STABILIZING ZEROTH-ORDER FINE-TUNING OF MIXTURE-OF-EXPERTS VIA ONE-SIDED ESTIMATION AND EXPERT REPLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Zeroth-order (ZO) optimization enables memory-efficient fine-tuning through forward loss queries. In sparse mixture-of-experts (MoE) models, however, parameter perturbations can change the experts selected for a token. This path inconsistency mixes continuous parameter effects with discrete routing changes in finite-difference estimates. We propose ZOMoE, a ZO fine-tuning algorithm that combines expert replay with one-sided estimation. At each update, an unperturbed forward pass records the selected expert indices. Perturbed queries reuse these indices while recomputing routing coefficients and other continuous computations. The next update begins with a fresh unperturbed forward pass to refresh expert selections; router parameters remain frozen. The recording pass also supplies a shared loss baseline, so replay adds no forward queries to the one-sided estimator. Mathematically, we isolate the term introduced by expert switching in the finite-difference estimate and show that expert replay removes this term. Across two sparse MoE backbones and six downstream tasks, ZOMoE achieves the highest average accuracy among the evaluated ZO methods, accelerates loss descent, and adds negligible memory or runtime overhead.

1 INTRODUCTION

Sparse mixture-of-experts (MoE) models expand language-model capacity while activating only a few experts per token (Fedus et al., 2022; Jiang et al., 2024). Yet sparse computation does not eliminate the memory burden of fine-tuning: conventional first-order training still stores the large parameter set along with activations, gradients, and optimizer states. This memory burden motivates zeroth-order (ZO) methods such as MeZO and FZOO, which enable memory-efficient fine-tuning of dense language models by estimating updates through forward loss queries, without backpropagation (Malladi et al., 2023; Dang et al., 2025).

However, whether ZO methods can effectively extend from dense language models to sparse MoEs remains underexplored. To investigate this question, we first evaluate MeZO and FZOO on COPA (Roemmele et al., 2011) and observe optimization difficulties (Figure 1a,b). MeZO fails to reduce its initial validation loss on OLMoE (Muennighoff et al., 2024) and loses early progress on Qwen3.5 (Team, 2026). FZOO initially improves but later stalls or rebounds.

To understand these difficulties, we examine the loss comparisons used by ZO. Each estimate compares losses at nearby parameter values, but in an MoE, a perturbation can also change the top-k experts selected for a token. Loss queries within the same estimate may therefore execute different expert subnetworks, mixing continuous parameter effects with discrete changes in expert selection. We call this disagreement *path inconsistency* and hypothesize that it contributes to the observed optimization difficulties. To assess its prevalence, we measure expert switching under parameter perturbations of both pretrained backbones. At perturbation scale $\epsilon = 10^{-4}$, expert sets change at 11.72% of non-padding token positions on OLMoE and 28.91% on Qwen3.5, averaged across routed layers (Figure 1c).

A direct approach to addressing this issue is to freeze the router parameters. However, expert selection depends jointly on the router weights and its input hidden states. Perturbations to upstream layers can change these hidden states and alter the ranking of expert scores, even when the router

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.

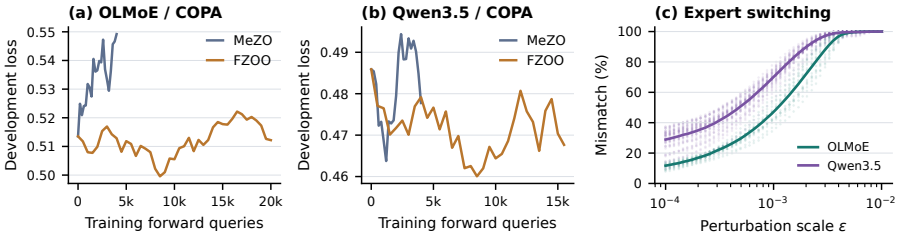

Figure 1: **Optimization difficulties and perturbation-induced expert switching in MoEs.** (a,b) MeZO and frozen-router FZOO development losses on COPA, plotted against training forward queries without smoothing. (c) Expert-set mismatch on SST-2 for both backbones, using 32 mini-batches and four perturbation directions. Dots show pooled per-layer rates; lines show layer means. At $\epsilon = 10^{-4}$, the means are 11.72% for OLMoE and 28.91% for Qwen3.5. Protocols are in Appendix C.1.

weights remain fixed. Preventing expert switching therefore requires controlling the selected expert indices within each ZO estimate.

We therefore propose ZOMoE, which maintains consistent expert selections through *expert replay* and combines replay with shared-baseline one-sided estimation. At each update, an unperturbed forward pass records the selected expert indices. All perturbed queries reuse these indices while recomputing expert outputs and continuous routing weights. A fresh unperturbed pass refreshes the indices at the next update, so expert selections remain adaptive across training even though router parameters stay frozen.

At the same time, we use *one-sided estimation* to turn these replayed loss queries into an update. For each sampled direction, we compare one perturbed loss with the unperturbed loss on the same mini-batch; the loss difference determines that direction's contribution to the update. Central differences query two opposite perturbations per direction, while one-sided estimation shares the unperturbed loss across directions. Since the replay recording pass already supplies this shared baseline, an update with N directions uses $N + 1$ forward queries, with no additional queries for replay. Our analysis shows that replay removes the expert-switching term from each finite-difference estimate. Under smoothness, mini-batch-sampling, and a condition controlling the loss change when expert indices are reselected between updates, we further establish an explicit convergence guarantee for ZOMoE.

To assess the resulting method's effectiveness and cost, we evaluate ZOMoE on two MoE backbones and six classification tasks. It achieves the highest average accuracy among the evaluated ZO methods, with the complete configurations improving on FZOO by 3.23 and 2.80 percentage points on OLMoE and Qwen3.5, respectively. ZOMoE also reaches FZOO's best development loss with fewer forward queries on 10 of 12 model–task pairs. Matched measurements show at most 0.04 GiB of additional peak memory and little throughput overhead. Controlled comparisons isolate replay's contribution, and additional QA evaluations examine its broader applicability.

Our contributions are threefold:

- We identify *path inconsistency* as a challenge in ZO fine-tuning of sparse MoEs, document perturbation-induced expert switching, and characterize its contribution to finite-difference estimates.
- We introduce ZOMoE, combining expert replay with one-sided estimation to maintain consistent expert selections within each update, and establish an explicit convergence guarantee for its normalized-and-clipped update.
- We evaluate ZOMoE on two MoE backbones across six classification and three QA tasks, demonstrating improved average accuracy and query efficiency with little additional memory or runtime cost.

2 Related Work

Zeroth-order optimization for language-model adaptation. ZO methods estimate search directions from function values (Spall, 1992; Nesterov & Spokoiny, 2017). MeZO applies this idea to language models with forward-only memory (Malladi et al., 2023). Later methods use sparse or low-rank perturbations, variance reduction, random or activation-guided subspaces, quantization, momentum, and one-sided normalization (Zhang et al., 2024; Liu et al., 2024; Gautam et al., 2024; Yu et al., 2024; Chen et al., 2024; Zhou et al., 2025; Peng et al., 2025; Dang et al., 2025; Lin et al., 2026). These methods improve estimator geometry or execution. ZOMoE addresses an orthogonal issue: nearby queries in a sparse MoE can execute different expert supports.

Mixture-of-Experts language models. Sparse MoEs activate a token-dependent subset of experts (Shazeer et al., 2017). GShard, Switch Transformer, DeepSpeed-MoE, Mixtral, OLMoE, and Qwen study scalable routing, load balance, training, and inference (Lepikhin et al., 2020; Fedus et al., 2022; Rajbhandari et al., 2022; Jiang et al., 2024; Muennighoff et al., 2024; Team, 2026). Routing replay has also been used to align rollout inference with RL training. R3 records routing distributions to reduce training–inference mismatch (Ma et al., 2025); the Composer 2 training system overrides rollout expert assignments while retaining current gating scores (Cursor Research Team, 2026). ZOMoE uses related expert-replay semantics for a different purpose: making the perturbed loss queries inside one ZO estimator comparable. Our novelty claim is this estimator-level control and its support-conditioned formulation, not the replay primitive in isolation.

3 Preliminaries

Notations: Unless stated otherwise, $\|\cdot\|$ denotes the Euclidean norm, $[u]_+ = \max\{u, 0\}$ for a scalar u, and L denotes a gradient-Lipschitz constant.

Zeroth-order estimation. Let $\theta \in \mathbb{R}^d$ denote the parameters being optimized, $\xi = (x, y) \sim \mathcal{D}$ a data example drawn from the training distribution, and $\ell(\theta; \xi)$ its loss. For a minibatch ξ_t, we use the same loss notation for the average over its examples. A *finite difference* estimates parameter sensitivity by comparing losses at nearby parameter values. For a random direction z and perturbation scale $\epsilon > 0$, the central and clean-baselined one-sided estimates are

$$\widehat{g}_{\mathrm{cd}} = \frac{\ell(\theta + \epsilon z; \xi) - \ell(\theta - \epsilon z; \xi)}{2\epsilon} z, \tag{1}$$

$$\widehat{g}_{\mathrm{os}} = \frac{\ell(\theta + \epsilon z; \xi) - \ell(\theta; \xi)}{\epsilon} z. \tag{2}$$

MeZO uses central differences. A multi-direction one-sided estimator shares the clean loss $\ell(\theta; \xi)$ across directions, as in FZOO. Let T denote the total number of parameter updates. Updates are indexed by $t = 0, \ldots, T-1$, with minibatch $\xi_t = (x_t, y_t)$ and directions $z_{t,i}$ for $i = 1, \ldots, N$. All queries in one estimate use the same minibatch. ZOMoE uses independent Rademacher directions, whose coordinates are -1 or $+1$ with equal probability. Thus ϵ is a coordinate scale; the Euclidean perturbation norm is $\epsilon\sqrt{d}$.

Expert selection in sparse MoEs. For token j at routed layer r, hidden state $h_{r,j}(\theta)$ produces router scores $W_r h_{r,j}(\theta)$. Here W_r is the router weight matrix at layer r. The router selects k experts with the largest scores. Write $\mathcal{S}_\theta(x)$ for the collection of these indices across layers and tokens, and $\ell(\theta; \xi, \mathcal{S})$ for the loss computed using prescribed indices $\mathcal{S}$. Prescribing indices fixes which experts execute; their outputs and the architecture-native routing weights are still recomputed from θ. The ordinary dynamically routed loss is

$$\ell_{\mathrm{moe}}(\theta; \xi) = \ell(\theta; \xi, \mathcal{S}_\theta(x)). \tag{3}$$

The *expert support* means this collection of selected experts. In ZOMoE, router weights are frozen and θ contains the trainable non-router parameters; changing these parameters can still change router inputs and expert support.

4 Method

We explain the two sources of loss change under a parameter perturbation, then introduce expert replay and its combination with one-sided estimation.

4.1 How Path Inconsistency Affects ZO Estimates

A parameter perturbation changes the model's continuous computations and can also change which experts execute. To separate these effects, let $\mathcal{S}_t(x) = \mathcal{S}_{\theta_t}(x)$ be the experts selected by the unperturbed model at step t. Keeping these selections fixed defines the replayed loss

$$f_{t,\xi}(\theta) = \ell(\theta; \xi, \mathcal{S}_t(x)). \tag{4}$$

This loss agrees with the ordinary MoE loss at θ_t. At a perturbed point, their difference is $\delta_t(z) = \ell_{\text{moe}}(\theta_t + \epsilon z; \xi_t) - f_{t,\xi_t}(\theta_t + \epsilon z)$, the loss change caused by switching experts. For the one-sided estimator in Eq. (2), which underlies ZOMoE, the loss difference in the numerator decomposes as

$$\ell_{\text{moe}}(\theta_t + \epsilon z; \xi_t) - \ell_{\text{moe}}(\theta_t; \xi_t) = \underbrace{f_{t,\xi_t}(\theta_t + \epsilon z) - f_{t,\xi_t}(\theta_t)}_{\text{parameter response}} + \underbrace{\delta_t(z)}_{\text{expert switching}}. \tag{5}$$

The first term measures the response to parameter changes while reusing the expert indices selected at θ_t. The second measures the additional effect of selecting different experts at the perturbed parameters. Path inconsistency mixes these two effects within a single finite-difference estimate.

Multiplying Eq. (5) by z/ϵ gives the one-sided estimate, with switching contribution $\delta_t(z)z/\epsilon$. This division can amplify a loss jump caused by switching experts. For example, Appendix D.1 derives the one-sided and central-difference decompositions and gives a lower bound on the switching term's second moment. Motivated by this, in our proposed method, replay removes this term exactly within each update by reusing the recorded expert indices, while still recomputing expert outputs and routing weights.

4.2 ZOMoE: Record, Replay, and Refresh

ZOMoE combines expert replay with shared-baseline one-sided estimation. Its guiding principle is to keep expert selections consistent within an update while allowing them to adapt across updates. An unperturbed forward pass records the expert indices and the baseline loss; the perturbed queries reuse those indices to estimate the update. The next update begins with a new unperturbed forward pass. Algorithm 1 summarizes this procedure.

Record the expert indices. At the start of an update, ZOMoE evaluates the unperturbed model on ξ_t. This pass returns the baseline loss $\ell_{0,t} = f_{t,\xi_t}(\theta_t)$ and caches the top-k expert indices for each routed token and layer.

Replay indices for every perturbation. Each perturbed forward pass sends tokens to the cached experts. Only the indices are reused: hidden states, router logits, expert outputs, and routing coefficients are recomputed from the perturbed model. Thus all queries evaluate f_{t,ξ_t} with consistent expert selections and retain the continuous response to parameter changes.

Estimate the update. For N independent directions $z_{t,i}$, let $\ell^{\text{rep}}_{t,i} = f_{t,\xi_t}(\theta_t + \epsilon z_{t,i})$ and $\Delta_{t,i} = \ell^{\text{rep}}_{t,i} - \ell_{0,t}$. The same baseline loss $\ell_{0,t}$ is shared by all directions. For the unnormalized one-sided estimator, subtracting a direction-independent baseline leaves its mean unchanged but changes its variance. With Rademacher directions and an L-smooth replayed loss, the unperturbed loss satisfies $|b^\star - \ell_{0,t}| \le L\epsilon^2 d/2$, where $b^\star$ is the variance-minimizing direction-independent scalar baseline (Appendix D.2). The unperturbed forward used to record expert indices also computes $\ell_{0,t}$, so using it as the baseline requires no additional forward query.

For the ZOMoE update, we adopt the loss normalization of FZOO (Dang et al., 2025) and clip the resulting coefficients. Let

$$\sigma_t = \max\{\text{Std}_i(\ell^{\text{rep}}_{t,i}), \tau\}, \qquad a_{t,i} = \text{clip}_c\left(\frac{\Delta_{t,i}}{N\sigma_t}\right), \tag{6}$$

where $\text{clip}_c(x) = \max\{-c, \min\{x, c\}\}$, $\tau > 0$ is the normalization floor, and $c > 0$ is the clipping threshold. The update is then

$$\widehat{g}_t = \sum_{i=1}^{N} a_{t,i} z_{t,i}, \qquad \theta_{t+1} = \theta_t - \eta_t \widehat{g}_t. \tag{7}$$

Replay determines which experts execute, while normalization and clipping determine how the resulting losses are combined into the update.

Algorithm 1 ZOMoE with T parameter updates

Require: Initial parameters θ_0, training distribution $\mathcal{D}$, number of updates T, scale ϵ, directions N, learning rates $\{\eta_t\}_{t=0}^{T-1}$, floor τ, and clip c.

for $t = 0, \dots, T-1$ **do**
  Sample minibatch $\xi_t = (x_t, y_t)$ from $\mathcal{D}$.
  Run the unperturbed forward; record expert indices $\mathcal{S}_t(x_t)$ and loss $\ell_{0,t}$.
  for $i = 1, \dots, N$ **do**
    Sample $z_{t,i}$ and perturb parameters to $\theta_t + \epsilon z_{t,i}$.
    Evaluate $\ell_{t,i}^{\mathrm{rep}}$ with cached indices and recomputed routing weights.
    Restore θ_t.
  end for
  Discard the index cache and compute $a_{t,i}$ using Eq. (6).
  Update $\theta_{t+1} = \theta_t - \eta_t \sum_i a_{t,i} z_{t,i}$.
end for
return θ_T.

Refresh for the next update. After collecting the perturbed losses, ZOMoE discards the cache. At iteration $t+1$, a new unperturbed forward with θ_{t+1} and ξ_{t+1} recomputes the hidden states and expert indices; router parameters remain frozen. Evaluation and inference use the ordinary dynamic router.

An update uses $N+1$ forward queries: one unperturbed and N perturbed. Replay adds no query to the shared-baseline one-sided estimator. The temporary cache stores $O(BT_{\mathrm{seq}}L_{\mathrm{MoE}}k)$ integer indices for batch size B, sequence length T_{seq}, and L_{MoE} routed layers, without saving activations or routing coefficients.

