ENGINE: Endogenous Variational MultiScale Optimization for Zeroth-Order LLM Fine-Tuning

Zhuoli Ouyang[1,2†] **Changxi Chi**[1,3] **Siyuan Li**[1,3] **Tailin Wu**[1‡]
[1]Department of Artificial Intelligence, School of Engineering, Westlake University
[2]Department of Electronic and Electrical Engineering,
Southern University of Science and Technology
[3]Zhejiang University
ouyangzl2023@mail.sustech.edu.cn wutailin@westlake.edu.cn

Abstract

Zeroth-order (ZO) optimization enables memory-efficient large language model (LLM) fine-tuning, and recent methods incorporate curvature-aware strategies. However, in the high-dimensional parameter space of LLMs, these methods inherently suffer from exploding gradient variance and prohibitive costs. Conversely, confining updates to a low-dimensional subspace reduces variance but discards critical high-dimensional information, degrading performance. This creates a fundamental dilemma: full-space optimization is informative but has high variance, while subspace optimization has low variance but loses information. To break this trade-off, we propose **ENGINE** (**En**do**g**enous Var**i**ation**n**al MultiScal**e**). Inspired by computational mechanics, we adapt static Variational MultiScale (VMS) theory to dynamic optimization, treating low- and high-dimensional spaces as resolved coarse and unresolved fine scales. Through a multigrid-style V-cycle, ENGINE alternates between spaces, dynamically probing fine-grained information in the high-dimensional space and injecting it to correct the efficient low-dimensional sampling process. Under appropriate assumptions, statistical analysis characterizes variance reduction and correction of curvature-induced bias, and convergence analysis establishes a stationarity bound for the corrected coarse updates. Evaluations on standard LLM benchmarks show that ENGINE achieves higher accuracy and faster convergence than existing ZO methods, achieving up to a $\mathbf{2.61}\times$ **speed-up in forward passes** with zero additional peak memory overhead. Code is available at https://github.com/AI4Science-WestlakeU/ENGINE.

1 Introduction

Large language model fine-tuning relies on first-order (FO) optimizers [Loshchilov et al., 2017]. However, FO training requires storing activations, gradients, and optimizer states, making it memory-intensive. Zeroth-order (ZO) optimization [Malladi et al., 2023] is therefore an appealing alternative: it uses only forward passes and reduces memory to near inference level. Still, in the full LLM parameter space, ZO methods suffer from high variance and high query cost [Nesterov and Spokoiny, 2017].

Table 1: Trade-offs between full-space and subspace ZO optimization. ✓ denotes a relative strength, and × a weakness.

Space	Effic.	Var.	Perf.	Curv.
Full space	×	×	✓	✓
Subspace	✓	✓	×	×
ENGINE (Ours)	✓	✓	✓	✓

[†]Work done as an intern at Westlake University.
[‡]Corresponding author.

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

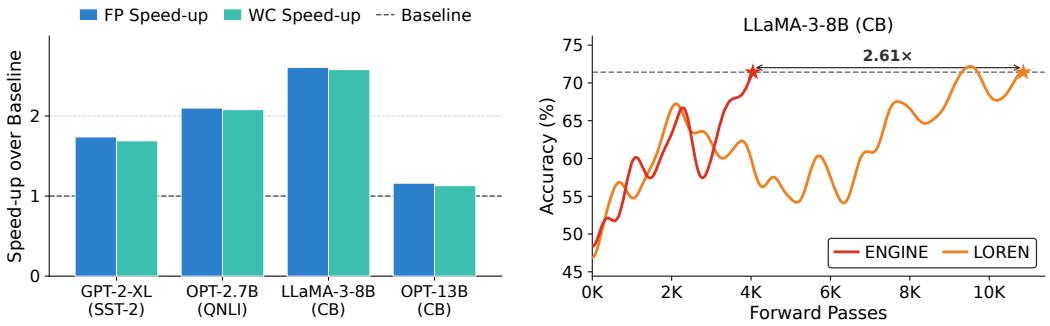


Figure 1: **Left:** Forward-pass and wall-clock speed-up of ENGINE over LOREN to reach LOREN's best accuracy, across four representative (model, task) settings. **Right:** Validation accuracy convergence on LLaMA-3-8B (CB).

LoRA [Hu et al., 2022] offers a lower-dimensional alternative by restricting updates to a low-rank subspace, improving efficiency and usually reducing variance. But this also discards fine-grained curvature information from the full space, which can slow convergence and degrade performance.

This leads to a central dilemma in zeroth-order optimization (Table 1): **full-space optimization is informative but high-variance, while subspace optimization is low-variance but blind.** Recent curvature-aware ZO methods, such as LOREN [Seung et al., 2026], reduce estimator variance through anisotropic perturbations [Kool et al., 2019]. Even so, existing methods usually stay in one space only: either the full parameter space or a low-rank subspace such as LoRA. They therefore gain only one side of the trade-off. Full-space methods retain richer information but are statistically costly, while subspace methods are more efficient but inevitably lose part of the curvature structure that is present in the full space. This motivates a simple question: **can we keep the efficiency of low-dimensional sampling while still using curvature information from the full space?** We approach this question through Variational MultiScale (VMS) theory [Hughes, 1995, Hughes et al., 1998] from computational mechanics. VMS separates a problem into a resolved coarse scale and an unresolved fine scale, then uses a correction term [Hughes et al., 2000] to account for the effect of the unresolved part. This viewpoint suggests a natural way to think about low-dimensional and high-dimensional spaces in zeroth-order optimization.

Based on this idea, we propose **ENGINE** (**En**dog**e**nous Vari**a**tio**n**al MultiScal**e**). We treat the low-dimensional LoRA parameterization as the resolved coarse scale and incorporate unresolved full-space curvature through a fine-scale correction. Unlike classical VMS, the optimization operator evolves with the optimization state, so the interaction between the two spaces must be updated dynamically. ENGINE addresses this with an endogenous V-cycle that alternates between the two spaces during training. Through a closed-form subgrid correction, it uses fine-scale information from the high-dimensional space to correct the efficient low-dimensional update.

Experiments on standard LLM optimization benchmarks show that ENGINE consistently improves over state-of-the-art zeroth-order baselines, including LOREN. It achieves better downstream accuracy, faster convergence, and up to a $2.61\times$ **speed-up in forward passes** (Figure 1), while keeping peak memory usage unchanged.

Our contributions are summarised as follows.

- **A cross-disciplinary perspective.** We introduce the idea of Variational MultiScale (VMS) from computational mechanics into zeroth-order LLM fine-tuning. This perspective clarifies why unresolved fine-scale curvature matters and provides a systematic way to describe the interaction between low-dimensional and high-dimensional optimization spaces.
- **The framework of ENGINE.** We propose ENGINE, an endogenous Variational MultiScale framework for zeroth-order optimization that combines low-rank sampling with a closed-form subgrid correction, thereby reintroducing unresolved full-space curvature into the coarse update. Under appropriate assumptions, we rigorously analyze the bias and variance of the corrected estimator. We also establish a stationarity bound for coarse updates.

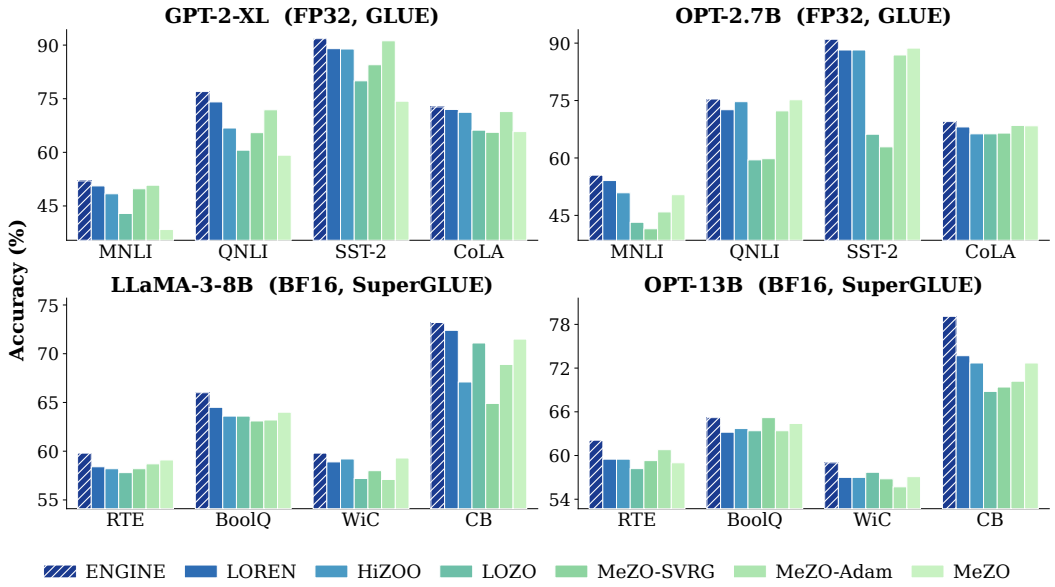


Figure 2: Test accuracy on all 16 (model, task) combinations. **Top row**: GPT-2-XL (GLUE, FP32) and OPT-2.7B (GLUE, FP32). **Bottom row**: LLaMA-3-8B (SuperGLUE, BF16) and OPT-13B (SuperGLUE, BF16). ENGINE is shown in **blue hatched**. Table 2 reports CoLA using MCC. Full numerical values are in Appendix E.1.

- **Consistent improvements.** Extensive experiments across diverse LLM fine-tuning settings demonstrate that ENGINE improves both downstream performance and convergence speed over state-of-the-art zeroth-order baselines (Figure 2). These gains come with up to a $2.61\times$ speed-up in forward passes and no additional peak memory overhead.

2 Preliminaries & Problem Setup

To address the single-scale limitation of ZO methods discussed in Sec. 1, ENGINE couples a low-dimensional LoRA scale with the high-dimensional full parameter space via Variational MultiScale (VMS). Sec. 2.1 first introduces the curvature-aware ZO framework adopted from LOREN, after which Sec. 2.2 recalls the Variational MultiScale (VMS) viewpoint underlying our method.

2.1 Zeroth-Order Setting & Notation Setup

Notation. We denote vectors by lowercase bold and matrices by uppercase bold, where $\otimes$ is the Kronecker product and $\|\cdot\|$ the Euclidean norm. For a weight matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, let $\mathbf{x} := \mathrm{vec}(\mathbf{X}) \in \mathbb{R}^d$ with $d = mn$ (multi-layer models are handled block-wise). We minimize $f(\mathbf{x}) := \mathbb{E}_{\xi\sim\mathcal{P}_{\mathrm{data}}}[\ell(\mathbf{x},\xi)]$ using only function-value queries, where $\mathcal{P}_{\mathrm{data}}$ is the data distribution.

Rank-1 Kronecker curvature. Full Hessian storage costs $O(m^2n^2)$ memory. K-FAC [Martens and Grosse, 2015] and Shampoo [Gupta et al., 2018] exploit matrix structure to reduce preconditioning costs. Following LOREN, we use the compact vector parameterization:

$$\widetilde{\mathbf{H}} \;=\; \mathbf{I}_m \otimes \big(\rho\mathbf{I}_n + \mathbf{v}\mathbf{v}^\top\big), \qquad \mathbf{v}\in\mathbb{R}^n,\ \rho>0, \tag{1}$$

where $\mathbf{v}\mathbf{v}^\top$ captures anisotropy shared across blocks, and damping ensures positive definiteness. Storing $\mathbf{v}$ costs $O(n)$. The inverse and inverse square root admit closed forms with $O(mn)$ application cost, avoiding dense matrices. Section 3 extends this approximation to multiple scales.

Curvature as covariance. For $\mathbf{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)$ and smoothing radius $\epsilon>0$, the symmetric finite-difference estimator [Spall, 2002] is $\widehat{\mathbf{g}}_\epsilon(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\epsilon\mathbf{u})-f(\mathbf{x}-\epsilon\mathbf{u})}{2\epsilon}\mathbf{u}$. Its expectation is the gradient of the Gaussian-smoothed loss. Generalized Gaussian smoothing extends this to anisotropic perturbations $\mathbf{u}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})$, and setting $\boldsymbol{\Sigma} = \widetilde{\mathbf{H}}^{-1}$ yields the NES identity [Wierstra et al., 2014]

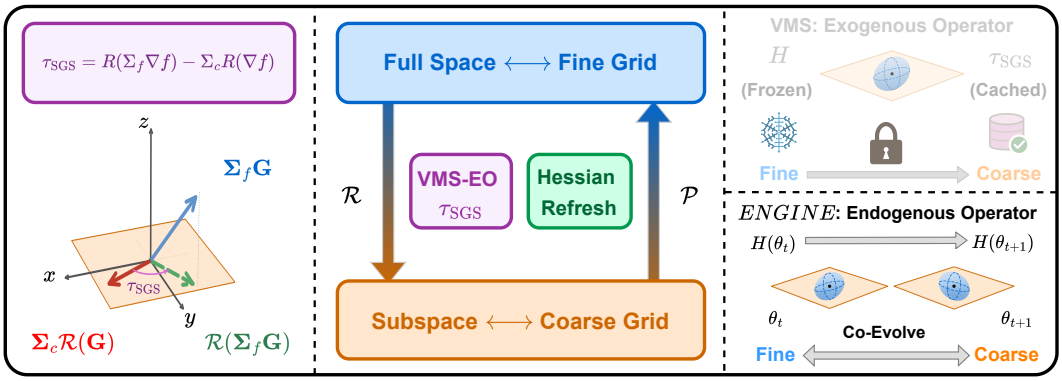


Figure 3: **ENGINE overview.** *Left*: the gap between the restricted fine direction $\mathcal{R}(\mathbf{\Sigma}_f \mathbf{G})$ and the coarse direction $\mathbf{\Sigma}_c \mathcal{R}(\mathbf{G})$ defines the subgrid correction $\boldsymbol{\tau}_{\text{sgs}}$. *Middle*: V-cycle alternates Coarse ($\boldsymbol{\phi}$) and Fine ($\mathbf{W}$) phases, with every S steps a fine probe refreshing $\mathbf{v}_f \leftarrow \mathbf{v}_f - \nu g(\mathbf{v}_f)$ and λ (Hessian refresh), while VMS-EO ($\boldsymbol{\tau}_{\text{sgs}}$) injects the subgrid correction into each coarse step. *Right*: classical VMS uses a fixed operator $\mathcal{L}$, while ENGINE uses the dynamic Hessian $\mathbf{H}(\boldsymbol{\theta})$.

