle\mathbf{z}$ with $\bar{\mathbf{g}} = \frac{1}{B}\sum_i \mathbf{g}_i$, giving $\text{Var}(\hat{\mathbf{g}}_{\text{MeZO}}) \approx (D+m_4-2)(\|\nabla F\|^2 + \nu^2/B)$. MeZO therefore already averages the *data* noise over the mini-batch; what GRZO additionally averages is the *directional* noise. Writing $r = \nu^2/\|\nabla F\|^2$, the ratio of the two bounds at the Rademacher default (m_4=1, where the shared-base term vanishes) is

$$\frac{\text{Var}(\hat{\mathbf{g}}_{\text{MeZO}})}{\text{Var}(\hat{\mathbf{g}}_{\text{can}})} \approx \frac{B+r}{1+r} = cB + (1-c) =: B_{\text{eff}}, \tag{9}$$

where $c = 1/(1+r)$ is the mean pairwise cosine between per-example gradients. This gives the informal "B effective directions" of Section 3 a measurable meaning: $B_{\text{eff}} = B$ when per-example gradients are perfectly aligned (c=1) and degrades to 1 when they are mutually orthogonal (c=0), with c estimable directly from per-example gradients on the target model; Appendix H measures both c and the resulting alignment with the exact backpropagation gradient. With $\eta \propto 1/\sqrt{T}$, (8) then improves MeZO's stationarity bound by $\sqrt{B_{\text{eff}}}$: GRZO reaches an ϵ-stationary point in $B_{\text{eff}}\times$ fewer steps at exactly the same two forward passes per step, without extra perturbation queries. Proposition 1 additionally requires B large enough for s to concentrate, consistent with the $B \ge 16$ recommendation of Section 5.4.

5 Experiments

We consider two architectural families. Masked language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) learn bidirectional representations under a masked-token objective; auto-regressive language models such as Llama (Grattafiori et al., 2024) and OPT (Zhang et al., 2022) predict the next token. We benchmark GRZO under full-parameter fine-tuning (Section 5.1), report memory and per-step time on Llama3-8B (Section 5.2), show GRZO composes with parameter-efficient ZO variants such as Sparse-MeZO, LOZO, and QuZO (Section 5.3), and ablate the components of GRZO (Section 5.4).

Setup. We compare against first-order baselines (Adam (Kingma and Ba, 2015), LoRA (Hu et al., 2022)) and zeroth-order baselines (MeZO (Malladi et al., 2023a), FZOO (Dang et al., 2026)) on classification tasks from GLUE (Wang et al., 2018) (SST-2) and SuperGLUE (Wang et al., 2019) (RTE, CB, BoolQ, WiC, MultiRC, COPA), and QA tasks SQuAD (Rajpurkar et al., 2016) and DROP (Dua et al., 2019). On the autoregressive models all methods train for $20k$ steps at batch size 16 in FP16; ZO methods use perturbation scale σ=10^{-3}. For RoBERTa-large we follow the k=512 few-shot protocol of Malladi et al. (2023a), including its batch size of 64. Hyperparameter details are in Appendix D.

5.1 Accuracy and Convergence

Masked Language Models. We evaluate GRZO on RoBERTa-large (350M) under the k=512 few-

Table 1: Results on RoBERTa-large (350M, k=512). FO methods are marked with orange bullets. Among ZO methods, the highest accuracy is highlighted in **bold**.

Method	Sentiment		NLI			Topic
	SST-2	SST-5	SNLI	MNLI	RTE	TREC
Zero-shot	79.0	35.5	50.2	48.8	51.4	32.0
• LP(FO)	91.3	51.7	80.9	71.5	73.1	89.4
• FT(FO)	91.9	47.5	77.5	70.0	66.4	85.0
MeZO	92.8	53.2	83.0	78.3	78.6	94.3
FZOO	93.0	54.2	84.6	**79.9**	78.1	**95.6**
GRZO	**93.3**	**54.8**	**85.4**	79.1	**79.0**	94.7

shot setting. Table 1 shows GRZO outperforms MeZO on all six tasks and FZOO on four of six, with the largest gains over FZOO on RTE ($+0.9$) and SNLI ($+0.8$). Averaged across tasks, GRZO reaches 81.1, vs. 80.9 for FZOO and 80.0 for MeZO.

Auto-Regressive Language Models. We expand to Llama3-8B and OPT-13B on the same SuperGLUE+QA suite. Table 2 shows GRZO is the best ZO method on 8/9 Llama3-8B tasks (average 78.6 vs. 76.5 FZOO and 75.6 MeZO, $+2.0$ under the same two-forward-pass budget) and 6/9 OPT-13B tasks (average 70.5 vs. 70.0 FZOO and 67.9 MeZO). The largest gains over FZOO appear on tasks requiring deeper understanding: $+3.4$ CB, $+5.0$ RTE, $+7.6$ DROP on Llama3-8B; $+4.1$ DROP, $+1.2$ RTE, $+1.0$ COPA on OPT-13B.

Convergence and Wall-Clock Efficiency. Figure 3 plots training-loss curves on Llama3-8B (RTE, MultiRC) and OPT-13B (SQuAD, DROP) against both training steps and wall-clock seconds. GRZO descends faster than FZOO and MeZO across all four panels and reaches a lower final loss on Llama-RTE, Llama-MultiRC, and OPT-SQuAD; on Llama-MultiRC in particular, GRZO matches the final loss of FZOO in roughly half the wall-clock time. MeZO fails to descend on OPT-DROP and makes no measurable progress on Llama-RTE or Llama-MultiRC.

5.2 Memory and Time Analysis

GRZO holds inference-level memory at a modest per-step time cost. Figure 5 reports a production profile on Llama3-8B; Figure 4 extends the memory picture across model sizes.

Memory. We take the *forward-only inference peak* as the reference point, since no forward-only optimizer can go below it; on Llama3-8B (RTE, fp16, B=16) it is 17.74 GB, which already includes activations and workspace on top of the 16.0 GB of weights. Fig. 5a shows that vanilla GRZO peaks at 17.82 GB, i.e. $+0.08$ GB or 0.5% above this floor, so extracting B per-example directions costs essentially nothing in memory. A memory-optimal in-place MeZO also sits at the floor (17.74 GB, $+0.00$ GB): GRZO does not use *less* memory than MeZO, it obtains B directions instead of one at the same footprint. We profile MeZO with a fully in-place perturb–restore–update implementation so that it attains its true inference-level optimum. The picture carries over to the combined variants (Section 5.3): the GRZO family stays within 0.7% of the floor wherever the paired variant does, and where a variant adds its own scratch storage—Sparse and quantized perturbations—GRZO is the cheaper of the pair by 0.3 and 1.5 GB, the latter within the fake-quant regime discussed in Appendix E. Figure 4 shows that GRZO tracks the inference footprint across OPT 1.3B–30B, staying 6–8$\times$ below full fine-tuning.

Per-Step Time. Figure 5b decomposes the per-step wall-clock cost. A fully in-place MeZO completes in 790 ms; GRZO fuses the per-example perturbation into the forward via per-Linear pre-hooks, making this fused forward $\sim$27% slower than MeZO's forward-plus-in-place-perturbation (712 vs. 561 ms). The update shrinks from 229 to 82 ms via sign-vector products, and GRZO totals 973 ms ($+23\%$ over MeZO). The added forward work is one GEMM per Linear layer of the same shape as that layer's own matmul, plus two elementwise sign products; since both scale with the layer's own cost, the relative overhead is largely invariant to batch size, hidden dimension, and model size. This per-step cost is more than offset by the variance reduction (factor of $1/B_{\text{eff}}$, Theorem 2), which translates into proportionally fewer optimization steps to a target loss (Figure 3). The same trade applies to combined variants; full per-method breakdowns are in Appendix E.

Why B Directions Are Memory-Free. The question is not why GRZO is cheaper than MeZO—at their memory optima both sit at the inference floor—but why moving from one perturbation direction to B does not multiply the footprint. Two mechanisms are responsible. First, GRZO never materializes a perturbed weight tensor: each layer's

Table 2: Results on Llama3-8B across SuperGLUE and QA tasks. FO methods are marked with orange bullets. Among ZO methods, the highest accuracy is highlighted in **bold**.

Method	SuperGLUE (Classification)							QA	
	SST-2	RTE	CB	BoolQ	WiC	MultiRC	COPA	SQuAD	DROP
Llama3-8B									
• Adam (FO)	96.0	92.0	92.0	86.6	72.6	84.7	89	90.4	59.4
• LoRA (FO)	95.0	80.9	73.2	86.4	70.7	82.4	89	89.4	58.2
MeZO	92.2	74.4	69.6	76.7	57.8	77.6	88.0	**86.7**	57.1
FZOO	93.0	76.6	68.6	81.2	59.4	77.6	**89.0**	86.0	57.4
GRZO (Ours)	**93.4**	**81.6**	**72.0**	**81.4**	**59.8**	**78.6**	**89.0**	86.2	**65.0**
OPT-13B									
• Adam (FO)	95.3	80.9	94.6	83.5	66.3	76.2	88	89.5	31.3
• LoRA (FO)	94.8	78.3	69.6	80.2	64.3	69.4	89	88.0	30.9
MeZO	91.4	66.1	66.0	67.6	**59.4**	57.3	**88.0**	84.7	30.9
FZOO	**93.8**	76.8	69.6	**72.2**	**59.4**	57.6	87.0	84.8	28.7
GRZO (Ours)	93.4	**78.0**	**70.2**	70.4	58.6	**57.8**	**88.0**	**85.2**	**32.8**

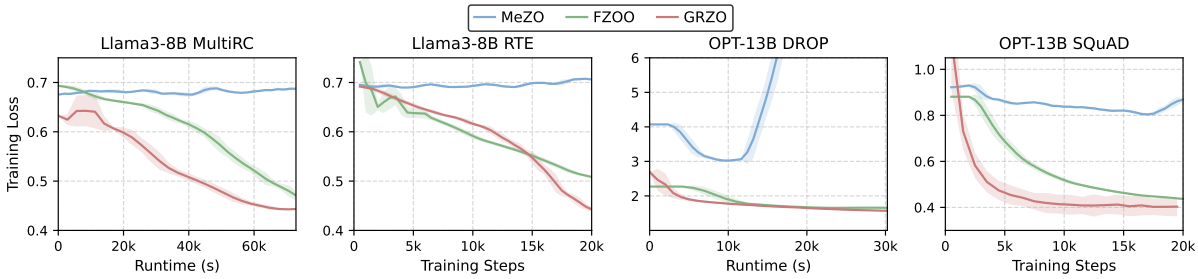


Figure 3: Training-loss curves on Llama3-8B (RTE, MultiRC) and OPT-13B (SQuAD, DROP) plotted against training steps and wall-clock time.

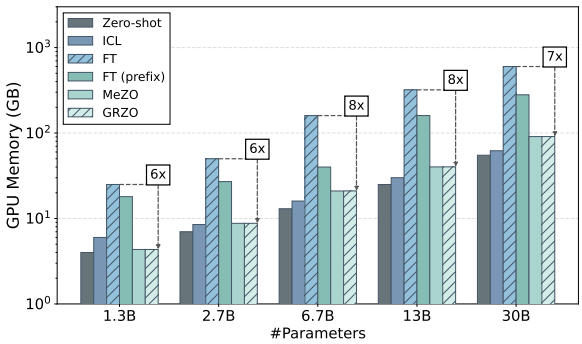


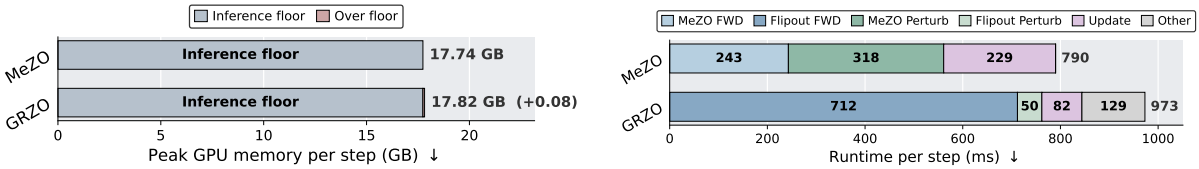
Figure 4: Peak GPU memory (GB) vs. model size for OPT (1.3B–30B). GRZO tracks the forward-only inference footprint across scales, so the B per-example directions are obtained without departing from inference-level memory.

perturbed weight is built transiently inside a forward pre-hook and freed before the next layer runs, so at most one layer's worth of perturbation is live at any moment, independent of depth. Second, the sign factorization $\Delta \mathbf{W}_i = \mathbf{U} \odot (\mathbf{r}_i \mathbf{s}_i^\top)$ compresses all per-example variation into ± 1 sign vectors of size $B(d_{\text{out}}+d_{\text{in}})$ per layer, so the B-direction estimator carries the same per-layer state as a single-direction one. The residual $+0.08$ GB is exactly these sign vectors and the transient per-layer buffer.

5.3 Combination with ZO Variants

GRZO as a drop-in base for other ZO variants. We frame GRZO not only as a stronger ZO estimator than MeZO, but as a base on which existing MeZO variants—sparse (Sparse-MeZO), low-rank (LOZO), and quantized (QuZO) perturbations—can be dropped in for further gains. GRZO intervenes only at the MeZO core, swapping the single-direction perturbation and scalar update for multi-directional per-example perturbation and group-relative normalization; sparsity, low-rank, and quantization act on orthogonal axes of the perturbation and remain complementary to GRZO.

Table 3 shows the composition is Pareto-favorable along two directions. First, every GRZO-combined variant outperforms its baseline on every shared task (the sole exception is LO-GRZO at -0.6 F1 on SQuAD); the most striking case is Qu-GRZO on DROP, where $52.3 \rightarrow 63.9$ F1 ($+11.6$)



(a) Peak GPU memory breakdown.

(b) Per-step time breakdown.

Figure 5: Production profile on Llama3-8B (RTE, fp16, B=16, 4×A100, mean over 20 steps × 4 ranks). **Left**: peak GPU memory per step, split into the forward-only inference floor and the overhead above it. **Right**: per-step time decomposition.

Table 3: GRZO with orthogonal ZO baselines on Llama3-8B. Parentheses: (Δ vs. paired baseline / Δ vs. vanilla GRZO). Best per task in **bold**. QuZO and Qu-GRZO both use 8-bit quantization for weights and perturbations.

