# LIZO: ENHANCING ZEROTH-ORDER LLM FINE-TUNING VIA LATERAL INHIBITION-INSPIRED PARAMETER PERTURBATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Fine-tuning is widely used to adapt large language models (LLMs) to downstream tasks, while zeroth-order (ZO) optimization provides a memory-efficient alternative through forward-only gradient estimation. However, existing ZO methods lack a principled characterization of parameter importance for guiding where perturbations should be introduced. In biological neural systems, lateral inhibition (LI) emphasizes the contrast between individual neuronal activity and its surrounding population. Inspired by this principle, we formulate an energy function to characterize the relative importance of individual parameters within their population and derive an LI score for parameter selection. Based on perturbation analysis, we show that parameters with higher LI scores are more sensitive to perturbations. We therefore propose Lateral-Inhibition-inspired Zeroth-Order Optimization (LIZO), which preserves parameters with higher LI scores and restricts perturbations and updates to parameters with lower scores. Extensive experiments on multiple LLMs and downstream tasks show that LIZO consistently improves over existing ZO methods across a range of settings.

## 1 INTRODUCTION

Large Language Models (LLMs) have demonstrated strong capabilities across a wide range of domains Brown et al. (2020); Touvron et al. (2023). Fine-tuning adapts pretrained LLMs to downstream tasks by updating their parameters. Full-parameter fine-tuning typically relies on backpropagation (BP) and first-order (FO) optimization Devlin et al. (2019); Ouyang et al. (2022), which requires storing intermediate activations and computing gradients. As model size increases, the resulting memory and computational costs make full-parameter fine-tuning increasingly challenging in resource-constrained environments.

Parameter-Efficient Fine-Tuning (PEFT) methods Houlsby et al. (2019); Hu et al. (2022); Dettmers et al. (2023) reduce the number of trainable parameters by updating only a subset of model parameters or introducing additional lightweight parameters. However, most PEFT methods still rely on BP and therefore require activation storage during training. Zeroth-order (ZO) optimization provides an alternative by estimating optimization directions from finite differences of objective function values without explicit gradient computation Malladi et al. (2023). MeZO Malladi et al. (2023) applies ZO optimization to LLM fine-tuning and reports up to a $12\times$ reduction in memory consumption compared with BP-based fine-tuning. Subsequent studies have explored different strategies to improve ZO efficiency and performance Chen et al. (2025); Dang et al. (2026). In particular, SubZero Yu et al. (2025) and S-MeZO Liu et al. (2025) restrict perturbations to lower-dimensional subspaces or selected parameters. S-MeZO selects parameters with relatively small magnitudes for perturbation and reports a $3.5\times$ convergence speedup over MeZO. These results suggest that parameter selection can play an important role in ZO fine-tuning, but also raise a fundamental question: **what parameter characteristics should be used to guide perturbation selection?**

In biological neural systems, lateral inhibition (LI) is a mechanism through which neuronal activity suppresses the responses of neighboring neurons, enhancing the contrast among neuronal responses Hartline & Ratliff (1957); Isaacson & Scanziani (2011). This mechanism emphasizes the response of an individual neuron relative to its surrounding population, rather than considering its



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

absolute response alone. Similar population-relative formulations have also been explored in artificial neural networks Yang et al. (2021); Liu et al. (2026). Motivated by this principle, we investigate whether the relative characteristics of parameters within their population can provide a more informative criterion for ZO perturbation selection.

In this paper, we propose Lateral-Inhibition-inspired Zeroth-Order Optimization (**LIZO**) for LLM fine-tuning. We first examine magnitude-based parameter selection and find that restricting ZO perturbations to small-magnitude parameters can lead to suboptimal performance. We then introduce an LI-inspired energy formulation to characterize the relative significance of individual parameters within their parameter population and derive an LI score for parameter selection. Our perturbation analysis shows that parameters with higher LI scores exhibit larger changes in their importance under perturbation. Based on this property, LIZO preserves parameters with higher LI scores and restricts ZO perturbations and updates to parameters with lower scores.

The main contributions of this paper are summarized as follows:

- We demonstrate that restricting ZO perturbations to small-magnitude parameters can lead to suboptimal performance, indicating that parameter magnitude alone is not a reliable criterion for parameter selection in ZO-based LLM fine-tuning.
- Motivated by this finding, we introduce an energy formulation inspired by LI and derive an LI score to quantify the importance of the parameters. We provide a theoretical sensitivity analysis of LI scores, showing that high-importance parameters are more sensitive to perturbations. Based on this observation, we propose **LIZO**, which preserves high-importance parameters while restricting perturbations and updates to parameters with lower LI scores.
- Extensive experiments on Llama3-8B and OPT-13B on multiple downstream benchmarks, including SuperGLUE and question-answer tasks (SQuAD and DROP), demonstrate that LIZO achieves the best performance in most cases.

## 2 Related Works

### 2.1 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) has emerged as an effective approach to adapting large language models (LLMs) to downstream tasks while reducing the computational and memory costs associated with full-parameter fine-tuning. Rather than updating all pretrained parameters, PEFT methods typically freeze the majority of the model and optimize a small set of task-specific parameters. Adapter-based methods Houlsby et al. (2019) introduce lightweight trainable modules into pretrained networks, while prefix tuning Li & Liang (2021) and prompt tuning Lester et al. (2021) adapt language models by optimizing continuous task-specific representations. LoRA Hu et al. (2022) exploits the low-rank structure of weight updates by introducing trainable low-rank matrices while keeping the pretrained weights frozen. Building upon LoRA, QLoRA Dettmers et al. (2023) further reduces memory consumption by combining low-rank adaptation with 4-bit quantization of pretrained model weights.

Although these methods substantially reduce the number of trainable parameters and associated optimizer states, they generally rely on backpropagation to optimize parameters, which still requires gradient computation and the storage or recomputation of intermediate activations. Zeroth-order optimization offers an alternative to backpropagation-based fine-tuning by estimating gradients solely through objective function evaluations, thereby providing a complementary approach to reducing the memory requirements of LLM adaptation.