4.3 Convergence Analysis

We now study the convergence behavior of ZOMoE. The main question is whether the replayed, normalized, and clipped zeroth-order update can still make consistent optimization progress when the model uses sparse expert routing. Unlike a standard ZO method, a perturbation in an MoE may change not only the model parameters but also the experts selected for a token. Expert replay removes this discrete change from the perturbed queries used in each update, while allowing the selected experts to evolve across iterations.

This distinction naturally separates the analysis into two parts. Within an iteration, all perturbed queries reuse the same expert indices, so the update can be analyzed through the corresponding smooth loss. Across iterations, the next unperturbed forward pass may select different experts, which introduces an additional change in the loss. Under standard smoothness and minibatch-noise conditions, together with a condition controlling this between-iteration change, we establish a non-convex stationarity guarantee for ZOMoE.

The normalized and clipped update introduces one additional difficulty. Its coefficients are computed from the same queried losses that determine the update direction, so this dependence must be handled directly.

For iteration t, define $F_t(\theta) = \mathbb{E}_{\xi\sim\mathcal{D}}\big[\ell(\theta; \xi, S_{\theta_t}(x))\big]$. Thus, F_t is the expected loss evaluated using the same expert indices reused by the perturbed queries in that iteration. Let $\mathcal{H}_t$ denote the history before the current minibatch and perturbation directions are drawn. Conditional on $\mathcal{H}_t$, θ_t, $S_{\theta_t}(\cdot)$, and F_t are fixed.

Assumption 4.1 (Smoothness)**.** *For any expert indices recorded by ZOMoE, the corresponding replayed loss is L-smooth along the perturbations and parameter updates used by the algorithm. Each F_t is lower bounded by the same constant $F_\star$. All expectations below are finite, and differentiation under the data expectation is valid.*

Assumption 4.2 (Minibatch noise)**.** *At iteration t, a fresh minibatch ξ_t of size B satisfies*

$$\mathbb{E}[\nabla f_{t,\xi_t}(\theta_t) \mid \mathcal{H}_t] = \nabla F_t(\theta_t), \qquad \mathbb{E}\big[\|\nabla f_{t,\xi_t}(\theta_t) - \nabla F_t(\theta_t)\|^2 \mid \mathcal{H}_t\big] \le \frac{\sigma^2}{B}. \tag{8}$$

Conditional on $\mathcal{H}_t$, the N perturbation directions are independent Rademacher vectors and are independent of the fresh minibatch. All queries within the iteration use the same minibatch.

Assumption 4.3 (Expert reselection). *There is a constant $\rho \geq 0$ such that*

$$\mathbb{E}\Big[\big(F_{t+1}(\theta_{t+1}) - F_t(\theta_{t+1})\big)_+\Big] \leq \rho\,\mathbb{E}\|\theta_{t+1} - \theta_t\|^2, \qquad (x)_+ = \max\{x, 0\}. \tag{9}$$

The first two assumptions are standard once the expert indices are fixed within an iteration. Assumption 4.3 captures the additional effect specific to sparse routing: the selected experts may change between consecutive iterations. Appendix D.4 gives sufficient conditions for this assumption in terms of routing margins and the loss change caused by expert switching.

Using more perturbation directions should reduce the variability of the zeroth-order update. Bounding the clipped contributions separately loses this averaging effect. Our analysis instead keeps the directional contributions together. To state the resulting bound compactly, define

$$m_{N,c} = \min\left\{Nc,\ \frac{5}{4}\sqrt{1 + \frac{2}{N}} + Nc\,2^{(1-N)/2}\right\}, \tag{10}$$

$$V_{N,d,c} = \min\left\{c^2 N[d + (N-1)\sqrt{d}],\ d\,m_{N,c}^2,\ \frac{\pi m_{N,c}^2}{2N}\big[(\sqrt{d} + \sqrt{N})^2 + 1\big]\right\}. \tag{11}$$

Here $m_{N,c}$ controls the aggregate size of the normalized and clipped coefficients, and $V_{N,d,c}$ controls the second moment of the update. The latter retains the averaging across perturbation directions while remaining valid when the empirical normalizer is small.

Theorem 4.1 (Convergence of ZOMoE). *Suppose Assumptions 4.1–4.3 hold. Start from a fixed θ_0 with $0 < F_0(\theta_0) - F_\star < \infty$. Run ZOMoE for $T \geq 1$ iterations with $N \geq 2$ perturbation directions per iteration and a constant learning rate $\eta_t \equiv \eta$. Let d be the number of trainable parameters, and let $c > 0$ and $\tau > 0$ be the clipping threshold and normalization floor in Eq. (6). Choose $\eta = \sqrt{\frac{2\,[F_0(\theta_0) - F_\star]}{(L+2\rho)V_{N,d,c}T}}$, $\epsilon = \sqrt{\frac{\min\{1, c\sqrt{N}\}\,\tau}{2Ld[\min\{1, c\sqrt{N}\} + m_{N,c}]}}$. If Assumption 4.1 holds over the queries and updates generated by these choices, then*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_t(\theta_t)\| \leq \frac{\sqrt{(N+2)/N}}{\min\{1, c\sqrt{N}\}}\left[\sqrt{\frac{2(L+2\rho)V_{N,d,c}[F_0(\theta_0) - F_\star]}{T}}\right.$$
$$\left. + \sqrt{\frac{Ld\,\min\{1, c\sqrt{N}\}[\min\{1, c\sqrt{N}\} + m_{N,c}]\tau}{2}} + \frac{m_{N,c}\sigma}{\sqrt{B}}\right]. \tag{12}$$

Theorem 4.1 shows that ZOMoE retains a standard nonconvex stationarity guarantee despite sparse expert routing. Replay makes the perturbed queries within each iteration evaluate the same loss function, while changes in expert selection between iterations are controlled by Assumption 4.3. The bound also captures the benefit of averaging multiple perturbation directions.

For fixed $c > 0$, $m_{N,c}$ remains bounded as N grows and converges to $5/4$, while $V_{N,d,c} = O_c(1 + d/N)$. Thus, for fixed L and ρ, Eq. (12) scales as $O_c\Big(\sqrt{(L+2\rho)(1+d/N)[F_0(\theta_0) - F_\star]/T} + \sqrt{Ld\tau} + \sigma/\sqrt{B}\Big)$, where O_c hides only numerical constants and constants depending on c. For a fixed number of iterations, increasing N therefore improves directional averaging and reduces the leading dimensional dependence from d to d/N, with diminishing returns as N grows. This improvement must be balanced against query cost. Each ZOMoE iteration uses $N+1$ forward queries, so a total query budget Q allows approximately $T = Q/(N+1)$ iterations. The leading optimization term then scales as $O_c\Big(\sqrt{(N+1)(1+d/N)/Q}\Big)$. Its N-dependent part is $N + d/N$, which is balanced at $N \asymp \sqrt{d}$. This scaling describes the leading optimization term; the best finite N can also depend on the clipping threshold c, the normalization floor τ, the minibatch noise level $\sigma/\sqrt{B}$, and the total query budget Q.

Finally, for fixed N, d, and c, the optimization term decays as $O(T^{-1/2})$. If the normalization floor and minibatch noise scale as $\tau = O(T^{-1})$ and $\sigma/\sqrt{B} = O(T^{-1/2})$, the full bound has the same rate. Otherwise, these two terms determine the limiting neighborhood.

Table 1: **ZOMoE achieves the highest average test accuracy among the ZO methods on both backbones.** Values are percentages; Avg. is the six-task mean and bold marks the best ZO result.

Method	SST-2	RTE	BoolQ	COPA	WiC	MultiRC	Avg.
OLMoE-1B-7B							
Zero-shot	66.86	65.70	68.20	75.00	50.63	68.60	65.83
MeZO	85.67	79.06	68.60	75.00	51.72	74.00	72.34
Sparse-MeZO	84.98	71.12	66.60	71.00	54.08	71.00	69.80
SubZero	84.86	77.26	**68.90**	75.00	55.49	71.90	72.23
MuZO-SGD	72.13	75.45	68.70	75.00	55.80	70.50	69.60
FZOO	86.70	76.90	68.20	81.00	56.27	73.30	73.73
ZOMoE	**91.63**	**82.67**	68.80	**86.00**	**58.46**	**74.20**	**76.96**
AdamW	93.69	81.95	77.30	88.00	65.67	81.20	81.30
Qwen3.5-35B-A3B							
Zero-shot	56.54	86.64	71.30	83.00	60.50	85.50	73.91
MeZO	87.16	88.81	86.50	79.00	**72.41**	**89.50**	83.90
Sparse-MeZO	81.65	87.73	85.70	79.00	69.91	88.00	82.00
SubZero	85.55	88.09	86.50	81.00	71.63	**89.50**	83.71
MuZO-SGD	77.18	88.09	81.50	81.00	71.32	84.30	80.56
FZOO	87.84	89.53	87.30	83.00	69.91	87.10	84.11
ZOMoE	**90.83**	**90.61**	**88.50**	**92.00**	70.85	88.70	**86.91**
AdamW	OOM	OOM	OOM	OOM	OOM	OOM	-

5 EXPERIMENTS

5.1 SETUP AND QUESTIONS

We fine-tune OLMoE-1B-7B (Muennighoff et al., 2024) and Qwen3.5-35B-A3B (Team, 2026) on six classification and three question-answering tasks, using at most 1,000 training examples per task. We compare against MeZO (Malladi et al., 2023), Sparse-MeZO, SubZero, MuZO-SGD, and FZOO (Dang et al., 2025), with AdamW as a first-order reference. All runs use seed 42. Checkpoints are selected on validation performance, and evaluation uses ordinary dynamic routing. Full protocols are provided in Appendix A.

Our experiments examine whether ZOMoE improves downstream performance and optimization progress, whether replay contributes beyond router freezing, and whether maintaining consistent expert paths preserves the memory and runtime efficiency of forward-only fine-tuning.

5.2 Q1: DOES ZOMOE IMPROVE FINE-TUNING?

Higher average accuracy with gains across tasks. ZOMoE achieves the highest average test accuracy among the evaluated ZO methods on both backbones (Table 1). It improves over FZOO, the strongest ZO baseline on average, by **3.23 percentage points on OLMoE and 2.80 on Qwen3.5**, with gains on all twelve classification model–task pairs. The gains are not confined to tasks on which existing ZO methods already adapt well. On COPA, where the ZO baselines other than FZOO do not improve over zero-shot accuracy on either backbone, ZOMoE exceeds FZOO by 5.0 and 9.0 points, respectively. ZOMoE also achieves the highest mean QA F1 on both backbones, extending the results to answer generation and entity selection (Appendix B.3).

Sustained progress within the forward-query budget. To examine how these end-to-end gains develop, Figure 2 tracks validation loss against training forward queries on three typical datasets (full trajectories are provided in Appendix C.2). MeZO often loses its initial progress, while FZOO moderates this behavior but can plateau or rebound. ZOMoE maintains a downward loss trend on SST-2 and COPA and reaches a lower minimum than this reference on all twelve model–task pairs (Appendix C.2). Eq. (5) helps interpret this contrast: without replay, the measured loss difference combines the parameter response with changes caused by switching experts. FZOO averages multiple directions, but each query can still follow a different expert path. ZOMoE removes this

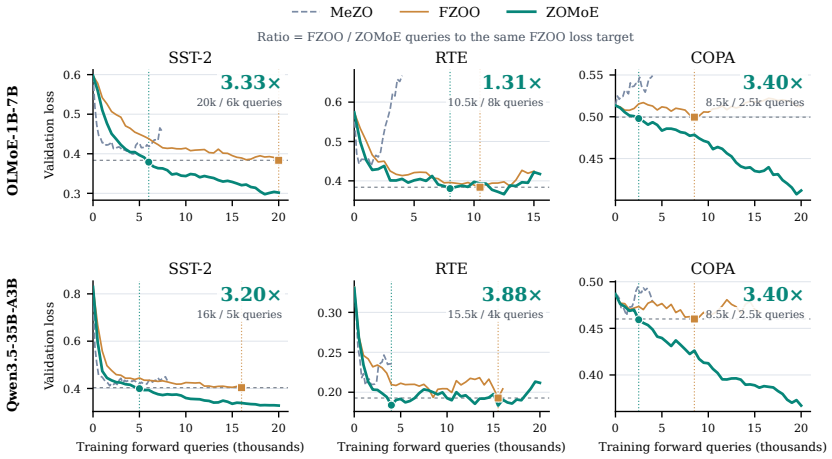

Figure 2: **ZOMoE reaches FZOO's best validation loss with fewer forward queries.** Unsmoothed validation loss evaluated with ordinary dynamic routing; rows show OLMoE and Qwen3.5. Dashed lines indicate its minimum post-update loss; squares (FZOO) and circles (ZOMoE) mark the first evaluations that reach this target. Annotations report query counts and their ratio, $Q_{\mathrm{FZOO}}/Q_{\mathrm{ZOMoE}}$.

switching contribution by reusing the same paths within an update. The resulting loss comparisons are therefore more consistent, offering an explanation for its more sustained descent.

To quantify query efficiency, we use each frozen-router FZOO run's minimum post-update validation loss as a common target and count the queries required to first reach it. ZOMoE requires fewer queries on ten of twelve model–task pairs. Across SST-2, RTE, and COPA, shown in Figure 2, the task-wise ratios average $\mathbf{2.68}\times$ on OLMoE and $\mathbf{3.49}\times$ on Qwen3.5. Together with the lower minima, these results distinguish two benefits: faster progress toward a common loss target on most tasks, and a lower loss reached within the observed budget across all twelve pairs. We next isolate the contribution of replay from that of router freezing.

5.3 Q2: Does Replay Contribute Beyond Router Freezing?

Table 2: **Router freezing and expert replay across tasks.** Test accuracy (%), seed 42. Gold, silver, and bronze mark the three highest distinct scores per column within each backbone; ties share a color. Bold row labels identify the main ZOMoE configurations.

Router	Replay	SST-2	RTE	BoolQ	COPA	WiC	MultiRC	Avg.
OLMoE-1B-7B								
Trainable	No	86.70	76.90	68.20	81.00	56.27	73.30	73.73
Trainable	Yes	91.06	80.51	68.20	82.00	52.66	73.60	74.67
Frozen	No	87.39	81.23	68.60	77.00	57.68	**74.90**	74.47
Frozen	**Yes**	**91.63**	**82.67**	**68.80**	**86.00**	**58.46**	74.20	**76.96**
Qwen3.5-35B-A3B								
Trainable	No	87.84	89.53	87.30	83.00	69.91	87.10	84.11
Trainable	Yes	**91.17**	**90.61**	**88.60**	87.00	71.16	**89.10**	86.27
Frozen	No	86.58	89.89	85.00	84.00	**72.10**	88.50	84.35
Frozen	**Yes**	90.83	**90.61**	88.50	**92.00**	70.85	88.70	**86.91**

Replay improves accuracy with a trainable router. Table 2 separates expert replay from router freezing. Under matched optimizer settings with a trainable router, adding replay improves average accuracy by 0.94 points on OLMoE and 2.16 on Qwen3.5. By comparison, freezing the router without replay yields gains of 0.74 and 0.24 points over the same trainable-router baseline. Thus,

replay can improve accuracy even when the router weights are updated during training; freezing them is not required.

Frozen router weights do not imply fixed expert paths. Replay also complements freezing on OLMoE: with router weights frozen in both runs, it raises average accuracy from 74.47% to 76.96%, a 2.5-point gain under identical optimizer settings. This is consistent with Eq. (5): frozen router weights do not prevent upstream perturbations from changing expert selections, whereas replay removes this switching term from the finite difference. The two controls are therefore not interchangeable: freezing fixes a subset of parameters, whereas replay fixes expert selections within each estimate.

5.4 Q3: What Does Replay Cost?

Path consistency preserves the forward-only memory footprint. Replay caches only selected expert indices, not intermediate activations or routing coefficients. Its storage therefore scales with the number of routed token positions and selected experts, rather than the size of the trainable parameter vector. Consistent with this design, Figure 3 shows nearly identical peak allocated memory for ZOMoE and FZOO. On SST-2, ZOMoE fine-tunes Qwen3.5-35B-A3B on a single H200 with 68.60 GiB of peak allocated memory, whereas our full-parameter AdamW baseline runs out of memory.

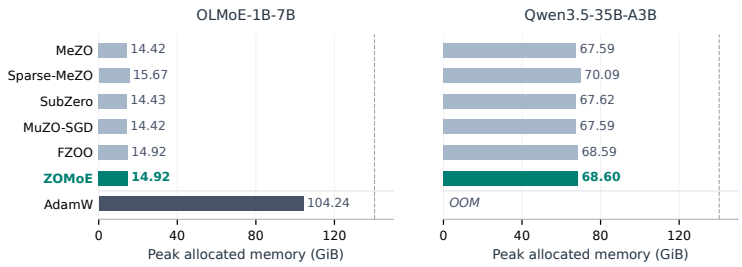

Figure 3: **ZOMoE preserves the memory efficiency of ZO fine-tuning.** Peak allocated GPU memory on SST-2; dashed lines mark device capacity. Qwen3.5 AdamW runs out of memory on one H200. Measurement settings are in Appendix B.1.