Method	**SuperGLUE (Classification)**			**QA (F1)**	
	BoolQ	**RTE**	**COPA**	**SQuAD**	**DROP**
ZO variants baselines					
Sparse-MeZO (Liu et al., 2025a)	80.5	73.4	83.0	87.5	48.4
LOZO (Chen et al., 2025)	79.4	72.1	84.0	**89.0**	65.4
QuZO (int8) (Zhou et al., 2025)	76.8	75.2	87.0	80.6	52.3
Vanilla GRZO					
GRZO	81.4	**81.6**	89.0	86.2	65.0
GRZO combined with orthogonal ZO variants					
Sparse-GRZO	**85.1** (+4.6/+3.7)	79.4 (+6.0/-2.2)	88.0 (+5.0/-1.0)	**89.0** (+1.5/+2.8)	59.3 (+10.9/-5.7)
LO-GRZO	84.4 (+5.0/+3.0)	75.1 (+3.0/-6.5)	90.0 (+6.0/+1.0)	88.4 (-0.6/+2.2)	**65.5** (+0.1/+0.5)
Qu-GRZO (int8)	79.3 (+2.5/-2.1)	80.5 (+5.3/-1.1)	**91.0** (+4.0/+2.0)	88.6 (+8.0/+2.4)	63.9 (+11.6/-1.1)

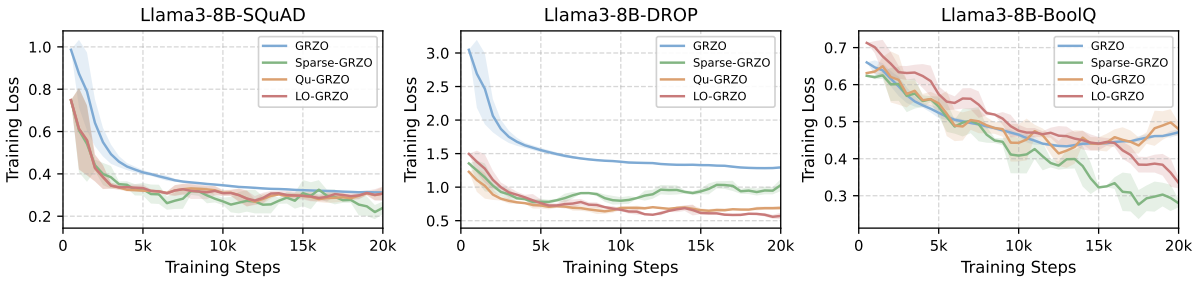


Figure 6: Training loss curves on Llama3-8B comparing vanilla GRZO with the three GRZO-combined variants.

brings the low-bit baseline within 1.1 F1 of vanilla GRZO (65.0). The quantized pairing benefits twice: on top of the $1/B$ variance reduction, low-bit forward passes inflate the scale of the loss differences, and a MeZO-style update—being proportional to δ—absorbs that inflation as an uncontrolled increase in effective step size, whereas the group-relative weights divide by an s inflated by the same factor and are therefore unaffected (Section 3.2). The gain thus reflects GRZO's scale invariance rather than any special interaction with quantization; Appendix I measures both effects. Second, the combinations also beat *vanilla* GRZO on 8 of 15 task–variant cells, consistently so on SQuAD and most reliably for LO-GRZO, showing that these ZO variants complement, rather than compete with, GRZO. Fig. 6 confirms both effects in the training dynamics; two per-variant comparison examples on SQuAD and BoolQ are in Figure 8 (Appendix F). The standard ZO taxonomy—sparsity, low-rank, quantization, and variance reduction—thus reads as independent axes that can be stacked, with GRZO supplying the variance-reduction axis that has no native solution at MeZO's forward budget.

Efficiency Benefit of the GRZO Core. Swapping the MeZO core for GRZO inside any ZO variant improves not only convergence quality but also

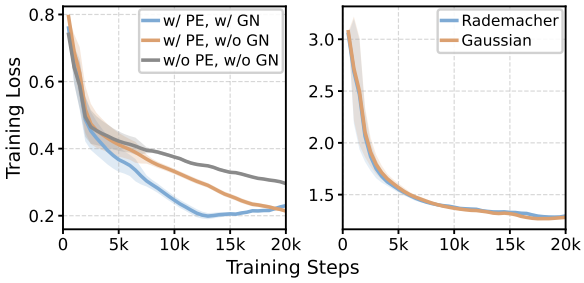


Figure 7: **Left**: GRZO components on SST-2. **Right**: Perturbation ablation on DROP.

resource efficiency. Across the three fp16 variant pairings (vanilla, LOZO, Sparse), GRZO+X is memory-neutral relative to MeZO+X—$+0.08$ GB on vanilla and LOZO, and 0.33 GB lower on Sparse—so the B per-example directions come at no memory premium. GRZO+X is 6–23% slower per step than MeZO+X—each forward fuses the per-example perturbation—but the variance reduction greatly reduces the number of optimization steps to a target loss (Figure 6). We exclude QuZO and Qu-GRZO from this efficiency claim because our fp16 implementations measure fake-quant overhead rather than the low-bit deployment regime targeted by Zhou et al. (2025); per-method numbers and a full discussion are in Appendix E.

5.4 Ablation Study

Two ablations isolate GRZO's design: which component—per-example (PE) perturbation vs. group-relative normalization (GN)—drives convergence, and whether the choice of perturbation noise distribution materially affects performance.

Component Ablation. Figure 7 (left) compares full GRZO, GRZO without GN, and GRZO without both PE and GN (a MeZO-style estimator) on SST-2. Removing GN alone slows descent on this task; further removing PE degrades it more, showing the two components are complementary—PE enables efficient batched estimation, GN stabilizes the gradient signal. GRZO beats FZOO by $+0.4$ on SST-2 ($+2.0$ on Llama3-8B average; Section 5.1), ruling out a normalization-only explanation.

Perturbation Type. Figure 7 (right) compares Gaussian and Rademacher perturbations on DROP; the two loss curves are nearly indistinguishable, confirming robustness to the choice of base distribution. We default to Rademacher for its lower memory overhead and because it is the choice for which the shared-base variance term of Theorem 2 vanishes exactly ($m_4{=}1$).

Batch Size Sensitivity. Figure 9 (Appendix G) shows $B{=}4$ diverges and $B{=}8$ is unstable, while $B{\geq}16$ converges smoothly. This matches Proposition 1: group-relative normalization perturbs the estimator by a relative $O(1/B)$, which is negligible at $B{\geq}16$ but no longer so once B falls to 4. We therefore recommend $B \geq 16$.

6 Conclusion

We have presented GRZO, a zeroth-order optimizer that treats the mini-batch as a source of perturbation directions rather than only loss-averaging samples. By combining pseudo-independent per-example perturbations with group-relative normalization, GRZO extracts a gradient direction from every mini-batch example under MeZO's two-forward-pass budget, greatly mitigating the variance bottleneck of single-direction ZO methods. We have provided theoretical guarantees on directional unbiasedness, variance reduction, and nonconvex convergence. Extensive experiments on multiple models have shown that GRZO outperforms state-of-the-art ZO baselines on the large majority of tasks at inference-level memory, and serves as a drop-in replacement for the MeZO core that composes with sparsity, low-rank, and quantization variants. Extending GRZO to further MeZO variants and full pre-training of large-scale models remains open.

Limitations

GRZO's main per-step cost comes from running two perturbed forwards (ℓ^+ and ℓ^-) to form a directionally unbiased two-sided estimate. A one-sided variant comparing a single perturbed forward to an unperturbed forward would be faster than MeZO but at the cost of a biased gradient estimate and slower descent; building a low-bias one-sided design is open future work.

Our empirical study covers four ZO families (vanilla, Sparse-MeZO, LOZO, QuZO) up to 13B parameters. Other MeZO variants—curvature-preconditioned (HiZOO), control-variate (MeZO-SVRG), and subspace-orthogonalization—and 70B+ scaling are the most immediate empirical extensions: the GRZO core swap is mechanically straightforward in each case, but the variance-versus-stability trade-off and downstream quality at those settings remain to be verified.

Ethical Considerations

This work uses publicly available pre-trained models and benchmarks for research purposes, follows their respective licenses and terms of use, and does not involve the collection or release of personally identifiable information. We do not foresee specific ethical concerns beyond the general risks associated with large language models. In particular, GRZO may inherit biases, hallucinations, or misleading patterns from the underlying models and data. We therefore do not recommend deploying it as a standalone decision-making system, especially in high-stakes settings. Any practical use should include human oversight and task-specific safety evaluation.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328.

Shun-ichi Amari. 1993. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196.

Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. 2022. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560.

Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. 2024. DeepZero: Scaling up zeroth-order optimization for deep model training. In *International Conference on Learning Representations*, volume 2024, pages 50185–50206.

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. 2025. Enhancing zeroth-order finetuning for language models with low-rank structures. In *International Conference on Learning Representations*, volume 2025, pages 62581–62607.

Sizhe Dang, Yangyang Guo, Yanjun Zhao, Haishan Ye, Xiaodong Zheng, Guang Dai, and Ivor W. Tsang. 2026. FZOO: Fast zeroth-order optimizer for fine-tuning large language models towards Adam-scale speed. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. 2015. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806.

Katelyn Gao and Ozan Sener. 2022. Generalizing gaussian smoothing for random search. In *International Conference on Machine Learning*, pages 7077–7101. PMLR.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. 2024. Variance-reduced zeroth-order methods for fine-tuning language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 15180–15208. PMLR.

Saeed Ghadimi and Guanghui Lan. 2013. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank adaptation of large language models. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Yicheng Lang, Changsheng Wang, Yihua Zhang, Mingyi Hong, Zheng Zhang, Wotao Yin, and Sijia Liu. 2026. Powering up zeroth-order training via

subspace gradient orthogonalization. *arXiv preprint arXiv:2602.17155*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. 2020. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. 2025a. Sparse MeZO: Less parameters for better performance in zeroth-order LLM fine-tuning. In *Advances in Neural Information Processing Systems*.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. 2025b. CoLA: Compute-efficient pre-training of LLMs via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4627–4645.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Shaocong Ma and Heng Huang. 2025. Revisiting zeroth-order optimization: Minimum-variance two-point estimators and directionally aligned perturbations. In *International Conference on Learning Representations*, volume 2025, pages 13267–13293.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. 2023a. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. 2023b. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pages 23610–23641. PMLR.

Yurii Nesterov and Vladimir Spokoiny. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2383–2392.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

James C Spall. 2002. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341.

Liyan Tan, Yequan Zhao, Jinming Lu, Ben F. Jamroz, Ari Feldman, and Zheng Zhang. 2026. ZOAF: Towards efficient zeroth-order optimization for analog/RF circuit design. *arXiv preprint arXiv:2606.02869*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. 2018. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *International Conference on Learning Representations*.

Yifan Yang, Zhen Zhang, Rupak Vignesh Swaminathan, Jing Liu, Nathan Susanj, and Zheng Zhang. 2025. SharpZO: Hybrid sharpness-aware vision language model prompt tuning via forward-only passes. *Advances in Neural Information Processing Systems*, 38:143695–143721.

Yifan Yang, Kai Zhen, Ershad Banijamali, Athanasios Mouchtaris, and Zheng Zhang. 2024. AdaZeta: Adaptive zeroth-order tensor-train adaption for memory-efficient large language models fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 977–995.

Ruijie Ray Zhang, Ziyue Alvin Liu, Zhengyang Wang, and Zheng Zhang. 2025a. LaX: Boosting low-rank training of foundation models via latent crossing. *Advances in Neural Information Processing Systems*, 38:142920–142948.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. OPT: Open Pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhen Zhang, Yifan Yang, Kai Zhen, Nathan Susanj, Athanasios Mouchtaris, Siegfried Kunzmann, and Zheng Zhang. 2025b. MaZO: Masked zeroth-order optimization for multi-task fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18537–18554.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *International Conference on Machine Learning*.

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor W. Tsang. 2025a. Second-order fine-tuning without pain for LLMs: A Hessian informed zeroth-order optimizer. In *International Conference on Learning Representations*.

Yequan Zhao, Hai Li, Ian Young, and Zheng Zhang. 2025b. Poor man's training on MCUs: A memory-efficient quantized back-propagation-free approach. *ACM Transactions on Design Automation of Electronic Systems*, 30(5):1–33.

Yequan Zhao, Xinling Yu, Zhixiong Chen, Ziyue Liu, Sijia Liu, and Zheng Zhang. 2023. Tensor-compressed back-propagation-free training for (physics-informed) neural networks. *arXiv preprint arXiv:2308.09858*.

Jiajun Zhou, Yifan Yang, Kai Zhen, Ziyue Liu, Yequan Zhao, Ershad Banijamali, Athanasios Mouchtaris, Ngai Wong, and Zheng Zhang. 2025. QuZO: Quantized zeroth-order fine-tuning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5341–5359.

A GRZO Algorithm

Algorithm 1 gives the per-step pseudocode for GRZO, fusing the per-example sign-factorized perturbation construction with group-relative normalization. The same random seeds are used for the perturbation in the forward pass and the seed-regenerated weight update, so the per-example perturbations $\mathbf{P}_i^{(\ell)}$ never have to be materialized between the two phases.

B Additional Background on MeZO and Flipout

B.1 MeZO as In-Place Two-Point Zeroth-Order Optimization

MeZO (Malladi et al., 2023a) adapts the classical two-point SPSA estimator to LLM fine-tuning with inference-level memory by perturbing parameters in place and regenerating the same noise direction from a random seed. Given parameters $\boldsymbol{\theta} \in \mathbb{R}^D$, mini-batch $\mathcal{B}$, perturbation scale σ, and seed-generated direction $\mathbf{z}(s)$, it evaluates

$$\widehat{\mathbf{g}}(\boldsymbol{\theta}; \mathcal{B}) = \frac{L(\boldsymbol{\theta} + \sigma \mathbf{z}; \mathcal{B}) - L(\boldsymbol{\theta} - \sigma \mathbf{z}; \mathcal{B})}{2\sigma} \mathbf{z}, \tag{10}$$

or equivalently the projected scalar

$$\begin{aligned} g_{\text{proj}} &= \frac{\ell^+ - \ell^-}{2\sigma}, \\ \ell^+ &= L(\boldsymbol{\theta} + \sigma \mathbf{z}(s); \mathcal{B}), \\ \ell^- &= L(\boldsymbol{\theta} - \sigma \mathbf{z}(s); \mathcal{B}), \end{aligned} \tag{11}$$

followed by the in-place update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\, g_{\text{proj}}\, \mathbf{z}(s)$. This design avoids storing activations or full perturbation tensors, but because each step uses only one direction, reducing estimator variance by averaging more directions increases forward cost linearly.

B.2 Flipout for Pseudo-Independent Per-Example Perturbations

Flipout (Wen et al., 2018) addresses the inefficiency of sharing one weight perturbation across an entire mini-batch. For a linear layer with weight matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, it samples a shared base perturbation $\mathbf{U}$ and constructs an effective perturbation for example i as

$$\Delta \mathbf{W}_i \;=\; \mathbf{U} \odot (\mathbf{r}_i \mathbf{s}_i^\top), \tag{12}$$

where $\mathbf{r}_i \in \{\pm 1\}^{d_{\text{out}}}$ and $\mathbf{s}_i \in \{\pm 1\}^{d_{\text{in}}}$ are independent sign vectors. Stacking a mini-batch of activations into X, and the sign vectors into matrices R and S, yields the vectorized form

$$Y = \phi\big(XW \;+\; \big((X \odot S)\mathbf{U}\big) \odot R\big), \tag{13}$$

which avoids materializing B separately perturbed weight matrices while still producing pseudo-independent example-level perturbations. GRZO uses this construction to obtain B perturbation-induced loss signals within the same two-forward-pass budget used by MeZO.