### 2.2 Zeroth-Order Optimization

Unlike gradient-based optimization, which relies on backpropagation to obtain exact gradients, zeroth-order (ZO) optimization estimates gradients solely from objective function evaluations. Given the model parameters $\theta \in \mathbb{R}^d$ and a random perturbation vector $z \sim \mathcal{N}(0, I_d)$, a commonly used two-point ZO gradient estimator is

$$\hat{g}(\theta) = \frac{\mathcal{L}(\theta + \varepsilon z) - \mathcal{L}(\theta - \varepsilon z)}{2\varepsilon} z, \tag{1}$$

where $\varepsilon > 0$ denotes the perturbation scale. Based on this estimator, MeZO Malladi et al. (2023) performs in-place perturbation and parameter updates using only forward passes, reducing the memory cost of LLM fine-tuning to approximately the inference level. However, due to the stochastic perturbation, ZO gradient estimates can suffer from substantial estimation error and high variance, particularly in the extremely high-dimensional parameter space of LLMs, resulting in slow convergence and suboptimal performance.

Several variants of MeZO have therefore been proposed to improve the efficiency and quality of ZO optimization. FZOO Dang et al. (2026) accelerates ZO fine-tuning through batched one-sided gradient estimates with adaptive step sizes and Rademacher perturbations, substantially reducing the number and cost of forward passes. LOZO Chen et al. (2025) exploits the intrinsic low-rank structure of gradients in LLM fine-tuning and constructs a low-rank ZO gradient estimator, effectively restricting gradient estimation to a lower-dimensional subspace. S-MeZO Liu et al. (2025) approaches the problem from parameter selection and observes that ZO estimation noise has a stronger detrimental effect on large-magnitude weights than on small-magnitude ones. It therefore selectively perturbs and updates parameters with smaller magnitudes, reducing the effective optimization dimension and improving both convergence and downstream performance.

# 3 METHOD

In this section, we first analyze the behavior of magnitude-based parameter perturbation. We then introduce a lateral-inhibition-inspired formulation for characterizing parameter importance in LLMs. Finally, we present *Lateral-Inhibition-Inspired Zeroth-Order Optimization* (LIZO), which leverages the resulting importance scores to guide parameter perturbation during LLM fine-tuning.

## 3.1 ANALYSIS OF MAGNITUDE-BASED PARAMETER SELECTION

Parameter magnitude provides a simple and efficient criterion for parameter selection in ZO optimization Liu et al. (2025). However, recent studies have shown that even small-magnitude weights can encode task-specific knowledge Yin et al. (2024), suggesting that magnitude alone may be insufficient for identifying parameters suitable for ZO perturbation. To examine this issue, we use the representative magnitude-based ZO method, S-MeZO, to fine-tune Llama3-8B on SST-2. Specifically, S-MeZO incorporates a sparsity mask $m$ into the ZO gradient estimator:

$$\hat{g}(\theta) = \frac{\mathcal{L}(\theta + \varepsilon m \odot z) - \mathcal{L}(\theta - \varepsilon m \odot z)}{2\varepsilon}(m \odot z), \quad (2)$$

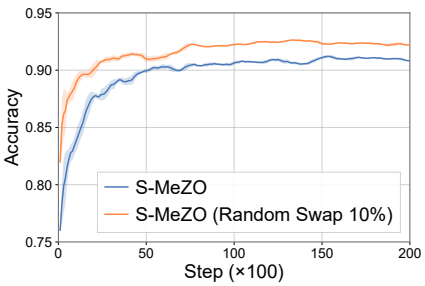


Figure 1: Effect of $10\%$ random swapping in S-MeZO on SST-2 accuracy.

where $\odot$ denotes the element-wise multiplication. We use a sparsity ratio of $25\%$ as the baseline. To evaluate the reliability of magnitude-based selection, we randomly select $10\%$ of the parameters from the original S-MeZO mask and replace them with the same number of parameters randomly sampled from outside the mask, keeping the overall perturbation ratio fixed. Interestingly, this simple random swapping improves the accuracy from $91.39\%$ to $92.66\%$, as shown in Fig. 1. This result suggests that parameter magnitude alone is not a reliable indicator of which parameters should be selected for ZO perturbation. This motivates us to develop an importance measure that can characterize the relative significance of individual parameters within their parameter populations, thereby guiding parameter selection for ZO perturbation.

## 3.2 MODELING PARAMETER IMPORTANCE WITH LATERAL INHIBITION

In biological neural systems, the functional significance of a neuron is influenced not only by its individual response but also by its interactions with surrounding neurons. Lateral inhibition (LI) is one mechanism that modulates neuronal responses through such local interactions Isaacson & Scanziani (2011). Inspired by this mechanism, we model the importance of parameters by considering the relationship between a target parameter and its surrounding parameters.

Specifically, for a parameter $x_t$, we formulate a least-square discrimination problem between $x_t$ and its surrounding parameter population $\{x_i\}_{i\neq t}$, where $i \in \{1, \ldots, d\}$:

$$\mathcal{E}_t(w_t, b_t) = (y_I - w_t x_t - b_t)^2 + \frac{1}{d-1}\sum_{i\neq t}(y_S - w_t x_i - b_t)^2, \tag{3}$$

where $w_t$ and $b_t$ are the coefficients of an affine discriminant, while $y_I$ and $y_S$ denote the target and surrounding labels, respectively. Setting $y_I = 1$ and $y_S = -1$, the minimum LI energy, as established in Lemmas 1 and 2, is

$$e_t^* = \frac{4\sigma_t^2}{(x_t - \mu_t)^2 + 2\sigma_t^2}. \tag{4}$$

where $\mu_t = \frac{1}{d-1}\sum_{i\neq t} x_i$, $\sigma_t^2 = \frac{1}{d-1}\sum_{i\neq t}(x_i - \mu_t)^2$. A lower energy indicates that the target parameter can be more distinctly discriminated from its surrounding population. We therefore define the LI importance score as the inverse of the minimum energy:

$$I_t = \frac{1}{e_t^*} = \frac{(x_t - \mu_t)^2}{4\sigma_t^2} + \frac{1}{2}, \tag{5}$$

where $\sigma_t^2 > 0$. The resulting score increases with the normalized deviation of the target parameter from its surrounding population, providing a parameter-wise measure of relative importance. The detailed derivation is provided in the appendix.