$\widetilde{\mathbf{H}}^{-1} \nabla f(\mathbf{x}) = \mathbb{E}_{\tilde{\mathbf{u}} \sim \mathcal{N}(\mathbf{0}, \widetilde{\mathbf{H}}^{-1})} \big[\langle \nabla f(\mathbf{x}), \tilde{\mathbf{u}} \rangle \, \tilde{\mathbf{u}} \big]$, so preconditioning by $\widetilde{\mathbf{H}}^{-1}$ corresponds to larger ZO perturbations along directions of lower estimated curvature.

2.2 Variational MultiScale Method

The Variational MultiScale (VMS) method decomposes problems into resolved (coarse) and unresolved (fine) scales. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, $\mathcal{V}$ a function space over Ω with dual $\mathcal{V}^*$, and $\mathcal{L} : \mathcal{V} \to \mathcal{V}^*$ a linear operator, where the symmetric bilinear form $a(w, u) := (w, \mathcal{L}u)$ uses the $L^2(\Omega)$ inner product $(w, u) := \int_\Omega w(x)\, u(x)\, \mathrm{d}x$. The weak problem seeks $u \in \mathcal{V}$ such that $a(w, u) = (w, f)$ for all test functions $w \in \mathcal{V}$. VMS decomposes $\mathcal{V} = \bar{\mathcal{V}} \oplus \mathcal{V}'$ into coarse $\bar{\mathcal{V}}$ and fine $\mathcal{V}'$ subspaces, so that $u = \bar{u} + u'$ and $w = \bar{w} + w'$ with $\bar{u}, \bar{w} \in \bar{\mathcal{V}}$ and $u', w' \in \mathcal{V}'$. The fine scale is driven by the coarse residual $\mathcal{L}u' = -(\mathcal{L}\bar{u} - f)$. Following the sign convention of Hughes [1995], the fine-scale solution operator $M : \mathcal{V}^* \to \mathcal{V}'$ includes the negative sign, so $u' = M(\mathcal{L}\bar{u} - f)$. Substitution yields the closed coarse equation:

$$a(\bar{w}, \bar{u}) \; + \; \big(\mathcal{L}^* \bar{w}, \; M(\mathcal{L}\bar{u} - f)\big) \; = \; (\bar{w}, f), \tag{2}$$

where $\mathcal{L}^* : \mathcal{V} \to \mathcal{V}^*$ is the adjoint of $\mathcal{L}$, and the second term in (2) is the *subgrid correction* encoding unresolved-scale feedback. In our setting, $\bar{u}$, u', and $\mathcal{L}$ correspond to the LoRA-space update, unresolved full-space curvature, and endogenous Hessian operator $\widetilde{\mathbf{H}}$, respectively. The subgrid correction corresponds to the VMS-EO fine-scale curvature injection developed in Sec. 3.

3 Method

Building on the rank-1 curvature primitive $(\widetilde{\mathbf{H}}, \mathbf{v})$ from Eq. (1) and the VMS formulation coupling coarse and fine scales from Eq. (2), this section develops and analyzes ENGINE (Figure 3). §3.1 contrasts the *rigid* linear operator $\mathcal{L}$ of classical VMS with the *dynamic* Hessian $\mathbf{H}(\boldsymbol{\theta})$ of optimization, motivating why classical VMS cannot be ported verbatim. §3.2 derives the central correction $\boldsymbol{\tau}_{\text{sgs}}$ and its rank-1 LoRA-branch surrogate. §3.3 embeds VMS-EO into a V-cycle so that the endogenous state $\mathbf{v}_f$ is refreshed across phases. §3.4 then quantifies how coarse sampling and exact subgrid correction affect variance and bias relative to a common target.

The multiscale state is carried by three rank-1 curvature vectors: the full-space $\mathbf{v}_f \in \mathbb{R}^n$ and the two LoRA-branch curvatures $\mathbf{v}_\mathbf{B} \in \mathbb{R}^r$, $\mathbf{v}_\mathbf{A} \in \mathbb{R}^n$. We instantiate the resolved coarse scale by a rank-r LoRA factorization $\Delta\mathbf{W} = \mathbf{BA}$ with $\mathbf{B} \in \mathbb{R}^{m \times r}$, $\mathbf{A} \in \mathbb{R}^{r \times n}$ and $r \ll \min(m, n)$, so the coarse parameter $\boldsymbol{\phi} = (\mathbf{B}, \mathbf{A})$ has dimension $d_c = r(m + n)$, much smaller than $d_f = mn$. Fine-scale information is obtained from the full parameter space and used to correct curvature effects not captured by the coarse model.

3.1 Endogenous Operator: Why Classical VMS Fails?

Classical VMS starts from a linear problem $\mathcal{L}u = f$ with operator $\mathcal{L}$ fixed by the underlying physics, which is what allows the unresolved component u' to be eliminated analytically in closed form (Figure 3 Right, top). In optimization the analogous linear equation is the Newton step $\mathbf{H}(\boldsymbol{\theta})\Delta\boldsymbol{\theta} = -\nabla f(\boldsymbol{\theta})$, giving the correspondence $(\mathcal{L}, u, f) \leftrightarrow (\mathbf{H}, \Delta\boldsymbol{\theta}, -\nabla f)$. Unlike $\mathcal{L}$, however, $\mathbf{H}(\boldsymbol{\theta})$ co-evolves with the iterate (Figure 3 Right, bottom), so the closure cannot be derived once and for all but must be refreshed periodically by the V-cycle of §3.3. We call this state-dependent operator an endogenous operator and the resulting subgrid correction VMS-EO.

Coarse increments are represented by changes in the LoRA factors. The *prolongation* operator $\mathcal{P} : (\Delta\mathbf{B}, \Delta\mathbf{A}) \longmapsto \mathbf{B}\Delta\mathbf{A} + \Delta\mathbf{B}\mathbf{A} \in \mathbb{R}^{m\times n}$ lifts a coarse update to the full space (the differential of the LoRA product), and the *restriction* operator $\mathcal{R} = (\mathcal{R}_{\mathbf{B}}, \mathcal{R}_{\mathbf{A}})$ with $\mathcal{R}_{\mathbf{B}}(\mathbf{X}) = \mathbf{X}\mathbf{A}^\top \in \mathbb{R}^{m\times r}$ and $\mathcal{R}_{\mathbf{A}}(\mathbf{X}) = \mathbf{B}^\top\mathbf{X} \in \mathbb{R}^{r\times n}$ maps a full-space matrix $\mathbf{X}$ to the two LoRA gradient blocks. The two operators are formal adjoints under the Frobenius inner product, $\mathcal{R} = \mathcal{P}^*$, i.e., $\langle \mathcal{P}(\Delta\mathbf{B}, \Delta\mathbf{A}), \mathbf{X}\rangle_F = \langle (\Delta\mathbf{B}, \Delta\mathbf{A}), \mathcal{R}(\mathbf{X})\rangle_F$ for all $\mathbf{X} \in \mathbb{R}^{m\times n}$.

At LLM scale, $\mathbf{H} \in \mathbb{R}^{d_f\times d_f}$ cannot be formed, so ENGINE inherits LOREN's rank-1 Kronecker $\widetilde{\mathbf{H}} = \mathbf{I}_m \otimes (\rho\mathbf{I}_n + \mathbf{v}\mathbf{v}^\top)$ as the practical operator with preconditioner $\boldsymbol{\Sigma} = \widetilde{\mathbf{H}}^{-1}$. Given parameter $\mathbf{x}$ and curvature $\mathbf{v}$, LOREN draws K anisotropic perturbations $z_k = \mathbf{x} + \varepsilon\boldsymbol{\Sigma}^{1/2}u_k$ where $u_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to simultaneously form RLOO estimators for both parameter and curvature gradients:

$$g(\mathbf{x}) \;=\; \frac{1}{\varepsilon(K-1)}\sum_{k=1}^{K}(f_k - \bar{f})\,\boldsymbol{\Sigma}^{1/2}u_k, \qquad g(\mathbf{v}) \;=\; \frac{1}{K-1}\sum_{k=1}^{K}(f_k - \bar{f})\,\nabla_{\mathbf{v}}\log p_{\boldsymbol{\theta}}(z_k), \quad (3)$$

where $f_k := f(z_k)$, $\bar{f} := \frac{1}{K}\sum_j f_j$, and the closed form of the score function $\nabla_{\mathbf{v}}\log p_{\boldsymbol{\theta}}(z)$ is given in LOREN Eq. (10). The two estimators *share the same batch* $\{u_k, f_k\}$, so each step costs only K forward passes. ENGINE samples in LoRA coordinates with (3) to avoid d_f-scale variance, while injecting full-space curvature feedback as a subgrid correction back into the coarse phase.

3.2 VMS-EO Subgrid Correction

To construct the correction in Figure 3 (Left), we first identify where coarse and fine directions differ. Lemma 3.1 establishes exact restriction of first-order gradients. Proposition 3.2 then gives the curvature mismatch introduced by preconditioning. This residual supplies the corrected LoRA updates in Eqs. (7) and (8).

Closed-Form Curvature Correction. Let $\mathbf{G} \in \mathbb{R}^{m\times n}$ denote a full-space gradient-like matrix. Deterministically $\mathbf{G} = \nabla_{\mathbf{W}}f$, while in the ZO implementation it is replaced by the RLOO estimator.

Lemma 3.1 (chain rule)**.** *LoRA restriction commutes with first-order differentiation:* $\mathcal{R}(\nabla_{\mathbf{W}}f) = \nabla_{(\mathbf{B},\mathbf{A})}f$. *Proof in Appendix B.1.*

Lemma 3.1 shows that restricting the exact fine-space gradient gives the coarse-space gradient. Curvature preconditioning introduces a mismatch between the two directions. Let $\boldsymbol{\Sigma}_f, \boldsymbol{\Sigma}_{\mathbf{B}}, \boldsymbol{\Sigma}_{\mathbf{A}}$ be the rank-1 Kronecker covariances built from $\mathbf{v}_f, \mathbf{v}_{\mathbf{B}}, \mathbf{v}_{\mathbf{A}}$ (Sherman-Morrison forms in Appendix B.2), and $\boldsymbol{\Sigma}_c := \mathrm{diag}(\boldsymbol{\Sigma}_{\mathbf{B}}, \boldsymbol{\Sigma}_{\mathbf{A}})$. In general $\mathcal{R}(\boldsymbol{\Sigma}_f\mathbf{G}) \neq \boldsymbol{\Sigma}_c\mathcal{R}(\mathbf{G})$, so the two preconditioned directions can differ in both magnitude and orientation. This gap defines the *subgrid residual*:

$$\boldsymbol{\tau}_{\mathrm{sgs}} \;:=\; \mathcal{R}(\boldsymbol{\Sigma}_f\mathbf{G}) - \boldsymbol{\Sigma}_c\mathcal{R}(\mathbf{G}), \quad (4)$$

with branch components $\boldsymbol{\tau}_{\mathrm{sgs},\mathbf{B}} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{\tau}_{\mathrm{sgs},\mathbf{A}} \in \mathbb{R}^{r\times n}$. The following result expresses each branch residual through the rank-1 curvature vectors.

Proposition 3.2 (closed-form subgrid residual)**.** *Under the rank-1 surrogates above with* $\rho > 0$*, the branch residuals are:*

$$\boldsymbol{\tau}_{\mathrm{sgs},\mathbf{B}} \;=\; \frac{1}{\rho}\left[\frac{\mathbf{G}\mathbf{A}^\top\mathbf{v}_{\mathbf{B}}\,\mathbf{v}_{\mathbf{B}}^\top}{\rho + \|\mathbf{v}_{\mathbf{B}}\|^2} - \frac{\mathbf{G}\mathbf{v}_f\,(\mathbf{A}\mathbf{v}_f)^\top}{\rho + \|\mathbf{v}_f\|^2}\right], \quad (5)$$

$$\boldsymbol{\tau}_{\mathrm{sgs},\mathbf{A}} \;=\; \frac{1}{\rho}\left[\frac{\mathbf{B}^\top\mathbf{G}\mathbf{v}_{\mathbf{A}}\,\mathbf{v}_{\mathbf{A}}^\top}{\rho + \|\mathbf{v}_{\mathbf{A}}\|^2} - \frac{\mathbf{B}^\top\mathbf{G}\mathbf{v}_f\,\mathbf{v}_f^\top}{\rho + \|\mathbf{v}_f\|^2}\right]. \quad (6)$$

Proof in Appendix B.2.

Lemma 3.1 and Proposition 3.2 together identify $\boldsymbol{\tau}_{\text{sgs}}$ as the second-order curvature gap that VMS-EO must inject. We now describe the corrected coarse update.

Rank-1 Surrogate for LoRA Branches. Let $\mathbf{s}_c := \boldsymbol{\Sigma}_c \mathcal{R}(\mathbf{G})$ be the pure coarse preconditioned direction, which equals the LoRA-space LOREN direction without any VMS-EO correction. VMS-EO simply adds a branch-weighted subgrid correction $\mathbf{s}_c + \lambda\, \boldsymbol{\tau}_{\text{sgs}}$, where $\lambda = 0$ recovers LoRA-space LOREN and a larger λ injects more fine-grid curvature feedback. The two branches carry independent weights $\lambda_{\mathbf{B}}, \lambda_{\mathbf{A}} \in [0, 1]$.

VMS-EO corrected updates. Let $\widehat{\mathbf{g}}_{c,\mathbf{B}} \in \mathbb{R}^{m\times r}$ and $\widehat{\mathbf{g}}_{c,\mathbf{A}} \in \mathbb{R}^{r\times n}$ denote the preconditioned RLOO directions for the two LoRA branches from Eq. (3). With branch step sizes $\eta_{\mathbf{B}}, \eta_{\mathbf{A}} > 0$, the corrected updates are:

$$\Delta\mathbf{B} \;=\; -\eta_{\mathbf{B}}\big(\widehat{\mathbf{g}}_{c,\mathbf{B}} + \lambda_{\mathbf{B}}\, \boldsymbol{\tau}_{\text{sgs},\mathbf{B}}\big), \tag{7}$$

$$\Delta\mathbf{A} \;=\; -\eta_{\mathbf{A}}\big(\widehat{\mathbf{g}}_{c,\mathbf{A}} + \lambda_{\mathbf{A}}\, \boldsymbol{\tau}_{\text{sgs},\mathbf{A}}\big). \tag{8}$$

Implementation details are given in Appendix D.1.