No additional queries and little runtime overhead. Collecting the cache also requires no additional forward pass: the unperturbed forward already required by the one-sided estimator supplies both the expert indices and the shared baseline loss. An update with N directions therefore still uses $N + 1$ forward queries. To isolate the cost of handling the cache, we compare FZOO and ZOMoE with frozen routers, identical optimizer settings, and identical token sequences. Throughput differs by at most 1.09% in our sequential implementation (Figure 4, Appendix B.1).

Additional experiments. Additional experiments show that replay adds little overhead and improves average performance under matched settings and on QA tasks, although its gains vary across optimizers (Appendix B). Routing diagnostics and full loss curves show more expert switching at larger perturbations and task-dependent loss reduction (Appendix C). Sensitivity studies support small perturbations and identify effective hyperparameter ranges (Appendices C.3 and C.4).

6 Conclusion

ZOMoE addresses path inconsistency by reusing expert indices from the unperturbed forward within each ZO update while recomputing continuous routing responses, with the same forward providing the one-sided loss baseline. Under smoothness, minibatch sampling, and controlled loss changes from expert reselection, we establish an explicit average-stationarity guarantee. Empirically, ZOMoE achieves the highest average downstream accuracy among evaluated ZO methods on both backbones, with negligible replay overhead and substantially lower memory use than first-order fine-tuning.

AI USE STATEMENT

We used generative AI tools to polish the language and improve the readability of the manuscript, and to assist in writing the experimental code.

REFERENCES

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order fine-tuning for language models with low-rank structures, 2024. URL https://arxiv.org/abs/2410.07698.

Cursor Research Team. Composer 2 technical report. Technical report, 2026. URL https://cursor.com/resources/Composer2.pdf.

Sizhe Dang, Yangyang Guo, Yanjun Zhao, Haishan Ye, Xiaodong Zheng, Guang Dai, and Ivor Tsang. Fzoo: Fast zeroth-order optimizer for fine-tuning large language models towards adam-scale speed, 2025. URL https://arxiv.org/abs/2506.09034.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://arxiv.org/abs/2101.03961.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models, 2024. URL https://arxiv.org/abs/2404.08080.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding, 2020. URL https://arxiv.org/abs/2006.16668.

Wei Lin, Yining Jiang, Qingyu Song, Qiao Xiang, and Hong Xu. AGZO: Activation-guided zeroth-order optimization for LLM fine-tuning, 2026. URL https://arxiv.org/abs/2601.17261. Accepted at ICML 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, 2024. URL https://arxiv.org/abs/2402.15751.

Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Zhifang Sui, and Fuli Luo. Stabilizing MoE reinforcement learning by aligning training and inference routers, 2025. URL https://arxiv.org/abs/2510.11370.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes, 2023. URL https://arxiv.org/abs/2305.17333.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoE: Open mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2409.02060.

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

Yuezhang Peng, Yuxin Liu, Fei Wen, and Xie Chen. MUZO: Leveraging multiple queries and momentum for zeroth-order fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8566–8584. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.432. URL https://aclanthology.org/2025.emnlp-main.432/.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *Proceedings of the 39th International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/rajbhandari22a.html.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2011. URL https://asgordon.github.io/publications/AAAI-SPRING11A.PDF.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Ramon van Handel. Probability in high dimension, 2016. URL https://web.math.princeton.edu/~rvan/APC550.pdf. APC 550 Lecture Notes, Princeton University, version of December 21, 2016.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018. doi: 10.1017/9781108231596.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of LLMs in random subspaces, 2024. URL https://arxiv.org/abs/2410.08989.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark, 2024. URL https://arxiv.org/abs/2402.11592.

Jiajun Zhou, Yifan Yang, Kai Zhen, Ziyue Liu, Yequan Zhao, Ershad Banijamali, Athanasios Mouchtaris, Ngai Wong, and Zheng Zhang. QuZO: Quantized zeroth-order fine-tuning for large language models, 2025. URL https://arxiv.org/abs/2502.12346.

A Experimental Setup

A.1 Hardware and software

Our main experiments use $1\times$ NVIDIA H200 GPU (141 GB) per run. The implementation uses Python, PyTorch for model execution and optimization, Hugging Face Transformers for pretrained models and tokenizers, Hugging Face Datasets for data loading, and NumPy for data sampling and numerical operations. ZO forward passes use bfloat16, and perturbation queries execute sequentially on one device. Our implementation follows the FZOO estimator without its parallel CUDA execution engine. The pretrained checkpoints are OLMoE-1B-7B-0125 and Qwen3.5-35B-A3B. The routing diagnostic records PyTorch 2.13.0 with CUDA 12.9 and Transformers 5.13.1.

A.2 Datasets and evaluation

Following the large-model evaluation setup of MeZO and FZOO (Malladi et al., 2023; Dang et al., 2025), we use a limited labeled training set and separate development and test subsets. We randomly sample 1,000 training examples and a disjoint 500-example development set from the original training split. We sample up to 1,000 test examples from the original validation split. This gives 872 test examples for SST-2, 277 for RTE, 638 for WiC, and 1,000 each for BoolQ, MultiRC and QA tasks. COPA has only 400 original training examples, so we use 300/100/100 examples for train/validation/test. All methods use the same task-specific subsets and seed 42.

We evaluate every 100 updates, restore the checkpoint with the highest validation accuracy (classification) or F1 (QA), and report its held-out test score. Each task runs allow at most 4,000 updates and stop after 15 evaluations without improvement following a 500-update warmup; MuZO-SGD uses 1,000 updates. Zero-shot evaluation uses the same test examples. Model selection and all reported evaluations use ordinary dynamic routing.

Classification training uses cross-entropy over candidate answers, with continuation scores computed from answer-token likelihoods. For the QA tasks, training uses gold-answer language-model loss, while evaluation follows the task-specific answer format. On SQuAD and DROP, we generate answers using greedy decoding with at most 50 new tokens, stopping at a newline or an end-of-sequence token. The first answer line is evaluated using token-overlap F1 against the reference answers. On ReCoRD, we select an answer from the supplied candidate entities rather than generating it freely, and evaluate the selected entity using token-overlap F1 against the reference answers.

A.3 HYPERPARAMETERS AND OPTIMIZER CONFIGURATIONS

We use grid search over the hyperparameter values in Table 3. The table covers both classification and question answering. Checkpoint selection and test evaluation follow Appendix A.2.

Table 3: **Hyperparameter search grids and fixed settings for ZO methods.** Braces denote candidate values; single values are fixed. Central ZO denotes MeZO, Sparse-MeZO, and SubZero.

Hyperparameter	Central ZO	FZOO	ZOMoE
Learning rate	$\{5\times10^{-7}, 10^{-6}, 5\times10^{-5}\}$	$\{5\times10^{-5}, 6\times10^{-5}\}$	
Perturbation scale ϵ	10^{-3}	$\{10^{-4}, 3\times10^{-4}\}$	
Directions N	1	4	4
Coefficient clip c	None	$\{0.7, 1\}$	
Training batch size	16		
Learning-rate schedule	Constant		
Weight decay	0		
Maximum updates	4,000		
Evaluation interval	100 updates		
Maximum input length	512 (classification); 768 (QA)		
Maximum generated tokens	50 (SQuAD and DROP)		

Central-difference methods. MeZO uses Gaussian directions. Sparse-MeZO perturbs 12.5% of expert parameters and 25% of shared parameters. SubZero uses rank-8 directions for matrix parameters and resamples its subspace every 500 updates. MuZO-SGD averages four central pairs per update with $\epsilon = 10^{-3}$ and learning rates 10^{-6} on OLMoE and 5×10^{-7} on Qwen3.5, using a 1,000-update budget. These methods do not clip their finite-difference coefficients. MeZO is the central-difference baseline for QA.

FZOO and ZOMoE. Both methods use Rademacher directions and population-standard-deviation normalization. The normalization floor is 10^{-8} for classification and ReCoRD, and 0.03125 for SQuAD and DROP. The main classification FZOO baseline trains the router; the frozen-router control changes only this setting.

AdamW. The first-order baseline uses learning rate 2×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, optimizer epsilon 10^{-8}, zero weight decay, and gradient-norm clipping at 1.0. It uses linear learning-rate decay over 4,000 updates with 120 warmup updates, gradient checkpointing, and microbatches of one example accumulated to an effective batch size of 16. Parameters and optimizer states are maintained in FP32, with bfloat16 forward computation. The evaluation and checkpoint-selection protocol matches the ZO runs.

A.4 PROMPT TEMPLATES AND ANSWER SPACES

Table 4 specifies the task templates. Italic angle-bracketed fields are replaced by example content; ↩ marks an actual newline, while other line wrapping is for display. Each candidate continuation starts with one space. BoolQ questions are capitalized and receive a final question mark if absent. SQuAD includes the title line only when a title is available. ReCoRD removes `@placeholder` from the query and scores the example's entity candidates.

For Qwen3.5 SQuAD and DROP, we place the template in one user message. The tokenizer's chat template uses `add_generation_prompt=True` and `enable_thinking=False`. OLMoE uses the plain template. These formatting choices are shared across optimizers within each model and task.

Table 4: **Task prompts and answer spaces.** Literal prompt text is monospaced; input fields are italicized. ↩ denotes a newline.

Task	Input template	Answer space
Classification and candidate selection		
SST-2	⟨*sentence*⟩ `It was`	`terrible` `great`
RTE	⟨*premise*⟩ ↩ `Does this mean that "`⟨*hypothesis*⟩`" is true? Yes or No?`	`Yes`, `No`
BoolQ	⟨*passage*⟩ ⟨*question*⟩`?`	`No`, `Yes`
COPA	⟨*premise*⟩ `What was the` ⟨*cause/effect*⟩`?`	⟨*choice 1*⟩ ⟨*choice 2*⟩
WiC	`Does the word "`⟨*word*⟩`" have the same meaning in the following two sentences?` ↩ ⟨*sentence 1*⟩ ↩ ⟨*sentence 2*⟩ ↩ `Yes or No?` ↩ `Answer:`	`No`, `Yes`
MultiRC	⟨*paragraph*⟩ ↩ `Question:` ⟨*question*⟩ ↩ `I found this answer "`⟨*answer*⟩`". Is that correct? Yes or No?` ↩ `Answer:`	`No`, `Yes`
ReCoRD	⟨*passage*⟩ ↩ ⟨*query without placeholder*⟩	Candidate entities
Answer generation		
SQuAD	`Title:` ⟨*title*⟩ ↩ `Context:` ⟨*context*⟩ ↩ `Question:` ⟨*question*⟩ ↩ `Answer:`	Gold answer text
DROP	`Passage:` ⟨*passage*⟩ ↩ `Question:` ⟨*question*⟩ ↩ `Answer:`	Gold answer text

B Additional Experimental Results

This section supplements Sections 5.2–5.4 with absolute throughput.

B.1 Memory and absolute throughput

For Figures 3 and 4, each frozen-router FZOO–ZOMoE pair processes identical active-token sequences with $(\eta, \epsilon, N, c) = (5 \times 10^{-5}, 10^{-4}, 4, 1)$. Replay is the only difference. Both methods use the common sequential implementation described in Appendix A.1; the comparison measures replay overhead and does not benchmark FZOO's parallel CUDA engine. We measure throughput over updates 100–999, excluding warmup and evaluation, with five sequential forward queries per update. Figure 4 shows the relative throughput change summarized in Section 5.4, and Table 5 gives the absolute rates underlying it. Rates count active tokens once per training forward query: $\sum_t q_t T_t / \sum_t s_t$, where $q_t = 5$, T_t counts active tokens and s_t is update time. Unique-input-token throughput is one fifth of the displayed rate; relative changes are identical.

The memory benchmark uses SST-2, batch size 16, evaluation batch size 4, and maximum input length 512. FZOO and ZOMoE use the matched runs; other ZO methods use a 100-update benchmark. AdamW's OLMoE peak is measured during its full SST-2 run. We report peak allocated memory, and assign no numerical value to an out-of-memory run.

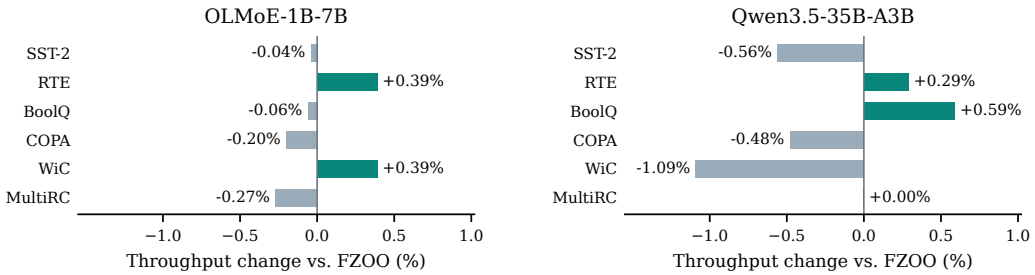

Figure 4: **Replay preserves training throughput.** Relative throughput change from frozen-router FZOO to ZOMoE on H200, with identical optimizer settings and token sequences. Absolute rates are in Table 5.

Table 5: **Absolute training throughput across six tasks.** Forward-query tokens/s for the matched FZOO–ZOMoE runs used in Figure 4; Δ is the relative change from FZOO.

Model	Method	SST-2	RTE	BoolQ	COPA	WiC	MultiRC
OLMoE	FZOO	787.8	3,973.1	5,666.4	1,227.7	2,571.4	11,674.2
	ZOMoE	787.5	3,988.4	5,663.1	1,225.3	2,581.5	11,642.8
	Δ	-0.04%	$+0.39\%$	-0.06%	-0.20%	$+0.39\%$	-0.27%
Qwen3.5	FZOO	481.2	2,571.6	4,046.5	630.9	1,483.1	8,994.0
	ZOMoE	478.5	2,578.9	4,070.2	627.9	1,467.0	8,994.4
	Δ	-0.56%	$+0.29\%$	$+0.59\%$	-0.48%	-1.09%	$+0.00\%$

B.2 Central-difference replay across tasks

To investigate whether expert replay extends to central-difference ZO methods and assess the respective contributions of router freezing and replay, we compare three MeZO variants across six tasks on both backbones: standard MeZO with a trainable router, MeZO with a frozen router, and frozen-router MeZO with expert replay. The last variant reuses the clean expert indices for both positive and negative perturbation queries (Table 6).

Table 6: **Task-level effects of adding replay to MeZO.** Test accuracy (%); Avg. is the six-task mean and Δ compares Replay with frozen-router MeZO.

Variant	SST-2	RTE	BoolQ	COPA	WiC	MultiRC	Avg.	Δ
OLMoE-1B-7B								
Standard MeZO	85.67	79.06	68.60	75.00	51.72	74.00	72.34	–
Frozen, no replay	86.01	82.31	68.20	79.00	54.55	73.80	73.98	–
Replay	87.96	81.23	69.50	77.00	58.78	71.80	**74.38**	+0.40
Qwen3.5-35B-A3B								
Standard MeZO	87.16	88.81	86.50	79.00	72.41	89.50	83.90	–
Frozen, no replay	87.96	87.36	85.80	80.00	70.38	88.00	83.25	–
Replay	89.56	89.53	86.70	82.00	71.94	87.70	**84.57**	+1.32

Router freezing alone improves mean accuracy on the smaller OLMoE model but slightly reduces it on the larger Qwen3.5 model. The replay configurations achieve the highest mean accuracy on both backbones, exceeding frozen-router MeZO by 0.40 and 1.32 percentage points, respectively. On Qwen3.5, replay also recovers the loss from freezing and surpasses standard MeZO.

These results support extending expert replay beyond one-sided estimation to central differences. Freezing router parameters does not fix expert selection, since perturbations to upstream parameters can still change the router inputs. Replay directly keeps the selected experts consistent across the two perturbed forwards. Thus, replay addresses perturbation-induced expert switching that router freezing alone does not prevent.

B.3 Question answering

Table 7: **ZOMoE achieves the highest average QA F1 on both backbones.** We report test F1 (%) using seed 42. Zero-shot references use the same prompts, test subsets, and scoring protocol. SQuAD and DROP use free-form generation, while ReCoRD uses entity selection. Avg. denotes the mean across the three tasks, and bold indicates the best result.