### 3.3 Lateral-Inhibition-Inspired Zeroth-Order Optimization

In this section, we investigate how perturbations affect the LI importance scores and use this analysis to derive an importance-guided parameter selection strategy for ZO optimization.

**Efficient importance estimation.** Computing the target-specific statistics (*i.e.*, $\mu_t$ and $\sigma_t$) in Eq. 5 introduces additional computational overhead during fine-tuning. To simplify importance estimation, we approximate the surrounding population statistics using the mean and variance of the entire parameter population Yang et al. (2021). The resulting importance score is

$$\widetilde{I}_t = \frac{(x_t - \mu)^2}{4\sigma^2} + \frac{1}{2}, \tag{6}$$

where

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i, \qquad \sigma^2 = \frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2$$

are shared across the parameter population. Thus, $\widetilde{I}_t$ increases with the squared deviation of $x_t$ from the population mean, assigning higher scores to larger deviations.

**Parameter selection for ZO fine-tuning.** In ZO optimization, parameter perturbations serve as probes for estimating the optimization direction rather than as the final parameter updates. When perturbations are applied to only a subset of parameters, the choice of this subset determines where the probes are introduced. Since a pretrained LLM already has a well-established parameter configuration, we aim to avoid unnecessary perturbations to parameters that play a more distinctive role within the parameter population. This motivates us to consider parameters with lower relative importance as perturbation candidates. We therefore investigate whether the LI importance score can characterize how individual parameters respond to ZO perturbations.

**Perturbation sensitivity of LI importance.** To this end, we analyze how the LI importance score changes when an individual parameter is perturbed. Consider a parameter population with mean $\mu$ and variance $\sigma^2 > 0$, which are treated as fixed references when evaluating the effect of a perturbation. For a target parameter $x_t$, we introduce a random perturbation as

$$x_t' = x_t + \varepsilon z_t, \qquad z_t \sim \mathcal{N}(0, 1), \tag{7}$$

where $\varepsilon > 0$ is the perturbation scale. The corresponding importance score after perturbation is

$$\widetilde{I}'_t = \frac{(x'_t - \mu)^2}{4\sigma^2} + \frac{1}{2}. \tag{8}$$

We define perturbation sensitivity as the expected squared change in the LI importance score:

$$D_t = \mathbb{E}_{z_t}\left[\left(\widetilde{I}'_t - \widetilde{I}_t\right)^2\right]. \tag{9}$$

The following proposition relates perturbation sensitivity to LI importance.

**Proposition 1.** *Let $\widetilde{I}_t$ be the LI importance score defined in Eq. 6, with population statistics $\mu$ and $\sigma^2 > 0$. For $z_t \sim \mathcal{N}(0,1)$, the expected squared change in the importance score is*

$$D_t = \frac{\varepsilon^2}{\sigma^2}\left(\widetilde{I}_t - \frac{1}{2}\right) + \frac{3\varepsilon^4}{16\sigma^4}. \tag{10}$$

*Consequently,*

$$\frac{\partial D_t}{\partial \widetilde{I}_t} = \frac{\varepsilon^2}{\sigma^2} > 0. \tag{11}$$

*Furthermore, for a binary mask $m \in \{0,1\}^d$ selecting $k$ parameters, i.e., $\sum_{t=1}^{d} m_t = k$, the expected squared change is minimized by parameters with the lowest $k$ LI importance scores.*

Proposition 1 shows that parameters with higher LI importance exhibit greater sensitivity of their importance scores to perturbations. For a fixed perturbation budget, selecting parameters with lower LI importance therefore minimizes the expected squared change in their importance scores. Motivated by this result, LIZO restricts random perturbations to parameters with lower LI importance, while leaving higher-importance parameters unperturbed during ZO gradient estimation.

**Importance-guided parameter perturbation.** Based on the LI importance criterion, we propose LIZO, which selects low-importance parameters for ZO perturbations. We compute the LI importance scores and use the score at the $r$-th percentile as the selection threshold. A binary mask $m \in \{0,1\}^d$ is then constructed as

$$m_t = \begin{cases} 1, & \widetilde{I}_t < \tau_r, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\widetilde{I}_t$ denotes the approximated LI importance score of parameter $x_t$, and $\tau_r$ is the threshold corresponding to the lowest $r\%$ of parameters. Thus, parameters with $\widetilde{I}_t < \tau_r$ are selected for perturbation. Given a random perturbation vector $z \in \mathbb{R}^d$, we define the masked perturbation as

$$\hat{z} = m \odot z. \tag{13}$$

The resulting LIZO gradient estimator is

$$\hat{g}(\theta) = \frac{\mathcal{L}(\theta + \varepsilon\hat{z}) - \mathcal{L}(\theta - \varepsilon\hat{z})}{2\varepsilon}\hat{z}. \tag{14}$$

Only the selected parameters are perturbed during ZO gradient estimation, while the remaining parameters remain unchanged. The overall procedure is summarized in Algorithm 1.

**Relation to magnitude-based selection.** Recent work on S-MeZO Liu et al. (2025) selects small-magnitude parameters for ZO fine-tuning. In contrast, LIZO ranks parameters by their normalized deviation from the population mean rather than their absolute magnitudes. As illustrated in Fig. 6, parameters with similar magnitudes can have substantially different LI importance scores.