C Comparison of ZO Fine-Tuning Methods

Table 4 compares representative zeroth-order fine-tuning methods across four dimensions: per-step forward-pass count, extra memory relative to MeZO, variance reduction mechanism, and whether backpropagation is required.

D Detailed Experimental Settings

Hardware. All experiments are conducted on servers equipped with 8× NVIDIA A100 (40 GB) or 8× NVIDIA A6000 (48 GB) GPUs. Each Llama3-8B or OPT-13B fine-tuning run on a single task takes approximately 4–8 hours of wall-clock time on one such node, and longer for the quantized variants.

Aggregation. All accuracy and loss numbers reported in this paper are means across 3 runs (different random seeds) per task-method configuration.

Implementation. All experiments use PyTorch with the Hugging Face `transformers` library and standard model checkpoints (`roberta-large`, `meta-llama/Llama-3-8B`, `facebook/opt-13b`). LoRA adapters are implemented via the `peft` library; AdamW is from `torch.optim`. F1 scorers for SQuAD and DROP follow the official scripts from the respective dataset releases.

RoBERTa-large. We follow the experimental protocol of Malladi et al. (2023a) exactly, including data sampling, evaluation splits, and prompt templates. Each task uses k=512 labeled examples per class.

Llama3-8B and OPT-13B. Each task is trained on 1,000 examples (200 for CB, 350 for COPA), with a held-out development set of 500 examples used for learning rate selection and a test set of up to 1,000 examples. We train for 20,000 steps with a linear warmup over 500 steps and a constant learning rate thereafter. The batch size is 16 and

Algorithm 1 GRZO (Group-Relative Zeroth-Order Optimization)

Require: Parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(\ell)}\}$; scale σ; batch size B; steps T; learning rates $\{\eta_t\}$; $\epsilon > 0$

for $t = 1, \ldots, T$ **do**
  Sample $\mathcal{B}_t$; draw per-example sign vectors $\{(\mathbf{r}_i^{(\ell)}, \mathbf{s}_i^{(\ell)})\}$ and layer seeds $\{seed^{(\ell)}\}$
  for each layer ℓ **do** ▷ *Fused forward: two forward passes, per-example perturbations*
    Regenerate $\mathbf{U}^{(\ell)}$ from $seed^{(\ell)}$; $\mathbf{P}_i^{(\ell)} \leftarrow ((\mathbf{X}_i \odot \mathbf{S}_i^{(\ell)})\,\mathbf{U}^{(\ell)}) \odot \mathbf{R}_i^{(\ell)}$
    $\mathbf{X}_i \leftarrow \textsc{Concat}\big(\phi(\mathbf{X}_i\mathbf{W}^{(\ell)} + \sigma\mathbf{P}_i^{(\ell)}),\ \phi(\mathbf{X}_i\mathbf{W}^{(\ell)} - \sigma\mathbf{P}_i^{(\ell)})\big)$
  end for
  $\delta_i \leftarrow \ell_i^+ - \ell_i^-$ for all $i = 1, \ldots, B$ ▷ *Group-relative normalization*
  $s \leftarrow \sqrt{\frac{1}{B}\sum_i(\delta_i - \bar{\delta})^2}$ with $\bar{\delta} = \frac{1}{B}\sum_i \delta_i$; $a_i \leftarrow \delta_i/(s+\epsilon)$
  for each layer ℓ **do** ▷ *Seed-regenerated weight update*
    Regenerate $\mathbf{U}^{(\ell)}$ from $seed^{(\ell)}$; $\bar{\mathbf{M}}^{(\ell)} \leftarrow \frac{1}{B}\sum_i a_i\,\mathbf{r}_i^{(\ell)}(\mathbf{s}_i^{(\ell)})^\top$
    $\mathbf{W}^{(\ell)} \leftarrow \mathbf{W}^{(\ell)} - \dfrac{\eta_t}{2\sigma}\,\mathbf{U}^{(\ell)} \odot \bar{\mathbf{M}}^{(\ell)}$
  end for
end for

Method	Fwd passes/step	Extra memory vs. inference	Variance reduction	BP-free	Update space
MeZO (Malladi et al., 2023a)	2	none (inference level)*	—	✓	Full param
HiZOO (Zhao et al., 2025a)†	3	fp32 diagonal Hessian	Curvature preconditioning	✓	Full param
MeZO-SVRG (Gautam et al., 2024)‡	2+ periodic sweep	reference copy + full-batch est.	SVRG control variates	✓	Full param
FZOO (Dang et al., 2026)§	N+1 (one-sided)	activations from N parallel fwds	N indep. directions	✓	Full param
Sparse-MeZO (Liu et al., 2025a)	2	mask + mutation	Reduced update dimension	✓	Sparse subset
LOZO (Chen et al., 2025)	2	inherits MeZO mutation	Low-rank subspace constraint	✓	Low-rank
LoRA + Adam (Hu et al., 2022)	2 (fwd+bwd)	full-net activations + Adam state on adapter	—	×	Low-rank adapter
GRZO (ours)	**2**	none (inference level)**	**B pseudo-indep. directions per step**	✓	Full param

Table 4: Comparison of representative ZO fine-tuning methods, listing the qualitative source of each method's extra memory; quantitative measurements appear in Appendix E. *We profile MeZO with a fully in-place perturb–restore–update implementation, which reaches the inference floor. †HiZOO uses one extra forward per step and stores an fp32 diagonal Hessian. ‡MeZO-SVRG additionally performs periodic full-dataset sweeps. §Parallel FZOO runs N perturbed forwards concurrently; a sequential variant trades activation memory for $N\times$ wall-clock. **GRZO applies perturbations via forward pre-hooks, so at most one layer's perturbation is live at a time; it reaches the same inference floor while providing B directions rather than one.

we use FP16 precision. The perturbation scale is $\sigma{=}10^{-3}$. Learning rates for GRZO are selected from {1e-7, 3e-7, 5e-7} based on development set accuracy; baseline settings follow their respective sources (Malladi et al., 2023a; Dang et al., 2026).

Perturbed Parameters. We apply full-parameter zeroth-order fine-tuning: all learnable parameters are perturbed, including linear projection weights (sign-factorized), embedding matrices (sparse row-wise perturbation indexed by active tokens), and normalization layer parameters (LayerNorm/RMSNorm). This matches the full-parameter MeZO setting.

Prompts and Task Formulation. We adopt the same prompt templates as Malladi et al. (2023a). For multiple-choice tasks (CB, COPA, WiC, BoolQ, MultiRC), inference uses candidate log-likelihood scoring: each candidate is appended to the prompt and the candidate with the highest mean per-token log-likelihood is selected. During

training, we apply teacher forcing on the correct candidate only, computing the loss solely on candidate tokens while excluding prompt tokens.

Hyperparameters. Tables 5 and 6 summarize the hyperparameter settings for all methods. For all methods, learning rates are selected by grid search on the development set; the best value per task is reported. Adam and LoRA use AdamW with β_1=0.9, β_2=0.999.

Adam (FO) and LoRA (FO) hyperparameters follow Malladi et al. (2023a) and prior work.

E Detailed Time and Memory Breakdown

This appendix tabulates the per-step wall-clock decomposition and peak GPU memory measurements that underlie Section 5.2 and Figure 5. All numbers are from the same production profile: Llama-3-8B (fp16), RTE, batch size B=16, 4×A100-40GB, mean of 80 samples (20 steps × 4 ranks). In Table 7, **Forward** sums the per-step forward passes (two standard for MeZO-family; two fused-perturbation for GRZO-family); **Perturb/Setup** covers perturbation handling (three in-place operations for MeZO-family; per-Linear sign-vector and base-noise setup for GRZO-family); **Update** is the weight-update step; **Other** is the residual (loss reduce + HF Trainer/DDP overhead).

Observations. Two patterns dominate. First, MeZO-family in-place perturbation cost scales steeply with the variant's per-parameter work—LOZO (1.32×), Sparse-MeZO (1.64×), QuZO (3.78×) over plain MeZO—because each variant traverses all ∼8B parameters during perturb, restore, and update. GRZO removes this scaling by representing each per-example perturbation as $B(d_{\text{out}}+d_{\text{in}})$ int-8 sign vectors per layer rather than a per-parameter modification; consequently the GRZO-family update step is uniformly 82 ms across all four pairings, and combined-variant perturb/setup cost is $\leq$ 270 ms. Second, on memory, what separates the rows of Table 8 is the variant's own scratch storage, not the choice of ZO core: vanilla and LO-GRZO sit within 0.7% of the inference floor, whereas Sparse-GRZO and Qu-GRZO carry the sparse mask and quantization buffers of their paired variant. Pairwise, GRZO costs +0.08 GB over MeZO on the vanilla and LOZO pairings and is 0.33 and 1.52 GB *cheaper* on the Sparse and quantized pairings, where avoiding a mutated weight copy alongside the variant's own buffers pays off.

What the B Directions Cost. Implemented at their respective optima, MeZO and GRZO both reach the inference floor, so the question Table 8 answers is not which core is cheaper but what the extra $B-1$ directions cost: +0.08 GB, or 0.5%. Two mechanisms hold it there. Each layer's perturbed weight is constructed transiently inside a forward pre-hook and freed before the next layer runs, so at most one layer's perturbation is live at any moment regardless of depth; and the sign factorization compresses all per-example variation into $B(d_{\text{out}}+d_{\text{in}})$ int-8 sign vectors per layer, so the per-layer state of the B-direction estimator matches that of a single-direction one. Neither term scales with $B\times$ the parameter count, which is what a naive B-direction estimator would pay. The LOZO pairing reproduces this exactly: +0.04 GB for LOZO and +0.12 GB for LO-GRZO over the floor, again +0.08 GB for the extra directions. Sparse and quantized pairings are instead dominated by the variant's own per-parameter scratch storage, which is common to both families rather than attributable to the ZO core.

Caveat: FZOO Variant Ambiguity. FZOO (Dang et al., 2026) admits both sequential and batched-parallel perturbation variants with substantially different wall-clock and memory profiles (the latter trading $O(N)$ activation memory for near-MeZO wall-clock). Reporting either single number would mischaracterize the method, and the choice is implementation-dependent rather than algorithmic. We therefore restrict the FZOO comparison to accuracy, which is independent of the parallel/sequential choice.

Caveat: QuZO and Qu-GRZO Profile Numbers. The peak-memory and per-step wall-clock numbers reported by Zhou et al. (2025) for QuZO (their Tables 4 and 5; Appendix C) come from a Cutlass INT8 kernel that stores weights in packed int4/int8 format and dispatches GEMMs onto integer Tensor Core paths; this kernel is not part of the public release. The released code (`qft` mode) simulates low-bit fine-tuning by fake-quantizing weights inside each forward pass while retaining fp16 storage and fp16 compute, and our QuZO and Qu-GRZO implementations follow the same paradigm. Profiling either method on A100 in fp16 therefore captures fake-quant overhead, not the weight footprint and INT8-GEMM throughput that drive the

Table 5: Hyperparameter settings for RoBERTa-large (k=512).

Method	Hyperparameter	Value
MeZO	Batch size	64
	Learning rate	{1e−7, 5e−7, 1e−6}
	Perturbation σ	1e−3
	Weight decay	0
FZOO	Batch size	64
	Learning rate	{1e−5, 1e−4, 5e−4}
	Perturbation σ	1e−3
	Weight decay	0
GRZO (Ours)	Batch size	64
	Learning rate	{1e−6, 1e−5, 5e−5}
	Perturbation σ	1e−3
	Weight decay	0

Table 6: Hyperparameter settings for Llama3-8B and OPT-13B.

Method	Hyperparameter	Llama3-8B	OPT-13B
MeZO	Batch size	16	16
	Learning rate	{1e−7, 1e−6, 1e−5}	{1e−7, 1e−6, 1e−5}
	Perturbation σ	1e−3	1e−3
	Weight decay	0	0
	Training steps	20,000	20,000
FZOO	Batch size	16	16
	Learning rate	{1e−5, 5e−5, 1e−4}	{1e−5, 5e−5, 1e−4}
	Perturbation σ	1e−3	1e−3
	Weight decay	0	0
	Training steps	20,000	20,000
GRZO (Ours)	Batch size	16	16
	Learning rate	{1e−7, 3e−7, 5e−7}	{1e−7, 3e−7, 5e−7}
	Perturbation σ	1e−3	1e−3
	Weight decay	0	0
	Training steps	20,000	20,000

Table 7: Per-step wall-clock breakdown (ms) across all profiled methods. **Total**s are production measurements; column definitions are given in the text above.

Method	Forward	Perturb/Setup	Update	Other	Total
MeZO family (in-place perturbation, two forward passes)					
MeZO	243	318	229	0.1	790
LOZO	243	421	268	0.1	932
Sparse-MeZO	216	520	292	0.1	1028
QuZO	243	1203	538	0.1	1984
GRZO family (fused per-example perturbation, two forward passes)					
GRZO	712	50	82	129	973
LO-GRZO	713	61	82	134	990
Sparse-GRZO	702	170	82	235	1189
Qu-GRZO	709	208	82	1567	2566

deployment numbers of Zhou et al. (2025); the QuZO and Qu-GRZO rows in Tables 7–8 should be read as fake-quant simulation, not as comparable to QuZO's Table 5. We therefore omit QuZO and Qu-GRZO from the cross-variant memory and time claim in Section 5.3. Their accuracy entries in Table 3 apply paper Algorithm 1 with identical W8 weight and perturbation quantization to both meth-

Table 8: Peak GPU memory per step (GB), same setup as Table 7. The forward-only inference peak is 17.74 GB; **Over floor** = peak − inference floor, in GB and as a percentage. MeZO-family methods are profiled with fully in-place implementations.