Model	Method	SQuAD	DROP	ReCoRD	Avg.
OLMoE	Zero-shot	62.16	20.99	24.69	35.95
	MeZO	80.22	50.99	22.55	51.25
	FZOO	78.69	53.52	**27.03**	53.08
	ZOMoE	**83.22**	**57.40**	26.25	**55.62**
Qwen3.5	Zero-shot	23.18	3.41	25.67	17.42
	MeZO	84.73	69.78	32.84	62.45
	FZOO	86.79	75.82	34.27	65.63
	ZOMoE	**87.89**	**77.10**	**52.89**	**72.63**

We evaluate whether ZOMoE's benefits extend from classification to question answering on SQuAD, DROP, and ReCoRD. SQuAD and DROP require free-form answers, whereas ReCoRD requires selecting an entity from the supplied candidates. We fine-tune with gold-answer supervision and compare held-out test F1 against zero-shot, MeZO, and FZOO under the same task-specific evaluation protocol (Table 7).

Qwen3.5's low zero-shot scores on SQuAD and DROP reflect the difficulty of producing concise answers under this protocol. Its non-thinking chat responses are limited to 50 tokens and scored only up to the first newline: explanations can dilute token-overlap F1 or precede an answer that is cut off. The sharp improvement after fine-tuning is consistent with learning both the task and the required answer format. With candidate-based evaluation on ReCoRD, Qwen3.5's zero-shot F1 is instead comparable to OLMoE's (25.67 versus 24.69).

ZOMoE achieves the highest average F1 on both backbones, outperforming FZOO by 2.54 points on OLMoE and 7.00 points on Qwen3.5. These gains are consistent with the motivation for expert

replay: keeping expert selection fixed during perturbation queries reduces routing-induced variation in the loss differences used for ZO updates. The improvements on both generated answers and entity selection show that the benefit extends across answer formats.

C Optimization Diagnostics

C.1 Preliminary loss and routing measurements

The COPA curves in Figure 1a,b use the same evaluations and query-count convention as Appendix C.2. MeZO retains its backbone-specific baseline configuration; FZOO freezes the router and uses $(\epsilon, N, c) = (10^{-4}, 4, 1)$. They compare complete optimization recipes and do not isolate switching alone.

Multi-minibatch routing sweep. Figure 1c compares full-model Gaussian perturbations of the pretrained OLMoE and Qwen3.5 checkpoints. Both models use the same 32 disjoint SST-2 training minibatches of 8 examples (256 distinct examples, sampling seed 42). Four independent directions use seeds 101, 202, 303, and 404. Each direction is held fixed across 50 logarithmically spaced perturbation scales from 10^{-4} to 10^{-2}. Inputs consist of the task prompt and gold verbalizer, truncated at 256 tokens using the native tokenizer of each model. We compare unordered top-k expert sets at non-padding positions, pooling changed-token and valid-token counts across batches and directions separately for each layer and scale. Native tokenization yields 3,821 valid tokens for OLMoE and 3,700 for Qwen3.5. Each backbone contributes 6,400 perturbed forwards: 102,400 layer observations for OLMoE and 256,000 for Qwen3.5. Both sweeps perturb the full pretrained text model, including router parameters. The four directions are perturbation replicates, not independent fine-tuning seeds.

Forward computation uses bfloat16. Each perturbed tensor is constructed in float32 from the exact original checkpoint values and the same Gaussian direction, avoiding cumulative add/subtract drift. OLMoE retains the original tensors and direction on GPU; Qwen3.5 stores original values on CPU and regenerates the identical direction at each scale, using one parameter tensor at a time. This storage change leaves the perturbation definition unchanged. Repeated clean forwards produce identical expert selections, all Qwen router logits and final hidden states are finite, and parameter restoration is checked exactly. The layer-averaged mismatch at $\epsilon = 10^{-2}$ is 99.9951% on OLMoE and 99.9995% on Qwen3.5. Figures 5 and 6 show the respective layer-level measurements.

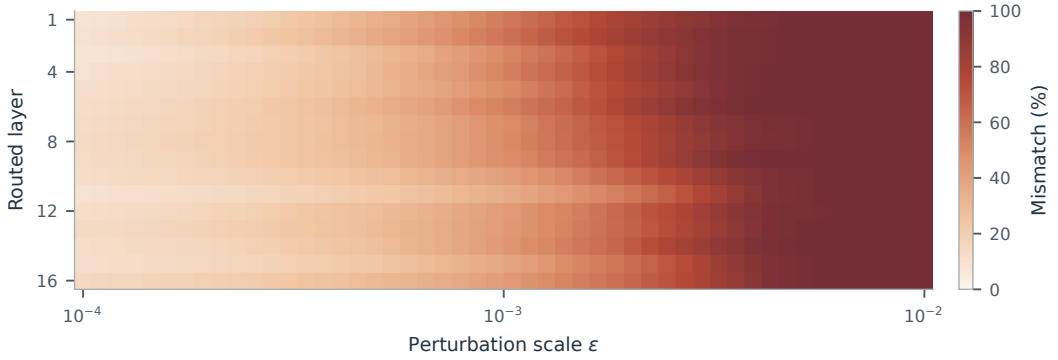

Figure 5: **Expert mismatch grows across the MoE stack as perturbations increase.** Color shows the percentage of tokens whose selected expert set differs from the clean forward; rows denote layers.

The layer heatmaps supplement the averages in Figure 1c. At the smallest scale, $\epsilon = 10^{-4}$, mean layer mismatch is 11.72% on OLMoE and 28.91% on Qwen3.5; at 10^{-2}, almost all valid-token expert sets differ. Rates vary across layers, so the mean summarizes distinct local routing responses. This is a fixed-checkpoint perturbation diagnostic: it does not directly measure gradient error or the switching rate of the frozen-router fine-tuning runs.

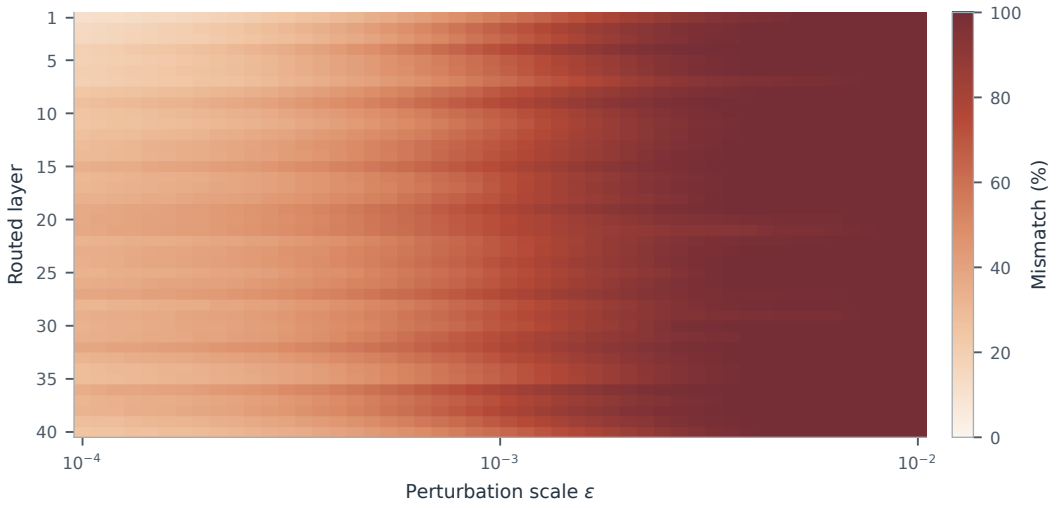

Figure 6: **Perturbation-induced expert switching across Qwen3.5 layers.** Each cell pools valid-token expert-set mismatch across the same 32 minibatches and four directions as Figure 1c.

C.2 Additional optimization trajectories and measurement details

Figure 7 extends the SST-2, RTE, and COPA comparison in Figure 2 to all six tasks, with the same query-to-target annotations. Figures 8 and 9 below add all five ZO baselines using the exact runs underlying Table 1. Their FZOO curves therefore use the trainable-router main baseline. All validation evaluations use unperturbed parameters and ordinary dynamic routing, with no smoothing; each curve stops at its recorded endpoint. Each task uses a fixed validation subset throughout training. The horizontal axis counts training forward queries and excludes validation evaluation: two per MeZO update and five per FZOO/ZOMoE update.

Queries to a common loss target. For each panel, we take the minimum post-update FZOO validation loss as the target and find each method's first recorded evaluation at or below it. The annotated ratio is therefore a training-query reduction factor, rather than a wall-clock speedup. Evaluation every 100 updates resolves the FZOO and ZOMoE crossing counts in increments of 500 queries. The target is defined retrospectively from the baseline trajectory and is used only to summarize optimization progress. For each backbone, the main-text mean is the unweighted arithmetic mean of the three task-specific ratios for SST-2, RTE, and COPA, yielding 2.6819 for OLMoE and 3.4917 for Qwen3.5, rounded to $2.68\times$ and $3.49\times$. Including all six tasks gives 1.7733 and 3.6436, respectively ($1.77\times$ and $3.64\times$). The full six-task calculation includes ratios below one. Both statistics average task-wise query-to-target speedups; neither is a ratio of total query counts or a wall-clock measurement.

ZOMoE reaches this target with fewer queries in 10 of 12 pairs. The strongest reduction is Qwen3.5 WiC: 1,000 versus 7,500 queries ($7.50\times$), followed by Qwen3.5 RTE at 4,000 versus 15,500 ($3.88\times$). WiC reaches the baseline's best level during the initial descent, although its later curve still fluctuates. Its large crossing ratio also coexists with lower test accuracy than MeZO in Table 1. Thus reaching a validation-loss threshold quickly does not imply a proportional improvement in held-out accuracy.

The two exceptions are OLMoE BoolQ and WiC. ZOMoE first reaches the FZOO target at 2,500 versus 2,000 queries on BoolQ ($0.80\times$), and 5,500 versus 4,000 on WiC ($0.73\times$). Their initial descent is slower, but both eventually attain a lower minimum loss. Across all 12 pairs, ZOMoE's minimum is below FZOO's. The curves therefore distinguish faster early progress from a better eventual loss, rather than suggesting uniform acceleration at every stage.

Late-training behavior. To summarize rebound without extrapolating a stopped run, let H be the smallest final update among the three methods in a panel. We compute $\ell_{\mathrm{dev}}(H) - \min_{0<t\le H} \ell_{\mathrm{dev}}(t)$

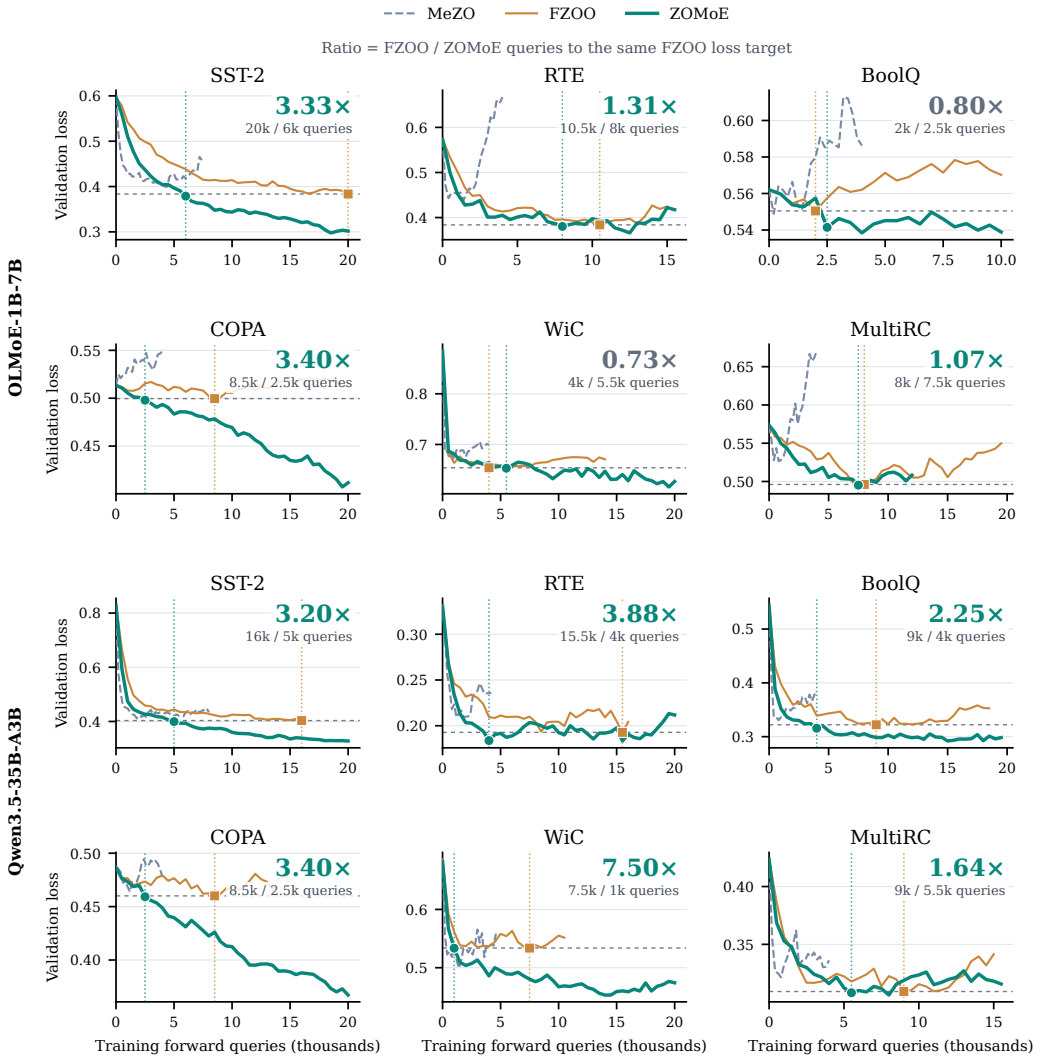

Figure 7: **Full query-efficiency comparison across all six tasks.** Unsmoothed validation loss under ordinary dynamic routing. The FZOO reference freezes the router. Dashed horizontal lines show its minimum post-update loss; squares and circles mark the first FZOO and ZOMoE evaluations reaching that target. Each panel reports query counts and $Q_{\mathrm{FZOO}}/Q_{\mathrm{ZOMoE}}$. The six-task arithmetic means are $1.77\times$ on OLMoE and $3.64\times$ on Qwen3.5.

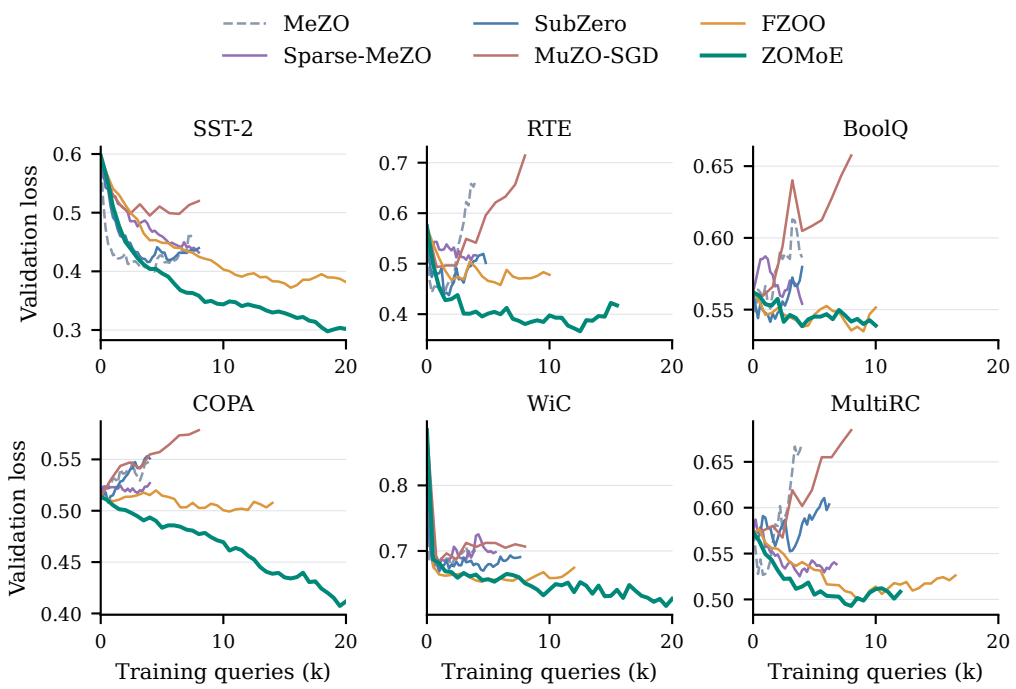

Figure 8: **All five ZO baselines on OLMoE.** Full unsmoothed validation-loss trajectories from the exact runs underlying Table 1. Here FZOO uses a trainable router, matching the main table. Training queries per update are two for MeZO/Sparse-MeZO/SubZero, eight for MuZO-SGD, and five for FZOO/ZOMoE. Curves stop at each recorded endpoint.