# 4 Experiments

## 4.1 Experimental Setup

**Models and Datasets.** We evaluate LIZO on SuperGLUE Wang et al. (2019) tasks (WiC, MultiRC, CB, COPA, SST-2, RTE, and BoolQ) and question answering tasks (SQuAD Rajpurkar et al. (2016)

**Algorithm 1** Lateral-Inhibition-Inspired Zeroth-Order Optimization (LIZO)

**Require:** Training dataset $\mathcal{D}$; pretrained parameters $\theta^0$; learning rate $\eta$; perturbation scale $\varepsilon$; selection ratio $r$; regularization coefficient $\lambda$; total steps $K$.
**Ensure:** Fine-tuned parameters $\theta^K$.

```
for k = 0, ..., K − 1 do
    Compute LI-based parameter importance:
    for all eligible weight tensors W ⊆ θ^k do
        Calculate Ĩ_t based on Eq. 6.
    end for
    Generate the perturbation mask:
    Determine the LI score threshold τ_r corresponding to the lowest r% of {Ĩ_t}
    m_t ← 𝕀[Ĩ_t < τ_r]
    Estimate the ZO gradient:
    Sample mini-batch B_k ∼ D
    Sample z_k ∼ N(0, I_d)
    ẑ_k ← m ⊙ z_k
    p_k ← (L_{B_k}(θ^k + εẑ_k) − L_{B_k}(θ^k − εẑ_k)) / (2ε)
    Update the selected parameters:
    θ^{k+1} ← θ^k − ηp_k ẑ_k
end for
return θ^K
```

and DROP Dua et al. (2019)). We use Llama3-8B Grattafiori et al. (2024) and OPT-13B Zhang et al. (2022) to cover different model sizes and architectures.

**Baselines and Evaluation.** We compare LIZO with four ZO fine-tuning methods: MeZO Malladi et al. (2023), FZOO Dang et al. (2026), LOZO Chen et al. (2025), and S-MeZO Liu et al. (2025). Adam Kingma & Ba (2015) and LoRA Hu et al. (2022) are included as first-order baselines.

**Implementation Details.** We use 20,000 training steps and a batch size of 16 for all model–dataset pairs. For LIZO, the parameter sampling ratio is set to $r = 25\%$. For each task, we sample 1,000 training examples, except for CB and COPA, for which we use 200 and 350 examples, respectively. The remaining training settings are listed in Tab. 2 in the Appendix.

## 4.2 Performance

### 4.2.1 Comparison with Previous Methods

**Overall Performance.** Tab. 1 compares LIZO with existing methods on Llama3-8B and OPT-13B. Among ZO methods, LIZO obtains the highest average accuracy on both models, reaching 82.22% on Llama3-8B and 76.33% on OPT-13B, which are 3.06 and 2.03 percentage points higher than the strongest ZO baselines, respectively. On question answering, LIZO achieves an average F1 score of 75.96% on Llama3-8B, 8.01 percentage points above S-MeZO, and 58.08% on OPT-13B, the highest among the ZO methods.

**Comparison with ZO Methods.** LIZO performs consistently well at the task level. On Llama3-8B, it performs the best among ZO methods on all seven language understanding tasks, with gains of 4.70, 3.62, and 3.57 percentage points on WiC, RTE, and CB, respectively. On OPT-13B, LIZO ranks highest on five of the seven tasks, improving over the strongest ZO results by 3.20, 2.51, and 2.00 percentage points on BoolQ, WiC, and COPA, respectively. For question answering, LIZO also performs the best, reaching 85.91% F1 on SQuAD with OPT-13B.

**Comparison with First-Order Methods.** We compare LIZO with two first-order methods, Adam and LoRA. On Llama3-8B, LIZO achieves an average accuracy of 82.22%, close to LoRA (82.51%). LIZO outperforms both methods on COPA and DROP, reaching 90.00% accuracy and 63.15% F1, respectively. It also surpasses LoRA on CB and RTE by 1.80 and 3.58 percentage points, respectively, while achieving comparable accuracy on SST-2 (94.95% vs. 95.00%). On question answering, LIZO achieves an average F1 score of 75.96%, exceeding Adam and LoRA by 1.06 and

Table 1: Performance on language understanding and question answering tasks. Bold and underline indicate the best and second-best results, respectively.

| Method | Accuracy (%) ↑ | | | | | | | | F1 (%) ↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | WiC | MultiRC | CB | COPA | SST-2 | RTE | BoolQ | Avg. | SQuAD | DROP | Avg. |
| **Llama3–8B** | | | | | | | | | | | |
| FO (Adam) | 72.60 | 84.70 | 92.00 | 89.00 | 96.00 | 92.00 | 86.60 | 87.56 | 90.40 | 59.40 | 74.90 |
| FO (LoRA) | 70.70 | 82.40 | 73.20 | 89.00 | 95.00 | 80.90 | 86.40 | 82.51 | 89.40 | 58.20 | 73.80 |
| MeZO | 57.80 | 77.60 | 69.60 | 88.00 | 92.20 | 74.40 | 76.70 | 76.61 | 86.70 | 57.10 | 71.90 |
| FZOO | 60.80 | 81.50 | 69.60 | 88.00 | 94.30 | 77.60 | 81.80 | 79.09 | 87.90 | 56.50 | 72.20 |
| LOZO | 55.80 | 78.10 | 69.60 | 84.00 | 92.50 | 66.80 | 79.40 | 75.17 | **89.00** | **65.40** | **77.20** |
| S-MeZO | 62.54 | 78.30 | 71.43 | 89.00 | 91.39 | 80.86 | 80.60 | 79.16 | 87.50 | 48.40 | 67.95 |
| LIZO | **67.24** | **80.70** | **75.00** | **90.00** | **94.95** | **84.48** | **83.20** | **82.22** | 88.76 | 63.15 | 75.96 |
| **OPT–13B** | | | | | | | | | | | |
| FO (Adam) | 66.30 | 76.20 | 94.60 | 88.00 | 95.30 | 80.90 | 83.50 | 83.54 | 89.50 | 31.30 | 60.40 |
| FO (LoRA) | 64.30 | 69.40 | 69.60 | 89.00 | 94.80 | 78.30 | 80.20 | 77.94 | 88.00 | 30.90 | 59.45 |
| MeZO | 59.40 | 57.30 | 66.00 | 88.00 | 91.40 | 66.10 | 67.60 | 70.83 | 84.70 | **30.90** | 57.80 |
| FZOO | 60.50 | **66.00** | 69.60 | 87.00 | **93.70** | 71.10 | 72.20 | 74.30 | 84.80 | 28.70 | 56.75 |
| LOZO | 60.20 | 63.00 | 69.60 | 89.00 | 91.70 | 70.40 | 71.90 | 73.69 | 84.90 | 30.70 | 57.80 |
| S-MeZO | 58.80 | 64.70 | 71.43 | 89.00 | 90.40 | 74.70 | 66.40 | 73.60 | 84.20 | 29.55 | 56.88 |
| LIZO | **63.01** | 65.40 | **73.21** | **91.00** | 93.58 | **75.81** | **75.40** | **76.33** | **85.91** | 30.25 | **58.08** |