Adaptive mixing weights. Proposition C.7 characterizes an oracle mixing weight that minimizes the estimation MSE. Since it depends on unknown correction and error statistics, we use normalized angular disagreement between coarse and fine curvature directions as a practical proxy for selecting $\lambda_{\mathbf{B}}, \lambda_{\mathbf{A}} \in [0, 1]$. For the B-branch we use the formula below, and the A-branch is analogous after replacing $\mathbf{v}_{\mathbf{B}}$ with $\mathbf{v}_{\mathbf{A}}$ and the projection $\mathbf{A}\mathbf{v}_f$ with $\mathbf{v}_f$:

$$\delta_{\mathbf{B}} := \frac{1}{2}\left\| \frac{\mathbf{v}_{\mathbf{B}}}{\|\mathbf{v}_{\mathbf{B}}\|} - \frac{\mathbf{A}\mathbf{v}_f}{\|\mathbf{A}\mathbf{v}_f\|} \right\|, \qquad \lambda_{\mathbf{B}} = \text{clip}(\delta_{\mathbf{B}}, \lambda_{\min}, \lambda_{\max}). \tag{9}$$

When δ approaches zero the two curvature directions become aligned, so λ falls to $\lambda_{\min}$ and the correction receives its minimum mixing weight. When δ approaches one the directions diverge, so λ rises to $\lambda_{\max}$ and injects more fine-grid feedback.

3.3 ENGINE: V-cycle Coupling

ENGINE alternates between M-epoch coarse ($\boldsymbol{\phi}$) and N-epoch fine ($\mathbf{W}$) phases (Figure 3 Middle), updating the base $\mathbf{W}_0$ in $\mathbf{W} = \mathbf{W}_0 + \mathbf{B}\mathbf{A}$ only at phase boundaries. A cyclic schedule couples them, using $\mathcal{P}$ and $\mathcal{R}$ operators to transfer parameters, momentum, and curvature.

Two-phase schedule. *Coarse phase.* ENGINE updates $\boldsymbol{\phi} = (\mathbf{B}, \mathbf{A})$ via Eqs. (7) and (8) and uses (3) to take one curvature step per branch with learning rate $\nu > 0$ (Algorithm 1, line 6). Crucially, the K RLOO samples $\{u_k, f_k\}$ are reused from those already drawn for $(\Delta\mathbf{B}, \Delta\mathbf{A})$, so this curvature step costs no additional forward passes. The operator $\widetilde{\mathbf{H}} = \widetilde{\mathbf{H}}(\boldsymbol{\theta}, \mathbf{v}_f)$ is endogenous, however, so $\mathbf{v}_f$ goes stale across coarse steps and causes $\boldsymbol{\tau}_{\text{sgs}}$ to inject outdated information through its $\mathbf{v}_f$ dependence. We therefore trigger a Hessian refresh every S coarse steps as illustrated in Algorithm 1, where K extra full-space perturbations update $\mathbf{v}_f$ using (3) with learning rate ν together with the projection scalars in Proposition 3.2 while $\boldsymbol{\theta}$, $\mathbf{B}$, and $\mathbf{A}$ remain unchanged. This refresh-and-reuse strategy parallels Lazy Hessian [Doikov et al., 2023], which also recomputes curvature every S steps and reuses it in between to cut arithmetic cost without sacrificing convergence.

Fine phase. ENGINE returns to the full parameter space and applies $\boldsymbol{\Sigma}_f$-preconditioned LOREN steps from Eq. (3) to $\mathbf{W}$, simultaneously refreshing $\mathbf{v}_f$ for the next coarse phase.

Prolongation $\mathcal{P}$ from coarse to fine. The coarse update $\mathbf{B}\mathbf{A}$ is absorbed into the base weight $\mathbf{W}_0$. We save the current $\mathbf{A}$ as $\mathbf{A}_{\text{old}}$ before reinitialization. This cached matrix maps vectors from $\mathbb{R}^n$ to $\mathbb{R}^r$ during the next transition from fine to coarse. Branch momenta combine into $\mathbf{m}_f$ via the LoRA product rule, and curvature is EMA-blended with weight β. Algorithm 1 specifies the transfers.

Restriction $\mathcal{R}$ from fine to coarse. We set the base weight $\mathbf{W}_0$ to $\mathbf{W}_{\text{fine_best}}$, rescale $\mathbf{B}$, and reset $\mathbf{A}$ to zero so that $\mathbf{W}_0 + \mathbf{B}\mathbf{A} = \mathbf{W}_0$ holds exactly at cycle start. Momentum is projected onto the two branches via $\mathcal{R}$ with default damping factor $\alpha = 0.5$, and the two branch curvatures use the cached basis $\mathbf{A}_{\text{old}}$. Algorithm 1 gives these transfer formulas.

Algorithm 1 ENGINE (see Appendix D for details).

Require: frozen $\mathbf{W}$, dataset $\mathcal{D}$, rank r, cycles V, coarse/fine epochs M, N, queries K, smoothing ϵ, damping ρ, learning rates $\eta_{\mathbf{B}}, \eta_{\mathbf{A}}, \eta_f, \nu$, mixing range $[\lambda_{\min}, \lambda_{\max}]$, transfer weight β, fine-probe interval S (every S coarse steps, refresh $\mathbf{v}_f$ and λ).

Initialize $\mathbf{B}$ randomly, set $\mathbf{A} \leftarrow \mathbf{0}$, and initialize $\mathbf{v}_{\mathbf{B}}, \mathbf{v}_{\mathbf{A}}, \mathbf{v}_f$.
for $v = 1, \ldots, V$ **do**
  Coarse phase (M epochs in $\phi = (\mathbf{B}, \mathbf{A})$): ▷ ■ Coarse space
  for $t = 1, \ldots, M \cdot$steps/epoch **do**
    Draw K LoRA perturbations and estimate $\widehat{\mathbf{g}}_{c,\mathbf{B}}, \widehat{\mathbf{g}}_{c,\mathbf{A}}$ via (3).
    $\mathbf{v}_{\mathbf{B}} \leftarrow \mathbf{v}_{\mathbf{B}} - \nu g(\mathbf{v}_{\mathbf{B}})$, $\mathbf{v}_{\mathbf{A}} \leftarrow \mathbf{v}_{\mathbf{A}} - \nu g(\mathbf{v}_{\mathbf{A}})$ on the same K samples.
    if $t \bmod S = 0$ **then**
      $\mathbf{v}_f \leftarrow \mathbf{v}_f - \nu g(\mathbf{v}_f)$ via (3). ▷ ■ Hessian refresh, K extra fwd
      Update $\lambda_{\mathbf{B}}, \lambda_{\mathbf{A}}$ via (9).
    end if
    Compute $\boldsymbol{\tau}_{\text{sgs}}$ via Prop. 3.2 (Eqs. (5) and (6)). ▷ ■ VMS-EO
    Update $(\mathbf{B}, \mathbf{A})$ via Eqs. (7) and (8).
  end for
  Prolongation $\mathcal{P}$ ($\mathbf{C} \to \mathbf{F}$): $\mathbf{W}_0 \leftarrow \mathbf{W}_0 + \mathbf{BA}$, $\mathbf{A}_{\text{old}} \leftarrow \mathbf{A}$, $\mathbf{m}_f = \mathbf{m}_{\mathbf{B}}\mathbf{A} + \mathbf{B}\mathbf{m}_{\mathbf{A}}$, $\mathbf{v}_f \leftarrow \text{normalize}((1-\beta)\mathbf{v}_f + \beta\,\text{normalize}(\mathbf{A}^\top \mathbf{v}_{\mathbf{B}} + \mathbf{v}_{\mathbf{A}}))$.
  Fine phase (N epochs in $\mathbf{W}$): LOREN on $\mathbf{W}$ with lr η_f, K queries via (3), updating $\mathbf{v}_f$. ▷ ■ Fine space
  Restriction $\mathcal{R}$ ($\mathbf{F} \to \mathbf{C}$): $\mathbf{W}_0 \leftarrow \mathbf{W}_{\text{fine_best}}$, rescale $\mathbf{B}$, $\mathbf{A} \leftarrow \mathbf{0}$, $\mathbf{m}_{\mathbf{A}} = \alpha \mathbf{B}^\top \mathbf{m}_f$, $\mathbf{m}_{\mathbf{B}} = \alpha \mathbf{m}_f \mathbf{A}_{\text{old}}^\top$, $\mathbf{v}_{\mathbf{A}} = \text{normalize}(\mathbf{v}_f)$, $\mathbf{v}_{\mathbf{B}} = \text{normalize}(\mathbf{A}_{\text{old}} \mathbf{v}_f)$.
end for
return $\mathbf{W}$.

ENGINE maintains only the LoRA factors $(\mathbf{B}, \mathbf{A})$ and the rank-1 curvature vectors $(\mathbf{v}_f, \mathbf{v}_{\mathbf{B}}, \mathbf{v}_{\mathbf{A}})$, never explicitly storing a full Hessian. Algorithm 1 summarizes the procedure, and implementation details are in Appendix D.

3.4 Statistical and Convergence Analysis

We provide a detailed analysis of the trade-off between bias and variance in ENGINE. Under appropriate assumptions, coarse sampling reduces the dimension-dependent variance bound and exact correction eliminates structural bias in the local-linear limit. We also establish convergence bounds for coarse updates.

Fix the current weights, LoRA factors, and covariances. Subscripts f and c denote fine and coarse spaces. In row-wise vector coordinates, let $\mathbf{g}_f^{\text{raw}}$ be the full-space gradient and $\mathbf{g}_c^{\text{raw}} = \mathcal{R}(\mathbf{g}_f^{\text{raw}})$ the gradient with respect to ϕ. The coarse direction is $\mathbf{s}_c = \boldsymbol{\Sigma}_c \mathbf{g}_c^{\text{raw}}$, and the common comparison target is $\mathbf{g}_\star = \mathcal{R}(\boldsymbol{\Sigma}_f \mathbf{g}_f^{\text{raw}})$. Thus Eq. (4) gives $\boldsymbol{\tau}_{\text{sgs}} = \mathbf{g}_\star - \mathbf{s}_c$.

We use a Gaussian estimator with b independent directions and smoothing radius ϵ. Write $F(\phi) = f(\mathbf{W}_0 + \mathbf{BA})$ and draw $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_c)$ independently. Define:

$$\widehat{\mathbf{g}}_{c,\epsilon,b} = \frac{1}{b} \sum_{i=1}^{b} \frac{F(\phi + \epsilon \mathbf{z}_i) - F(\phi)}{\epsilon} \mathbf{z}_i. \tag{10}$$

Here $\mathbf{z}_i = [\mathbf{z}_{\mathbf{B},i}^\top, \mathbf{z}_{\mathbf{A},i}^\top]^\top \in \mathbb{R}^{d_c}$ stacks B- and A-branch perturbations of dimensions mr and rn, with $d_c = r(m+n)$. The covariance $\boldsymbol{\Sigma}_c = \text{diag}(\boldsymbol{\Sigma}_{\mathbf{B}}, \boldsymbol{\Sigma}_{\mathbf{A}}) \in \mathbb{R}^{d_c \times d_c}$ has blocks of sizes $mr \times mr$ and $rn \times rn$, respectively. Implementation uses RLOO from Eq. (3). The next result isolates the effect of exact correction in the local-linear limit, where the smoothing bias vanishes.

Proposition 3.3 (Common-target error with exact correction)**.** *Suppose F has a Lipschitz continuous gradient and $\boldsymbol{\Sigma}_c$ is fixed and positive definite. As $\epsilon \to 0$ with b fixed, Eq.* (10) *converges in mean square to $\widehat{\mathbf{g}}_{c,0,b} = b^{-1} \sum_i \mathbf{z}_i \mathbf{z}_i^\top \mathbf{g}_c^{\text{raw}}$. Its mean is $\mathbf{s}_c$ and its trace variance is:*

$$V_{c,0} = \frac{\text{tr}(\boldsymbol{\Sigma}_c)(\mathbf{g}_c^{\text{raw}})^\top \boldsymbol{\Sigma}_c \mathbf{g}_c^{\text{raw}} + (\mathbf{g}_c^{\text{raw}})^\top \boldsymbol{\Sigma}_c^2 \mathbf{g}_c^{\text{raw}}}{b}. \tag{11}$$

With exact deterministic correction and both branch weights equal to one, the common-target errors satisfy:

$$\mathbb{E}\|\widehat{\mathbf{g}}_{c,0,b} - \mathbf{g}_\star\|^2 = V_{c,0} + \|\boldsymbol{\tau}_{\mathrm{sgs}}\|^2, \tag{12}$$

$$\mathbb{E}\|\widehat{\mathbf{g}}_{c,0,b} + \boldsymbol{\tau}_{\mathrm{sgs}} - \mathbf{g}_\star\|^2 = V_{c,0}. \tag{13}$$

Theorem C.3 proves the stated mean-square limit of the Gaussian estimator $\widehat{\mathbf{g}}_{c,\epsilon,b}$. Lemma C.2 gives Eq. (11)*, and Proposition C.4 yields Eqs.* (12) *and* (13) *in the limit. Proof in Appendix C.3.3.*

Variance reduction and bias correction. For isotropic sampling with $\boldsymbol{\Sigma}_c = I_{d_c}$, Eq. (11) reduces to $V_{c,0} = \frac{d_c+1}{b}\|\mathbf{g}_c^{\mathrm{raw}}\|^2$, compared with the fine-space factor $\frac{d_f+1}{b}$. For anisotropic sampling with fixed positive-definite $\boldsymbol{\Sigma}_c$ and $\|\boldsymbol{\Sigma}_c\|_{\mathrm{op}} \le M_c$, the same equation gives the upper bound $V_{c,0} \le \frac{d_c+1}{b}M_c^2\|\mathbf{g}_c^{\mathrm{raw}}\|^2$. Under comparable gradient and spectral scales, $d_c \ll d_f$ reduces the dimension-dependent variance bound. In the local-linear limit, Proposition C.4 shows that exact correction eliminates the structural bias relative to $\mathbf{g}_\star$ while preserving the coarse sampling variance, decreasing MSE by $\|\boldsymbol{\tau}_{\mathrm{sgs}}\|^2$. Ideal ENGINE thus combines coarse-space variance reduction with structural-bias elimination. Appendix C.3 gives finite-radius bounds and a common-target comparison in Table 4.