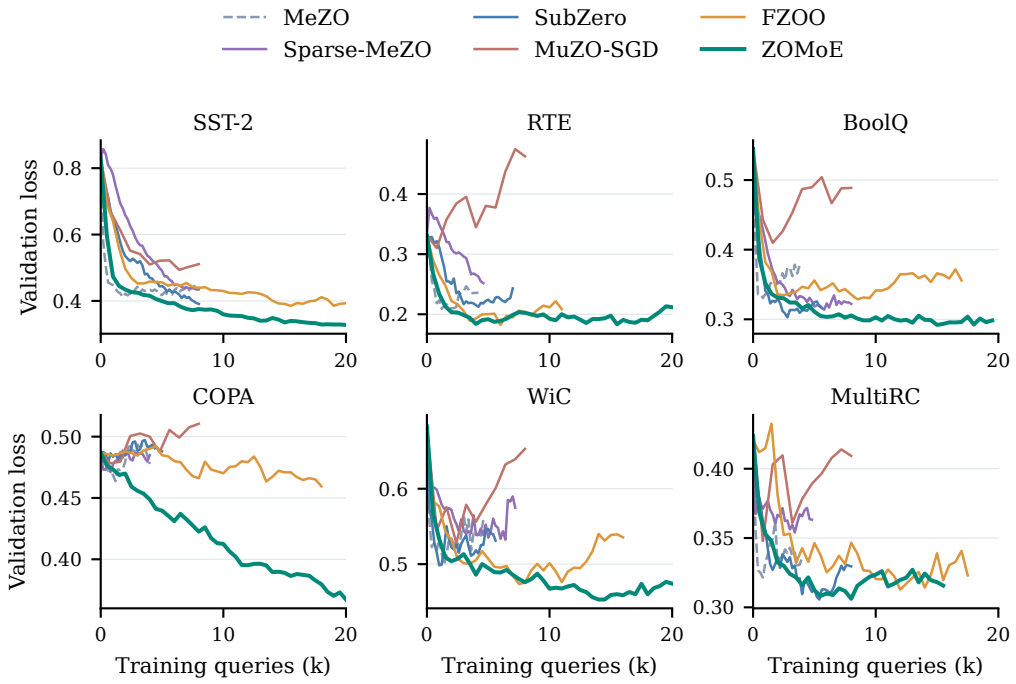

Figure 9: **All five ZO baselines on Qwen3.5.** Same protocol as Figure 8; each curve matches its test-score run in Table 1.

from recorded evaluations. This statistic uses a common *update* horizon; the plotted horizontal axis uses training *queries*. ZOMoE has less rebound than MeZO in 11 of 12 pairs. For example, OLMoE RTE drops from 0.2294 to 0.0172 and MultiRC from 0.1437 to 0.0184. Qwen3.5 MultiRC is the exception (0.0174 for ZOMoE versus 0.0147 for MeZO), and ZOMoE does not uniformly reduce rebound relative to FZOO. The near-overlapping, non-monotone MultiRC curves indicate a smaller and less consistent late-stage benefit than the clearer separations on SST-2 and COPA.

Clean training-loss diagnostic. Training dynamics can be measured using training or validation loss; the main text uses validation loss, which is identical to development loss in our data protocol. The logged `train_loss` field is not a common unperturbed loss: central-difference methods average positive and negative perturbed losses, whereas FZOO/ZOMoE average the one-sided perturbed losses. For an additional training-set diagnostic, Figure 10 instead uses the clean baseline loss recorded before each FZOO/ZOMoE update (the `loss_minus` field in this one-sided implementation). This measures the ordinary, unperturbed model on the current training minibatch. Both arms freeze routers and share $\epsilon = 10^{-4}$, $N = 4$, $\eta = 5 \times 10^{-5}$, and clipping 1.0.

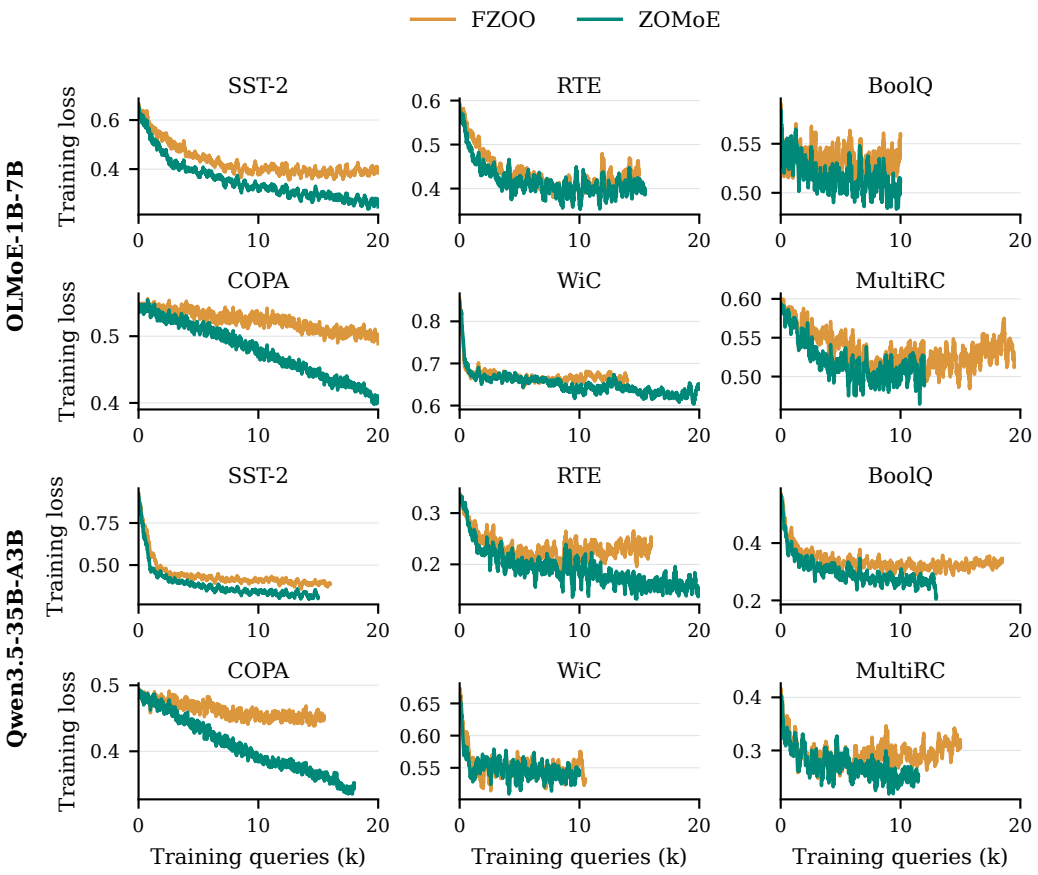

Figure 10: **Clean training-loss trajectories under matched settings.** Ordinary dynamic routing at the unperturbed parameters, measured before each update on its training minibatch. Lines use a centered 51-update moving average (shorter at the endpoints) and stop at each run's recorded endpoint. Both methods use five training queries per update.

C.3 Effect of Perturbation Strength on MoE Fine-Tuning

Measurement protocol. All points use SST-2 and seed 42. FZOO and ZOMoE share a frozen router, $N = 4$, learning rate 5×10^{-5}, and clipping threshold 1.0, differing only in index replay. MeZO uses one Gaussian central-difference pair per update, a trainable router, no replay or coeffi-

cient clipping, and learning rates 10^{-6} on OLMoE and 5×10^{-7} on Qwen3.5; its ϵ varies with the plotted value.

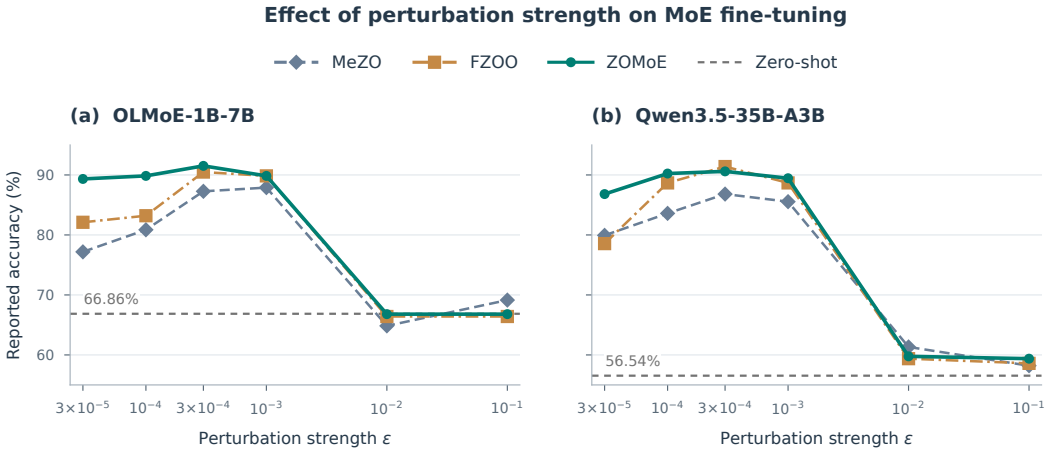

Figure 11: **Small perturbations support effective ZO fine-tuning.** Gray dashed lines mark the zero-shot SST-2 test accuracies from Table 1: 66.86% for OLMoE and 56.54% for Qwen3.5. At $\epsilon = 10^{-2}$ and 10^{-1}, reported accuracy remains near these levels.

Large perturbations yield little useful adaptation. At $\epsilon = 10^{-2}$ and 10^{-1}, the three methods reach approximately 65–69% on OLMoE and 58–61% on Qwen3.5, close to the zero-shot SST-2 test accuracies of 66.86% and 56.54% in Table 1, respectively. The near-zero-shot performance suggests little effective task-specific learning at these large perturbation scales. Within the same coarse sweep, all three methods exceed 80% at $\epsilon = 10^{-4}$ and 10^{-3}, showing that the choice of perturbation scale is critical to learning.

This pattern is consistent with the local nature of finite-difference estimation: ZO methods use small perturbations to approximate local loss sensitivity (Malladi et al., 2023). Excessively large perturbations can make the loss differences less informative about a useful local update. Expert replay removes within-estimate changes in expert selection, but does not remove finite-perturbation error in the continuous computation. The fact that ZOMoE also remains near zero-shot performance at large ϵ is consistent with this limitation.

Replay benefits within the small-perturbation regime. At 3×10^{-5}, ZOMoE improves over matched frozen-router FZOO by 7.22 and 8.26 percentage points on OLMoE and Qwen3.5 (Figure 11). At 3×10^{-4}, the differences are $+1.03$ and -0.80 points. ZOMoE's own test accuracy increases from 89.33% to 91.51% on OLMoE and from 86.81% to 90.60% on Qwen3.5 between these two scales. Thus replay provides clear gains at the smaller tested scale, while its relative benefit varies within the useful small-perturbation range. The results support choosing a small ϵ that preserves informative local loss comparisons, with the precise value depending on the backbone and optimizer configuration.

C.4 One-factor sensitivity on OLMoE SST-2

We vary one factor at a time around the core recipe $(\eta, \epsilon, N, c) = (5 \times 10^{-5}, 10^{-4}, 4, 1.0)$ while fixing the data split, seed, batch size, and 4,000-step budget. Checkpoints are selected on validation accuracy and evaluated once on the 872-example test split. The three five-point axes reported below share one baseline and therefore comprise 13 unique configurations.

C.4.1 LEARNING RATE

Table 8: **Accuracy plateaus once the learning rate is sufficiently large.** OLMoE SST-2 test accuracy (%).

Learning rate	Test accuracy	Change from base
5×10^{-7}	66.97	-24.66
10^{-6}	66.97	-24.66
10^{-5}	74.54	-17.09
5×10^{-5} (base)	91.63	0.00
10^{-4}	**91.74**	$+0.11$

The learning-rate response shows a clear optimization threshold. Values inherited from the central-difference baselines, 5×10^{-7} and 10^{-6}, remain near the pretrained accuracy after 4,000 steps, and 10^{-5} still under-trains. Performance saturates between 5×10^{-5} and 10^{-4}, where the 0.11-point difference is negligible at the resolution of this single run. Thus the main recipe is not selected at an isolated optimum; it lies at the start of the effective learning-rate plateau.

C.4.2 NUMBER OF DIRECTIONS

Table 9: **Averaging beyond four directions yields little additional accuracy.** N is the number of perturbed forward queries per update; accuracy is in percent.

N	Test accuracy	Change from base
1	88.76	-2.87
2	88.53	-3.10
4 (base)	91.63	0.00
8	**91.74**	$+0.11$
16	90.94	-0.69

Using fewer than four directions reduces accuracy in these runs. At $N = 1$, the empirical loss standard deviation is zero, so the normalizer uses its floor; this comparison changes normalization behavior as well as averaging. Beyond four directions, however, the return saturates: doubling to eight changes accuracy by only 0.11 points, and sixteen is slightly worse despite four times the perturbation-query cost of the base setting. This compute–accuracy trade-off motivates $N = 4$ rather than the largest direction count.

C.4.3 COEFFICIENT CLIPPING

Table 10: **Aggressive clipping reduces accuracy in this comparison.** OLMoE SST-2 test accuracy (%); None denotes no coefficient clipping.

Clip c	Test accuracy	Change from base
None	**92.78**	$+1.15$
0.3	89.22	-2.41
0.7	92.66	$+1.03$
1.0 (base)	91.63	0.00
2.0	90.94	-0.69

Coefficient clipping has a non-monotone effect in this one-factor study. Aggressive clipping at $c = 0.3$ reduces accuracy to 89.22%, whereas $c = 0.7$ and no clipping reach 92.66% and 92.78%. These scores do not validate the theoretical alignment bound: that bound concerns local optimization under additional assumptions, not test accuracy. The convergence theorem assumes a finite clipping

threshold and therefore does not cover the no-clipping limit. We retain the original model-level choices rather than selecting a threshold using these test results.

Overall, the controlled study identifies a stable operating region while exposing meaningful failure regimes. We retain the model-level recipes used in Table 1 rather than selecting separate test-optimal values from these diagnostics.

D THEORETICAL ANALYSIS

The theoretical appendix follows the proof path behind the main result. We first make precise what expert replay changes in a finite-difference query: it removes the loss variation caused by perturbation-induced expert switching. We then justify the unperturbed loss that ZOMoE already obtains when recording expert indices as the shared scalar baseline of its one-sided estimator. With these two ingredients in place, Appendix D.3 analyzes the normalized and clipped update, derives a one-step decrease, and sums that decrease across iterations. Appendix D.4 finally gives concrete conditions under which selecting new experts between consecutive updates satisfies Assumption 4.3.

D.1 WHAT EXPERT REPLAY REMOVES FROM A ZO QUERY

The convergence argument starts from the role of replay itself. Without replay, changing the parameters from θ to $\theta+\epsilon z$ can change both the continuous network computation and the selected experts. The following decomposition separates these two effects.

Fix (θ, ξ) and suppress the input x in the expert-index notation. Using the expert indices selected at θ, define

$$f_0 = \ell(\theta; \xi, \mathcal{S}_\theta), \qquad f_\pm(z) = \ell(\theta \pm \epsilon z; \xi, \mathcal{S}_\theta). \tag{13}$$

At the perturbed points, ordinary dynamic routing may select different experts. Let

$$\begin{aligned} \delta_+(z) &= \ell(\theta + \epsilon z; \xi, \mathcal{S}_{\theta+\epsilon z}) - f_+(z), \\ \delta_-(z) &= \ell(\theta - \epsilon z; \xi, \mathcal{S}_{\theta-\epsilon z}) - f_-(z). \end{aligned} \tag{14}$$

These quantities measure only the additional loss change caused by using the newly selected experts at the same perturbed parameter. Substitution into the one-sided and central finite differences gives

$$\widehat{g}^{\mathrm{dyn}}_{\mathrm{os}}(z) = \frac{f_+(z) + \delta_+(z) - f_0}{\epsilon} z = \frac{f_+(z) - f_0}{\epsilon} z + \frac{\delta_+(z)}{\epsilon} z, \tag{15}$$

$$\begin{aligned} \widehat{g}^{\mathrm{dyn}}_{\mathrm{cd}}(z) &= \frac{f_+(z) + \delta_+(z) - f_-(z) - \delta_-(z)}{2\epsilon} z \\ &= \frac{f_+(z) - f_-(z)}{2\epsilon} z + \frac{\delta_+(z) - \delta_-(z)}{2\epsilon} z. \end{aligned} \tag{16}$$

Thus replay has a precise effect: every perturbed query uses the same recorded expert indices, so the terms involving δ_+ and δ_- disappear exactly. Replay does not otherwise alter the finite-difference rule. In particular, ZOMoE remains a one-sided method; replay simply prevents expert switching from entering that one-sided loss comparison.

The removed term can be large when switching remains likely as the perturbation radius shrinks. The next lemma records this observation because it explains why eliminating the term is useful before any convergence argument begins.