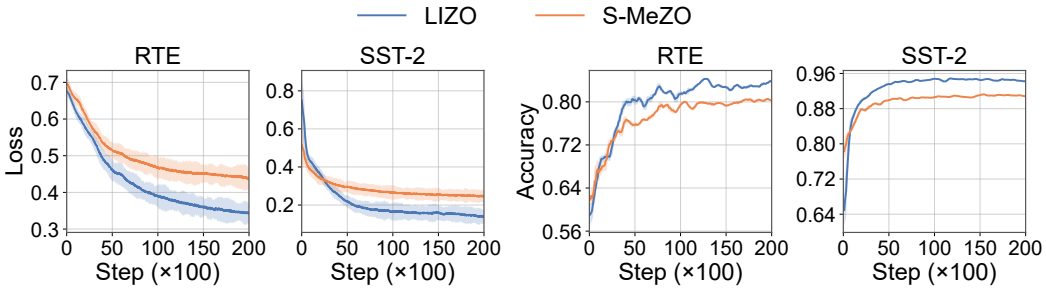


Figure 2: Loss and accuracy of LIZO and S-MeZO on RTE and SST-2 (Llama3-8B with $r = 25\%$).

2.16 percentage points, respectively. On OPT-13B, LIZO achieves 91.00% accuracy on COPA, outperforming Adam and LoRA by 3.00 and 2.00 percentage points, respectively, while its average accuracy of 76.33% remains close to LoRA (77.94%). Overall, LIZO remains competitive with FO methods while retaining the forward-only training paradigm of ZO optimization.

### 4.2.2 Convergence

We investigate the convergence behavior of LIZO on RTE and SST-2 using Llama3-8B with a parameter sampling ratio of $r = 25\%$. S-MeZO uses magnitude-based parameter selection and serves as the baseline. As shown in Fig. 2, both methods improve in accuracy and reduce loss during the early stages of training, followed by slower gains. LIZO reaches higher final accuracy and lower loss than S-MeZO in both tasks.

We further compare the iterations needed to reach selected accuracy thresholds. On RTE, LIZO reaches 80% accuracy in 3,800 iterations, while S-MeZO reaches the same threshold in 7,800 iterations. On SST-2, LIZO reaches 90% accuracy in 2,100 iterations, compared with 5,200 iterations for S-MeZO. Thus, LIZO reaches these accuracy targets in 48.7% and 40.4% of the iterations required by S-MeZO, corresponding to $2.05\times$ and $2.48\times$ faster convergence speeds respectively.

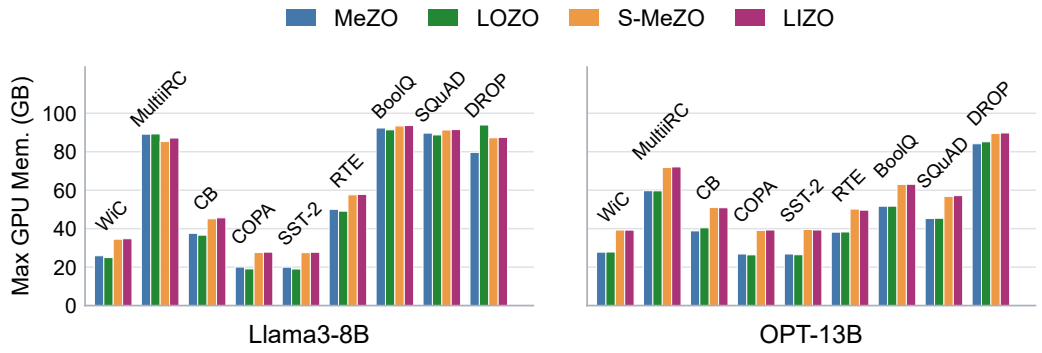


Figure 3: GPU memory usage of MeZO, LOZO, S-MeZO, and LIZO with a batch size of 16.

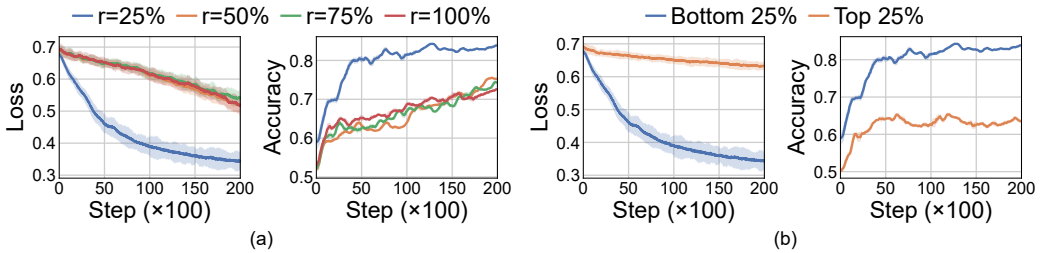


Figure 4: **(a) Effect of the perturbation ratio.** Loss and accuracy of LLaMA3-8B for $r \in \{25\%, 50\%, 75\%, 100\%\}$ on RTE. **(b) Effect of parameter selection.** Loss and accuracy when perturbing the bottom 25% versus the top 25% of parameters.