Method	**Peak (GB)**	**Over floor**	
		GB	**%**
Forward-only inference	17.74	—	—
MeZO family			
MeZO	17.74	+0.00	0.0
LOZO	17.78	+0.04	0.2
Sparse-MeZO	24.64	+6.90	38.9
QuZO	20.66	+2.92	16.5
GRZO family			
GRZO	17.82	+0.08	0.5
LO-GRZO	17.86	+0.12	0.7
Sparse-GRZO	24.31	+6.57	37.0
Qu-GRZO	19.14	+1.40	7.9

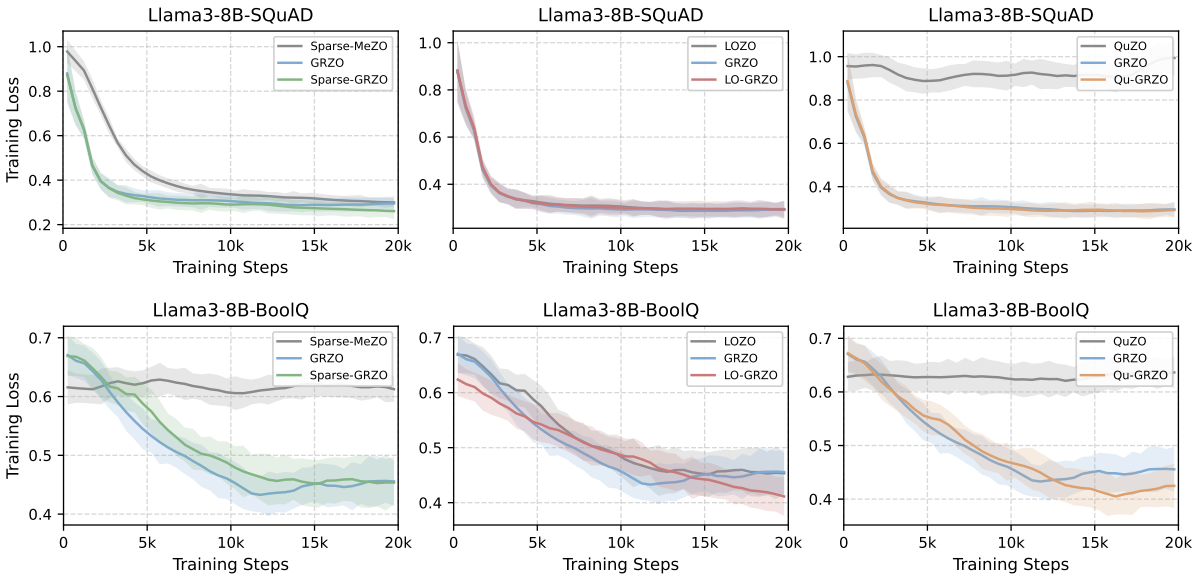


Figure 8: Per-variant GRZO+X vs MeZO+X training-loss curves on Llama3-8B. **Left**: SQuAD. **Right**: BoolQ.

ods and remain apples-to-apples. Within the fake-quant regime on A100, Qu-GRZO is still empirically more memory-efficient than QuZO (though 29% slower per step, the largest per-step premium of the four pairings); this memory advantage stems from forward-hook streaming rather than storage precision, and would persist or amplify on true low-bit hardware.

F GRZO-combined Variants Convergence Examples

Figure 8 shows per-task training-dynamics views of GRZO+X versus its paired MeZO variants and vanilla GRZO baseline on SQuAD and BoolQ (Llama3-8B).

G Batch Size Sensitivity

Figure 9 shows training loss curves for GRZO across batch sizes $B \in \{4, 8, 16, 32\}$ on two tasks: SST-2 (Llama3-8B) and COPA (OPT-13B). The results corroborate the theoretical prediction in Section 5.4: the group-relative normalizer requires a stable within-batch loss standard deviation s to produce reliable advantage weights, and this stability breaks down at very small batch sizes.

H Alignment with the True Gradient

We compare each single ZO gradient estimate against the exact backpropagation gradient on one pretrained Llama3-8B decoder block (218M pa-

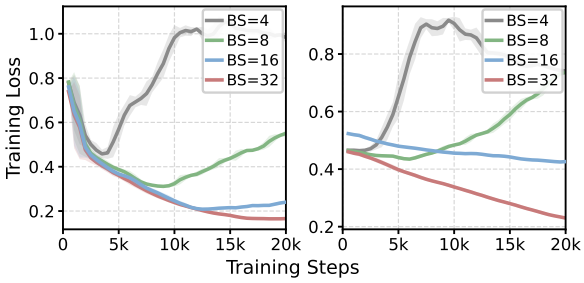


Figure 9: Batch size sensitivity of GRZO. **Left**: Llama3-8B/SST-2. **Right**: OPT-13B/COPA. B=4 diverges; B=8 unstable; $B \geq 16$ converges stably.

rameters), over 3,000 paired trials per batch size. Table 9 reports the cosine similarity between the estimate and the true gradient. In 218M dimensions the absolute cosines are small for both methods, but GRZO's grows with B while MeZO's is flat, which is the behaviour predicted by (9): MeZO draws one direction regardless of B, whereas GRZO draws one per example. On the same block, the mean pairwise cosine between per-example gradients is $c = 0.83$, which by (9) gives $B_{\text{eff}} \approx 13$ at B=16.

Table 9: Cosine similarity to the exact backpropagation gradient on a Llama3-8B decoder block (218M parameters), in units of 10^{-5}, mean $\pm$ standard error over 3,000 paired trials per B.

B	GRZO	MeZO	Ratio
1	5.280 ± 0.072	5.420 ± 0.073	0.97
4	11.87 ± 0.091	5.330 ± 0.074	2.23
8	17.21 ± 0.094	5.329 ± 0.075	3.23
16	24.51 ± 0.096	5.346 ± 0.075	4.58
32	34.80 ± 0.099	5.339 ± 0.075	6.52

I Effect of Quantization on the Loss-Difference Scale

To separate the effect of quantization on the loss differences from its effect on the update, we run a controlled MLP probe (B=16, 20k trials) with and without W4 weight quantization. Quantization raises the dispersion of the per-example loss differences from $\text{std}(\delta) = 0.0119$ to 0.0493, a factor of 4.15. Table 10 reports how this propagates to the mean update norm for the MeZO estimator, the unnormalized canonical estimator, and GRZO. MeZO and the canonical estimator scale by $4.13\times$ and $4.15\times$ respectively, matching the dispersion increase; GRZO changes by $1.00\times$, since the group-relative weights $a_i = \delta_i/(s+\epsilon)$ divide by a within-batch standard deviation that is inflated by the same factor.

Table 10: Mean update norm with and without W4 weight quantization, on a controlled MLP probe (B=16, 20k trials).

Estimator	No quant.	W4	Change
MeZO	66.6	274.8	$4.13\times$
Canonical (unnorm.)	75.6	313.9	$4.15\times$
GRZO	13.42	13.44	$1.00\times$

J Unbiasedness and Smoothing Bias of GRZO

J.1 Estimator Definition and Scope

Joint Parameterization. GRZO perturbs all trainable blocks at every step, and each loss difference δ_i is produced by perturbing them *simultaneously*. The analysis is therefore carried out on the concatenated parameter vector rather than one block at a time; Appendix J.8 explains why a per-block treatment is unavailable. Let

$$\boldsymbol{\theta} = \big(\mathrm{vec}(\mathbf{W}^{(1)}), \ldots, \mathrm{vec}(\mathbf{W}^{(L)})\big) \in \mathbb{R}^D, \tag{14}$$

with $D = \sum_{\ell=1}^{L} D_{\text{out}}^{(\ell)} D_{\text{in}}^{(\ell)}$ the total number of perturbed parameters, and for each example $i \in \{1, \ldots, B\}$ let

$$\mathbf{z}_i = \big(\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(L)}\big), \quad \mathbf{z}_i^{(\ell)} = \mathrm{vec}\big(\mathbf{U}^{(\ell)} \odot (\mathbf{r}_i^{(\ell)} (\mathbf{s}_i^{(\ell)})^\top)\big), \tag{15}$$

with an independent base $\mathbf{U}^{(\ell)}$ per block and independent Rademacher sign pairs $(\mathbf{r}_i^{(\ell)}, \mathbf{s}_i^{(\ell)})$ per block and example. Writing $\mathbf{u} \in \mathbb{R}^D$ for the concatenated base entries and $\mathbf{v}_i \in \{\pm 1\}^D$ for the concatenated sign pattern, (15) reads compactly as $\mathbf{z}_i = \mathbf{u} \odot \mathbf{v}_i$. Every result below uses only this form, so all statements hold verbatim for a single layer (L=1, $D = D_{\text{out}} D_{\text{in}}$).

The two-sided losses and their differences are

$$\ell_i^{\pm} := \ell(\boldsymbol{\theta} \pm \sigma \mathbf{z}_i; \xi_i), \qquad \delta_i := \ell_i^{+} - \ell_i^{-}. \tag{16}$$

Canonical Two-Sided ZO Estimator. The standard (canonical) two-sided estimator is

$$\widehat{\mathbf{g}}_{\text{can}}(\boldsymbol{\theta}) := \frac{1}{2\sigma B} \sum_{i=1}^{B} \delta_i \, \mathbf{z}_i. \tag{17}$$

This serves as the theoretical baseline from which GRZO departs.

GRZO Estimator (Group-Relative Normalization). GRZO computes the within-batch standard deviation

$$s = \sqrt{\frac{1}{B} \sum_{i=1}^{B} (\delta_i - \bar{\delta})^2}, \qquad \bar{\delta} = \frac{1}{B} \sum_{i=1}^{B} \delta_i, \tag{18}$$

defines group-relative weights $a_i := \delta_i \,/\, (s + \epsilon)$, and forms

$$\widehat{\mathbf{g}}_{\text{GRZO}}(\boldsymbol{\theta}) := \frac{1}{2\sigma B} \sum_{i=1}^{B} a_i \, \mathbf{z}_i. \tag{19}$$

The scaling by $1/(s + \epsilon)$ makes the update invariant to the magnitude of $\{\delta_i\}$, analogous to advantage normalization in GRPO (Shao et al., 2024). Because two-sided finite differences satisfy $\mathbb{E}[\delta_i] = 0$ by symmetry of $\mathbf{z}_i \leftrightarrow -\mathbf{z}_i$, no explicit mean baseline is subtracted from the numerator.

J.2 Marginal Distribution of Flipout Perturbations

We only need the *marginal* distribution of $\mathbf{z}_i$ for unbiasedness. Let $\mathbf{U}$ have i.i.d. entries $\mathbf{U}_{jk}$ and let $\mathbf{r}_i \in \{\pm 1\}^{D_{\text{out}}}$, $\mathbf{s}_i \in \{\pm 1\}^{D_{\text{in}}}$ be independent Rademacher vectors. Then each coordinate of $\mathbf{z}_i$ is of the form

$$(\mathbf{z}_i)_{(j,k)} = \mathbf{U}_{jk} \cdot (\mathbf{r}_i)_j \cdot (\mathbf{s}_i)_k, \tag{20}$$

i.e., a sign flip of $\mathbf{U}_{jk}$.

Gaussian Base Noise (Exact). If $\mathbf{U}_{jk} \sim \mathcal{N}(0, 1)$, then $(\mathbf{z}_i)_{(j,k)} \sim \mathcal{N}(0, 1)$ as well, because multiplying a standard normal by an independent ± 1 does not change its distribution. Hence $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I}_D)$ marginally.

Rademacher Base Noise (Isotropic, Symmetric). If $\mathbf{U}_{jk} \sim \mathrm{Rad}(\pm 1)$, then $(\mathbf{z}_i)_{(j,k)} \sim \mathrm{Rad}(\pm 1)$ marginally. In either case, $\mathbf{z}_i$ is symmetric ($\mathbf{z}_i \overset{d}{=} -\mathbf{z}_i$) and isotropic ($\mathbb{E}[\mathbf{z}_i] = 0$, $\mathbb{E}[\mathbf{z}_i \mathbf{z}_i^\top] = \mathbf{I}_D$).

J.3 Unbiasedness w.r.t. a Smoothed Objective (Gaussian Case: Exact)

In the Gaussian case, define the Gaussian-smoothed population objective

$$F_\sigma(\boldsymbol{\theta}) := \mathbb{E}_{u \sim \mathcal{N}(0, \mathbf{I}_D)} \big[F(\boldsymbol{\theta} + \sigma \mathbf{u}) \big]. \tag{21}$$

By Stein's identity, we have

$$\nabla F_\sigma(\boldsymbol{\theta}) = \frac{1}{\sigma} \mathbb{E}_u \big[F(\boldsymbol{\theta} + \sigma \mathbf{u})\, \mathbf{u} \big]. \tag{22}$$

Using symmetry of the Gaussian, $\mathbf{u} \overset{d}{=} -\mathbf{u}$, we obtain the antithetic form:

$$\nabla F_\sigma(\boldsymbol{\theta}) = \frac{1}{2\sigma} \mathbb{E}_u \Big[\big(F(\boldsymbol{\theta} + \sigma \mathbf{u}) - F(\boldsymbol{\theta} - \sigma \mathbf{u}) \big)\, \mathbf{u} \Big]. \tag{23}$$

Now consider a single-example objective $f_i(\boldsymbol{\theta}) := \mathbb{E}_{\xi_i}[\ell(\boldsymbol{\theta}; \xi_i)]$. If $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I}_D)$ marginally (Appendix J.2), then applying (23) to f_i yields

$$\mathbb{E}_{z_i} \left[\frac{f_i(\boldsymbol{\theta} + \sigma \mathbf{z}_i) - f_i(\boldsymbol{\theta} - \sigma \mathbf{z}_i)}{2\sigma} \mathbf{z}_i \right] = \nabla f_{i,\sigma}(\boldsymbol{\theta}), \tag{24}$$

where $f_{i,\sigma}$ is the Gaussian smoothing of f_i.

Taking expectation over the mini-batch sampling and averaging over i, linearity of expectation gives the unbiasedness of the canonical estimator:

$$\mathbb{E}\big[\widehat{\mathbf{g}}_{\text{can}}(\boldsymbol{\theta}) \big] = \nabla F_\sigma(\boldsymbol{\theta}). \tag{25}$$

GRZO as Self-Normalized Estimator. GRZO applies a positive self-normalization to the canonical estimator. Since $a_i = \delta_i / (s + \epsilon)$, the estimator factors as

$$\widehat{\mathbf{g}}_{\text{GRZO}}(\boldsymbol{\theta}) = \frac{1}{s + \epsilon} \cdot \frac{1}{2\sigma B} \sum_{i=1}^{B} \delta_i\, \mathbf{z}_i = \frac{1}{s + \epsilon} \cdot \widehat{\mathbf{g}}_{\text{can}}(\boldsymbol{\theta}). \tag{26}$$

The scalar $1/(s + \epsilon)$ depends on $\{\mathbf{z}_i\}_{i=1}^B$ through $\{\delta_i\}$; it is therefore random and correlated with $\widehat{\mathbf{g}}_{\text{can}}$, and cannot be factored out of the expectation or absorbed into the learning rate by taking expectations. Equivalently, the GRZO update can be written as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \tilde{\eta}_t\, \widehat{\mathbf{g}}_{\text{can}}(\boldsymbol{\theta}_t)$ with a *random* effective step size $\tilde{\eta}_t := \eta_t / (s_t + \epsilon)$, paralleling the FZOO (Dang et al., 2026) convergence framing. Rather than assume that s concentrates, we bound its finite-batch effect directly in Appendix J.6.6: with $c_\star = 1/(s_\star + \epsilon)$, $s_\star^2 = \mathbb{E}[s^2]$, Lemma 3 and Corollary 1 give

$$\mathbb{E}\big[\widehat{\mathbf{g}}_{\text{GRZO}}(\boldsymbol{\theta}) \big] \;=\; c_\star \big(1 + O(B^{-1} + d_{\text{eff}}^{-1}) \big)\, \nabla F_\sigma(\boldsymbol{\theta}) + O(\sigma^2). \tag{27}$$

GRZO is thus direction-preserving up to an explicit finite-batch correction that vanishes as B grows, and the convergence analysis (Appendix J.7) runs on the effective step size $\tilde{\eta} = \eta\, c_\star$ with this correction carried through the constants.