For finite smoothing radius ϵ and L_c-smooth F, Theorem C.3 in Appendix C.2 gives $\|\mathbf{b}_{c,\epsilon}\| = \mathcal{O}(\epsilon\sqrt{d_c})$ with L_c and M_c held fixed. Here $\mathbf{b}_{c,\epsilon} = \mathbb{E}\widehat{\mathbf{g}}_{c,\epsilon,b} - \mathbf{s}_c$ is the residual smoothing bias. Proposition C.4 gives the MSE reduction $\|\boldsymbol{\tau}_{\mathrm{sgs}}\|^2 - 2\langle \mathbf{b}_{c,\epsilon}, \boldsymbol{\tau}_{\mathrm{sgs}}\rangle$, quantifying the effect of this remaining bias. For approximate corrections, Proposition C.5 accounts for their bias, variance, and dependence on the coarse estimate. Proposition C.6 in Appendix C.4 separates row compression error from scalar coefficient mismatch.

Convergence analysis. Consider the ideal Gaussian update $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t - \eta(\widehat{\mathbf{g}}_{c,\epsilon,b,t} + \boldsymbol{\tau}_{\mathrm{sgs},t})$ on a fixed coarse objective F. Under the assumptions in Appendix C.6, Theorem C.9 gives the following bound for the prescribed $\eta = \Theta(T^{-1/2})$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\boldsymbol{\phi}_t)\|^2 \le O(T^{-1/2}) + O(\epsilon^2). \tag{14}$$

Here T counts iterations. The bound separates optimization decay from the smoothing residual, with coarse-dimensional sampling constants. Corollary C.10 extends the bound to approximate corrections, with correction bias affecting the residual and correction variance affecting the sampling constants.

4 Experiments

We evaluate ENGINE on 16 (model, task) combinations, assessing accuracy, efficiency, and the contributions of V-cycle and VMS-EO.

4.1 Experimental Setup and Main Results

Setup. We evaluate four autoregressive LLMs of size 1.5B to 13B, namely GPT-2-XL [Radford et al., 2022], OPT-2.7B and OPT-13B [Zhang et al., 2022], and LLaMA-3-8B [Grattafiori et al., 2024], on 16 prompt-free classification tasks drawn from GLUE [Wang et al., 2018] and SuperGLUE [Wang et al., 2019]. We compare ENGINE against LOREN together with five other ZO baselines, whose full list and background appear in Appendix A. All baselines are independently reproduced under LOREN's setup. On top of this protocol, ENGINE introduces only two new hyperparameters. The C60F60 V-cycle alternates 60 coarse epochs with 60 fine epochs, and the curvature refresh interval S is set per (model, task) combination as listed in Table 8. The LoRA rank r scales with model size. We report five-seed means using Matthews correlation coefficient (MCC) scaled by 100 for CoLA and accuracy (%) for the other tasks. See Appendices E and F for details.

Main results. Table 2 and Figure 2 show that ENGINE outperforms LOREN and other baseline methods on every (model, task) cell. Representative training curves appear in Figure 4.

Efficiency and memory. Figure 1 further shows that on every setting, ENGINE reaches LOREN's best accuracy with substantially fewer forward passes and less wall-clock time, achieving up to a

Table 2: Main results using MCC $\times 100$ for CoLA and accuracy (%) for other tasks. **Bold** and **cyan** mark the best and second best per column. Highlighted rows give ENGINE and its absolute gain over LOREN. Full results in Appendix E.1.

GLUE (FP32)	**GPT-2-XL** (1.5B)					**OPT-2.7B**				
Method	MNLI	QNLI	SST-2	CoLA	Avg.	MNLI	QNLI	SST-2	CoLA	Avg.
MeZO	38.4	59.2	74.3	20.1	48.0	50.4	**75.2**	**88.7**	19.8	58.5
MeZO-Adam	**50.8**	71.9	**91.2**	31.6	61.4	45.9	72.3	86.9	28.3	58.4
MeZO-SVRG	49.8	65.5	84.5	25.6	56.4	41.5	59.8	62.9	26.1	47.6
LOZO	42.9	60.6	80.0	18.1	50.4	43.2	59.5	66.2	16.9	46.5
HiZOO	48.4	66.8	88.9	27.9	58.0	50.9	74.7	88.2	22.3	59.0
LOREN	50.6	**74.1**	89.0	**32.9**	**61.7**	**54.1**	72.6	88.2	**28.6**	**60.9**
ENGINE (Ours)	**52.0**	**77.0**	**91.8**	**34.8**	**63.9**	**55.5**	**75.4**	**91.0**	**31.0**	**63.2**
Δ vs LOREN	+1.4	+2.9	+2.8	+1.9	+2.3	+1.4	+2.8	+2.8	+2.4	+2.4
SuperGLUE (BF16)	**LLaMA-3-8B**					**OPT-13B**				
Method	RTE	BoolQ	WiC	CB	Avg.	RTE	BoolQ	WiC	CB	Avg.
MeZO	**59.1**	64.0	**59.3**	71.5	63.5	59.0	64.4	57.1	72.7	63.3
MeZO-Adam	58.7	63.2	57.1	68.9	62.0	**60.8**	63.4	55.7	70.2	62.2
MeZO-SVRG	58.2	63.1	58.0	64.9	61.1	59.3	**65.2**	56.8	69.4	62.7
LOZO	57.8	63.6	57.2	71.1	62.4	58.2	63.4	**57.7**	68.8	62.0
HiZOO	58.2	63.6	59.2	67.1	62.0	59.5	63.7	57.0	72.7	63.2
LOREN	58.4	**64.5**	58.9	**72.4**	**63.6**	59.5	63.2	57.0	**73.7**	**63.4**
ENGINE (Ours)	**59.8**	**66.0**	**59.8**	**73.2**	**64.7**	**62.1**	**65.2**	**59.0**	**79.1**	**66.4**
Δ vs LOREN	+1.4	+1.5	+0.9	+0.8	+1.1	+2.6	+2.0	+2.0	+5.4	+3.0

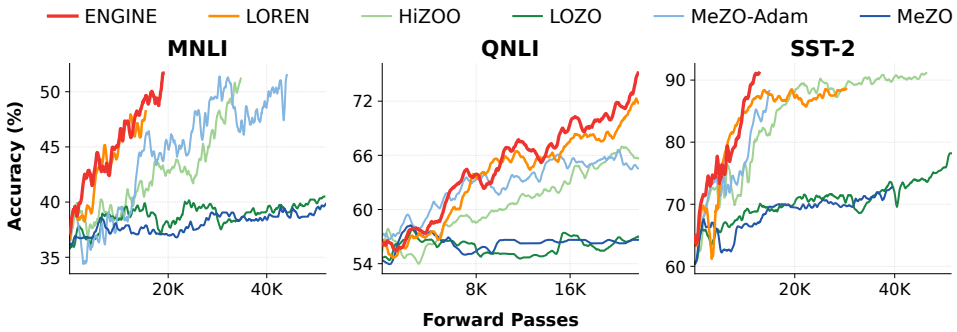


Figure 4: **Visualization of validation accuracy on GPT-2-XL across three GLUE tasks (MNLI, QNLI, SST-2).** ENGINE reaches higher peak accuracy than all baselines on every task. See Appendix E.2 for training loss curves and all four models.

$2.61\times$ forward-pass speed-up and a $2.58\times$ wall-clock speed-up on LLaMA-3-8B (CB). Meanwhile, ENGINE's peak GPU memory matches LOREN's, because the fine phase is equivalent to vanilla LOREN and determines the peak, while the coarse phase only stores LoRA-scale optimizer and curvature states. Table 11 compares all methods at common target accuracies.

4.2 Ablation: Where does ENGINE's gain originate from?

Setup and design. We compare LOREN, ENGINE w/o VMS-EO, and ENGINE on the same 16 settings, with other settings matched (Appendix F). The variant without VMS-EO retains the V-cycle and disables the correction. Its comparison with LOREN isolates the V-cycle's contribution, while its comparison with full ENGINE isolates the correction's contribution to performance.

Table 3: Ablation results on all 16 settings, using MCC $\times 100$ for CoLA and accuracy (%) for the other tasks. **Bold** marks the best per column, and **cyan** marks the second best. The highlighted row is our method, with the green Δ vs LOREN row showing the absolute gain. Full results in Appendix E.

GLUE (FP32)	**GPT-2-XL** (1.5B)					**OPT-2.7B**				
Method	MNLI	QNLI	SST-2	CoLA	Avg.	MNLI	QNLI	SST-2	CoLA	Avg.
LOREN	**50.6**	**74.1**	89.0	**32.9**	**61.7**	54.1	72.6	88.2	28.6	60.9
ENGINE w/o VMS-EO	48.8	72.7	**90.2**	32.1	61.0	**54.7**	**73.8**	**89.5**	**29.4**	**61.9**
ENGINE (Ours)	**52.0**	**77.0**	**91.8**	**34.8**	**63.9**	**55.5**	**75.4**	**91.0**	**31.0**	**63.2**
Δ vs LOREN	+1.4	+2.9	+2.8	+1.9	+2.3	+1.4	+2.8	+2.8	+2.4	+2.4
SuperGLUE (BF16)	**LLaMA-3-8B**					**OPT-13B**				
Method	RTE	BoolQ	WiC	CB	Avg.	RTE	BoolQ	WiC	CB	Avg.
LOREN	**58.4**	64.5	**58.9**	**72.4**	**63.6**	**59.5**	63.2	57.0	73.7	63.4
ENGINE w/o VMS-EO	58.2	**65.2**	57.8	71.4	63.2	59.0	**64.1**	**57.8**	**75.0**	**64.0**
ENGINE (Ours)	**59.8**	**66.0**	**59.8**	**73.2**	**64.7**	**62.1**	**65.2**	**59.0**	**79.1**	**66.4**
Δ vs LOREN	+1.4	+1.5	+0.9	+0.8	+1.1	+2.6	+2.0	+2.0	+5.4	+3.0

4.3 Analysis: Attributing the Dual-Scale Benefits of ENGINE

Result. Table 3 yields two observations. First, ENGINE w/o VMS-EO has a mix of wins and losses against LOREN across the 16 settings. Second, ENGINE achieves higher accuracy than both LOREN and ENGINE w/o VMS-EO across all four models. Table 6 gives the full mean$\pm$std results, and Figure 9 shows the ablation convergence curves.

Analysis. The two observations identify distinct contributions.

- *The V-cycle alone achieves mixed gains against LOREN.* Coarse sampling in $d_c \ll d_f$ dimensions reduces the variance bound under comparable gradient and spectral scales (Proposition 3.3). The second-order restriction mismatch in Proposition 3.2 biases the uncorrected coarse direction and can offset this variance benefit. Alternating fine phases retain full-space curvature-aware optimization, while coarse phases reduce per-step cost.
- *Incorporating VMS-EO yields consistent performance gains.* By feeding full-space curvature information back into the subspace through Eqs. (7) and (8), VMS-EO corrects the structural bias. Proposition 3.3 shows that exact correction eliminates this bias without increasing coarse sampling variance in the local-linear limit. Combining coarse sampling with curvature correction thus helps explain ENGINE's gains over LOREN and the uncorrected V-cycle.

5 Conclusions

In this paper, we propose ENGINE, a zeroth-order LLM fine-tuning framework inspired by Variational MultiScale (VMS) theory from computational mechanics. From the VMS perspective, we treat the LoRA parameter space as the coarse space and the full parameter space as the fine space. By reformulating VMS with an endogenous, state-dependent Hessian, we derive the VMS-EO subgrid correction to incorporate fine-space curvature information into coarse updates. ENGINE integrates this correction into an alternating V-cycle that couples the two spaces through mutual feedback. Our statistical analysis shows that, under the stated assumptions, coarse sampling reduces the dimension-dependent variance bound and exact subgrid correction eliminates curvature-mismatch bias in the local-linear limit. We further derive nonconvex stationarity bounds for corrected coarse updates. Across 16 model-task settings, ENGINE consistently outperforms state-of-the-art ZO baselines, achieving substantial forward-pass and wall-clock speed-ups with no additional peak memory overhead. More broadly, this work provides a multiscale perspective on optimization, showing how efficient low-dimensional updates can be guided by full-space curvature information. We hope this perspective inspires further connections between multiscale modeling and optimization. Appendix H discusses limitations and future work.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the support of Westlake University Research Center for Industries of the Future and Westlake University Center for High-performance Computing.

The authors declare no competing interests.

References

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order fine-tuning for language models with low-rank structures. *arXiv preprint arXiv:2410.07698*, 2024.

Nikita Doikov, Martin Jaggi, et al. Second-order optimization with lazy hessians. In *International Conference on Machine Learning*, pages 8138–8161. PMLR, 2023.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models. *arXiv preprint arXiv:2404.08080*, 2024.

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

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 2018. URL `https://proceedings.mlr.press/v80/gupta18a.html`.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Thomas JR Hughes. Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer methods in applied mechanics and engineering*, 127(1-4):387–401, 1995.

Thomas JR Hughes, Gonzalo R Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering*, 166(1-2):3–24, 1998.

Thomas JR Hughes, Luca Mazzei, and Kenneth E Jansen. Large eddy simulation and the variational multiscale method. *Computing and visualization in science*, 3(1):47–59, 2000.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReloRA: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=DLJznSp6X3`.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

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

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417. PMLR, 2015. URL `https://proceedings.mlr.press/v37/martens15.html`.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners (2019), 2022.

Hyunseok Seung, Jaewoo Lee, and Hyunsuk Ko. Low-rank curvature for zeroth-order optimization in llm fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 25235–25242, 2026.