#### 4.2.3 Memory Usage

We compare the GPU memory usage of different ZO fine-tuning methods on Llama3-8B and OPT-13B. All experiments use a single NVIDIA RTX PRO 6000 GPU with 96 GB memory and a batch size of 16. We record the peak memory usage during the first 200 training iterations. As shown in Fig. 3, Parameter-selection-based methods generally use additional memory to store information for perturbation. LIZO has memory usage close to S-MeZO on both models. The small difference between the two methods suggests that LI-based parameter selection adds little GPU memory overhead while retaining the memory efficiency of sparse ZO fine-tuning.

### 4.3 Ablation Study

#### 4.3.1 Effect of Perturbation Ratio

We investigate the effect of the perturbation ratio $r$ by fine-tuning Llama3-8B on RTE using LIZO with $r \in \{25\%, 50\%, 75\%, 100\%\}$. As shown in Fig. 4(a), all four configurations exhibit progressively decreasing training loss and improving validation accuracy throughout fine-tuning. Notably, $r = 25\%$ achieves substantially faster convergence, consistently lower training loss, and higher validation accuracy than the other configurations. At the end of training, validation accuracy generally decreases as the perturbation ratio increases, suggesting that perturbing a smaller subset of parameters is more effective than applying perturbations to a larger parameter space. Based on these observations, we adopt $r = 25\%$ as the default perturbation ratio in all subsequent experiments.

#### 4.3.2 Effect of Parameter Selection

To investigate the effect of parameter selection in LIZO, we compare two perturbation strategies on Llama3-8B fine-tuned on RTE. Specifically, we select either the bottom 25% or the top 25% of

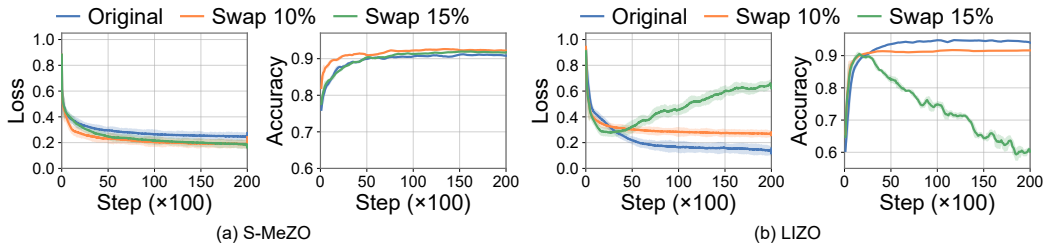


Figure 5: Effects of random mask swapping on S-MeZO and LIZO. We randomly replace 10% or 15% of the selected parameters with parameters outside the corresponding mask while keeping the overall perturbation ratio unchanged.

parameters ranked by their LI scores for perturbation and update, while keeping the perturbation ratio and other training settings unchanged.

As shown in Fig. 4(b), perturbing the bottom 25% of parameters leads to a more pronounced reduction in training loss and a higher validation accuracy than perturbing the top 25%. The training loss of the bottom-25% strategy decreases steadily from approximately 0.68 to 0.35, whereas that of the top-25% strategy declines only slightly, from approximately 0.69 to 0.63. A similar difference is observed in validation accuracy. The bottom-25% strategy rapidly improves accuracy during the early training stages and reaches approximately 83%, while the top-25% strategy increases more slowly and stabilizes at around 64%.

These results favor perturbing the bottom 25% of parameters over the top 25% under the same perturbation budget. This is consistent with our strategy of perturbing lower-importance parameters and preserving those with higher LI scores.

#### 4.3.3 Sensitivity to Mask Perturbation

To examine the effect of the parameter selection criterion, we test the sensitivity of S-MeZO and LIZO to perturbations of their parameter masks. Specifically, we randomly replace $10\%$ or $15\%$ of the selected parameters with parameters outside the corresponding mask, while keeping the overall perturbation ratio unchanged.

As shown in Fig. 5, S-MeZO is relatively insensitive to mask perturbations. Replacing $10\%$ or $15\%$ of the selected parameters causes a small performance change, with the $10\%$ random-swap setting even slightly outperforming the original mask. This is consistent with the analysis in Sec. 3.1, where magnitude-based selection shows limited discrimination among perturbation candidates.

In contrast, LIZO is much more sensitive to changes in its LI-based mask. Replacing $10\%$ of the selected parameters already reduces the final accuracy, while a $15\%$ replacement leads to a clear degradation in both accuracy and loss. These results suggest that the LI-based mask identifies more specific perturbation candidates, and that disrupting this selection can impair ZO fine-tuning.

## 5 Conclusion

In this work, we propose LIZO, a lateral inhibition-inspired zeroth-order optimization method for efficient LLM fine-tuning. LIZO introduces a bio-inspired energy criterion to characterize parameter importance for ZO perturbation. It preserves high-importance parameters while selectively perturbing low-importance parameters for model updates. Extensive experiments demonstrate that LIZO consistently improves the performance of ZO-based fine-tuning.

### AI USE STATEMENT

In this work, we used generative AI tools for polish writing and improve the clarity of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of the results presented in this work. Detailed descriptions of the methodology, experimental settings, datasets, evaluation protocols, and hyperparameter configurations are provided in the main paper and Appendix. Additional implementation details and theoretical derivations are also included where appropriate. The source code and scripts for reproducing the experiments will be made publicly available.

## REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order fine-tuning for language models with low-rank structures. In *ICLR*, 2025.

Sizhe Dang, Yanjun Zhao, Xiaodong Zheng, Guang Dai, Ivor Tsang, and Haishan Ye. Fzoo: Fast zeroth-order optimizer for fine-tuning large language models towards adam-scale speed. In *ICLR*, 2026.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *NeurIPS*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*, 2019.