J.4 Taylor Expansion View (General Symmetric Isotropic Directions)

This subsection provides an alternative derivation via Taylor expansion, which applies to any symmetric isotropic direction distribution (including Rademacher).

Assume $f(\boldsymbol{\theta})$ is three-times differentiable and its third derivative tensor is bounded:

$$\|\nabla^3 f(\boldsymbol{\theta})\|_{\text{op}} \le \rho, \quad \forall \boldsymbol{\theta}. \tag{28}$$

For a fixed direction $\mathbf{z}$, a third-order Taylor expansion gives

$$f(\boldsymbol{\theta}+\sigma\mathbf{z}) = f(\boldsymbol{\theta}) + \sigma\langle\nabla f(\boldsymbol{\theta}),\mathbf{z}\rangle + \frac{\sigma^2}{2}\mathbf{z}^\top\nabla^2 f(\boldsymbol{\theta})\,\mathbf{z} + \frac{\sigma^3}{6}\nabla^3 f(\boldsymbol{\theta})[\mathbf{z},\mathbf{z},\mathbf{z}] + O(\sigma^4\|\mathbf{z}\|^4), \tag{29}$$

$$f(\boldsymbol{\theta}-\sigma\mathbf{z}) = f(\boldsymbol{\theta}) - \sigma\langle\nabla f(\boldsymbol{\theta}),\mathbf{z}\rangle + \frac{\sigma^2}{2}\mathbf{z}^\top\nabla^2 f(\boldsymbol{\theta})\,\mathbf{z} - \frac{\sigma^3}{6}\nabla^3 f(\boldsymbol{\theta})[\mathbf{z},\mathbf{z},\mathbf{z}] + O(\sigma^4\|\mathbf{z}\|^4). \tag{30}$$

Subtracting,

$$f(\boldsymbol{\theta}+\sigma\mathbf{z}) - f(\boldsymbol{\theta}-\sigma\mathbf{z}) = 2\sigma\langle\nabla f(\boldsymbol{\theta}),\mathbf{z}\rangle + \frac{\sigma^3}{3}\nabla^3 f(\boldsymbol{\theta})[\mathbf{z},\mathbf{z},\mathbf{z}] + O(\sigma^5\|\mathbf{z}\|^5). \tag{31}$$

Plugging (31) into the two-sided estimator for a single direction,

$$\frac{f(\boldsymbol{\theta}+\sigma\mathbf{z}) - f(\boldsymbol{\theta}-\sigma\mathbf{z})}{2\sigma}\,\mathbf{z} = \langle\nabla f(\boldsymbol{\theta}),\mathbf{z}\rangle\mathbf{z} + \frac{\sigma^2}{6}\nabla^3 f(\boldsymbol{\theta})[\mathbf{z},\mathbf{z},\mathbf{z}]\,\mathbf{z} + O(\sigma^4\|\mathbf{z}\|^6). \tag{32}$$

Taking expectation over a symmetric isotropic $\mathbf{z}$ with $\mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \mathbf{I}$ yields

$$\mathbb{E}\big[\langle\nabla f(\boldsymbol{\theta}),\mathbf{z}\rangle\mathbf{z}\big] = \mathbb{E}[\mathbf{z}\mathbf{z}^\top]\nabla f(\boldsymbol{\theta}) = \nabla f(\boldsymbol{\theta}). \tag{33}$$

For the remainder, symmetry implies cancellation of odd moments and (28) implies

$$\left\|\mathbb{E}\left[\frac{\sigma^2}{6}\nabla^3 f(\boldsymbol{\theta})[\mathbf{z},\mathbf{z},\mathbf{z}]\,\mathbf{z}\right]\right\| \le \frac{\sigma^2}{6}\rho\,\mathbb{E}\|\mathbf{z}\|^4 = O(\sigma^2). \tag{34}$$

Therefore,

$$\mathbb{E}\left[\frac{f(\boldsymbol{\theta}+\sigma\mathbf{z}) - f(\boldsymbol{\theta}-\sigma\mathbf{z})}{2\sigma}\,\mathbf{z}\right] = \nabla f(\boldsymbol{\theta}) + O(\sigma^2). \tag{35}$$

Averaging over examples yields $\mathbb{E}[\widehat{\mathbf{g}}_{\text{can}}(\boldsymbol{\theta})] = \nabla F(\boldsymbol{\theta}) + O(\sigma^2)$.

J.5 Smoothing Bias: Smoothed vs. Original Gradient

We bound $\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\|$. Assume $\nabla^2 F$ is ρ-Lipschitz (equivalently, (28) holds for F). Then for any $\mathbf{u}$,

$$\nabla F(\boldsymbol{\theta}+\sigma\mathbf{u}) = \nabla F(\boldsymbol{\theta}) + \nabla^2 F(\boldsymbol{\theta})\,\sigma\mathbf{u} + R(\boldsymbol{\theta},\mathbf{u}), \qquad \|R(\boldsymbol{\theta},\mathbf{u})\| \le \frac{\rho}{2}\sigma^2\|\mathbf{u}\|^2. \tag{36}$$

Taking expectation over $\mathbf{u} \sim \mathcal{N}(0,\mathbf{I}_D)$ (or any symmetric isotropic distribution), the linear term vanishes since $\mathbb{E}[\mathbf{u}] = 0$, giving

$$\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\| = \|\mathbb{E}_u[\nabla F(\boldsymbol{\theta}+\sigma\mathbf{u})] - \nabla F(\boldsymbol{\theta})\| = \|\mathbb{E}_u[R(\boldsymbol{\theta},\mathbf{u})]\| \le \frac{\rho}{2}\sigma^2\,\mathbb{E}\|\mathbf{u}\|^2. \tag{37}$$

For Gaussian $\mathbf{u} \sim \mathcal{N}(0,\mathbf{I}_D)$, $\mathbb{E}\|\mathbf{u}\|^2 = D$, hence

$$\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\| \le \frac{\rho}{2}\sigma^2 D. \tag{38}$$

Remark (Weaker Assumption). If we only assume F is L-smooth (i.e., ∇F is L-Lipschitz), then one can obtain the weaker bound $\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\| \le L\sigma\,\mathbb{E}\|\mathbf{u}\| = O(\sigma\sqrt{d})$ for Gaussian $\mathbf{u}$.

J.6 Second Moment and Variance Bound of GRZO

Setup. Let $\boldsymbol{\theta} \in \mathbb{R}^D$ be the concatenated parameter vector of Appendix J.1. For each flattened example $i \in \{1,\ldots,B\}$, define the Flipout perturbation $\mathbf{z}_i = \mathrm{vec}(\Delta\mathbf{W}_i)$ with $\Delta\mathbf{W}_i = \mathbf{U} \odot (\mathbf{r}_i\mathbf{s}_i^\top)$ as in (15). Consider the two-sided loss difference $\delta_i = \ell(\boldsymbol{\theta}+\sigma\mathbf{z}_i;\xi_i) - \ell(\boldsymbol{\theta}-\sigma\mathbf{z}_i;\xi_i)$ and the canonical estimator

$$\widehat{g}(\boldsymbol{\theta}) := \widehat{g}_{\text{can}}(\boldsymbol{\theta}) = \frac{1}{2\sigma B}\sum_{i=1}^{B}\delta_i\,\mathbf{z}_i. \tag{39}$$

Assumptions for This Subsection. We assume: (i) $\ell(\cdot;\xi)$ is three-times differentiable and $\|\nabla^3\ell(\boldsymbol{\theta};\xi)\|_{\mathrm{op}} \le \rho$; (ii) the per-example gradients are bounded in second moment: $\mathbb{E}\|\nabla\ell(\boldsymbol{\theta};\xi)\|^2 \le \|\nabla F(\boldsymbol{\theta})\|^2 + \nu^2$; (iii) the base noise has i.i.d. entries with $\mathbb{E}[\mathbf{U}_{jk}^2] = 1$ and fourth moment $m_4 := \mathbb{E}[\mathbf{U}_{jk}^4]$ (Gaussian: $m_4 = 3$, Rademacher: $m_4 = 1$).

J.6.1 Key Conditional Independence

Let $\mathbf{u} = \mathrm{vec}(\mathbf{U}) \in \mathbb{R}^D$ and define the sign vector $\mathbf{v}_i = \mathrm{vec}(\mathbf{r}_i\mathbf{s}_i^\top) \in \{\pm 1\}^d$. Then

$$\mathbf{z}_i = \mathbf{u} \odot \mathbf{v}_i. \tag{40}$$

Since $\{(\mathbf{r}_i, \mathbf{s}_i)\}_{i=1}^B$ are independent across i, the sign vectors $\{\mathbf{v}_i\}_{i=1}^B$ are independent across i. Therefore:

Lemma 2 (Conditional independence and moments)**.** *Conditioned on* $\mathbf{U}$ *(equivalently, on* $\mathbf{u}$*), the perturbations* $\{\mathbf{z}_i\}_{i=1}^B$ *are independent. Moreover,*

$$\mathbb{E}[\mathbf{z}_i \mid \mathbf{U}] = 0, \qquad \mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top \mid \mathbf{U}] = \mathrm{diag}(\mathbf{u}^2) =: \boldsymbol{\Lambda}_{\mathbf{U}}, \qquad \mathbb{E}[\mathbf{z}_i\mathbf{z}_j^\top \mid \mathbf{U}] = 0 \ (i \ne j).$$

Unconditionally, $\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top] = \mathbf{I}_D$.

Proof. Given $\mathbf{U}$, each coordinate satisfies $(\mathbf{z}_i)_k = u_k(\mathbf{v}_i)_k$ with $\mathbb{E}[(\mathbf{v}_i)_k] = 0$ and $\mathbb{E}[(\mathbf{v}_i)_k(\mathbf{v}_i)_{k'}] = 0$ for $k \ne k'$ (factorized Rademacher signs). Independence across i follows from independence of $\{\mathbf{v}_i\}$. □

J.6.2 Taylor Reduction from Perturbation Products to Quadratic Forms

Fix ξ_i and denote $\mathbf{g}_i := \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\xi_i)$. By the third-order Taylor expansion (see Appendix J.4), for each i there exists a remainder R_i such that

$$\delta_i = 2\sigma\langle \mathbf{g}_i, \mathbf{z}_i\rangle + R_i, \qquad |R_i| \le \frac{\rho}{3}\sigma^3\|\mathbf{z}_i\|^3. \tag{41}$$

Plugging into $\delta_i\mathbf{z}_i$ gives the decomposition

$$\delta_i\mathbf{z}_i = 2\sigma\ \underbrace{\big(\langle \mathbf{g}_i, \mathbf{z}_i\rangle\mathbf{z}_i\big)}_{:=\mathbf{q}_i} \ + \ \underbrace{R_i\mathbf{z}_i}_{:=\mathbf{e}_i}. \tag{42}$$

Hence

$$\widehat{g}(\boldsymbol{\theta}) = \frac{1}{B}\sum_{i=1}^B \mathbf{q}_i \ + \ \frac{1}{2\sigma B}\sum_{i=1}^B \mathbf{e}_i. \tag{43}$$

J.6.3 Expanding the Estimator Second Moment: Diagonal and Cross Terms

Conditioning convention. Throughout this subsubsection and the next, we compute second-moment expressions treating the per-example gradients $\mathbf{g}_i = \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\xi_i)$ as held fixed, i.e. all expectations are conditioned implicitly on $\{\xi_i\}_{i=1}^B$. The outer data expectation $\mathbb{E}_\xi$ is applied at the final substitution step (59) via the standard bounded-variance bound $\sum_i \mathbb{E}_{\xi_i}\|\mathbf{g}_i\|^2 \le B(\|\nabla F(\boldsymbol{\theta})\|^2 + \nu^2)$ from assumption (ii). Inner conditional expectations $\mathbb{E}[\cdot \mid \mathbf{U}]$ average over the per-example sign vectors $\{\mathbf{v}_i\}$ only; the outer expectations $\mathbb{E}[\cdot]$ average additionally over $\mathbf{U}$.