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

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

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

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

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

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor Tsang. Second-order fine-tuning without pain for llms: A hessian informed zeroth-order optimizer. In *The Thirteenth International Conference on Learning Representations*.

Appendix — Table of Contents

A Related Work **14**

B Structural Derivations and Proofs **14**
B.1 Proof of Lemma 3.1 (LoRA chain rule) 14
B.2 Proof of Proposition 3.2 (subgrid residual) 15

C Theoretical Analysis **15**
C.1 Setup and assumptions 15
C.2 Bias and variance of Gaussian estimators 16
C.3 Common-target MSE and exact correction 18
C.4 Scalar approximation error 21
C.5 Oracle-optimal mixing 21
C.6 Convergence under positive alignment 22

D Implementation Details **25**
D.1 Scalar correction and update 25

E Detailed Experiments and Results **26**
E.1 Full Results and Ablation 26
E.2 Additional Convergence Curves 26

F Experimental Setup and Hyperparameters **29**
F.1 Hyperparameter Sensitivity 30

G Efficiency Analysis **31**

H Limitations and Future Work **31**

A Related Work

ENGINE connects research on zeroth-order optimization, low-rank adaptation, and Variational MultiScale theory. These three lines of research motivate its sampling strategy, parameter spaces, and correction mechanism.

Zeroth-order LLM fine-tuning. Zeroth-order methods estimate gradients from function evaluations and reduce the memory required for LLM fine-tuning. MeZO uses in-place parameter perturbations to achieve memory usage comparable to inference. Its gradient estimates have high variance, which can slow optimization. MeZO-SVRG [Gautam et al., 2024] uses periodic reference gradients to reduce this variance, with additional memory for stored parameters and gradients. LOZO [Chen et al., 2024] exploits the low-rank structure of gradients to reduce the dimension of gradient estimation. Curvature information provides another way to improve the search direction. HiZOO [Zhao et al.] estimates the diagonal Hessian and uses it to precondition the update. LOREN uses a low-rank Kronecker approximation to incorporate curvature into perturbation sampling. ENGINE builds on this curvature representation, combining LoRA sampling with full-parameter updates. Its subgrid correction injects full-space curvature information into coarse updates.

Low-rank adaptation. LoRA represents a weight update as the product of two small matrices and optimizes their entries while keeping the base weight fixed. This reduces the number of trainable parameters and provides the coarse coordinates used by ENGINE. ReLoRA [Lialin et al., 2024] periodically merges the low-rank update into the base weight and reinitializes the low-rank factors. Successive updates can therefore accumulate in the full weight matrix. ReLoRA uses this procedure within first-order training. ENGINE combines parameter merging with alternating LoRA and full-space ZO updates, transferring momentum and curvature between the two spaces.

Variational MultiScale theory. Variational MultiScale (VMS) theory originates in computational mechanics. It separates a problem into resolved coarse scales and unresolved fine scales. The effect of the fine scales enters the coarse equation through a subgrid correction. In the classical linear formulation, the governing operator is fixed by the physical problem and determines this correction. ENGINE brings this perspective into zeroth-order optimization. The LoRA coordinates define the coarse space, and the correction accounts for curvature information lost when forming the coarse update. The curvature operator is estimated online and evolves during optimization. This state dependence motivates VMS with an Endogenous Operator.

B Structural Derivations and Proofs

In this appendix, we prove the LoRA chain rule and derive the subgrid residual formulas.

B.1 Proof of Lemma 3.1 (LoRA chain rule)

Proof. With $\mathbf{W}_0$ fixed, differentiating $\mathbf{W} = \mathbf{W}_0 + \mathbf{B}\mathbf{A}$ gives:

$$\mathrm{d}\mathbf{W} = \mathrm{d}\mathbf{B}\,\mathbf{A} + \mathbf{B}\,\mathrm{d}\mathbf{A}.$$

The Frobenius inner product and cyclic trace identity give:

$$\begin{aligned}\mathrm{d}f &= \langle \nabla_{\mathbf{W}} f, \mathrm{d}\mathbf{W}\rangle_F \\ &= \langle (\nabla_{\mathbf{W}} f)\mathbf{A}^\top, \mathrm{d}\mathbf{B}\rangle_F + \langle \mathbf{B}^\top \nabla_{\mathbf{W}} f, \mathrm{d}\mathbf{A}\rangle_F.\end{aligned}$$

Matching the coefficients of the independent differentials gives the components of $\mathcal{R}(\nabla_{\mathbf{W}} f)$:

$$\begin{aligned}\frac{\partial f}{\partial \mathbf{B}} &= (\nabla_{\mathbf{W}} f)\mathbf{A}^\top, \\ \frac{\partial f}{\partial \mathbf{A}} &= \mathbf{B}^\top \nabla_{\mathbf{W}} f.\end{aligned}$$

□

B.2 Proof of Proposition 3.2 (subgrid residual)

Proof. For $\rho > 0$ and $\mathbf{v} \in \mathbb{R}^d$, the Sherman-Morrison identity gives:

$$(\rho \mathbb{I}_d + \mathbf{v}\mathbf{v}^\top)^{-1} = \frac{1}{\rho}\left(\mathbb{I}_d - \frac{\mathbf{v}\mathbf{v}^\top}{\rho + \|\mathbf{v}\|^2}\right). \tag{B.1}$$

The Kronecker structure of $\mathbf{\Sigma}_f$ applies $(\rho \mathbb{I}_n + \mathbf{v}_f \mathbf{v}_f^\top)^{-1}$ independently to each row of $\mathbf{G}$. Equation (B.1) therefore gives:

$$\mathbf{\Sigma}_f \mathbf{G} = \frac{1}{\rho}\left(\mathbf{G} - \frac{\mathbf{G}\mathbf{v}_f \mathbf{v}_f^\top}{\rho + \|\mathbf{v}_f\|^2}\right).$$

Applying $\mathcal{R}_{\mathbf{B}}(\mathbf{X}) = \mathbf{X}\mathbf{A}^\top$ yields:

$$\mathcal{R}_{\mathbf{B}}(\mathbf{\Sigma}_f \mathbf{G}) = \frac{1}{\rho}\left(\mathbf{G}\mathbf{A}^\top - \frac{\mathbf{G}\mathbf{v}_f (\mathbf{A}\mathbf{v}_f)^\top}{\rho + \|\mathbf{v}_f\|^2}\right).$$

For the coarse direction, apply Eq. (B.1) with $d = r$ and $\mathbf{v} = \mathbf{v}_{\mathbf{B}}$ to each row of $\mathcal{R}_{\mathbf{B}}(\mathbf{G})$ to obtain:

$$\mathbf{\Sigma}_{\mathbf{B}} \mathcal{R}_{\mathbf{B}}(\mathbf{G}) = \frac{1}{\rho}\left(\mathbf{G}\mathbf{A}^\top - \frac{\mathbf{G}\mathbf{A}^\top \mathbf{v}_{\mathbf{B}} \mathbf{v}_{\mathbf{B}}^\top}{\rho + \|\mathbf{v}_{\mathbf{B}}\|^2}\right).$$

Subtracting the coarse direction from the restricted fine direction cancels the $\mathbf{G}\mathbf{A}^\top$ terms and gives:

$$\boldsymbol{\tau}_{\mathrm{sgs},\mathbf{B}} = \frac{1}{\rho}\left[\frac{\mathbf{G}\mathbf{A}^\top \mathbf{v}_{\mathbf{B}} \mathbf{v}_{\mathbf{B}}^\top}{\rho + \|\mathbf{v}_{\mathbf{B}}\|^2} - \frac{\mathbf{G}\mathbf{v}_f (\mathbf{A}\mathbf{v}_f)^\top}{\rho + \|\mathbf{v}_f\|^2}\right],$$

which proves Eq. (5).

The same calculation with $\mathcal{R}_{\mathbf{A}}(\mathbf{X}) = \mathbf{B}^\top \mathbf{X}$ and the A-branch covariance gives:

$$\boldsymbol{\tau}_{\mathrm{sgs},\mathbf{A}} = \frac{1}{\rho}\left[\frac{\mathbf{B}^\top \mathbf{G}\mathbf{v}_{\mathbf{A}} \mathbf{v}_{\mathbf{A}}^\top}{\rho + \|\mathbf{v}_{\mathbf{A}}\|^2} - \frac{\mathbf{B}^\top \mathbf{G}\mathbf{v}_f \mathbf{v}_f^\top}{\rho + \|\mathbf{v}_f\|^2}\right],$$

which proves Eq. (6). □

C Theoretical Analysis

This appendix presents the statistical and convergence analysis of ENGINE. Section C.1 defines the notation and estimation setting. Section C.2 derives Gaussian bias and variance bounds. Section C.3 compares estimation errors at a common target. Section C.4 bounds scalar projection errors. Section C.5 derives the optimal mixing weight. Section C.6 establishes convergence bounds for exact and approximate correction.

C.1 Setup and assumptions

Fine and coarse spaces. Subscripts f and c denote fine and coarse spaces, corresponding to full parameters and LoRA factors, respectively. Write $W = W_0 + BA$, with fixed W_0, $B \in \mathbb{R}^{m\times r}$, and $A \in \mathbb{R}^{r\times n}$. Using row-wise vectorization, define $w = \mathrm{vec}(W) \in \mathbb{R}^{d_f}$ and $\boldsymbol{\phi} = (\mathrm{vec}(B)^\top, \mathrm{vec}(A)^\top)^\top \in \mathbb{R}^{d_c}$, where $d_f = mn$ and $d_c = r(m+n)$. The fine-space loss is $f(w)$, and its LoRA parameterization is $F(\boldsymbol{\phi}) = f(w(\boldsymbol{\phi}))$. We omit boldface on matrices and directions in this appendix.

Gradients and restriction. The superscript raw denotes exact gradients before preconditioning. We write $g_f^{\mathrm{raw}} = \nabla_w f(w) \in \mathbb{R}^{d_f}$ and $g_c^{\mathrm{raw}} = \nabla_{\boldsymbol{\phi}} F(\boldsymbol{\phi}) \in \mathbb{R}^{d_c}$. Let $J = \frac{\partial w}{\partial \boldsymbol{\phi}}$ and $R = J^\top \in \mathbb{R}^{d_c \times d_f}$ represent prolongation and restriction. The chain rule gives $g_c^{\mathrm{raw}} = R g_f^{\mathrm{raw}}$. For $G \in \mathbb{R}^{m\times n}$, restriction acts as $R\,\mathrm{vec}(G) = (\mathrm{vec}(GA^\top)^\top, \mathrm{vec}(B^\top G)^\top)^\top$. For the gradient matrix $G = \nabla_W f$, we have $g_f^{\mathrm{raw}} = \mathrm{vec}(G)$.

Preconditioned directions and correction. Let $\Sigma_f \in \mathbb{R}^{d_f \times d_f}$ and $\Sigma_c = \mathrm{diag}(\Sigma_B, \Sigma_A) \in \mathbb{R}^{d_c \times d_c}$ be symmetric positive-definite preconditioners and perturbation covariances. The branch blocks have sizes $mr \times mr$ and $rn \times rn$, respectively. Define the preconditioned directions $g_f = \Sigma_f g_f^{\mathrm{raw}}$ and $g_c = \Sigma_c g_c^{\mathrm{raw}}$, the common target $g_\star = R g_f$, and the exact subgrid residual $\tau = g_\star - g_c$. The common target $g_\star \in \mathbb{R}^{d_c}$ applies fine-space preconditioning before restriction, while g_c applies coarse-space preconditioning after restriction. Here g_c and τ are vector forms of $\mathbf{s}_c$ and $\boldsymbol{\tau}_{\mathrm{sgs}}$.

Estimation setting and errors. Let $\widehat{g}_c$ denote the preconditioned coarse estimate and $\widehat{d} = \widehat{g}_c + \lambda\tau$ the corrected direction. The mean squared error and bias relative to $g_\star$ are $\mathbb{E}\|\widehat{d} - g_\star\|^2$ and $\mathbb{E}\widehat{d} - g_\star$, respectively. For a random vector X, $\mathrm{Var}(X) = \mathbb{E}\|X - \mathbb{E}X\|^2$ denotes trace variance. Vector norms are Euclidean. Expectations condition on current parameters and preconditioners and average over fresh perturbations. For simplicity, we analyze Gaussian estimators with perturbation radius ε and b independent directions, where b plays the role of K in the method. Exact correction uses conditionally deterministic τ and a fixed mixing weight $\lambda \in [0, 1]$ shared by both branches. Section C.6 specifies the conditions for history-dependent weights.

C.2 Bias and variance of Gaussian estimators

Consider an L-smooth objective φ on $\mathbb{R}^d$, $g = \nabla\varphi(x)$, a fixed SPD covariance Σ with $\|\Sigma\|_{\mathrm{op}} \le M$, and independent $z_i \sim \mathcal{N}(0, \Sigma)$. With deterministic function evaluations, define:

$$\widehat{g}_{\varepsilon,b} = \frac{1}{b}\sum_{i=1}^{b} \frac{\varphi(x + \varepsilon z_i) - \varphi(x)}{\varepsilon} z_i, \qquad \varphi_{\varepsilon,\Sigma}(x) = \mathbb{E}\varphi(x + \varepsilon z). \tag{15}$$

For fixed x, Σ, and b, define the actual trace variance by $V_\varepsilon = \mathbb{E}\|\widehat{g}_{\varepsilon,b} - \mathbb{E}\widehat{g}_{\varepsilon,b}\|^2$. This quantity is defined at every positive smoothing radius. The subscript 0 below refers to the estimator obtained in the limit $\varepsilon \to 0$, with the number of random directions b held fixed. We first derive the mean and bound the smoothing bias separately from sampling variance.

Lemma C.1 (Gaussian smoothing identity)**.** *Under the setting above, the estimator has mean* $\mathbb{E}\widehat{g}_{\varepsilon,b} = \Sigma\nabla\varphi_{\varepsilon,\Sigma}(x)$. *Its bias* $b_\varepsilon = \mathbb{E}\widehat{g}_{\varepsilon,b} - \Sigma g$ *obeys* $\|b_\varepsilon\| \le L\varepsilon\|\Sigma\|_{\mathrm{op}}\sqrt{\mathrm{tr}(\Sigma)} \le L\varepsilon\sqrt{dM^3}$.