Lemma D.1 (Second moment of the switching term)**.** *For $z \in \{-1, +1\}^d$, let the switching event E_ϵ have probability π_ϵ. If $|\delta_+(z)| \geq J_{\mathrm{s}} > 0$ on E_ϵ, then*

$$\mathbb{E}\|\delta_+(z) z/\epsilon\|^2 \geq \frac{J_{\mathrm{s}}^2 d \pi_\epsilon}{\epsilon^2}. \tag{17}$$

Proof of Lemma D.1. Because every Rademacher direction satisfies $\|z\|^2 = d$,

$$\begin{aligned}\mathbb{E}\|\delta_+(z)z/\epsilon\|^2 &= \frac{d}{\epsilon^2}\mathbb{E}[\delta_+(z)^2] \\ &\geq \frac{d}{\epsilon^2}\mathbb{E}[\delta_+(z)^2\mathbf{1}_{E_\epsilon}] \\ &\geq \frac{dJ_{\mathrm{s}}^2}{\epsilon^2}\mathbb{E}\mathbf{1}_{E_\epsilon} = \frac{dJ_{\mathrm{s}}^2\pi_\epsilon}{\epsilon^2}. \end{aligned} \tag{18}$$

If $\pi_\epsilon = \Theta(\epsilon)$ and J_{s} is bounded below independently of ϵ, then there exist $a > 0$ and $\epsilon_0 > 0$ such that $\pi_\epsilon \geq a\epsilon$ for $0 < \epsilon \leq \epsilon_0$. The lower bound is then at least $adJ_{\mathrm{s}}^2/\epsilon$, which is $\Omega(1/\epsilon)$ for fixed d and J_{s}. □

The rest of the convergence proof can therefore study the loss evaluated with the recorded expert indices, without carrying an additional within-query switching term. Expert indices are still selected again at the next iteration; that separate effect is handled only when consecutive iterations are connected in Appendix D.3.

D.2 Why the Unperturbed Loss Is Used as the Shared Baseline

Replay already requires an unperturbed forward pass to record the expert indices. ZOMoE reuses the loss from this same pass as the common baseline for all one-sided perturbation queries. We next show two facts that justify this choice: subtracting any scalar baseline independent of the direction leaves the estimator mean unchanged, and the unperturbed loss is a second-order approximation to the scalar baseline that minimizes estimator variance.

Condition on $(\theta, \xi, \mathcal{S})$ and write $f(\vartheta) = \ell(\vartheta; \xi, \mathcal{S})$ and $Y(z) = f(\theta + \epsilon z)$. For a scalar b independent of z, define

$$G_b(z) = \frac{Y(z) - b}{\epsilon}z, \qquad \mathcal{V}(b) = \mathbb{E}\|G_b - \mathbb{E}G_b\|^2. \tag{19}$$

Proposition D.1 (Variance-minimizing scalar baseline)**.** *Suppose* $\mathbb{E}z = 0$, $0 < \mathbb{E}\|z\|^2 < \infty$, *and* $\mathbb{E}[Y(z)^2\|z\|^2] < \infty$. *Then* $\mathbb{E}G_b$ *does not depend on* b, *and* $\mathcal{V}(b)$ *is uniquely minimized at*

$$b^\star = \frac{\mathbb{E}[Y(z)\|z\|^2]}{\mathbb{E}\|z\|^2}. \tag{20}$$

Moreover,

$$\mathcal{V}(b) - \mathcal{V}(b^\star) = \frac{\mathbb{E}\|z\|^2}{\epsilon^2}(b - b^\star)^2. \tag{21}$$

Proof of Proposition D.1. First,

$$\mathbb{E}G_b = \frac{1}{\epsilon}\mathbb{E}[Y(z)z] - \frac{b}{\epsilon}\mathbb{E}z = \frac{1}{\epsilon}\mathbb{E}[Y(z)z] =: \mu. \tag{22}$$

Thus the mean is the same for every direction-independent scalar baseline. Expanding the centered second moment gives

$$\begin{aligned}\mathcal{V}(b) &= \mathbb{E}\|G_b\|^2 - \|\mu\|^2 \\ &= \epsilon^{-2}\left[\mathbb{E}(Y^2\|z\|^2) - 2b\mathbb{E}(Y\|z\|^2) + b^2\mathbb{E}\|z\|^2\right] - \|\mu\|^2. \end{aligned} \tag{23}$$

Let $A = \mathbb{E}\|z\|^2 > 0$, $B = \mathbb{E}(Y\|z\|^2)$, and $C = \mathbb{E}(Y^2\|z\|^2)$. Completing the square,

$$\begin{aligned}C - 2bB + b^2A &= A(b - B/A)^2 + C - B^2/A, \\ \mathcal{V}(b) &= \frac{A}{\epsilon^2}(b - B/A)^2 + \frac{C - B^2/A}{\epsilon^2} - \|\mu\|^2. \end{aligned} \tag{24}$$

The only b-dependent term is nonnegative and vanishes exactly at $b = B/A$, proving both claims. □

The variance-minimizing value still depends on perturbed losses and is not directly available from the recording pass. Smoothness shows that the unperturbed loss already returned by that pass approximates it to second order.

Proposition D.2 (Accuracy of the unperturbed loss as a baseline)**.** *Under the preceding moment conditions, suppose additionally that $z \overset{d}{=} -z$, $\mathbb{E}\|z\|^4 < \infty$, and f is L-smooth. Then*

$$|b^\star - f(\theta)| \le \frac{L\epsilon^2}{2} \frac{\mathbb{E}\|z\|^4}{\mathbb{E}\|z\|^2}. \tag{25}$$

For Rademacher directions this becomes $|b^\star - f(\theta)| \le L\epsilon^2 d/2$.

Proof of Proposition D.2. The integral form of the first-order expansion is

$$\begin{aligned} f(\theta + \epsilon z) &= f(\theta) + \epsilon \langle \nabla f(\theta), z \rangle + r(z), \\ r(z) &= \epsilon \int_0^1 \langle \nabla f(\theta + s\epsilon z) - \nabla f(\theta), z \rangle ds. \end{aligned} \tag{26}$$

By L-smoothness,

$$|r(z)| \le \epsilon \int_0^1 Ls\epsilon \|z\|^2 ds = \frac{L\epsilon^2}{2} \|z\|^2. \tag{27}$$

Central symmetry cancels the weighted linear term. Indeed, with $v = \mathbb{E}[\langle \nabla f(\theta), z \rangle \|z\|^2]$, replacing z by $-z$ gives $v = -v$, hence $v = 0$. Therefore

$$\begin{aligned} b^\star - f(\theta) &= \frac{\mathbb{E}[(f(\theta + \epsilon z) - f(\theta))\|z\|^2]}{\mathbb{E}\|z\|^2} \\ &= \frac{\mathbb{E}[r(z)\|z\|^2]}{\mathbb{E}\|z\|^2}. \end{aligned} \tag{28}$$

Taking absolute values and substituting the remainder bound gives

$$|b^\star - f(\theta)| \le \frac{L\epsilon^2}{2} \frac{\mathbb{E}\|z\|^4}{\mathbb{E}\|z\|^2}. \tag{29}$$

For Rademacher directions, $\|z\|^2 = d$ and $\|z\|^4 = d^2$ pathwise, so the ratio equals d. □

Thus the same unperturbed forward that records the expert indices also provides an appropriate shared one-sided baseline without an additional query. The convergence proof below now works directly with the normalized and clipped ZOMoE update built from these loss differences.

D.3 Proof of Theorem 4.1

We now prove the main convergence result. The proof follows the ZOMoE update itself. We first show that, despite normalization and clipping, the update retains a positive component along the minibatch gradient. We then transfer this alignment to the expected gradient and use smoothness to obtain a one-step decrease. Finally, we connect consecutive iterations, account for expert reselection, sum the resulting inequalities, and balance the learning rate and perturbation scale.

The main additional difficulty is to control the size of the normalized update without losing the benefit of using multiple perturbation directions. Bounding the clipped coefficients one at a time is valid but too loose for this purpose. We therefore begin with two auxiliary results that control the coefficients jointly and translate this control into a second-moment bound for the update. These estimates are used only where the number of directions enters the proof.

Lemma D.2 (Second moment of normalized coefficients)**.** *Let $N \ge 2$, $c > 0$, $\lambda > 0$, and $0 \le b \le \lambda/8$. Fix $v, w_1, \dots, w_N \in \mathbb{R}^d$, and let $s_1, \dots, s_N$ be independent uniform signs. For arbitrary remainders $e_i(s)$ satisfying $|e_i(s)| \le b$, define*

$$q_i(s) = s_i v^\top w_i + e_i(s), \qquad D(s) = \max\{\mathrm{Std}_i(q_i(s)), \lambda\}, \qquad A_i(s) = \mathrm{clip}\left(\frac{q_i(s)}{ND(s)}, -c, c\right). \tag{30}$$

The standard deviation uses divisor N. With $m_{N,c}$ from Eq. (10),

$$\mathbb{E}_s \sum_{i=1}^{N} A_i(s)^2 \le \frac{m_{N,c}^2}{N}. \tag{31}$$

The remainders may depend on all signs; no independence or zero-mean assumption on them is required.

Proof of Lemma D.2. Even-moment comparison. For any $\alpha \in \mathbb{R}^N$ with $\sum_i \alpha_i^2 = 1$, set

$$X = \frac{1}{\sqrt{N}} \sum_i \alpha_i s_i, \qquad Y = \frac{1}{N} \sum_i s_i. \tag{32}$$

Both variables lie in $[-1, 1]$. We first prove that, for every integer $j \ge 0$,

$$\mathbb{E}_s X^{2j} \le \mathbb{E}_s Y^{2j}. \tag{33}$$

Signs of the α_i can be absorbed into the s_i, so suppose $\alpha_i \ge 0$. Fix two coefficients p, q and keep $p^2 + q^2 = u$ fixed. For each integer $k \ge 0$,

$$\begin{aligned} \mathbb{E}(ps_1 + qs_2)^{2k} &= \tfrac{1}{2}\left[(p+q)^{2k} + (p-q)^{2k}\right] \\ &= \tfrac{1}{2}\left[(u+2pq)^{k} + (u-2pq)^{k}\right] \\ &= \sum_{r=0}^{\lfloor k/2 \rfloor} \binom{k}{2r} u^{k-2r} (2pq)^{2r}. \end{aligned} \tag{34}$$

Every term is nonnegative and nondecreasing in pq. Consequently, replacing (p^2, q^2) by $(u/2, u/2)$ cannot decrease any of these moments. If R is the independent sum of the remaining signed coefficients, then

$$\mathbb{E}(R + ps_1 + qs_2)^{2j} = \sum_{k=0}^{j} \binom{2j}{2k} \mathbb{E}R^{2j-2k}\, \mathbb{E}(ps_1 + qs_2)^{2k}; \tag{35}$$

odd powers of the pair sum vanish, and all terms displayed are nonnegative. The same replacement therefore cannot decrease the full even moment.

Repeated pairwise averaging drives the squared coefficients to the uniform vector while preserving the monotonicity above. Write $x_i = \alpha_i^2$ and $D_0 = \sum_i (x_i - 1/N)^2$. Averaging a largest and a smallest x_i decreases this sum by $(x_{\max} - x_{\min})^2/2$. Since $D_0 \le N(x_{\max} - x_{\min})^2$, each such operation contracts the sum by a factor at most $1 - 1/(2N)$. Thus $x_i \to 1/N$ for every i. Continuity of each finite even moment and the preceding monotonicity prove Eq. (33).

The binomial reciprocal sum. The variable Y equals $1 - 2k/N$ with probability $2^{-N}\binom{N}{k}$. Its two extreme values have total probability

$$p_N = 2^{1-N}. \tag{36}$$

The contribution of all other values to the reciprocal is

$$S_N = 2^{-N} \sum_{k=1}^{N-1} \binom{N}{k} \frac{N^2}{4k(N-k)}. \tag{37}$$

The binomial sum satisfies

$$0 < S_N \le 1 + \frac{2}{N}. \tag{38}$$

To see this, put $H_n = \sum_{k=1}^{n} \binom{n}{k}/k$. The integral representation

$$H_n = \int_0^1 \frac{(1+x)^n - 1}{x}\, dx \tag{39}$$

shows that $H_n - H_{n-1} = (2^n - 1)/n$. Using $1/[k(n-k)] = (1/k + 1/(n-k))/n$ gives

$$S_n = \frac{n}{2^{n+1}} \left(H_n - \frac{1}{n}\right), \qquad S_n = \frac{n}{2(n-1)} S_{n-1} + \frac{1}{2} - \frac{n-2}{(n-1)2^{n+1}} \quad (n \ge 3). \tag{40}$$

Direct evaluation yields

$$S_2 = \frac{1}{2}, \quad S_3 = \frac{27}{32}, \quad S_4 = \frac{25}{24}, \quad S_5 = \frac{875}{768}, \quad S_6 = \frac{1507}{1280}, \tag{41}$$

each of which is at most $1+2/n$ at its corresponding index. For $n \geq 7$, assuming the bound at $n-1$ and dropping the negative term in Eq. (40) gives

$$\begin{aligned} S_n &\leq \frac{n}{2(n-1)}\left(1+\frac{2}{n-1}\right)+\frac{1}{2} = 1+\frac{3n-1}{2(n-1)^2} \\ &\leq 1+\frac{2}{n}, \end{aligned} \tag{42}$$

because $4(n-1)^2 - n(3n-1) = n^2 - 7n + 4 \geq 0$ for $n \geq 7$. This proves Eq. (38) by induction.

A truncated reciprocal bound. Fix $K > 0$ and interpret $1/(1-X^2)$ as $+\infty$ when $|X| = 1$. For every $\delta > 0$ and $y \in [0,1]$,

$$\min\left\{\frac{1}{1-y}, K\right\} \leq \frac{1+\delta}{1-y+\delta/K}. \tag{43}$$

Indeed, for $1-y \geq 1/K$ the right-hand denominator is at most $(1+\delta)(1-y)$; for $1-y \leq 1/K$ it is at most $(1+\delta)/K$. Also,

$$\begin{aligned} \mathbb{E}\frac{1}{1-X^2+\delta/K} &= \sum_{j=0}^{\infty}(1+\delta/K)^{-j-1}\mathbb{E}X^{2j} \\ &\leq \sum_{j=0}^{\infty}(1+\delta/K)^{-j-1}\mathbb{E}Y^{2j} = \mathbb{E}\frac{1}{1-Y^2+\delta/K}. \end{aligned} \tag{44}$$

The series have nonnegative terms and ratio strictly below one, so the exchanges with expectation are justified. For the two extreme values of Y, the reciprocal is K/δ; for the others it is at most $1/(1-Y^2)$. Combining Eqs. (43)–(44) gives

$$\mathbb{E}\min\left\{\frac{1}{1-X^2}, K\right\} \leq (1+\delta)\left(S_N + \frac{Kp_N}{\delta}\right). \tag{45}$$

Set $\delta = \sqrt{Kp_N/S_N} > 0$ and expand the right-hand side:

$$S_N + Kp_N + 2\sqrt{S_N K p_N} = \left(\sqrt{S_N} + \sqrt{Kp_N}\right)^2. \tag{46}$$

This bound keeps the clipped contribution of the extreme values instead of assuming that the empirical standard deviation is nonzero.