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

H Keffer Hartline and Floyd Ratliff. Inhibitory interaction of receptor units in the eye of limulus. *The Journal of general physiology*, 40(3):357–376, 1957.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Jeffry S Isaacson and Massimo Scanziani. How inhibition shapes cortical activity. *Neuron*, 72(2): 231–243, 2011.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2015.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL-IJCNLP*, 2021.

Peiyu Liu, Jianhao Ding, and Zhaofei Yu. Training deep normalization-free spiking neural networks with lateral inhibition. In *ICLR*, 2026.

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. In *NeurIPS*, 2025.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *NeurIPS*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*, 2019.

Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie. Simam: A simple, parameter-free attention module for convolutional neural networks. In *ICML*, 2021.

Lu Yin, Ajay Kumar Jaiswal, Shiwei Liu, Souvik Kundu, and Zhangyang Wang. Junk DNA hypothesis: Pruning small pre-trained weights *Irreversibly* and *Monotonically* impairs "difficult" downstream tasks in LLMs. In *ICML*, 2024.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of llms in random subspaces. In *ICCV*, 2025.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

# A Appendix

## A.1 Theoretical Analysis

We first establish the relationship between the LI energy and parameter importance, and then analyze the sensitivity of the resulting importance score to stochastic perturbations.

**Lemma 1** (Closed-Form LI Energy)**.** *Consider a target parameter $x_t \in \mathbb{R}$ and its surrounding parameter population $\{x_i\}_{i\neq t}$, where $i \in \{1, \dots, d\}$ and $d \geq 2$. The LI energy is defined as*

$$\mathcal{E}_t(w, b) = (1 - wx_t - b)^2 + \frac{1}{d-1}\sum_{i\neq t}(-1 - wx_i - b)^2. \tag{15}$$

*Let*

$$\mu_t = \frac{1}{d-1}\sum_{i\neq t} x_i, \quad \sigma_t^2 = \frac{1}{d-1}\sum_{i\neq t}(x_i - \hat{\mu}_t)^2. \tag{16}$$

*Then, the minimum LI energy is given by*

$$\boxed{e_t^* = \min_{w,b} \mathcal{E}_t(w, b) = \frac{4\sigma_t^2}{(x_t - \mu_t)^2 + 2\sigma_t^2}}. \tag{17}$$

*Proof.* Using the mean-variance decomposition, the objective becomes

$$\mathcal{E}_t(w,b) = (1 - wx_t - b)^2 + (-1 - w\mu_t - b)^2 + \sigma_t^2 w^2. \tag{18}$$

For simplicity, define

$$\Delta_t = x_t - \mu_t, \qquad v_t = \sigma_t^2. \tag{19}$$

Setting the derivative with respect to $b$ to zero gives

$$\frac{\partial \mathcal{E}_t}{\partial b} = 0 \quad \Longrightarrow \quad b^* = -\frac{w(x_t + \mu_t)}{2}. \tag{20}$$

Substituting $b^*$ into the objective yields

$$\begin{aligned} \mathcal{E}_t(w, b^*) &= 2\left(1 - \frac{w\Delta_t}{2}\right)^2 + v_t w^2 \\ &= 2 - 2w\Delta_t + \left(\frac{\Delta_t^2}{2} + v_t\right) w^2. \end{aligned} \tag{21}$$

The stationary condition with respect to $w$ gives

$$\frac{\partial \mathcal{E}_t}{\partial w} = -2\Delta_t + (\Delta_t^2 + 2v_t)w = 0. \tag{22}$$

Therefore,

$$w^* = \frac{2\Delta_t}{\Delta_t^2 + 2v_t}. \tag{23}$$

Substituting $w^*$ into the objective, we obtain

$$\begin{aligned} e_t^* &= 2 - \frac{2\Delta_t^2}{\Delta_t^2 + 2v_t} \\ &= \frac{4v_t}{\Delta_t^2 + 2v_t}, \end{aligned} \tag{24}$$

which completes the proof. □

**Lemma 2** (Energy-Based Parameter Importance)**.** *Given the minimum LI energy $e_t^*$, define the importance score of parameter $x_t$ as*

$$I_t = \frac{1}{e_t^*}. \tag{25}$$

*Then*

$$\boxed{I_t = \frac{(x_t - \mu_t)^2}{4\sigma_t^2} + \frac{1}{2}.} \tag{26}$$

*For fixed local population statistics, the importance score increases monotonically with the squared deviation $(t - \mu_t)^2$ and decreases monotonically with the LI energy.*

*Proof.* Taking the reciprocal of Eq. 17 gives

$$\begin{aligned} I_t &= \frac{(x_t - \mu_t)^2 + 2\sigma_t^2}{4\sigma_t^2} \\ &= \frac{(x_t - \mu_t)^2}{4\sigma_t^2} + \frac{1}{2}. \end{aligned} \tag{27}$$

Let $q_t = (x_t - \mu_t)^2$ and $v_t = \sigma_t^2 > 0$. Then

$$\frac{\partial I_t}{\partial q_t} = \frac{1}{4v_t} > 0. \tag{28}$$

Moreover, since $I_t = 1/e_t^*$,

$$\frac{\partial I_t}{\partial e_t^*} = -\frac{1}{(e_t^*)^2} < 0. \tag{29}$$

Thus, a lower LI energy corresponds to a higher importance score, completing the proof. □

**Proposition 1.** *Let $\widetilde{I}_t$ be the LI importance score defined in Eq. 6, with population statistics $\mu$ and $\sigma^2 > 0$. For $z_t \sim \mathcal{N}(0,1)$, the expected squared importance distortion satisfies*

$$D_t = \frac{\varepsilon^2}{\sigma^2}\left(\widetilde{I}_t - \frac{1}{2}\right) + \frac{3\varepsilon^4}{16\sigma^4}. \tag{30}$$

*Consequently,*

$$\frac{\partial D_t}{\partial \widetilde{I}_t} = \frac{\varepsilon^2}{\sigma^2} > 0. \tag{31}$$