Proof. Write $z = \Sigma^{\frac{1}{2}}u$ with $u \sim \mathcal{N}(0, I_d)$. Since $\mathbb{E}z = 0$, the constant baseline $\varphi(x)$ contributes zero to the mean. Gaussian integration by parts, applied componentwise, gives:

$$\mathbb{E}\left[u\varphi(x + \varepsilon\Sigma^{\frac{1}{2}}u)\right] = \varepsilon\Sigma^{\frac{1}{2}}\mathbb{E}\nabla\varphi(x + \varepsilon\Sigma^{\frac{1}{2}}u).$$

Global L-smoothness bounds gradient growth linearly and function growth quadratically, justifying integration by parts and differentiation under the Gaussian expectation. Therefore, we have:

$$\begin{aligned}\mathbb{E}\widehat{g}_{\varepsilon,b} &= \varepsilon^{-1}\Sigma^{\frac{1}{2}}\mathbb{E}\left[u\varphi(x + \varepsilon\Sigma^{\frac{1}{2}}u)\right] \\ &= \Sigma\mathbb{E}\nabla\varphi(x + \varepsilon z) \\ &= \Sigma\nabla\varphi_{\varepsilon,\Sigma}(x).\end{aligned}$$

Averaging b samples preserves this mean. Subtracting Σg and using the Lipschitz continuity of the gradient yields:

$$\begin{aligned}\|b_\varepsilon\| &= \|\Sigma\mathbb{E}[\nabla\varphi(x + \varepsilon z) - \nabla\varphi(x)]\| \\ &\le \|\Sigma\|_{\mathrm{op}}\mathbb{E}\|\nabla\varphi(x + \varepsilon z) - \nabla\varphi(x)\| \\ &\le L\varepsilon\|\Sigma\|_{\mathrm{op}}\mathbb{E}\|z\| \\ &\le L\varepsilon\|\Sigma\|_{\mathrm{op}}\sqrt{\mathbb{E}\|z\|^2}.\end{aligned}$$

Using $\mathbb{E}\|z\|^2 = \mathrm{tr}(\Sigma) \le dM$ and $\|\Sigma\|_{\mathrm{op}} \le M$ completes the bias bound. □

The next identity gives the moments of the linear term in the finite-difference estimator. These moments enter the finite-radius variance bound.

Lemma C.2 (Gaussian covariance identity)**.** *For $z \sim \mathcal{N}(0, \Sigma)$ and fixed g, let $Y = (g^\top z)z$. Then we have:*

$$\mathbb{E}Y = \Sigma g, \qquad \mathrm{Cov}(Y) = (g^\top \Sigma g)\Sigma + \Sigma g g^\top \Sigma.$$

The covariance of the average of b independent copies is $\frac{\mathrm{Cov}(Y)}{b}$.

Proof. Since $\mathbb{E}[zz^\top] = \Sigma$ and g is fixed, the mean is $\mathbb{E}Y = \Sigma g$. For coordinates i, j, expand the second moment and use the Gaussian fourth-moment identity to obtain:

$$\begin{aligned}
\mathbb{E}[Y_i Y_j] &= \sum_{k,l} g_k g_l \mathbb{E}[z_i z_j z_k z_l] \\
&= \sum_{k,l} g_k g_l (\Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \\
&= \Sigma_{ij}(g^\top \Sigma g) + 2(\Sigma g)_i (\Sigma g)_j.
\end{aligned}$$

Subtracting $(\Sigma g)(\Sigma g)^\top$ gives the stated covariance. For independent copies $Y^{(1)}, \ldots, Y^{(b)}$, cross-covariances vanish, so $\mathrm{Cov}(b^{-1}\sum_i Y^{(i)}) = \mathrm{Cov}(Y)/b$. Taking its trace yields:

$$\mathbb{E}\left\| \frac{1}{b}\sum_{i=1}^{b} Y^{(i)} - \Sigma g \right\|^2 = \frac{\mathrm{tr}(\Sigma) g^\top \Sigma g + g^\top \Sigma^2 g}{b}.$$

□

These identities and a Taylor remainder bound yield the finite-radius estimation error bound.

Theorem C.3 (Gaussian estimation error)**.** *The mean is $\Sigma \nabla \varphi_{\varepsilon,\Sigma}(x)$ and its bias relative to Σg satisfies $\|b_\varepsilon\| \le L\varepsilon\sqrt{dM^3}$. For finite ε, the MSE satisfies:*

$$\mathbb{E}\|\widehat{g}_{\varepsilon,b} - \Sigma g\|^2 \le \frac{2(d+1)M^2}{b}\|g\|^2 + L^2\varepsilon^2 M^3 \left[\frac{d(d+2)(d+4)}{2b} + d\right]. \tag{16}$$

For fixed x, Σ, and b, the estimator converges in mean square as $\varepsilon \to 0$ to $\widehat{g}_{0,b} = \frac{1}{b}\sum_{i=1}^{b}(g^\top z_i)z_i$.

Proof. The mean and bias bound follow from Lemma C.1. We use Lemma C.2 to control the linear term and bound the remaining finite-difference error below.

Step 1. Control the Taylor remainder. For one sample, let $Y = (g^\top z)z$ and write:

$$\begin{aligned}
X_\varepsilon &= \frac{\varphi(x+\varepsilon z) - \varphi(x)}{\varepsilon} z \\
&= Y + U_\varepsilon, \\
U_\varepsilon &= \left(\frac{\varphi(x+\varepsilon z) - \varphi(x)}{\varepsilon} - g^\top z \right) z.
\end{aligned}$$

The smoothness inequality gives $|\varphi(x+\varepsilon z) - \varphi(x) - \varepsilon g^\top z| \le \frac{L\varepsilon^2\|z\|^2}{2}$, and consequently:

$$\begin{aligned}
\mathbb{E}\|U_\varepsilon\|^2 &\le \frac{L^2\varepsilon^2}{4}\mathbb{E}\|z\|^6 \\
&\le \frac{L^2\varepsilon^2 M^3}{4} d(d+2)(d+4).
\end{aligned}$$

Here we use $\|\Sigma^{\frac{1}{2}}u\|^2 \le M\|u\|^2$ and the Gaussian moment $\mathbb{E}\|u\|^6 = d(d+2)(d+4)$. Using the same sampled directions, Jensen's inequality gives:

$$\begin{aligned}
\mathbb{E}\|\widehat{g}_{\varepsilon,b} - \widehat{g}_{0,b}\|^2 &= \mathbb{E}\left\| \frac{1}{b}\sum_{i=1}^{b} U_{\varepsilon,i} \right\|^2 \\
&\le \frac{1}{b}\sum_{i=1}^{b} \mathbb{E}\|U_{\varepsilon,i}\|^2 \longrightarrow 0 \quad \text{as } \varepsilon \to 0.
\end{aligned}$$

For fixed b, this proves mean-square convergence to $\widehat{g}_{0,b}$ and $V_\varepsilon \to V_0$ as $\varepsilon \to 0$.

Step 2. Bound the centered variance. Applying $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ to the centered Y and U_ε gives:

$$\begin{aligned}\mathrm{Var}(X_\varepsilon) &\le 2\,\mathrm{Var}(Y) + 2\,\mathrm{Var}(U_\varepsilon)\\ &\le 2[\mathrm{tr}(\Sigma) g^\top \Sigma g + g^\top \Sigma^2 g] + 2\mathbb{E}\|U_\varepsilon\|^2\\ &\le 2(d+1)M^2\|g\|^2 + \frac{L^2\varepsilon^2 M^3}{2} d(d+2)(d+4).\end{aligned}$$

Here $g^\top \Sigma g \le M\|g\|^2$, $g^\top \Sigma^2 g \le M^2\|g\|^2$, and $\mathrm{tr}(\Sigma) \le dM$. Averaging b independent X_ε samples divides this variance bound by b.

Step 3. Add the squared smoothing bias. Write $\widehat{g}_{\varepsilon,b} - \Sigma g = (\widehat{g}_{\varepsilon,b} - \mathbb{E}\widehat{g}_{\varepsilon,b}) + b_\varepsilon$. The centered term has zero mean, so:

$$\begin{aligned}\mathbb{E}\|\widehat{g}_{\varepsilon,b} - \Sigma g\|^2 &= \mathrm{Var}(\widehat{g}_{\varepsilon,b}) + \|b_\varepsilon\|^2\\ &\le \frac{2(d+1)M^2}{b}\|g\|^2 + \frac{L^2\varepsilon^2 M^3}{2b} d(d+2)(d+4) + L^2\varepsilon^2 dM^3.\end{aligned}$$

This is (16). The variance decreases with b, while the smoothing-bias term is controlled by ε. □

Local-linear limit. By Theorem C.3, $b_\varepsilon \to 0$ and $\widehat{g}_{\varepsilon,b} \to \widehat{g}_{0,b}$ in mean square as $\varepsilon \to 0$ with b fixed. Applying Lemma C.2 to this limiting estimator gives $\mathbb{E}\widehat{g}_{0,b} = \Sigma g$. Its covariance and trace variance are:

$$\begin{aligned}\mathrm{Cov}(\widehat{g}_{0,b}) &= \frac{(g^\top \Sigma g)\Sigma + \Sigma g g^\top \Sigma}{b},\\ V_0 &= \frac{\mathrm{tr}(\Sigma) g^\top \Sigma g + g^\top \Sigma^2 g}{b}.\end{aligned} \tag{17}$$

The limiting MSE relative to Σg is therefore V_0. For $\Sigma = I_d$, substitution gives $V_0 = \frac{(d+1)\|g\|^2}{b}$. All constants here concern the Gaussian estimator in Eq. (15).

C.3 Common-target MSE and exact correction

Section C.3.1 treats finite perturbations with exact correction, and Section C.3.2 includes correction error. Section C.3.3 then specializes these results to the local-linear limit.

Which estimators and which targets? We distinguish isotropic LoRA ZO from LoRA+LOREN. The former samples with covariance I_{d_c} and targets g_c^{raw}, whereas the latter samples with covariance Σ_c and targets $g_c = \Sigma_c g_c^{\mathrm{raw}}$. Full-space LOREN samples with covariance Σ_f and targets $g_f = \Sigma_f g_f^{\mathrm{raw}}$. Ideal ENGINE uses the same coarse estimator as LoRA+LOREN, shifted by the exact residual τ. The comparison uses the Gaussian estimator models of (15), which isolate the effects of sampling covariance and subgrid correction. For $s \in \{f, c\}$, superscripts iso and ani denote sampling covariances I_{d_s} and Σ_s, respectively. We abbreviate the Gaussian coarse estimator as $\widehat{g}_c = \widehat{g}_c^{\mathrm{ani}}$, suppressing ε and b. This is $\widehat{\mathbf{g}}_{c,\epsilon,b}$ from Eq. (10).

C.3.1 Finite-radius analysis

We first keep $\varepsilon > 0$. Let f and F have smoothness constants L_f, L_c, and let $\|\Sigma_s\|_{\mathrm{op}} \le M_s$ for $s \in \{f, c\}$. Theorem C.3 gives the variance and smoothing-bias bounds:

$$\mathrm{Var}(\widehat{g}_s^{\mathrm{ani}}) \le \frac{2(d_s+1)M_s^2}{b}\|g_s^{\mathrm{raw}}\|^2 + \frac{L_s^2\varepsilon^2 M_s^3}{2b} d_s(d_s+2)(d_s+4), \tag{18}$$

$$\|\mathbb{E}\widehat{g}_s^{\mathrm{ani}} - g_s\| \le L_s\varepsilon\sqrt{d_s M_s^3}. \tag{19}$$

For isotropic sampling, set $\Sigma_s = I_{d_s}$ and $M_s = 1$, with target g_s^{raw}. All comparisons use the same direction count b and radius ε.

For any deterministic target a, expanding the centered error gives:

$$\mathbb{E}\|X - a\|^2 = \mathrm{Var}(X) + \|\mathbb{E}X - a\|^2, \tag{20}$$

since $\mathbb{E}(X - \mathbb{E}X) = 0$ makes the cross term vanish. For isotropic full-space and coarse estimators, use targets g_f^{raw} and g_c^{raw}, respectively. Full-space anisotropic sampling targets g_f. To compare coarse updates with and without correction, we use the common target $g_\star$ below.

For fixed b and $\varepsilon > 0$, the variance bound contains a term linear in d_s and a remainder of order $\varepsilon^2 d_s^3$. With comparable gradient, covariance, and smoothness scales bounded independently of dimension, coarse sampling reduces these dimension factors from $d_f = mn$ to $d_c = r(m+n)$ when $d_c \ll d_f$.

Exact correction at a common target. We next quantify how exact correction changes the bias and MSE while preserving the coarse sampling variance. Let $\widehat{g}_c$ denote the finite-radius coarse estimator, with smoothing bias $b_{c,\varepsilon} = \mathbb{E}\widehat{g}_c - g_c$ and trace variance $V_c = \operatorname{Var}(\widehat{g}_c)$. For fixed $\lambda \in [0,1]$, the corrected direction is $\widehat{d}_\lambda = \widehat{g}_c + \lambda\tau$.