We first analyze the leading term $\frac{1}{B}\sum_i \mathbf{q}_i$. Using $\|\sum_i a_i\|^2 = \sum_i \|a_i\|^2 + \sum_{i\ne j} a_i^\top a_j$,

$$\mathbb{E}\left\|\frac{1}{B}\sum_{i=1}^B \mathbf{q}_i\right\|^2 = \frac{1}{B^2}\sum_{i=1}^B \mathbb{E}\|\mathbf{q}_i\|^2 \ + \ \frac{1}{B^2}\sum_{i\ne j}\mathbb{E}\big[\mathbf{q}_i^\top\mathbf{q}_j\big]. \tag{44}$$

Diagonal Term. Condition on $\mathbf{U}$ and $\mathbf{q}_i$. Note that $\|\mathbf{z}_i\|^2 = \|\mathbf{u}\|^2$ does not depend on $\mathbf{v}_i$. Moreover, $\mathbf{q}_i = \langle \mathbf{g}_i, \mathbf{z}_i\rangle\mathbf{z}_i$ implies

$$\|\mathbf{q}_i\|^2 = (\langle \mathbf{g}_i, \mathbf{z}_i\rangle)^2\|\mathbf{z}_i\|^2.$$

Using Lemma 2 and $\mathbb{E}[(\langle \mathbf{g}_i, \mathbf{z}_i\rangle)^2 \mid \mathbf{U}] = \mathbf{g}_i^\top\boldsymbol{\Lambda}_{\mathbf{U}}\mathbf{g}_i = \sum_{k=1}^D g_{i,k}^2 u_k^2$, we get

$$\mathbb{E}\|\mathbf{q}_i\|^2 = \mathbb{E}\Big[\|\mathbf{u}\|^2 \cdot \mathbf{g}_i^\top\boldsymbol{\Lambda}_{\mathbf{U}}\mathbf{g}_i\Big] = \sum_{k=1}^D g_{i,k}^2\,\mathbb{E}\big[\|\mathbf{u}\|^2 u_k^2\big]. \tag{45}$$

For i.i.d. coordinates with $\mathbb{E}[u_k^2] = 1$ and $\mathbb{E}[u_k^4] = m_4$,

$$\mathbb{E}\big[\|\mathbf{u}\|^2 u_k^2\big] = \mathbb{E}[u_k^4] + \sum_{\ell \neq k} \mathbb{E}[u_\ell^2]\mathbb{E}[u_k^2] = m_4 + (D-1). \tag{46}$$

Thus

$$\mathbb{E}\|\mathbf{q}_i\|^2 = (D - 1 + m_4)\,\|\mathbf{g}_i\|^2. \tag{47}$$

Cross Term (Second Moment). For $i \neq j$, conditioned on $\mathbf{U}$ the vectors $\mathbf{z}_i$ and $\mathbf{z}_j$ are independent (Lemma 2), hence $\mathbf{q}_i$ and $\mathbf{q}_j$ are conditionally independent. Therefore

$$\mathbb{E}[\mathbf{q}_i^\top \mathbf{q}_j \mid \mathbf{U}] = \mathbb{E}[\mathbf{q}_i \mid \mathbf{U}]^\top \mathbb{E}[\mathbf{q}_j \mid \mathbf{U}]. \tag{48}$$

Moreover, $\mathbb{E}[\mathbf{q}_i \mid \mathbf{U}] = \mathbb{E}[\langle \mathbf{g}_i, \mathbf{z}_i\rangle \mathbf{z}_i \mid \mathbf{U}] = \mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top \mid \mathbf{U}]\mathbf{g}_i = \mathbf{\Lambda}_\mathbf{U}\mathbf{g}_i$. Thus, per the conditioning convention above (gradients $\mathbf{g}_i$ held fixed in expectation),

$$\mathbb{E}[\mathbf{q}_i^\top \mathbf{q}_j] = \mathbb{E}\big[\mathbf{g}_i^\top \mathbf{\Lambda}_\mathbf{U}^2 \mathbf{g}_j\big] = \sum_{k=1}^{D} g_{i,k} g_{j,k}\, \mathbb{E}[u_k^4] = m_4\, \langle \mathbf{g}_i, \mathbf{g}_j\rangle. \tag{49}$$

Under i.i.d. data sampling, the subsequent outer data expectation gives $\mathbb{E}_\xi[\langle \mathbf{g}_i, \mathbf{g}_j\rangle] = \|\nabla F(\boldsymbol{\theta})\|^2$ for $i \neq j$. Centering removes the part of this contribution proportional to $\|\bar{\mathbf{g}}\|^2$, but—because the B perturbations share one base $\mathbf{U}$—not all of it: the residual reappears as the shared-base term (56) of Subsubsection J.6.4.

Putting Diagonal + Cross Together. Combining (44), (47), and (49), we obtain

$$\begin{aligned}\mathbb{E}\left\|\frac{1}{B}\sum_{i=1}^{B}\mathbf{q}_i\right\|^2 &= \frac{D-1+m_4}{B^2}\sum_{i=1}^{B}\|\mathbf{g}_i\|^2 \;+\; \frac{m_4}{B^2}\sum_{i\neq j}\langle \mathbf{g}_i, \mathbf{g}_j\rangle \\ &= \frac{D-1}{B^2}\sum_{i=1}^{B}\|\mathbf{g}_i\|^2 \;+\; \frac{m_4}{B^2}\left\|\sum_{i=1}^{B}\mathbf{g}_i\right\|^2.\end{aligned} \tag{50}$$

J.6.4 Variance via the Law of Total Variance

The quantity above is a second moment. We now bound the variance of $\mathbf{Q} := \frac{1}{B}\sum_i \mathbf{q}_i$ around its mean $\mathbb{E}[\mathbf{Q}] = \bar{\mathbf{g}}$, where $\bar{\mathbf{g}} := \frac{1}{B}\sum_i \mathbf{g}_i$. Conditioning on the shared base $\mathbf{U}$ and applying the law of total variance,

$$\mathrm{Var}(\mathbf{Q}) = \underbrace{\mathbb{E}\big[\mathrm{Var}(\mathbf{Q} \mid \mathbf{U})\big]}_{\text{(I) per-example noise}} \;+\; \underbrace{\mathrm{Var}\big(\mathbb{E}[\mathbf{Q} \mid \mathbf{U}]\big)}_{\text{(II) shared-base noise}}. \tag{51}$$

Both terms must be retained: conditioning on $\mathbf{U}$ removes the cross-example covariance, but the fluctuation of $\mathbf{U}$ itself is common to all B examples and is therefore not reduced by per-example averaging.

Term (I): Per-Example Noise. Let $\bar{\mathbf{q}}_i := \mathbf{q}_i - \mathbb{E}[\mathbf{q}_i \mid \mathbf{U}]$. Conditioned on $\mathbf{U}$, the $\{\bar{\mathbf{q}}_i\}$ are independent and zero-mean by Lemma 2, so for $i \neq j$,

$$\mathbb{E}\big[\bar{\mathbf{q}}_i^\top \bar{\mathbf{q}}_j \mid \mathbf{U}\big] = 0, \tag{52}$$

and hence

$$\mathrm{Var}(\mathbf{Q} \mid \mathbf{U}) = \frac{1}{B^2}\sum_{i=1}^{B}\mathbb{E}\big[\|\bar{\mathbf{q}}_i\|^2 \mid \mathbf{U}\big]. \tag{53}$$

Taking the outer expectation over $\mathbf{U}$ and using (47) together with $\mathbb{E}\|\mathbb{E}[\mathbf{q}_i \mid \mathbf{U}]\|^2 = \mathbb{E}\|\mathbf{\Lambda}_\mathbf{U}\mathbf{g}_i\|^2 = m_4\|\mathbf{g}_i\|^2$,

$$\begin{aligned}\mathbb{E}\|\bar{\mathbf{q}}_i\|^2 &= \mathbb{E}\|\mathbf{q}_i\|^2 - \mathbb{E}\|\mathbb{E}[\mathbf{q}_i \mid \mathbf{U}]\|^2 \\ &= (D-1+m_4)\|\mathbf{g}_i\|^2 - m_4\|\mathbf{g}_i\|^2 = (D-1)\|\mathbf{g}_i\|^2,\end{aligned} \tag{54}$$

so that

$$\text{(I)} \;=\; \frac{D-1}{B^2}\sum_{i=1}^{B}\|\mathbf{g}_i\|^2. \tag{55}$$

This is the term that Flipout decorrelation reduces by a factor $1/B$.

Term (II): Shared-Base Noise. By Lemma 2, $\mathbb{E}[\mathbf{q}_i \mid \mathbf{U}] = \mathbf{\Lambda}_{\mathbf{U}}\mathbf{g}_i$, so $\mathbb{E}[\mathbf{Q} \mid \mathbf{U}] = \mathbf{\Lambda}_{\mathbf{U}}\bar{\mathbf{g}}$ with $\mathbb{E}[\mathbf{\Lambda}_{\mathbf{U}}] = \mathbf{I}_D$. Therefore

$$\begin{aligned}\text{(II)} &= \mathbb{E}\|\mathbf{\Lambda}_{\mathbf{U}}\bar{\mathbf{g}}\|^2 - \|\bar{\mathbf{g}}\|^2 = \sum_{k=1}^{D}\bar{g}_k^2\big(\mathbb{E}[u_k^4]-1\big)\\ &= (m_4-1)\,\|\bar{\mathbf{g}}\|^2.\end{aligned} \tag{56}$$

Term (II) carries no $1/B$ factor: it depends on the mini-batch only through the mean gradient $\bar{\mathbf{g}}$.

Remark (Gaussian vs. Rademacher). For Rademacher base noise $m_4 = 1$, term (II) vanishes identically and $\mathrm{Var}(\mathbf{Q})$ is exactly the $1/B$-scaled expression (55). For Gaussian base noise $m_4 = 3$ and term (II) equals $2\|\bar{\mathbf{g}}\|^2$, a floor that persists as $B \to \infty$. The degenerate case $D = 1$ separates the two terms unambiguously: (55) is then identically zero, whereas for Gaussian base noise the estimator reduces to $\bar{g}\,u^2$ with variance $2\bar{g}^2$, which is exactly (56). Retaining term (II) is thus necessary rather than merely conservative.

J.6.5 Remainder Control and Final Bound with Explicit Dimensions

We now incorporate the Taylor remainder $\mathbf{e}_i = R_i\mathbf{z}_i$ from (42). Using $|R_i| \le \frac{\rho}{3}\sigma^3\|\mathbf{z}_i\|^3$, we have

$$\|\mathbf{e}_i\|^2 \le \frac{\rho^2}{9}\sigma^6\|\mathbf{z}_i\|^8. \tag{57}$$

Hence

$$\mathbb{E}\left\|\frac{1}{2\sigma B}\sum_{i=1}^{B}\mathbf{e}_i\right\|^2 \le \frac{1}{4\sigma^2 B}\sum_{i=1}^{B}\mathbb{E}\|\mathbf{e}_i\|^2 \le \frac{1}{4\sigma^2}\cdot\frac{\rho^2}{9}\sigma^6\,\mathbb{E}\|\mathbf{z}\|^8 = O\big(\rho^2\sigma^4\,\mathbb{E}\|\mathbf{z}\|^8\big)\,. \tag{58}$$

For Rademacher $\mathbf{U}$ (hence $\|\mathbf{z}\|^2 = \|\mathbf{u}\|^2 = D$ deterministically), $\mathbb{E}\|\mathbf{z}\|^8 = D^4$. For Gaussian $\mathbf{U}$, $\|\mathbf{u}\|^2$ is $\chi^2(D)$ and $\mathbb{E}\|\mathbf{z}\|^8 = \mathbb{E}\|\mathbf{u}\|^8 = O(D^4)$. Thus the remainder contributes $O\big(\rho^2\sigma^4 D^4\big)$.

Final Variance Bound with Explicit Dimensions. Writing $\widehat{g} = \frac{1}{B}\sum_i \mathbf{q}_i + \frac{1}{2\sigma B}\sum_i \mathbf{e}_i$ via (43), Young's inequality $\|a+b\|^2 \le (1+\tau)\|a\|^2 + (1+1/\tau)\|b\|^2$ with any $\tau > 0$, together with $\mathbb{E}\|X - \mathbb{E}X\|^2 \le \mathbb{E}\|X\|^2$, controls $\mathrm{Var}(\widehat{g}_{\text{can}})$ by $(1+\tau)$ times the leading-term variance (51) plus $(1+1/\tau)$ times the remainder second moment (58). Choosing any fixed $\tau \in (0,1]$ (e.g. $\tau = 1$) keeps the leading coefficient at $O(1)$, and the resulting $(1+1/\tau)$ constant is absorbed into the $O(\rho^2\sigma^4 D^4)$ remainder. Substituting terms (I) and (II) from (55) and (56) and taking the outer data expectation with $\mathbb{E}_\xi\sum_i\|\mathbf{g}_i\|^2 \le B(\|\nabla F\|^2 + \nu^2)$ and $\mathbb{E}_\xi\|\bar{\mathbf{g}}\|^2 = \|\nabla F\|^2 + \nu^2/B$ (assumption (ii) and i.i.d. sampling), we obtain the corrected bound

$$\begin{aligned}\mathrm{Var}\big(\widehat{g}_{\text{can}}(\boldsymbol{\theta})\big) \le\; & \frac{D-1}{B}\big(\|\nabla F(\boldsymbol{\theta})\|^2 + \nu^2\big)\\ & + (m_4-1)\Big(\|\nabla F(\boldsymbol{\theta})\|^2 + \frac{\nu^2}{B}\Big)\\ & + O\big(\rho^2\sigma^4 D^4\big)\,.\end{aligned} \tag{59}$$

The mini-batch data-sampling variance ν^2/B contributed by $\mathrm{Var}_\xi(\bar{\mathbf{g}})$ is dominated by $(D-1)\nu^2/B$ for $D \ge 2$ and is absorbed into the first term. This is Theorem 2 of the main body.

Numerical Verification. Table 11 checks (59) against the empirical variance of the estimator on a linear objective, for which the Taylor remainder is exactly zero (2×10^5 trials per configuration). The corrected bound is tight in every configuration. The D=1 row isolates the shared-base term: the previous expression, which retained only (55), predicts zero variance, whereas the estimator reduces to $\bar{g}\, u^2$ and is genuinely random. The Rademacher row shows that for m_4=1 the correction is inactive and the two expressions coincide. The last column confirms the MeZO reference (87): at D=100 its variance is ≈ 101 independently of B, while GRZO's falls as $99/B + 2$.

Table 11: Empirical variance vs. the corrected bound (59) and the uncorrected expression (55), on a linear objective (ν^2=0, $\|\nabla F\|^2$=1).

D	Base noise	B	**Empirical**	**Corrected**	Uncorrected	MeZO
1	Gaussian	any	1.9988	**2.0000**	0.0000	2.018
100	Gaussian	16	8.197	**8.1875**	6.1875	100.99
100	Gaussian	64	3.550	**3.5469**	1.5469	100.99
100	Rademacher	64	1.5469	**1.5469**	1.5469	99.30

J.6.6 Finite-Batch Effect of Group-Relative Normalization

The bound (59) is for the canonical estimator. The implemented estimator factors as

$$\widehat{g}_{\mathrm{GRZO}}(\boldsymbol{\theta}) \;=\; \frac{1}{2\sigma B}\sum_{i=1}^{B}\frac{\delta_i}{s+\epsilon}\,\mathbf{z}_i \;=\; \frac{1}{s+\epsilon}\cdot\widehat{g}_{\mathrm{can}}(\boldsymbol{\theta}). \tag{60}$$

The prefactor $1/(s+\epsilon)$ is a single global scalar, but it is *random* and correlated with $\widehat{g}_{\mathrm{can}}$ through the same $\{\delta_i\}$, so it cannot be absorbed into the learning rate by taking expectations. We instead quantify its finite-B effect.