*Furthermore, for a binary mask $m \in \{0,1\}^d$ selecting $k$ parameters, i.e., $\sum_{t=1}^d m_t = k$, the expected squared change is minimized by parameters with the lowest $k$ LI importance scores.*

*Proof.* Let $\delta_t = x_t - \mu$. Under the fixed-reference-statistics assumption, the importance score change is

$$\Delta\widetilde{I}_t = \frac{2\varepsilon\delta_t z_t + \varepsilon^2 z_t^2}{4\sigma^2}. \tag{32}$$

Using the Gaussian moments $\mathbb{E}[z_t^2] = 1$, $\mathbb{E}[z_t^3] = 0$, and $\mathbb{E}[z_t^4] = 3$, we obtain

$$\begin{aligned} D_t &= \mathbb{E}\left[(\Delta\widetilde{I}_t)^2\right] \\ &= \frac{\varepsilon^2\delta_t^2}{4\sigma^4} + \frac{3\varepsilon^4}{16\sigma^4} \\ &= \frac{\varepsilon^2}{\sigma^2}\left(\widetilde{I}_t - \frac{1}{2}\right) + \frac{3\varepsilon^4}{16\sigma^4}. \end{aligned} \tag{33}$$

For a binary mask $m$, the masked perturbation of parameter $x_t$ is $\varepsilon m_t z_t$. Since $m_t \in \{0,1\}$, the corresponding total expected squared distortion is

$$\begin{aligned} D(m) &= \sum_{t=1}^d \mathbb{E}\left[(\Delta\widetilde{I}_t(m_t))^2\right] \\ &= \frac{\varepsilon^2}{\sigma^2}\sum_{t=1}^d m_t\left(\widetilde{I}_t - \frac{1}{2}\right) + \frac{3k\varepsilon^4}{16\sigma^4}. \end{aligned} \tag{34}$$

For a fixed $k$, the second term is constant. The first term is minimized by selecting the $k$ smallest importance scores. □

## B Training Setting

For LIZO, we fix the parameter sampling ratio to $r = 25\%$ across all experiments. Following previous work, we randomly sample 1,000 training examples for each downstream task, except for CB and COPA, where 200 and 350 examples are used, respectively. For Llama3-8B, we set the perturbation scale to $\epsilon = 1 \times 10^{-3}$ and the learning rate to $5 \times 10^{-6}$. For OPT-13B, we use $\epsilon = 1 \times 10^{-4}$ and a learning rate of $5 \times 10^{-7}$, while $\epsilon$ and learning rate are set to $1 \times 10^{-3}$ and $1 \times 10^{-6}$ for DROP. For the compared methods, we follow their publicly available implementations and recommended configurations, and report results from the original papers when applicable.

Table 2: Training settings of LIZO.

| Model | Learning Rate | $\epsilon$ | Sampling Ratio $r$ | Training Samples |
|---|---|---|---|---|
| Llama3-8B | $5 \times 10^{-6}$ | $1 \times 10^{-3}$ | 25% | 1,000 |
| OPT-13B | $5 \times 10^{-7}$ | $1 \times 10^{-4}$ | 25% | 1,000 |
| OPT-13B (DROP) | $1 \times 10^{-6}$ | $1 \times 10^{-3}$ | 25% | 1,000 |

*Note:* LIZO uses a sampling ratio of $r = 25\%$ for all experiments. Following previous work, we randomly sample 1,000 training examples for each task, except for CB and COPA, where 200 and 350 examples are used, respectively.

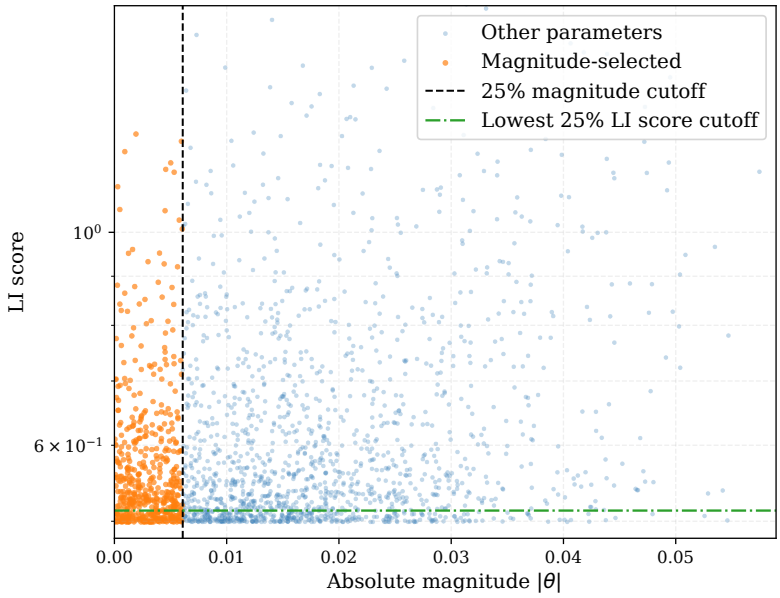


Figure 6: Distribution of parameter magnitudes and LI scores. Orange points denote the 25% of parameters with the smallest absolute magnitudes, while blue points denote the remaining parameters. The black vertical dashed line indicates the magnitude-based selection threshold, and the green horizontal dash-dotted line indicates the threshold corresponding to the lowest 25% LI scores . The LIscore axis is shown on a logarithmic scale.

## C Magnitude and LI Score Distributions

We further examine the relationship between parameter magnitude and LI score using the $up_proj$ weight matrix from the 15th transformer layer of Llama3-8B. Both magnitude-based and LI-score-based parameter selection use a fixed selection ratio of $r = 25\%$. All statistics are computed over the entire weight matrix, while 2,000 parameters are randomly sampled for visualization. As shown in Fig. 6, parameters with small absolute magnitudes do not necessarily correspond to low LI scores. In particular, some parameters selected by the magnitude criterion exhibit relatively high LI scores, indicating that parameter magnitude and LI score can characterize parameters differently.