Proposition C.4 (Exact-correction benefit)**.** *The ideal estimator satisfies:*

$$\operatorname{Var}(\widehat{d}_\lambda) = V_c, \qquad \mathbb{E}\|\widehat{d}_\lambda - g_\star\|^2 = V_c + \|b_{c,\varepsilon} - (1-\lambda)\tau\|^2. \tag{21}$$

The MSE reduction relative to the uncorrected estimator is:

$$\mathbb{E}\|\widehat{g}_c - g_\star\|^2 - \mathbb{E}\|\widehat{d}_\lambda - g_\star\|^2 = \lambda(2-\lambda)\|\tau\|^2 - 2\lambda\langle b_{c,\varepsilon}, \tau\rangle.$$

Proof. Conditional on the state, τ and λ are deterministic, so:

$$\operatorname{Var}(\widehat{d}_\lambda) = \operatorname{Var}(\widehat{g}_c),$$
$$\mathbb{E}\widehat{d}_\lambda - g_\star = b_{c,\varepsilon} - (1-\lambda)\tau.$$

Applying the bias-variance identity in Eq. (20) proves Eq. (21). Subtracting the result at λ from that at $\lambda = 0$ gives:

$$\|b_{c,\varepsilon} - \tau\|^2 - \|b_{c,\varepsilon} - (1-\lambda)\tau\|^2 = \lambda(2-\lambda)\|\tau\|^2 - 2\lambda\langle b_{c,\varepsilon}, \tau\rangle.$$

□

For finite ε and full correction $\lambda = 1$, the same identity gives:

$$\begin{aligned}\text{MSE}_{\text{LoRA+LOREN}} - \text{MSE}_{\text{ideal}} &= \|\tau\|^2 - 2\langle b_{c,\varepsilon}, \tau\rangle \\ &\geq \|\tau\|(\|\tau\| - 2L_c\varepsilon\sqrt{d_c M_c^3}).\end{aligned}$$

The inequality follows from Cauchy-Schwarz and Eq. (19). Thus $\|\tau\| > 2L_c\varepsilon\sqrt{d_c M_c^3}$ is sufficient for strictly smaller squared bias and MSE, with unchanged coarse sampling variance.

C.3.2 Approximate correction

At finite ε, an approximate correction can add bias and sampling noise. The following decomposition includes their dependence on the coarse estimate and provides the objective for choosing the mixing weight. Here $\widehat{g}_c$ denotes the finite-radius coarse estimator, with trace variance V_c, smoothing bias $b_{c,\varepsilon}$, and centered fluctuation $\xi_c = \widehat{g}_c - \mathbb{E}\widehat{g}_c$.

For a general correction $\widehat{\tau} = \tau + b_\tau + \xi_\tau$, define $b_\tau = \mathbb{E}\widehat{\tau} - \tau$ and $\xi_\tau = \widehat{\tau} - \mathbb{E}\widehat{\tau}$, so $\mathbb{E}\xi_\tau = 0$. Let $V_\tau = \mathbb{E}\|\xi_\tau\|^2$ and $C_{c,\tau} = \mathbb{E}\langle\xi_c, \xi_\tau\rangle$.

Proposition C.5 (MSE with an approximate correction)**.** *For fixed $\lambda \in [0,1]$ and square-integrable estimates, the MSE is:*

$$\begin{aligned}\mathcal{E}(\lambda) &:= \mathbb{E}\|\widehat{g}_c + \lambda\widehat{\tau} - g_\star\|^2 \\ &= V_c + \lambda^2 V_\tau + 2\lambda C_{c,\tau} + \|b_{c,\varepsilon} - (1-\lambda)\tau + \lambda b_\tau\|^2.\end{aligned} \tag{22}$$

Proof. The error splits into a centered fluctuation and a deterministic bias:

$$\widehat{g}_c + \lambda\widehat{\tau} - g_\star = \xi_c + \lambda\xi_\tau + b_{c,\varepsilon} - (1-\lambda)\tau + \lambda b_\tau.$$

The first two terms sum to a zero-mean vector. Their squared norm has expectation:

$$\begin{aligned}\mathbb{E}\|\xi_c + \lambda\xi_\tau\|^2 &= \mathbb{E}\|\xi_c\|^2 + \lambda^2\mathbb{E}\|\xi_\tau\|^2 + 2\lambda\mathbb{E}\langle\xi_c, \xi_\tau\rangle \\ &= V_c + \lambda^2 V_\tau + 2\lambda C_{c,\tau}.\end{aligned}$$

The expected cross term vanishes, so adding the squared bias yields (22). □

The covariance term $C_{c,\tau}$ accounts for dependence between the two estimates. For history-measurable weights, the formula applies conditional on that history. For weights computed from the same samples, the analysis applies directly to the joint quantity $\lambda\widehat{\tau}$.

C.3.3 Local-linear limit and exact comparisons

All estimators and variances in this subsection refer to the limit $\varepsilon \to 0$ with b fixed.

Proof of Proposition 3.3. Apply Theorem C.3 with $\varphi = F$ and $\Sigma = \Sigma_c$ to obtain mean-square convergence and $b_{c,\varepsilon} \to 0$. Lemma C.2 gives the limiting mean g_c and trace variance $V_{c,0}$ in Eq. (11). Setting $b_{c,\varepsilon} = 0$ in Proposition C.4 gives:

$$\mathbb{E}\|\widehat{d}_\lambda - g_\star\|^2 = V_c + (1-\lambda)^2\|\tau\|^2.$$

Here V_c denotes $V_{c,0}$ in the main text. Setting $\lambda = 0$ gives Eq. (12), and $\lambda = 1$ gives Eq. (13). □

More generally, the MSE reduction is $\lambda(2-\lambda)\|\tau\|^2$, which is positive for $0 < \lambda \leq 1$ and $\tau \neq 0$. For approximate corrections with mean-square limits, substituting $b_{c,\varepsilon} = 0$ and the limiting correction moments into Proposition C.5 gives the corresponding local result.

Dimension dependence of sampling variance. For fixed b and comparable gradient and covariance scales bounded independently of dimension, the local-linear variance bounds give $V_f^{\mathrm{ani}} = O(d_f)$ and $V_c^{\mathrm{ani}} = O(d_c)$. Here V_f^{ani} is the variance measured in full parameter coordinates. The linear dimension factor explains the sampling advantage of $d_c \ll d_f$. We next restrict the full-space output to compare all estimators in the same coarse coordinates.

Bias and variance at a common target. To compare full-space LOREN with coarse estimators, first restrict its output to the same coordinates using $R\widehat{g}_f^{\mathrm{ani}}$. Define $\delta_{\mathrm{iso}} = g_\star - g_c^{\mathrm{raw}}$. Let V_c^{iso} and V_c^{ani} denote the trace variances of $\widehat{g}_{c,0,b}^{\mathrm{iso}}$ and $\widehat{g}_{c,0,b}^{\mathrm{ani}}$, respectively, and let $V_\star^{\mathrm{direct}}$ denote that of $R\widehat{g}_{f,0,b}^{\mathrm{ani}}$. Substituting the coarse gradients and covariances into Eq. (17) gives the first two expressions below. For the third, apply the same identity with the fine-space quantities and use $\mathrm{Cov}(RX) = R\,\mathrm{Cov}(X)R^\top$. Their exact expressions are:

$$V_c^{\mathrm{iso}} = \frac{(d_c+1)\|g_c^{\mathrm{raw}}\|^2}{b}, \tag{23}$$

$$V_c^{\mathrm{ani}} = \frac{\mathrm{tr}(\Sigma_c)(g_c^{\mathrm{raw}})^\top\Sigma_c g_c^{\mathrm{raw}} + (g_c^{\mathrm{raw}})^\top\Sigma_c^2 g_c^{\mathrm{raw}}}{b}, \tag{24}$$

$$V_\star^{\mathrm{direct}} = \frac{((g_f^{\mathrm{raw}})^\top\Sigma_f g_f^{\mathrm{raw}})\,\mathrm{tr}(R\Sigma_f R^\top) + \|g_\star\|^2}{b}. \tag{25}$$

Here $R\Sigma_f g_f^{\mathrm{raw}} = g_\star$, and taking the trace of the restricted covariance gives $V_\star^{\mathrm{direct}}$.

Table 4: Exact comparison in the local-linear limit $\varepsilon \to 0$, with b held fixed, against the *same* target $g_\star$, conditional on the current state. Bias is mean minus target. All methods use b independent directions. ENGINE rows use exact, conditionally deterministic correction.

Estimator	Bias	Trace variance	MSE
Isotropic LoRA RGE	$-\delta_{\mathrm{iso}}$	V_c^{iso}	$V_c^{\mathrm{iso}} + \|\delta_{\mathrm{iso}}\|^2$
LoRA+LOREN	$-\tau$	V_c^{ani}	$V_c^{\mathrm{ani}} + \|\tau\|^2$
Restricted full-space LOREN	0	$V_\star^{\mathrm{direct}}$	$V_\star^{\mathrm{direct}}$
Ideal ENGINE ($\lambda \in [0,1]$)	$-(1-\lambda)\tau$	V_c^{ani}	$V_c^{\mathrm{ani}} + (1-\lambda)^2\|\tau\|^2$
Ideal ENGINE ($\lambda = 1$)	0	V_c^{ani}	V_c^{ani}

Table 4 follows by inserting the respective means g_c^{raw}, g_c, $g_\star$, and $g_c + \lambda\tau$ into (20). Ideal ENGINE with $\lambda = 1$ has smaller MSE than restricted full-space LOREN precisely when $V_c^{\mathrm{ani}} < V_\star^{\mathrm{direct}}$, and smaller MSE than isotropic LoRA precisely when $V_c^{\mathrm{ani}} < V_c^{\mathrm{iso}} + \|\delta_{\mathrm{iso}}\|^2$. These conditions account for gradients, covariances, and restriction in each comparison.

Combined benefit of ideal ENGINE. The two benefits act on separate terms of the MSE decomposition. Coarse sampling reduces the dimension-dependent variance under the stated scale assumptions. Exact correction recovers $g_\star$ and removes the structural bias, reducing $V_c^{\mathrm{ani}} + \|\tau\|^2$ to V_c^{ani}.

C.4 Scalar approximation error

The exact branch residual has the form $\tau_\ell = \rho^{-1}(\mathbf{c}_\ell q_\ell^\top - \mathbf{c}_{f,\ell} s_\ell^\top)$, where $p_B = m$, $p_A = r$, $\mathbf{c}_\ell, \mathbf{c}_{f,\ell} \in \mathbb{R}^{p_\ell}$, and $\rho > 0$ is the same damping parameter as in the rank-1 preconditioner. Here v_B, v_A, and v_f denote the method's curvature vectors $\mathbf{v_B}$, $\mathbf{v_A}$, and $\mathbf{v}_f$, respectively. The branch residual τ_ℓ is the matrix form of $\boldsymbol{\tau}_{\mathrm{sgs},\ell}$, for $\ell \in \{B, A\}$. Specifically:

$$
\begin{aligned}
&q_B = v_B, \quad s_B = A v_f, \quad q_A = v_A, \quad s_A = v_f, \\
&\mathbf{c}_B = \frac{G A^\top v_B}{\rho + \|v_B\|^2}, \qquad \mathbf{c}_{f,B} = \frac{G v_f}{\rho + \|v_f\|^2}, \\
&\mathbf{c}_A = \frac{B^\top G v_A}{\rho + \|v_A\|^2}, \qquad \mathbf{c}_{f,A} = \frac{B^\top G v_f}{\rho + \|v_f\|^2}.
\end{aligned}
$$

These are the exact coefficients of Proposition 3.2. Their row averages define the scalar-broadcast approximation studied below. Let $\mathbf{1}_p$ be the all-ones vector, $\Pi_p = \frac{\mathbf{1}_p \mathbf{1}_p^\top}{p}$ and $P_p = I_p - \Pi_p$. For $p = p_\ell$, define the arithmetic coefficient means $\bar{c}_\ell = \frac{\mathbf{1}_p^\top \mathbf{c}_\ell}{p}$ and $\bar{c}_{f,\ell} = \frac{\mathbf{1}_p^\top \mathbf{c}_{f,\ell}}{p}$.

The next proposition separates row compression error from scalar coefficient mismatch.

Proposition C.6 (Row compression and scalar approximation error)**.** *For $p = p_\ell$, the row mean $\tau_\ell^{\mathrm{mean}} = \Pi_p \tau_\ell$ is the unique closest matrix to τ_ℓ among matrices with identical rows, in the Frobenius norm. Let $D_\ell = \|P_p \tau_\ell\|_F$. For any matrix Z with identical rows, orthogonality gives:*

$$
\|\tau_\ell - Z\|_F^2 = D_\ell^2 + \|\tau_\ell^{\mathrm{mean}} - Z\|_F^2. \tag{26}
$$

In particular, for scalar coefficients $\widetilde{c}_\ell, \widetilde{c}_{f,\ell}$, set $\widetilde{\tau}_\ell = \frac{1}{\rho}\mathbf{1}_p(\widetilde{c}_\ell q_\ell^\top - \widetilde{c}_{f,\ell} s_\ell^\top)$. Then we have:

$$
\|\tau_\ell - \widetilde{\tau}_\ell\|_F^2 = D_\ell^2 + \frac{p}{\rho^2}\|(\widetilde{c}_\ell - \bar{c}_\ell) q_\ell - (\widetilde{c}_{f,\ell} - \bar{c}_{f,\ell}) s_\ell\|^2. \tag{27}
$$

Proof. Row averaging gives $\tau_\ell^{\mathrm{mean}} = \frac{1}{\rho}\mathbf{1}_p(\bar{c}_\ell q_\ell^\top - \bar{c}_{f,\ell} s_\ell^\top)$. The rows of $P_p \tau_\ell$ sum to zero, so this residual is orthogonal to every matrix with identical rows. Therefore, we have:

$$
\begin{aligned}
\|\tau_\ell - Z\|_F^2 &= \|P_p \tau_\ell + (\tau_\ell^{\mathrm{mean}} - Z)\|_F^2 \\
&= \|P_p \tau_\ell\|_F^2 + \|\tau_\ell^{\mathrm{mean}} - Z\|_F^2.
\end{aligned}
$$

The second term vanishes exactly at $Z = \tau_\ell^{\mathrm{mean}}$, proving optimality and uniqueness. For $Z = \widetilde{\tau}_\ell$, its value is:

$$
\|\widetilde{\tau}_\ell - \tau_\ell^{\mathrm{mean}}\|_F^2 = \frac{p}{\rho^2}\|(\widetilde{c}_\ell - \bar{c}_\ell) q_\ell - (\widetilde{c}_{f,\ell} - \bar{c}_{f,\ell}) s_\ell\|^2,
$$

since $\|\mathbf{1}_p\|^2 = p$. Substitution proves Eq. (27). □

The two branches have minimum total compression error $D_B^2 + D_A^2$. For general scalar coefficients, the total squared correction error is the sum of both right-hand sides of Eq. (27). Appendix D.1 specifies how the implementation's normalization and scaling determine these coefficients.

C.5 Oracle-optimal mixing

The MSE is quadratic in λ. Define $\bar{\tau} = \mathbb{E}\widehat{\tau} = \tau + b_\tau$.