Lemma 3 (Concentration of the group-relative scale)**.** *Assume the per-example loss differences have finite fourth moment and let $\kappa_\delta = \mathbb{E}[\delta_i^4]/(\mathbb{E}[\delta_i^2])^2$, $s_\star^2 = \mathbb{E}[s^2]$, and let $d_{\mathrm{eff}} = \|\nabla F\|^4/\sum_k(\partial_k F)^4$ be the participation ratio of the gradient. Then*

$$\frac{\mathrm{std}(s)}{s_\star} \;\le\; \sqrt{\frac{\kappa_\delta - 1}{4B}} \;+\; O\big(d_{\mathrm{eff}}^{-1/2}\big). \tag{61}$$

Proof. Conditioned on $\mathbf{U}$, the $\{\delta_i\}$ are i.i.d. with $\mathbb{E}[\delta_i \mid \mathbf{U}] = 0$ by the $\mathbf{z}_i \leftrightarrow -\mathbf{z}_i$ symmetry, so $Bs^2 = \sum_i \delta_i^2 - B\bar{\delta}^2$ is a sum of B i.i.d. terms up to an $O(1/B)$ correction, giving $\mathrm{Var}(s^2 \mid \mathbf{U})/(\mathbb{E}[s^2 \mid \mathbf{U}])^2 = (\kappa_\delta - 1)/B + O(B^{-2})$. The delta method applied to $s = \sqrt{s^2}$ yields $\mathrm{std}(s)/\mathbb{E}[s] = \frac{1}{2}\,\mathrm{std}(s^2)/\mathbb{E}[s^2] + O(B^{-1})$, which is the first term. The second term is the fluctuation of the conditional scale across $\mathbf{U}$: by (41), $\mathbb{E}[\delta_i^2 \mid \mathbf{U}] = 4\sigma^2\mathbf{g}_i^\top\boldsymbol{\Lambda}_{\mathbf{U}}\mathbf{g}_i + O(\sigma^4)$, and $\mathrm{Var}_{\mathbf{U}}(\mathbf{g}^\top\boldsymbol{\Lambda}_{\mathbf{U}}\mathbf{g}) = (m_4 - 1)\sum_k g_k^4$, so its relative standard deviation is $\sqrt{m_4 - 1}\, d_{\mathrm{eff}}^{-1/2}$. □

Corollary 1 (Normalization changes constants, not rates)**.** *Let $c_\star = 1/(s_\star + \epsilon)$ and $\Delta = O(B^{-1} + d_{\mathrm{eff}}^{-1})$. Then*

$$\mathbb{E}\big[\langle \nabla F_\sigma, \widehat{g}_{\mathrm{GRZO}}\rangle\big] = c_\star(1+\Delta)\,\|\nabla F_\sigma\|^2 + O(\sigma^2),$$
$$\mathrm{Var}\big(\widehat{g}_{\mathrm{GRZO}}\big) \le c_\star^2(1+\Delta)\,\mathrm{Var}\big(\widehat{g}_{\mathrm{can}}\big).$$

Proof. Write $X = \langle \nabla F_\sigma, \widehat{g}_{\mathrm{can}}\rangle$ and decompose the descent-relevant error as

$$\mathbb{E}\Big[\frac{X}{s+\epsilon}\Big] - c_\star\mathbb{E}[X] = \mathrm{Cov}\Big(\frac{1}{s+\epsilon}, X\Big) + \Big(\mathbb{E}\Big[\frac{1}{s+\epsilon}\Big] - c_\star\Big)\mathbb{E}[X]. \tag{62}$$

Cauchy–Schwarz bounds the first term by $\mathrm{std}(1/(s+\epsilon))\,\mathrm{std}(X)$; Lemma 3 gives $\mathrm{std}(1/(s+\epsilon))/c_\star = O(B^{-1/2} + d_{\mathrm{eff}}^{-1/2})$, and $\mathrm{std}(X)/\mathbb{E}[X] = O(B^{-1/2})$ because $\widehat{g}_{\mathrm{can}}$ averages B conditionally independent

terms, so the product is a relative $O(B^{-1} + d_{\text{eff}}^{-1})$. A second-order expansion of $1/(s+\epsilon)$ around $s_\star$ bounds the second term by $\mathrm{Var}(s)/(s_\star+\epsilon)^3$, of the same relative order. Finally $1/(s+\epsilon) \le 1/\epsilon$ bounds all moments of the prefactor, so no tail term is lost and the same rate transfers to the second moment. □

Corollary 1 is the formal version of Proposition 1: group-relative normalization perturbs the constants of (59) by a relative $O(1/B)$ but leaves the $1/B$ scaling—and hence the convergence rate of Appendix J.7—unchanged. Because the correction grows as B decreases, it also predicts the loss of stability at small batch sizes observed in Appendix G.

J.7 Nonconvex Convergence of GRZO

Objective and Update. Let $F(\boldsymbol{\theta}) = \mathbb{E}_\xi[\ell(\boldsymbol{\theta};\xi)]$ be the population objective and F_σ be the smoothed objective induced by the (Flipout) perturbation distribution, as defined in Appendix J.3 (Gaussian case) or in the Taylor-based view (Appendix J.4). We analyze the GRZO update with group-relative normalization:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\,\widehat{\mathbf{g}}_t, \qquad \widehat{\mathbf{g}}_t = \frac{1}{2\sigma B}\sum_{i=1}^{B} a_{t,i}\,\mathbf{z}_{t,i}, \tag{63}$$

where $\delta_{t,i} = \ell(\boldsymbol{\theta}_t + \sigma\mathbf{z}_{t,i};\xi_{t,i}) - \ell(\boldsymbol{\theta}_t - \sigma\mathbf{z}_{t,i};\xi_{t,i})$, $\bar{\delta}_t = \frac{1}{B}\sum_i \delta_{t,i}$, $s_t = \sqrt{\frac{1}{B}\sum_i(\delta_{t,i}-\bar{\delta}_t)^2}$, and $a_{t,i} = \delta_{t,i}/(s_t+\epsilon)$.

Assumptions. We assume: (A1) F_σ is lower bounded by $F_\sigma^\star$. (A2) F_σ is $\mathcal{L}$-smooth: $\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F_\sigma(\boldsymbol{\theta}')\| \le \mathcal{L}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|$. (A3) Data noise: $\mathbb{E}\|\nabla\ell(\boldsymbol{\theta};\xi) - \nabla F(\boldsymbol{\theta})\|^2 \le \nu^2$. (A4) Direction preservation with an explicit finite-batch correction (from (27)):

$$\mathbb{E}\big[\widehat{\mathbf{g}}_t \mid \boldsymbol{\theta}_t\big] = c_t\,\nabla F_\sigma(\boldsymbol{\theta}_t) + O(\sigma^2), \tag{64}$$

where, by Corollary 1, $c_t = c_\star(\boldsymbol{\theta}_t)\big(1 + O(B^{-1} + d_{\text{eff}}^{-1})\big) > 0$ with $c_\star = 1/(s_\star+\epsilon)$. For the remainder of this section we absorb c_t into the effective step size $\tilde{\eta} := \eta\,c_t$ and write η for it, so that $\mathbb{E}[\widehat{\mathbf{g}}_t/c_t \mid \boldsymbol{\theta}_t] = \nabla F_\sigma(\boldsymbol{\theta}_t) + O(\sigma^2)$. Unlike a concentration *assumption*, the relative correction $O(B^{-1} + d_{\text{eff}}^{-1})$ is bounded in Corollary 1 and affects only the constants below. (A5) Second-moment / variance bound (from Appendix J.6): there exist explicit constants $A_{\text{var}}(\boldsymbol{\theta}_t)$ such that

$$\mathbb{E}\big\|\widehat{\mathbf{g}}_t - \nabla F_\sigma(\boldsymbol{\theta}_t)\big\|^2 \le A_{\text{var}}(\boldsymbol{\theta}_t), \tag{65}$$

with $A_{\text{var}}(\boldsymbol{\theta}_t)$ depending on $(B, D_{\text{in}}, D_{\text{out}})$ through D is the total perturbed dimension.

J.7.1 One-Step Descent via Smoothness

By $\mathcal{L}$-smoothness of F_σ, for any random direction $\mathbf{u}$,

$$F_\sigma(\boldsymbol{\theta}_{t+1}) \le F_\sigma(\boldsymbol{\theta}_t) + \langle\nabla F_\sigma(\boldsymbol{\theta}_t), \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\rangle + \frac{\mathcal{L}}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2. \tag{66}$$

Substitute $\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t = -\eta\widehat{\mathbf{g}}_t$:

$$F_\sigma(\boldsymbol{\theta}_{t+1}) \le F_\sigma(\boldsymbol{\theta}_t) - \eta\langle\nabla F_\sigma(\boldsymbol{\theta}_t), \widehat{\mathbf{g}}_t\rangle + \frac{\mathcal{L}\eta^2}{2}\|\widehat{\mathbf{g}}_t\|^2. \tag{67}$$

Take conditional expectation given $\boldsymbol{\theta}_t$. After absorbing the positive scalar c_t into the effective step size as in (A4), we may write $\mathbb{E}[\widehat{\mathbf{g}}_t \mid \boldsymbol{\theta}_t] = \nabla F_\sigma(\boldsymbol{\theta}_t) + b_t$ with $\|b_t\| = O(\sigma^2)$ (smoothing bias only).

$$\mathbb{E}\big[F_\sigma(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t\big] \le F_\sigma(\boldsymbol{\theta}_t) - \eta\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 - \eta\langle\nabla F_\sigma(\boldsymbol{\theta}_t), b_t\rangle + \frac{\mathcal{L}\eta^2}{2}\mathbb{E}\big[\|\widehat{\mathbf{g}}_t\|^2 \mid \boldsymbol{\theta}_t\big]. \tag{68}$$

By Cauchy–Schwarz and Young's inequality, the smoothing-bias cross term satisfies

$$|\eta\langle\nabla F_\sigma(\boldsymbol{\theta}_t), b_t\rangle| \;\le\; \tfrac{\eta}{2}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \tfrac{\eta}{2}\|b_t\|^2 \;=\; \tfrac{\eta}{2}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + O(\eta\sigma^4),$$

absorbed into the leading $-\eta\|\nabla F_\sigma\|^2$ term and the $O(\sigma^2)$ remainder. Therefore, with effective gradient coefficient $\frac{\eta}{2}$:

$$\mathbb{E}\big[F_\sigma(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t\big] \le F_\sigma(\boldsymbol{\theta}_t) - \tfrac{\eta}{2}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{\mathcal{L}\eta^2}{2}\mathbb{E}\big[\|\widehat{\mathbf{g}}_t\|^2 \mid \boldsymbol{\theta}_t\big] + O(\eta\sigma^4). \tag{69}$$

Next decompose the second moment:

$$\mathbb{E}\big[\|\widehat{\mathbf{g}}_t\|^2 \mid \boldsymbol{\theta}_t\big] = \|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \mathbb{E}\big[\|\widehat{\mathbf{g}}_t - \nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 \mid \boldsymbol{\theta}_t\big]. \tag{70}$$

Plugging (70) and (65) into (69) gives

$$\mathbb{E}\big[F_\sigma(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t\big] \le F_\sigma(\boldsymbol{\theta}_t) - \frac{\eta}{2}\big(1-\mathcal{L}\eta\big)\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{\mathcal{L}\eta^2}{2}\,A_{\text{var}}(\boldsymbol{\theta}_t). \tag{71}$$

Assuming $\eta \le 1/(2\mathcal{L})$, we have $1-\mathcal{L}\eta \ge \frac{1}{2}$ and thus

$$\mathbb{E}\big[F_\sigma(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t\big] \le F_\sigma(\boldsymbol{\theta}_t) - \frac{\eta}{4}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{\mathcal{L}\eta^2}{2}\,A_{\text{var}}(\boldsymbol{\theta}_t). \tag{72}$$

J.7.2 Handling Randomness: Data Noise + ZO Noise

We now instantiate $A_{\text{var}}(\boldsymbol{\theta}_t)$ using Appendix J.6.

From (59) (Appendix J.6), collecting the per-example and shared-base contributions into

$$\kappa \;:=\; \frac{D-1}{B} \;+\; m_4 - 1, \tag{73}$$

we have

$$A_{\text{var}}(\boldsymbol{\theta}_t) \;\le\; \kappa\,\|\nabla F(\boldsymbol{\theta}_t)\|^2 \;+\; \frac{D+m_4-2}{B}\,\nu^2 \;+\; C_{\text{ZO}}(D,B,\sigma), \tag{74}$$

where C_{ZO} collects the finite-difference remainder (ZO noise) terms, e.g.,

$$C_{\text{ZO}}(D,B,\sigma) = O\big(\rho^2\sigma^4 D^4\big)\,. \tag{75}$$

For Rademacher base noise $m_4 = 1$ and $\kappa = (D-1)/B$ exactly; for Gaussian base noise κ additionally carries the batch-independent floor $m_4 - 1 = 2$.

To express everything in terms of $\nabla F_\sigma(\boldsymbol{\theta}_t)$, we use the smoothing bias bound (Appendix J.5): assuming $\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\| \le c_{\text{bias}}\sigma^2 D$ for all $\boldsymbol{\theta}$, we have

$$\|\nabla F(\boldsymbol{\theta}_t)\|^2 \le 2\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + 2c_{\text{bias}}^2\sigma^4 D^2. \tag{76}$$

Plug (76) into (74):

$$A_{\text{var}}(\boldsymbol{\theta}_t) \le \underbrace{2\kappa}_{:=\alpha}\,\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 \;+\; \underbrace{2\kappa c_{\text{bias}}^2\sigma^4 D^2 + \frac{D+m_4-2}{B}\nu^2 + C_{\text{ZO}}}_{:=\beta}\,. \tag{77}$$

J.7.3 Telescoping and Average Gradient Norm Bound

Take full expectation of (72) and sum from $t=0$ to $T-1$:

$$\mathbb{E}[F_\sigma(\boldsymbol{\theta}_T)] \le F_\sigma(\boldsymbol{\theta}_0) - \frac{\eta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{\mathcal{L}\eta^2}{2}\sum_{t=0}^{T-1}\mathbb{E}\big[A_{\text{var}}(\boldsymbol{\theta}_t)\big]. \tag{78}$$

Using the bound (77) and rearranging gives

$$\begin{aligned}\frac{\eta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 &\le F_\sigma(\boldsymbol{\theta}_0) - \mathbb{E}[F_\sigma(\boldsymbol{\theta}_T)] + \frac{\mathcal{L}\eta^2}{2}\sum_{t=0}^{T-1}\mathbb{E}\left[\alpha\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \beta\right] \\ &\le F_\sigma(\boldsymbol{\theta}_0) - F_\sigma^\star + \frac{\mathcal{L}\eta^2\alpha}{2}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{\mathcal{L}\eta^2 T}{2}\beta.\end{aligned} \tag{79}$$

Move the gradient-sum term to the left:

$$\left(\frac{\eta}{4} - \frac{\mathcal{L}\eta^2\alpha}{2}\right)\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 \le F_\sigma(\boldsymbol{\theta}_0) - F_\sigma^\star + \frac{\mathcal{L}\eta^2 T}{2}\beta. \tag{80}$$

Choose step size satisfying

$$\eta \le \min\left\{\frac{1}{2\mathcal{L}}, \frac{1}{4\mathcal{L}\alpha}\right\} = \min\left\{\frac{1}{2\mathcal{L}}, \frac{1}{8\mathcal{L}\kappa}\right\}. \tag{81}$$

Remark 1 (Regime condition). Since $\kappa \approx (D-1)/B$ whenever $D \gg B$—the regime of full-parameter LLM fine-tuning, where D is of order 10^9 and $B = 16$—the second term is the binding constraint, and the admissible step size grows linearly in B. This is the step-size counterpart of the variance reduction: a larger batch buys both a smaller variance and a larger stable step. The bound in (82) remains meaningful provided $\eta T \to \infty$, i.e. $T = \omega(\mathcal{L}\kappa)$ steps.