Applying the bounds to the coefficients. Write $h_i = v^\top w_i$, $u_i = s_i h_i$, $r = (N^{-1}\sum_i h_i^2)^{1/2}$, and $\chi = b/\lambda \leq 1/8$. First suppose $r > 0$ and set

$$X = \frac{\bar{u}}{r} = \frac{1}{\sqrt{N}}\sum_i \frac{h_i}{(\sum_j h_j^2)^{1/2}} s_i. \tag{47}$$

This is the variable in Eq. (32). Since $q = u + e$ and $\|e\| \leq \sqrt{N}b$, the triangle inequality for centered vectors gives

$$\begin{aligned} \mathrm{Std}_i(u_i) \leq \mathrm{Std}_i(q_i) + b \leq D + b \leq (1+\chi)D, \\ \frac{\|q\|}{\sqrt{N}} \leq r + b, \qquad \mathrm{Std}_i(u_i) = r\sqrt{1-X^2}. \end{aligned} \tag{48}$$

When $|X| < 1$, these inequalities imply

$$\frac{\|q\|}{\sqrt{N}D} \leq \frac{1+\chi}{\sqrt{1-X^2}} + \chi \leq \frac{1+2\chi}{\sqrt{1-X^2}} \leq \frac{5/4}{\sqrt{1-X^2}}. \tag{49}$$

Clipping cannot increase a squared coefficient, and $|A_i| \leq c$, so

$$\sum_i A_i^2 \leq \min\left\{\frac{\|q\|^2}{N^2D^2}, Nc^2\right\} \leq \frac{1}{N}\min\left\{\frac{(5/4)^2}{1-X^2}, N^2c^2\right\}. \tag{50}$$

The last expression also bounds the left side when $|X| = 1$, by its finite clipping branch. Apply Eq. (46) with $K = N^2c^2/(5/4)^2$ and then Eq. (38):

$$\mathbb{E}_s \sum_i A_i^2 \le \frac{1}{N}\left(\frac{5}{4}\sqrt{S_N} + Nc\sqrt{p_N}\right)^2 \le \frac{1}{N}\left(\frac{5}{4}\sqrt{1+\frac{2}{N}} + Nc\,2^{(1-N)/2}\right)^2. \tag{51}$$

Taking the minimum with Nc^2 proves the claim for $r > 0$. If $r = 0$, then $q = e$ and

$$\sum_i A_i^2 \le \frac{1}{N}\min\{\chi^2, N^2c^2\} \le \frac{m_{N,c}^2}{N}, \tag{52}$$

because $\chi \le 1/8$ while the second branch defining $m_{N,c}$ is greater than one. This completes the proof. □

Lemma D.3 (Second moment of a Rademacher matrix)**.** *If $W \in \mathbb{R}^{d\times N}$ has independent Rademacher entries, then*

$$\mathbb{E}\|W\|_{\mathrm{op}}^2 \le \min\Big\{dN,\ \frac{\pi}{2}\big[(\sqrt{d}+\sqrt{N})^2+1\big]\Big\}. \tag{53}$$

Proof of Lemma D.3. The first bound is immediate from $\|W\|_{\mathrm{op}}^2 \le \|W\|_{\mathrm{F}}^2 = dN$. For the second, let G have independent standard Gaussian entries, and let R be their entrywise signs. Then R has the same distribution as W and

$$\mathbb{E}[G \mid R] = \sqrt{2/\pi}\,R. \tag{54}$$

Convexity of the squared operator norm and conditional Jensen give

$$\frac{2}{\pi}\|R\|_{\mathrm{op}}^2 \le \mathbb{E}[\|G\|_{\mathrm{op}}^2 \mid R]. \tag{55}$$

A Gaussian comparison gives the required estimate on the right. For unit vectors $x, x' \in \mathbb{R}^d$ and $y, y' \in \mathbb{R}^N$, define $X_{x,y} = x^\top Gy$ and $Y_{x,y} = g^\top x + h^\top y$, where g, h are independent standard Gaussian vectors. Writing $u = \langle x, x'\rangle$ and $v = \langle y, y'\rangle$ gives

$$\begin{aligned}\mathbb{E}(X_{x,y} - X_{x',y'})^2 &= 2 - 2uv,\\ \mathbb{E}(Y_{x,y} - Y_{x',y'})^2 &= 4 - 2u - 2v,\\ (4-2u-2v) - (2-2uv) &= 2(1-u)(1-v) \ge 0.\end{aligned} \tag{56}$$

The Sudakov–Fernique comparison inequality, applied first on finite nets and then by continuity on the two unit spheres, yields

$$\mathbb{E}\|G\|_{\mathrm{op}} = \mathbb{E}\sup_{x,y} X_{x,y} \le \mathbb{E}\sup_{x,y} Y_{x,y} = \mathbb{E}\|g\| + \mathbb{E}\|h\| \le \sqrt{d} + \sqrt{N}; \tag{57}$$

see Vershynin (2018, Theorems 7.2.11 and 7.3.1). Finally,

$$|\|G\|_{\mathrm{op}} - \|G'\|_{\mathrm{op}}| \le \|G - G'\|_{\mathrm{op}} \le \|G - G'\|_{\mathrm{F}}, \tag{58}$$

so the Gaussian Poincaré inequality for Lipschitz functions gives $\mathrm{Var}(\|G\|_{\mathrm{op}}) \le 1$ (van Handel, 2016, Corollary 2.27). Thus

$$\mathbb{E}\|G\|_{\mathrm{op}}^2 = \mathrm{Var}(\|G\|_{\mathrm{op}}) + (\mathbb{E}\|G\|_{\mathrm{op}})^2 \le 1 + (\sqrt{d}+\sqrt{N})^2. \tag{59}$$

Taking expectations in Eq. (55) proves the second bound. □

Proof of Theorem 4.1. Fix an iteration t and condition on the history $\mathcal{H}_t$ before the current minibatch and directions are drawn. Then θ_t, the routing map $S_{\theta_t}(\cdot)$, and F_t are fixed. For brevity, write

$$g_t = \nabla F_t(\theta_t), \quad v_t = \nabla f_{t,\xi_t}(\theta_t), \quad \mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{H}_t], \tag{60}$$

so Assumption 4.2 gives

$$\mathbb{E}_t v_t = g_t, \qquad \mathbb{E}_t\|v_t - g_t\|^2 \le \frac{\sigma^2}{B}. \tag{61}$$

Use the deterministic abbreviations

$$a = \min\{1, c\sqrt{N}\}, \quad \kappa = \sqrt{\frac{N}{N+2}}, \quad m = m_{N,c}, \quad V = V_{N,d,c}, \qquad b = \frac{L\epsilon d}{2}, \quad \lambda = \frac{\tau}{\epsilon}. \tag{62}$$

Both branches defining m are at least a: $Nc \geq c\sqrt{N} \geq a$, and the other branch is greater than one. In particular, $m \geq a > 0$. The perturbation scale chosen in Theorem 4.1 therefore satisfies

$$\frac{b}{\lambda} = \frac{Ld\epsilon^2}{2\tau} = \frac{a}{4(a+m)} \leq \frac{1}{8}. \tag{63}$$

Thus the coefficient lemma applies to the stated parameter choice. The intermediate bound below also holds for any $\eta > 0$ and $\epsilon > 0$ satisfying $Ld\epsilon^2/(2\tau) \leq 1/8$.

For each direction $z_{t,i}$, define the one-sided slope

$$q_{t,i} = \frac{f_{t,\xi_t}(\theta_t + \epsilon z_{t,i}) - f_{t,\xi_t}(\theta_t)}{\epsilon}. \tag{64}$$

Because adding the same constant to all queried losses does not change their standard deviation,

$$\begin{aligned} \mathrm{Std}_i(\ell^{\mathrm{rep}}_{t,i}) &= \mathrm{Std}_i\big(f_{t,\xi_t}(\theta_t) + \epsilon q_{t,i}\big) \\ &= \epsilon\, \mathrm{Std}_i(q_{t,i}). \end{aligned} \tag{65}$$

Hence, if

$$s_t = \max\{\mathrm{Std}_i(q_{t,i}), \lambda\}, \qquad A_{t,i} = \mathrm{clip}\left(\frac{q_{t,i}}{N s_t}, -c, c\right), \tag{66}$$

then $\sigma_t = \epsilon s_t$ and the algorithmic coefficient in Eq. (6) is exactly $A_{t,i}$:

$$\frac{\ell^{\mathrm{rep}}_{t,i} - \ell_{0,t}}{N\sigma_t} = \frac{\epsilon q_{t,i}}{N\epsilon s_t} = \frac{q_{t,i}}{N s_t}. \tag{67}$$

Therefore the ZOMoE update direction can be written without any approximation as

$$\widehat{g}_t = \sum_{i=1}^{N} A_{t,i} z_{t,i}. \tag{68}$$

We first control the effect of normalization and clipping on the directional signal. A deterministic alignment bound handles the signal itself, while Lemma D.2 controls the finite-difference remainder without discarding the averaging across directions. For a generic vector $q = (q_1, \ldots, q_N) \in \mathbb{R}^N$, define

$$r(q) = \frac{\|q\|}{\sqrt{N}}, \qquad s(q) = \max\{\mathrm{Std}_i(q_i), \lambda\}, \qquad A_i(q) = \mathrm{clip}\left(\frac{q_i}{N s(q)}, -c, c\right). \tag{69}$$

We first show the pointwise inequality

$$\sum_{i=1}^{N} A_i(q) q_i \geq a\left(r(q) - \frac{\lambda}{4}\right). \tag{70}$$

If $q = 0$, the left side is zero and the right side is nonpositive, so the claim is immediate. Assume $q \neq 0$ and put $r = r(q)$. Since centering cannot increase Euclidean norm,

$$\mathrm{Std}_i(q_i) = \frac{\|q - \bar{q}\mathbf{1}\|}{\sqrt{N}} \leq \frac{\|q\|}{\sqrt{N}} = r. \tag{71}$$

Thus

$$s(q) \leq \max\{r, \lambda\} =: M. \tag{72}$$

Moreover $|q_i| \leq \|q\| = \sqrt{N} r$ and $\sum_i q_i^2 = N r^2$. Coordinates with $q_i = 0$ contribute zero. For every coordinate with $q_i \neq 0$,

$$\begin{aligned} A_i(q) q_i &= |q_i| \min\left\{\frac{|q_i|}{N s(q)}, c\right\} \\ &= q_i^2 \min\left\{\frac{1}{N s(q)}, \frac{c}{|q_i|}\right\} \\ &\geq q_i^2 \min\left\{\frac{1}{NM}, \frac{c}{\sqrt{N} r}\right\}. \end{aligned} \tag{73}$$

Summing over i gives

$$\sum_i A_i(q)q_i \geq Nr^2 \min\left\{\frac{1}{NM}, \frac{c}{\sqrt{N}r}\right\}$$
$$= \min\left\{\frac{r^2}{M}, c\sqrt{N}\, r\right\}. \tag{74}$$

Let $x = r^2/M$. Since $M \geq r$, we have $0 \leq x \leq r$. If $c\sqrt{N} \geq 1$, the second argument in the minimum in Eq. (74) is at least $r \geq x$, so the minimum equals x. If $c\sqrt{N} < 1$, then both x and $c\sqrt{N}\, r$ are at least $c\sqrt{N}\, x$. In both cases,

$$\sum_i A_i(q)q_i \geq a\frac{r^2}{\max\{r, \lambda\}}. \tag{75}$$

It remains to lower bound the scalar ratio. If $r \geq \lambda$, it equals r. If $0 \leq r \leq \lambda$, then

$$\frac{r^2}{\lambda} - r + \frac{\lambda}{4} = \frac{(r - \lambda/2)^2}{\lambda} \geq 0. \tag{76}$$

Therefore $r^2/\max\{r, \lambda\} \geq r - \lambda/4$, proving Eq. (70).

The preceding inequality depends on the root-mean-square size of the directional slopes. We next show that, on average over the Rademacher directions, this quantity is proportional to the Euclidean norm of the underlying gradient. Fix a vector $v \in \mathbb{R}^d$ and let $u_i = v^\top z_{t,i}$. Put

$$r_u = \left(\frac{1}{N}\sum_{i=1}^N u_i^2\right)^{1/2}. \tag{77}$$

We claim

$$\mathbb{E}_z r_u \geq \kappa\|v\|. \tag{78}$$

The case $v = 0$ is immediate, so assume $v \neq 0$. Let $X = v^\top z$ for one Rademacher vector. Coordinate independence gives

$$\mathbb{E}X^2 = \|v\|^2 \tag{79}$$

and

$$\mathbb{E}X^4 = \sum_j v_j^4 + 6\sum_{j<k} v_j^2 v_k^2$$
$$= 3\left(\sum_j v_j^2\right)^2 - 2\sum_j v_j^4 \leq 3\|v\|^4. \tag{80}$$

Using independence across the N directions,

$$\mathbb{E}_z r_u^2 = \frac{1}{N}\sum_i \mathbb{E}u_i^2 = \|v\|^2,$$
$$\mathbb{E}_z r_u^4 = \frac{1}{N^2}\left[\sum_i \mathbb{E}u_i^4 + \sum_{i\neq j}\mathbb{E}(u_i^2 u_j^2)\right]$$
$$\leq \frac{3N + N(N-1)}{N^2}\|v\|^4 = \left(1 + \frac{2}{N}\right)\|v\|^4. \tag{81}$$

For any nonnegative random variable R, Hölder's inequality applied to $R^2 = R^{2/3}R^{4/3}$ gives

$$\mathbb{E}R^2 \leq (\mathbb{E}R)^{2/3}(\mathbb{E}R^4)^{1/3}. \tag{82}$$

Rearranging with $R = r_u$ and using Eq. (81),

$$\mathbb{E}_z r_u \geq \frac{(\mathbb{E}_z r_u^2)^{3/2}}{(\mathbb{E}_z r_u^4)^{1/2}}$$
$$\geq \frac{\|v\|^3}{\sqrt{1 + 2/N}\,\|v\|^2} = \sqrt{\frac{N}{N+2}}\|v\| = \kappa\|v\|, \tag{83}$$

which proves Eq. (78).

We can now apply these two deterministic and probabilistic ingredients to the actual one-sided slopes produced by ZOMoE. The only additional error comes from the finite perturbation radius. Condition now on both $\mathcal{H}_t$ and the current minibatch. Write $v = v_t$ and $f = f_{t,\xi_t}$. By the fundamental theorem of calculus,

$$\begin{aligned} q_{t,i} &= v^\top z_{t,i} + e_{t,i}, \\ e_{t,i} &= \int_0^1 \langle \nabla f(\theta_t + s\epsilon z_{t,i}) - \nabla f(\theta_t), z_{t,i} \rangle \, ds. \end{aligned} \tag{84}$$

Because f is L-smooth and every Rademacher direction satisfies $\|z_{t,i}\|^2 = d$,

$$|e_{t,i}| \le \int_0^1 Ls\epsilon \|z_{t,i}\|^2 ds = \frac{L\epsilon d}{2} = b. \tag{85}$$

Let $u_i = v^\top z_{t,i}$, and let $q = (q_{t,1}, \dots, q_{t,N})$ and $u = (u_1, \dots, u_N)$. From $\|q - u\| \le \sqrt{N}\, b$,

$$|r(q) - r(u)| = \frac{|\|q\| - \|u\||}{\sqrt{N}} \le \frac{\|q - u\|}{\sqrt{N}} \le b. \tag{86}$$

We now apply Lemma D.2 without changing the distribution of the algorithm's directions. Conditional on $(\mathcal{H}_t, \xi_t)$, generate independent Rademacher vectors w_i and independent uniform signs s_i, and set

$$z_{t,i} = s_i w_i, \qquad W = [w_1, \dots, w_N], \qquad Z = [z_{t,1}, \dots, z_{t,N}] = W \operatorname{diag}(s). \tag{87}$$

For every prescribed vector $z \in \{-1, 1\}^d$, $\Pr(s_i w_i = z) = 2^{-d}$; different columns are independent. Thus Z has exactly the original conditional distribution, and $\|Z\|_{\mathrm{op}} = \|W\|_{\mathrm{op}}$ for every realization. For fixed W, the one-sided remainders in Eq. (84) may depend on the signs, but their bound $|e_{t,i}| \le b$ holds for every sign realization. Equation (63) and Lemma D.2 give

$$\mathbb{E}_s[\|A_t\|^2 \mid \mathcal{H}_t, \xi_t, W] \le \frac{m^2}{N}, \qquad A_t = (A_{t,1}, \dots, A_{t,N}). \tag{88}$$

Averaging over W and applying Cauchy–Schwarz and Jensen yields

$$\begin{aligned} \mathbb{E}_z[\|A_t\|^2 \mid \mathcal{H}_t, \xi_t] &\le \frac{m^2}{N}, \\ \mathbb{E}_z\left[\sum_i |A_{t,i}| \mid \mathcal{H}_t, \xi_t\right] &\le \sqrt{N}\, \mathbb{E}_z[\|A_t\| \mid \mathcal{H}_t, \xi_t] \le m. \end{aligned} \tag{89}$$

Averaging first over the signs and then over W preserves this dependence and gives the required conditional estimate.

Using the deterministic alignment bound and Eq. (86), we have pathwise

$$\begin{aligned} \langle v_t, \widehat{g}_t \rangle &= \sum_i A_{t,i} q_{t,i} - \sum_i A_{t,i} e_{t,i} \\ &\ge a\left(r(q) - \frac{\lambda}{4}\right) - b\sum_i |A_{t,i}| \\ &\ge a r(u) - \frac{a\lambda}{4} - ab - b\sum_i |A_{t,i}|. \end{aligned} \tag{90}$$

Taking expectation over the directions, and using Eqs. (78) and (89), gives

$$\mathbb{E}_z[\langle v_t, \widehat{g}_t \rangle \mid \mathcal{H}_t, \xi_t] \ge a\kappa\|v_t\| - \frac{a\lambda}{4} - (a + m)b. \tag{91}$$

We next transfer this alignment from the minibatch gradient to the expected gradient. Set $\delta_t = v_t - g_t$. Conditional on $(\mathcal{H}_t, \xi_t)$, both v_t and δ_t are fixed, while Z has independent Rademacher

columns. Therefore

$$\mathbb{E}_z[\|Z^\top \delta_t\|^2 \mid \mathcal{H}_t, \xi_t] = \sum_{i=1}^{N} \sum_{j,k} \delta_{t,j}\delta_{t,k}\mathbb{E}_z[z_{t,i,j}z_{t,i,k}] = N\sum_j \delta_{t,j}^2 = N\|\delta_t\|^2. \tag{92}$$

Apply Cauchy–Schwarz first to the inner product and then to expectation:

$$\begin{aligned}\left|\mathbb{E}_z[\langle \delta_t, \widehat{g}_t\rangle \mid \mathcal{H}_t, \xi_t]\right| &= \left|\mathbb{E}_z[\langle Z^\top \delta_t, A_t\rangle \mid \mathcal{H}_t, \xi_t]\right| \\ &\le \Big(\mathbb{E}_z[\|Z^\top \delta_t\|^2 \mid \mathcal{H}_t, \xi_t]\Big)^{1/2}\Big(\mathbb{E}_z[\|A_t\|^2 \mid \mathcal{H}_t, \xi_t]\Big)^{1/2} \\ &\le \sqrt{N}\,\|\delta_t\|\frac{m}{\sqrt{N}} = m\|\delta_t\|. \end{aligned} \tag{93}$$

Averaging over the minibatch and applying Eq. (61) gives

$$\left|\mathbb{E}_t\langle \delta_t, \widehat{g}_t\rangle\right| \le m\,\mathbb{E}_t\|\delta_t\| \le m\big(\mathbb{E}_t\|\delta_t\|^2\big)^{1/2} \le \frac{m\sigma}{\sqrt{B}}. \tag{94}$$

Because the coefficients depend on the same minibatch as the update, we bound this coupling term directly. Since $g_t = v_t - \delta_t$ and $\mathbb{E}_t\|v_t\| \ge \|\mathbb{E}_t v_t\| = \|g_t\|$, Eq. (91) implies

$$\begin{aligned}\mathbb{E}_t\langle g_t, \widehat{g}_t\rangle &= \mathbb{E}_t\langle v_t, \widehat{g}_t\rangle - \mathbb{E}_t\langle \delta_t, \widehat{g}_t\rangle \\ &\ge a\kappa\|g_t\| - \frac{a\lambda}{4} - (a+m)b - \frac{m\sigma}{\sqrt{B}}. \end{aligned} \tag{95}$$

It remains to control the size of the update. The joint coefficient bound now gives the tighter second-moment estimate that preserves directional averaging. By Eq. (87), $\|\widehat{g}_t\|^2 = \|ZA_t\|^2 \le \|W\|_{\mathrm{op}}^2\|A_t\|^2$. The matrix factor is fixed while averaging over s, so Eq. (88) gives

$$\mathbb{E}_s[\|\widehat{g}_t\|^2 \mid \mathcal{H}_t, \xi_t, W] \le \frac{m^2}{N}\|W\|_{\mathrm{op}}^2. \tag{96}$$

Lemma D.3, followed by averaging over W and the minibatch, yields

$$\mathbb{E}_t\|\widehat{g}_t\|^2 \le \min\left\{dm^2,\ \frac{\pi m^2}{2N}\left[(\sqrt{d}+\sqrt{N})^2+1\right]\right\}. \tag{97}$$

We can also use clipping directly. Since $|A_{t,i}| \le c$ and $\|z_{t,i}\|^2 = d$,

$$\begin{aligned}\|\widehat{g}_t\|^2 &= \sum_i A_{t,i}^2 d + \sum_{i\ne j} A_{t,i}A_{t,j}z_{t,i}^\top z_{t,j} \\ &\le Nc^2 d + c^2\sum_{i\ne j}|z_{t,i}^\top z_{t,j}|. \end{aligned} \tag{98}$$

For $i \ne j$, independence of the two Rademacher directions implies

$$\mathbb{E}_z(z_{t,i}^\top z_{t,j})^2 = \sum_{p,q}\mathbb{E}_z[z_{t,i,p}z_{t,i,q}]\mathbb{E}_z[z_{t,j,p}z_{t,j,q}] = d. \tag{99}$$

Thus $\mathbb{E}_z|z_{t,i}^\top z_{t,j}| \le \sqrt{d}$. There are $N(N-1)$ ordered cross pairs, and Eq. (98) gives

$$\mathbb{E}_t\|\widehat{g}_t\|^2 \le c^2 N[d + (N-1)\sqrt{d}]. \tag{100}$$

Combining the two estimates proves

$$\mathbb{E}_t\|\widehat{g}_t\|^2 \le V_{N,d,c} = V. \tag{101}$$

This minimum retains the direct clipping estimate when it is smaller for a particular N, d, c. The update also remains bounded pathwise by $\|\widehat{g}_t\| \le Nc\sqrt{d}$.

With the alignment and second-moment bounds in place, smoothness gives a one-step decrease. We then connect consecutive iterations, where the selected experts may change. The update uses a constant η, so L-smoothness along its segment gives

$$F_t(\theta_{t+1}) \le F_t(\theta_t) - \eta\langle g_t, \widehat{g}_t\rangle + \frac{L\eta^2}{2}\|\widehat{g}_t\|^2. \tag{102}$$

Taking $\mathbb{E}_t$ and substituting Eqs. (95) and (101) yields

$$\begin{aligned}\eta a\kappa\|g_t\| \le F_t(\theta_t) - \mathbb{E}_t F_t(\theta_{t+1}) + \frac{L\eta^2 V}{2} \\ + \eta\left[(a+m)b + \frac{m\sigma}{\sqrt{B}} + \frac{a\lambda}{4}\right].\end{aligned} \tag{103}$$

At the next iterate, define F_{t+1} using the routing map selected at θ_{t+1}, whether or not another update will be performed. Adding and subtracting $F_{t+1}(\theta_{t+1})$ gives the pathwise identity

$$\begin{aligned}F_t(\theta_t) - F_t(\theta_{t+1}) = F_t(\theta_t) - F_{t+1}(\theta_{t+1}) \\ + F_{t+1}(\theta_{t+1}) - F_t(\theta_{t+1}).\end{aligned} \tag{104}$$

Summing from $t = 0$ to $T-1$ telescopes the first differences:

$$\begin{aligned}\sum_{t=0}^{T-1}[F_t(\theta_t) - F_t(\theta_{t+1})] = F_0(\theta_0) - F_T(\theta_T) \\ + \sum_{t=0}^{T-1}[F_{t+1}(\theta_{t+1}) - F_t(\theta_{t+1})].\end{aligned} \tag{105}$$

Since $F_T(\theta_T) \ge F_\star$, taking full expectation and using $x \le (x)_+$ gives

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}[F_t(\theta_t) - F_t(\theta_{t+1})] \le F_0(\theta_0) - F_\star \\ + \sum_{t=0}^{T-1}\mathbb{E}\Big[\big(F_{t+1}(\theta_{t+1}) - F_t(\theta_{t+1})\big)_+\Big].\end{aligned} \tag{106}$$

Assumption 4.3 bounds the accumulated change from expert reselection by

$$\rho\sum_{t=0}^{T-1}\mathbb{E}\|\theta_{t+1} - \theta_t\|^2 = \rho\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\widehat{g}_t\|^2 \le \rho\eta^2 TV. \tag{107}$$

Taking full expectations in Eq. (103), summing, and applying Eqs. (106)–(107) gives

$$\begin{aligned}\eta a\kappa\sum_{t=0}^{T-1}\mathbb{E}\|g_t\| \le F_0(\theta_0) - F_\star + \frac{(L+2\rho)\eta^2 TV}{2} \\ + \eta T\left[(a+m)b + \frac{m\sigma}{\sqrt{B}} + \frac{a\lambda}{4}\right].\end{aligned} \tag{108}$$

All factors η, T, a, κ are positive, so division yields

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|g_t\| \le \frac{1}{a\kappa}\Bigg[\frac{F_0(\theta_0) - F_\star}{\eta T} + \frac{(L+2\rho)\eta V}{2} \\ + \frac{Ld(a+m)}{2}\epsilon + \frac{m\sigma}{\sqrt{B}} + \frac{a\tau}{4\epsilon}\Bigg].\end{aligned} \tag{109}$$

Finally, we balance the two scalar parameters in Eq. (109). Write $\Delta = F_0(\theta_0) - F_\star > 0$ and $Q = (L+2\rho)V > 0$. The learning-rate terms are

$$\psi(\eta) = \frac{\Delta}{\eta T} + \frac{Q\eta}{2}, \quad \psi'(\eta) = -\frac{\Delta}{T\eta^2} + \frac{Q}{2}, \quad \psi''(\eta) = \frac{2\Delta}{T\eta^3} > 0. \tag{110}$$

Their unique minimizer and value are

$$\eta^2 = \frac{2\Delta}{QT}, \qquad \frac{\Delta}{\eta T} = \frac{Q\eta}{2} = \sqrt{\frac{Q\Delta}{2T}},$$
$$\psi(\eta) = \sqrt{\frac{2Q\Delta}{T}}. \tag{111}$$

This is exactly the learning-rate choice in Theorem 4.1 and yields the first term of Eq. (12). The perturbation terms are

$$\phi(\epsilon) = \frac{Ld(a+m)}{2}\epsilon + \frac{a\tau}{4\epsilon}, \quad \phi'(\epsilon) = \frac{Ld(a+m)}{2} - \frac{a\tau}{4\epsilon^2}, \quad \phi''(\epsilon) = \frac{a\tau}{2\epsilon^3} > 0. \tag{112}$$

Their minimizer satisfies

$$\epsilon^2 = \frac{a\tau}{2Ld(a+m)}, \qquad \frac{Ld\epsilon^2}{2\tau} = \frac{a}{4(a+m)} \le \frac{1}{8}. \tag{113}$$

Thus the balanced choice of ϵ lies within the range required by Lemma D.2 and Eq. (109). At this value,

$$\phi(\epsilon) = 2\sqrt{\frac{Ld(a+m)}{2}\frac{a\tau}{4}} = \sqrt{\frac{Ld\,a(a+m)\tau}{2}}. \tag{114}$$

Substitute Eqs. (111) and (114) into Eq. (109) and use

$$\frac{1}{a\kappa} = \frac{\sqrt{(N+2)/N}}{\min\{1, c\sqrt{N}\}}, \qquad m = m_{N,c}, \qquad V = V_{N,d,c}. \tag{115}$$

This gives Eq. (12). □

The dependence on the number of directions stated after Theorem 4.1 can also be verified directly. For $N \ge 2$, $N2^{(1-N)/2} \le 3/2$: the first two values are $\sqrt{2}$ and $3/2$, and the ratio of successive values is $(1+1/N)/\sqrt{2} < 1$ for $N \ge 3$. Therefore

$$m_{N,c} \le \frac{5\sqrt{2}}{4} + \frac{3c}{2}, \qquad m_{N,c} \longrightarrow \frac{5}{4} \quad \text{for fixed } c > 0. \tag{116}$$

Moreover,

$$V_{N,d,c} \le \frac{\pi m_{N,c}^2}{2}\left[\frac{d}{N} + 1 + 2\sqrt{\frac{d}{N}} + \frac{1}{N}\right]$$
$$\le \frac{3\pi m_{N,c}^2}{2}\left(1 + \frac{d}{N}\right). \tag{117}$$

The last step uses $2\sqrt{d/N} \le 1 + d/N$ and $1/N \le 1 + d/N$. Also $a \ge \min\{1, c\sqrt{2}\} > 0$ and $\kappa \ge 1/\sqrt{2}$. Substitution into Eq. (12) gives the stated $O_c(1+d/N)$ dependence, with no dependence on $N, d, L, \rho, T, B, \tau$, or the initial loss gap hidden in O_c.

D.4 Sufficient Conditions for Assumption 4.3

The main theorem isolates one effect that replay does not remove: the next iteration selects its expert indices again. Assumption 4.3 controls the resulting increase in expected loss by the squared update size. This section explains when that condition is reasonable for hard top-k routing. The argument has two parts. First, an expert set can change only if the current top-k gap is small. Second, combining this boundary event with a bound on the loss change caused by switching experts yields the quadratic control used in the theorem.

We state the conditions uniformly over the histories and updates reached by the algorithm. Expectations below are over a fresh data point conditional on the current history. For one such input x, order the routing decisions in feed-forward order. Consider only decisions at which fewer than all

available experts are selected; if there are no such decisions, reselection makes no change and $\rho = 0$. At a decision a, let the router logits at θ_t be $s_{t,a}(\theta_t; x)$ and let

$$m_t(x) = \min_a \left[s_{t,a,(k)}(\theta_t; x) - s_{t,a,(k+1)}(\theta_t; x) \right] \tag{118}$$

be the smallest top-k gap at θ_t over all routed tokens and layers. For a perturbed parameter $\theta_t + h$, define $\widetilde{s}_{t,a}(\theta_t + h; x)$ as the logits obtained after replaying the expert selections made at θ_t at all routing events before a and recomputing all continuous operations.

Lemma D.4 (Changing experts requires a small top-k gap)**.** *Suppose that, for every routing event and every update h under consideration,*

$$\|\widetilde{s}_{t,a}(\theta_t + h; x) - \widetilde{s}_{t,a}(\theta_t; x)\|_\infty \le G_r \|h\|. \tag{119}$$

If the forwards at θ_t and $\theta_t + h$ select different expert sets for x, then

$$m_t(x) \le 2G_r \|h\|. \tag{120}$$

Proof of Lemma D.4. Choose the first routing event at which the selected expert set differs. All earlier selections are identical, so the actual updated logits at this first differing event equal the logits obtained by replaying the preceding selections in Eq. (119). Let S be the top-k set at θ_t and let $\delta = G_r\|h\|$. If the top-k gap at this event were strictly larger than 2δ, then for every selected expert $i \in S$ and every unselected expert $j \notin S$,

$$\begin{aligned} \widetilde{s}_i(\theta_t + h; x) - \widetilde{s}_j(\theta_t + h; x) &\ge [s_i(\theta_t; x) - \delta] - [s_j(\theta_t; x) + \delta] \\ &\ge s_{(k)}(\theta_t; x) - s_{(k+1)}(\theta_t; x) - 2\delta > 0. \end{aligned} \tag{121}$$

Every selected expert would therefore still outrank every unselected expert, so the top-k set could not change. This contradicts the choice of the event. Hence the margin at the first differing event is at most 2δ, and the minimum margin $m_t(x)$ is no larger. □

The next proposition shows two simple ways to obtain the quadratic refresh control used in the main theorem.

Proposition D.3 (Sufficient conditions for Assumption 4.3)**.** *Assume Eq. (119) holds uniformly over the updates reached by the algorithm. The constants C, J, G_r below are uniform over these histories and updates, and each boundary-mass inequality holds for all $u \ge 0$. Either of the following two pairs of conditions is sufficient for Assumption 4.3:*

1. *The boundary mass is linear,*

$$\Pr_{x \sim \mathcal{D}} \{m_t(x) \le u\} \le Cu, \tag{122}$$

   and, whenever the selected experts change, the positive loss jump at the updated parameter is at most $J\|h\|$. Then Eq. (9) holds with $\rho = 2JCG_r$.

2. *The boundary mass is quadratic,*

$$\Pr_{x \sim \mathcal{D}} \{m_t(x) \le u\} \le Cu^2, \tag{123}$$

   and the positive loss jump when the selected experts change is at most a constant J. Then Eq. (9) holds with $\rho = 4JCG_r^2$.

Proof of Proposition D.3. Fix the history and a realized update h. The expectation below is over a fresh data variable ξ, not over the minibatch that produced h. Let

$$\Delta\ell(\xi, h) = \ell(\theta_t + h; \xi, S_{\theta_t + h}(x)) - \ell(\theta_t + h; \xi, S_{\theta_t}(x)). \tag{124}$$

The corresponding change in the expected loss at the updated parameter is $\mathbb{E}_\xi \Delta\ell(\xi, h)$. Since $x \mapsto x_+$ is convex,

$$[\mathbb{E}_\xi \Delta\ell(\xi, h)]_+ \le \mathbb{E}_\xi [\Delta\ell(\xi, h)]_+. \tag{125}$$

By Lemma D.4, the event that the selected experts change is contained in $\{m_t(x) \le 2G_r\|h\|\}$.

Under the first pair of conditions,

$$\begin{aligned}[\mathbb{E}_\xi \Delta\ell(\xi, h)]_+ &\le J\|h\| \Pr\{\text{selected experts change}\} \\ &\le J\|h\| \Pr\{m_t(x) \le 2G_r\|h\|\} \\ &\le J\|h\| C(2G_r\|h\|) = 2JCG_r\|h\|^2. \end{aligned} \tag{126}$$

Under the second pair,

$$\begin{aligned}[\mathbb{E}_\xi \Delta\ell(\xi, h)]_+ &\le J \Pr\{m_t(x) \le 2G_r\|h\|\} \\ &\le JC(2G_r\|h\|)^2 = 4JCG_r^2\|h\|^2. \end{aligned} \tag{127}$$

Both inequalities hold for every realized update. Taking expectation over the random update therefore gives Eq. (9) with the stated value of ρ. □

The sufficient conditions make clear where Assumption 4.3 comes from. Small parameter updates can change the selected experts only for inputs close to a routing boundary, and the loss change on those inputs must itself be controlled. These are properties of the model and the training trajectory, not consequences of replay. Replay already removes expert switching inside each ZO estimate exactly, as shown in Appendix D.1; the present argument is needed only to connect one iteration to the next.