With $2\mathcal{L}\eta\alpha \le \frac{1}{2}$, we have $1 - 2\mathcal{L}\eta\alpha \ge \frac{1}{2}$, hence $\frac{\eta}{4}(1 - 2\mathcal{L}\eta\alpha) \ge \frac{\eta}{8}$, and (80) yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_\sigma(\boldsymbol{\theta}_t)\|^2 \le \frac{8\big(F_\sigma(\boldsymbol{\theta}_0) - F_\sigma^\star\big)}{\eta T} + 4\mathcal{L}\eta\,\beta. \tag{82}$$

Making β explicit in (B, D, m_4). From (77),

$$\beta = \frac{D + m_4 - 2}{B}\,\nu^2 + 2\kappa\, c_{\text{bias}}^2\sigma^4 D^2 + O\big(\rho^2\sigma^4 D^4\big)\,, \tag{83}$$

with κ as in (73) and $D = \sum_\ell D_{\text{out}}^{(\ell)} D_{\text{in}}^{(\ell)}$. Plugging (83) into (82) gives an explicit bound that cleanly separates the *data noise* term ν^2 and the *ZO noise* terms (proportional to σ^4).

GRZO Normalization Effect on Convergence. Direct algebra gives $\sum_i a_i^2 = B\big(1 + (\bar{\delta}/s)^2\big)\big(s/(s+\epsilon)\big)^2$, so $\frac{1}{B}\sum_i a_i^2 \approx 1$ at typical batch sizes *provided*: (i) $\mathbb{E}[s_t] \gg \epsilon$ (the within-batch SD dominates the numerical-stability constant, so $s/(s+\epsilon) \approx 1$); and (ii) the batch is large enough that $\mathbb{E}[(\bar{\delta}/s)^2] = O(1/B)$ (for two-sided ZO, $\mathbb{E}[\bar{\delta}] = 0$ by symmetry, so this follows from a standard $O(1/\sqrt{B})$ CLT-type estimate on $\bar{\delta}$). Under (i)–(ii), the update is invariant to the magnitude of $\{\delta_i\}$. Beyond this scale invariance, the *randomness* of the normalizer is what has to be controlled, and Corollary 1 does so: it perturbs c_t and A_{var} by a relative $O(B^{-1} + d_{\text{eff}}^{-1})$, which is absorbed into the constants α and β of (77) and leaves (82) unchanged in form. The correction is $O(1/B)$, so it degrades gracefully rather than abruptly as the batch shrinks, matching the progressive loss of stability from B=16 to B=4 in Appendix G.

J.8 Why the Analysis Must Be Joint Across Blocks

Appendices J–J.7 analyze the concatenated perturbation $\mathbf{z}_i \in \mathbb{R}^D$ directly. This subsection records why that is necessary: the corresponding per-block statement is false, so the network-level result cannot be obtained by proving a single-layer bound and summing it over layers.

Block-Wise Parameterization. Let $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(L)})$ with $\boldsymbol{\theta}^{(\ell)} = \text{vec}(\mathbf{W}^{(\ell)}) \in \mathbb{R}^{d_\ell}$, $d_\ell = D_{\text{out}}^{(\ell)} D_{\text{in}}^{(\ell)}$, and $D = \sum_\ell d_\ell$. At each step, block ℓ receives an independent base $\mathbf{U}_t^{(\ell)}$ and independent per-example sign pairs $(\mathbf{r}_{t,i}^{(\ell)}, \mathbf{s}_{t,i}^{(\ell)})$, so that $\mathbf{z}_{t,i} = (\mathbf{z}_{t,i}^{(1)}, \ldots, \mathbf{z}_{t,i}^{(L)})$. By Lemma 2 applied block-wise together with independence across blocks, $\mathbf{z}_{t,i}$ is symmetric with $\mathbb{E}[\mathbf{z}_{t,i}] = 0$ and $\mathbb{E}[\mathbf{z}_{t,i}\mathbf{z}_{t,i}^\top] = \mathbf{I}_D$. This is exactly the hypothesis used in Appendices J and J.6, so Theorems 1 and 2 apply verbatim with D the total perturbed dimension.

The Estimator Does Not Decompose Across Blocks. The squared norm of a concatenated vector is the sum of its per-block squared norms, so the centered second moment does split as

$$\mathbb{E}\big\|\widehat{\mathbf{g}}_t - \mathbb{E}[\widehat{\mathbf{g}}_t \mid \boldsymbol{\theta}_t]\big\|^2 = \sum_{\ell=1}^{L} \mathbb{E}\big\|\widehat{\mathbf{g}}_t^{(\ell)} - \mathbb{E}[\widehat{\mathbf{g}}_t^{(\ell)} \mid \boldsymbol{\theta}_t]\big\|^2. \tag{84}$$

What does *not* follow is that each summand obeys the single-layer bound with d_ℓ in place of D. GRZO produces one scalar δ_i per example by perturbing all blocks *simultaneously*: by (41),

$$\delta_i = 2\sigma \sum_{m=1}^{L} \big\langle \mathbf{g}_i^{(m)}, \mathbf{z}_i^{(m)} \big\rangle + R_i, \tag{85}$$

so the block-ℓ estimator $\widehat{\mathbf{g}}^{(\ell)} \propto \big(\sum_m \langle \mathbf{g}_i^{(m)}, \mathbf{z}_i^{(m)} \rangle\big)\mathbf{z}_i^{(\ell)}$ carries the perturbations and gradients of every *other* block. Substituting a single-layer bound into (84) discards these cross-block contributions. The failure is visible already at $L{=}2$ with $d_1{=}d_2{=}1$: a per-block bound would predict $(d_\ell - 1) = 0$, i.e. zero variance, whereas $\widehat{\mathbf{g}}^{(1)} = (g_1 z_1 + g_2 z_2) z_1$ has variance at least g_2^2 contributed by the second block alone.

Network-Level Bound. Working jointly instead, (59) yields the network-level bound directly:

$$\begin{aligned}
&\mathbb{E}\big\|\widehat{g}_{\mathrm{can},t} - \mathbb{E}[\widehat{g}_{\mathrm{can},t} \mid \boldsymbol{\theta}_t]\big\|^2 \\
&\quad \le \frac{D-1}{B}\big(\|\nabla F(\boldsymbol{\theta}_t)\|^2 + \nu^2\big) \\
&\qquad + (m_4 - 1)\Big(\|\nabla F(\boldsymbol{\theta}_t)\|^2 + \tfrac{\nu^2}{B}\Big) + O\big(\rho^2\sigma^4 D^4\big).
\end{aligned} \tag{86}$$

Relative to the block-wise expression, the joint treatment replaces $\sum_\ell (d_\ell - 1) = D - L$ by $D - 1$ and adds the shared-base term of (56). It therefore changes the decomposition and the constants, but preserves the $1/B$ scaling of the leading term—and with it the comparison against MeZO and the convergence rate below.

Theorem 4 (Network-level nonconvex convergence of GRZO)**.** *Let the full parameter vector be $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(L)}) \in \mathbb{R}^D$ with block dimensions $d_\ell = D_{\mathrm{out}}^{(\ell)} D_{\mathrm{in}}^{(\ell)}$ and $D = \sum_{\ell=1}^{L} d_\ell$ the total perturbed dimension. Let $F(\boldsymbol{\theta}) = \mathbb{E}_\xi[\ell(\boldsymbol{\theta};\xi)]$ be the population objective and F_σ be the smoothed objective induced by the joint Flipout perturbation distribution.*

Consider the GRZO update with group-relative normalization: let $\bar{\delta}_t = \frac{1}{B}\sum_i \delta_{t,i}$, $s_t = \sqrt{\frac{1}{B}\sum_i (\delta_{t,i} - \bar{\delta}_t)^2}$, $a_{t,i} = \delta_{t,i}/(s_t + \epsilon)$, and

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \widehat{\mathbf{g}}_t, \qquad \widehat{\mathbf{g}}_t = \frac{1}{2\sigma B}\sum_{i=1}^{B} a_{t,i}\, \mathbf{z}_{t,i},$$

where B is the batch size and $\delta_{t,i} = \ell(\boldsymbol{\theta}_t + \sigma \mathbf{z}_{t,i}; \xi_{t,i}) - \ell(\boldsymbol{\theta}_t - \sigma \mathbf{z}_{t,i}; \xi_{t,i})$. Assume:

(A1) *F_σ is lower bounded: $F_\sigma(\boldsymbol{\theta}) \ge F_\sigma^\star$ for all $\boldsymbol{\theta}$.*

(A2) *F_σ is $\mathcal{L}$-smooth.*

(A3) *Data noise: $\mathbb{E}\|\nabla \ell(\boldsymbol{\theta};\xi) - \nabla F(\boldsymbol{\theta})\|^2 \le \nu^2$.*

(A4) *Direction preservation: with the step size rescaled by the positive scale $c_t = c_\star(\boldsymbol{\theta}_t)(1 + O(B^{-1} + d_{\mathrm{eff}}^{-1}))$ of Corollary 1 (cf. (27)), $\mathbb{E}[\widehat{\mathbf{g}}_t \mid \boldsymbol{\theta}_t] = \nabla F_\sigma(\boldsymbol{\theta}_t) + O(\sigma^2)$.*

(A5) *Joint Flipout construction across blocks, so that the network-level variance admits the explicit bound (86):*

$$\mathbb{E}\big\|\widehat{\mathbf{g}}_t - \mathbb{E}[\widehat{\mathbf{g}}_t \mid \boldsymbol{\theta}_t]\big\|^2 \le \kappa\, \|\nabla F(\boldsymbol{\theta}_t)\|^2 + \frac{D + m_4 - 2}{B}\nu^2 + C_{\mathrm{ZO}},$$

with $\kappa = \frac{D-1}{B} + m_4 - 1$ as in (73) and $C_{\mathrm{ZO}} = O(\rho^2\sigma^4 D^4)$ collecting the finite-difference/Taylor remainder (with ρ bounding the third derivative).

Assume additionally the smoothing-bias bound $\|\nabla F_\sigma(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\| \le c_{\text{bias}}\sigma^2 D$ for all $\boldsymbol{\theta}$. Define

$$\alpha_{\text{net}} := 2\kappa, \qquad \beta_{\text{net}} := \frac{D+m_4-2}{B}\nu^2 + 2\kappa\, c_{\text{bias}}^2\sigma^4 D^2 + C_{\text{ZO}}.$$

Choose a constant step size

$$\eta \;\le\; \min\left\{\frac{1}{2\mathcal{L}},\; \frac{1}{4\mathcal{L}\alpha_{\text{net}}}\right\} = \min\left\{\frac{1}{2\mathcal{L}},\; \frac{1}{8\mathcal{L}\kappa}\right\}.$$

Then for any $T \ge 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla F_\sigma(\boldsymbol{\theta}_t)\big\|^2 \;\le\; \frac{8\big(F_\sigma(\boldsymbol{\theta}_0) - F_\sigma^\star\big)}{\eta T} \;+\; 4\mathcal{L}\eta\,\beta_{\text{net}}.$$

Equivalently, if $R \sim \text{Unif}\{0,\dots,T-1\}$, then

$$\mathbb{E}\big\|\nabla F_\sigma(\boldsymbol{\theta}_R)\big\|^2 \;\le\; \frac{8\big(F_\sigma(\boldsymbol{\theta}_0) - F_\sigma^\star\big)}{\eta T} \;+\; 4\mathcal{L}\eta\,\beta_{\text{net}}.$$

GRZO Normalization and Effective Step Size. *The group-relative weights $a_{t,i} = \delta_{t,i}/(s_t+\epsilon)$ have approximate unit empirical variance ($\frac{1}{B}\sum_i a_{t,i}^2 \approx 1$), and by Corollary 1 the randomness of the normalizer perturbs both (A4) and (A5) by a relative $O(B^{-1} + d_{\text{eff}}^{-1})$. The bound therefore holds with the same η after absorbing the positive scale $c_\star$ into the learning rate, with that correction carried in α_{net} and β_{net}.*

Remark (Scope of Theorem). The theorem covers the GRZO estimator with group-relative normalization $\widehat{\mathbf{g}}_t = \frac{1}{2\sigma B}\sum_i a_{t,i}\,\mathbf{z}_{t,i}$, $a_{t,i} = \delta_{t,i}/(s_t+\epsilon)$, which is the estimator used throughout the paper. The variance bound itself (Theorem 2) is stated for the canonical estimator $\widehat{\mathbf{g}}_{\text{can}}$; Corollary 1 transfers it to the normalized estimator at the cost of a relative $O(B^{-1} + d_{\text{eff}}^{-1})$ in the constants. The $O(\sigma^2)$ smoothing remainder in (A4) is the only residual bias.

Remark (Comparison with MeZO). At the same batch size, MeZO draws a single direction $\mathbf{z}$ and evaluates the *batch-mean* loss, so its estimator is $\langle\bar{\mathbf{g}},\mathbf{z}\rangle\mathbf{z}$ and the derivation of Appendix J.6.4 with $B{=}1$ and $\mathbf{g}_i$ replaced by $\bar{\mathbf{g}}$ gives

$$\text{Var}(\widehat{\mathbf{g}}_{\text{MeZO}}) = (D+m_4-2)\,\|\bar{\mathbf{g}}\|^2 \to (D{+}m_4{-}2)\Big(\|\nabla F\|^2 + \frac{\nu^2}{B}\Big) \tag{87}$$

after the outer data expectation. MeZO therefore already averages the *data* noise over the mini-batch, and it is the *directional* noise that GRZO additionally averages; comparing against an $O(D\|\nabla F\|^2 + D\nu^2)$ bound would overstate the gain by attributing to MeZO a batch size of one. With $r = \nu^2/\|\nabla F\|^2$, dividing (87) by (59) at the Rademacher default ($m_4{=}1$) gives the ratio

$$\frac{\text{Var}(\widehat{\mathbf{g}}_{\text{MeZO}})}{\text{Var}(\widehat{\mathbf{g}}_{\text{can}})} \;\approx\; \frac{B+r}{1+r} \;=\; cB + (1-c) \;=:\; B_{\text{eff}}, \tag{88}$$

where $c = 1/(1+r)$ is the mean pairwise cosine between per-example gradients, since $\mathbb{E}\langle\mathbf{g}_i,\mathbf{g}_j\rangle = \|\nabla F\|^2$ and $\mathbb{E}\|\mathbf{g}_i\|^2 = \|\nabla F\|^2 + \nu^2$ for $i \ne j$. The reduction is thus B-fold when per-example gradients are aligned and degrades smoothly to no reduction when they are mutually orthogonal, which is the precise content of the B_{eff} used in Section 4.

K LLM Usage

Large language models were used only as writing assistants for minor grammar and phrasing polish on author-drafted text. They played no role in research conception, experimental design, or interpretation of results. All technical content and claims are the authors' own and were independently verified.