Proposition C.7 (Oracle-optimal mixing weight)**.** *If $\|\bar{\tau}\|^2 + V_\tau > 0$, minimizing* (22) *over fixed $\lambda \in [0, 1]$ gives:*

$$
\lambda^\star = \mathrm{clip}_{[0,1]}\left(\frac{\langle \tau - b_{c,\varepsilon}, \bar{\tau}\rangle - C_{c,\tau}}{\|\bar{\tau}\|^2 + V_\tau}\right). \tag{28}
$$

If the denominator vanishes, $\widehat{\tau} = 0$ almost surely and every feasible weight is optimal.

Proof. Set $a_\tau = \|\bar{\tau}\|^2 + V_\tau$ and $q_\tau = \langle \tau - b_{c,\varepsilon}, \bar{\tau} \rangle - C_{c,\tau}$. Expanding the squared bias in Eq. (22) gives:

$$\mathcal{E}(\lambda) = \mathcal{E}(0) + a_\tau \lambda^2 - 2q_\tau \lambda.$$

For $a_\tau > 0$, this strictly convex quadratic is minimized at q_τ / a_τ on $\mathbb{R}$, and clipping gives the minimum on $[0, 1]$. The same argument applies to any closed subinterval $[\lambda_{\min}, \lambda_{\max}]$. If $a_\tau = \mathbb{E}\|\widehat{\tau}\|^2 = 0$, then $\widehat{\tau} = 0$ almost surely and $\mathcal{E}$ is independent of λ. □

For unbiased estimators with uncorrelated noise, set $b_{c,\varepsilon} = 0$, $b_\tau = 0$, and $C_{c,\tau} = 0$. When $\|\tau\|^2 + V_\tau > 0$, substituting these conditions into Proposition C.7 and Eq. (22) gives:

$$\lambda^\star = \frac{\|\tau\|^2}{\|\tau\|^2 + V_\tau}, \qquad \mathcal{E}(\lambda^\star) = V_c + \frac{\|\tau\|^2 V_\tau}{\|\tau\|^2 + V_\tau}.$$

For fixed positive correction energy, this optimal weight decreases as the correction variance grows.

C.6 Convergence under positive alignment

We analyze updates on the fixed coarse objective $F(\boldsymbol{\phi}) = f(w(\boldsymbol{\phi}))$ with W_0 held fixed. Stationarity is measured by the gradient with respect to the stacked LoRA factors, $g_{c,t}^{\mathrm{raw}} = \nabla_{\boldsymbol{\phi}} F(\boldsymbol{\phi}_t)$. Let $\mathcal{F}_t$ be the history before fresh step-t perturbations, and $\mathbb{E}_t = \mathbb{E}[\cdot \mid \mathcal{F}_t]$. All current parameters, preconditioners, and exact targets are $\mathcal{F}_t$-measurable.

Assumptions for ENGINE convergence. The following assumptions hold uniformly over the coarse steps under analysis.

(A1) **Smoothness and lower boundedness.** The coarse objective F is L_c-smooth with $L_c > 0$ and $F(\boldsymbol{\phi}) \geq F_{\inf}$.

(A2) **Bounded positive-definite preconditioning.** The coarse preconditioner satisfies $m_c I \preceq \Sigma_{c,t} \preceq M_c I$ for constants $0 < m_c \leq M_c$.

(A3) **Target alignment and norm control.** The restricted full-space target satisfies:

$$\langle g_{c,t}^{\mathrm{raw}}, g_{\star,t} \rangle \geq \kappa_\star \|g_{c,t}^{\mathrm{raw}}\|^2, \qquad \|g_{\star,t}\| \leq M_\star \|g_{c,t}^{\mathrm{raw}}\|, \qquad \kappa_\star > 0. \tag{29}$$

(A4) **Predictable mixing.** The weight $\lambda_t \in [0, 1]$ is $\mathcal{F}_t$-measurable, so it is fixed before the fresh step-t perturbations are sampled.

(A5) **Controlled estimator bias and variance.** For nonnegative constants β_c, ν_c, ω_c, the coarse estimator satisfies:

$$\|\mathbb{E}_t \widehat{g}_{c,t} - g_{c,t}\| \leq \beta_c, \qquad \mathrm{Var}_t(\widehat{g}_{c,t}) \leq \nu_c \|g_{c,t}^{\mathrm{raw}}\|^2 + \omega_c.$$

Empirical alignment proxy. In our diagnostic runs, the cosine between the raw gradient estimate and the deployed pre-momentum direction after correction and clipping was positive at every measured coarse step. This measures alignment with the estimated gradient, while Assumption A3 concerns uniform alignment and norm control of the exact target.

The following descent lemma provides a common bound for exact and approximate correction.

Lemma C.8 (Conditional descent with a biased stochastic direction)**.** *Suppose F is L_c-smooth, $L_c > 0$, and $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t - \eta \widehat{d}_t$. Let an $\mathcal{F}_t$-measurable reference direction d_t satisfy:*

$$\langle g_{c,t}^{\mathrm{raw}}, d_t \rangle \geq \kappa \|g_{c,t}^{\mathrm{raw}}\|^2,$$
$$\|d_t\| \leq M \|g_{c,t}^{\mathrm{raw}}\|, \qquad \kappa > 0.$$

Assume $\|\mathbb{E}_t \widehat{d}_t - d_t\| \leq \beta$ and $\mathrm{Var}_t(\widehat{d}_t) \leq \nu \|g_{c,t}^{\mathrm{raw}}\|^2 + \omega$, where β, ν, ω are nonnegative constants. Set $\mathcal{A} = 2M^2 + \nu > 0$ and $\mathcal{B} = 2\beta^2 + \omega$. For $0 < \eta \leq \frac{\kappa}{2L_c \mathcal{A}}$, the one-step bound is:

$$\mathbb{E}_t F(\boldsymbol{\phi}_{t+1}) \leq F(\boldsymbol{\phi}_t) - \frac{\eta\kappa}{4} \|g_{c,t}^{\mathrm{raw}}\|^2 + \frac{\eta\beta^2}{2\kappa} + \frac{L_c \eta^2 \mathcal{B}}{2}. \tag{30}$$

Proof. Step 1. Lower-bound the mean descent term. Let $e_t = \mathbb{E}_t\widehat{d}_t - d_t$. The Cauchy-Schwarz inequality and $\|e_t\| \le \beta$ imply:

$$\begin{aligned}\langle g_{c,t}^{\mathrm{raw}}, \mathbb{E}_t\widehat{d}_t\rangle &= \langle g_{c,t}^{\mathrm{raw}}, d_t\rangle + \langle g_{c,t}^{\mathrm{raw}}, e_t\rangle \\ &\ge \kappa\|g_{c,t}^{\mathrm{raw}}\|^2 - \beta\|g_{c,t}^{\mathrm{raw}}\|.\end{aligned}$$

Young's inequality in the form $\beta\|g_{c,t}^{\mathrm{raw}}\| \le \frac{\kappa\|g_{c,t}^{\mathrm{raw}}\|^2}{2} + \frac{\beta^2}{2\kappa}$ gives:

$$\langle g_{c,t}^{\mathrm{raw}}, \mathbb{E}_t\widehat{d}_t\rangle \ge \frac{\kappa}{2}\|g_{c,t}^{\mathrm{raw}}\|^2 - \frac{\beta^2}{2\kappa}.$$

Step 2. Upper-bound the second moment. Separating conditional variance from the squared conditional mean yields:

$$\begin{aligned}\mathbb{E}_t\|\widehat{d}_t\|^2 &= \|\mathbb{E}_t\widehat{d}_t\|^2 + \mathrm{Var}_t(\widehat{d}_t) \\ &= \|d_t + e_t\|^2 + \mathrm{Var}_t(\widehat{d}_t) \\ &\le 2\|d_t\|^2 + 2\|e_t\|^2 + \nu\|g_{c,t}^{\mathrm{raw}}\|^2 + \omega \\ &\le (2M^2 + \nu)\|g_{c,t}^{\mathrm{raw}}\|^2 + 2\beta^2 + \omega \\ &= \mathcal{A}\|g_{c,t}^{\mathrm{raw}}\|^2 + \mathcal{B}.\end{aligned}$$

Step 3. Apply smoothness and the step-size condition. The update and L_c-smoothness imply:

$$\begin{aligned}\mathbb{E}_t F(\boldsymbol{\phi}_{t+1}) &\le F(\boldsymbol{\phi}_t) - \eta\langle g_{c,t}^{\mathrm{raw}}, \mathbb{E}_t\widehat{d}_t\rangle + \frac{L_c\eta^2}{2}\mathbb{E}_t\|\widehat{d}_t\|^2 \\ &\le F(\boldsymbol{\phi}_t) - \left(\frac{\eta\kappa}{2} - \frac{L_c\eta^2\mathcal{A}}{2}\right)\|g_{c,t}^{\mathrm{raw}}\|^2 + \frac{\eta\beta^2}{2\kappa} + \frac{L_c\eta^2\mathcal{B}}{2}.\end{aligned}$$

Since $\frac{L_c\eta^2\mathcal{A}}{2} \le \frac{\eta\kappa}{4}$, the coefficient of the squared gradient is at least $\frac{\eta\kappa}{4}$. This proves (30). □

For ideal ENGINE, use the exact correction $\widehat{d}_t = \widehat{g}_{c,t} + \lambda_t\tau_t$. Set $\kappa = \min\{m_c, \kappa_\star\}$, $M = \max\{M_c, M_\star\}$, $\mathcal{A} = 2M^2 + \nu_c$ and $\mathcal{B} = 2\beta_c^2 + \omega_c$.

Let $T \ge 1$ be the number of optimization steps. The next theorem bounds the average squared coarse gradient after these steps and identifies the residual error due to estimator bias.

Theorem C.9 (Ideal ENGINE stationarity)**.** *Under Assumptions A1 to A5, for $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t - \eta\widehat{d}_t$, $\eta = \frac{\kappa}{2L_c\mathcal{A}\sqrt{T}}$ and $T \ge 1$, the iterates satisfy:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|g_{c,t}^{\mathrm{raw}}\|^2 \le \frac{8L_c\mathcal{A}\Delta_F}{\kappa^2\sqrt{T}} + \frac{\mathcal{B}}{\mathcal{A}\sqrt{T}} + \frac{2\beta_c^2}{\kappa^2}, \qquad \Delta_F = F(\boldsymbol{\phi}_0) - F_{\mathrm{inf}}. \tag{31}$$

Proof. Step 1. Verify alignment and the norm bound for the mixed target. The mixed target satisfies:

$$\begin{aligned}d_t &= g_{c,t} + \lambda_t\tau_t \\ &= (1-\lambda_t)g_{c,t} + \lambda_t g_{\star,t}.\end{aligned}$$

Since $g_{c,t} = \Sigma_{c,t}g_{c,t}^{\mathrm{raw}}$, the spectral lower bound and (29) give:

$$\begin{aligned}\langle g_{c,t}^{\mathrm{raw}}, d_t\rangle &= (1-\lambda_t)(g_{c,t}^{\mathrm{raw}})^\top\Sigma_{c,t}g_{c,t}^{\mathrm{raw}} + \lambda_t\langle g_{c,t}^{\mathrm{raw}}, g_{\star,t}\rangle \\ &\ge [(1-\lambda_t)m_c + \lambda_t\kappa_\star]\|g_{c,t}^{\mathrm{raw}}\|^2 \\ &\ge \kappa\|g_{c,t}^{\mathrm{raw}}\|^2.\end{aligned}$$

Similarly, the triangle inequality and the upper bounds yield:

$$\begin{aligned}\|d_t\| &\le (1-\lambda_t)\|\Sigma_{c,t}g_{c,t}^{\mathrm{raw}}\| + \lambda_t\|g_{\star,t}\| \\ &\le [(1-\lambda_t)M_c + \lambda_t M_\star]\|g_{c,t}^{\mathrm{raw}}\| \\ &\le M\|g_{c,t}^{\mathrm{raw}}\|.\end{aligned}$$

Both comparisons use $0 \le \lambda_t \le 1$.

Step 2. Apply the exact-correction result conditionally. Predictability makes λ_t fixed conditional on $\mathcal{F}_t$. The same holds for τ_t, so Proposition C.4 gives:

$$\mathbb{E}_t \widehat{d}_t - d_t = \mathbb{E}_t \widehat{g}_{c,t} - g_{c,t},$$
$$\mathrm{Var}_t(\widehat{d}_t) = \mathrm{Var}_t(\widehat{g}_{c,t}).$$

Thus Lemma C.8 applies with $\beta = \beta_c$, $\nu = \nu_c$, and $\omega = \omega_c$. The chosen step size is admissible since $T \ge 1$.

Step 3. Sum the expected descent inequalities. Take unconditional expectations in (30), use the tower property, and rearrange to get:

$$\frac{\eta\kappa}{4}\mathbb{E}\|g_{c,t}^{\mathrm{raw}}\|^2 \le \mathbb{E}F(\boldsymbol{\phi}_t) - \mathbb{E}F(\boldsymbol{\phi}_{t+1}) + \frac{\eta\beta_c^2}{2\kappa} + \frac{L_c\eta^2\mathcal{B}}{2}.$$

Summing from $t = 0$ to $T-1$ telescopes the objective terms. For a fixed initial state and $F \ge F_{\inf}$ this gives:

$$\frac{\eta\kappa}{4}\sum_{t=0}^{T-1}\mathbb{E}\|g_{c,t}^{\mathrm{raw}}\|^2 \le \Delta_F + \frac{T\eta\beta_c^2}{2\kappa} + \frac{TL_c\eta^2\mathcal{B}}{2}.$$

Dividing by $\frac{\eta\kappa T}{4}$ gives the general constant-step bound:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|g_{c,t}^{\mathrm{raw}}\|^2 \le \frac{4\Delta_F}{\eta\kappa T} + \frac{2\beta_c^2}{\kappa^2} + \frac{2L_c\eta\mathcal{B}}{\kappa}. \tag{32}$$

Substituting $\eta = \frac{\kappa}{2L_c\mathcal{A}\sqrt{T}}$ into Eq. (32) proves Eq. (31). □

For Gaussian RGE with b_c independent coarse-space directions, Theorem C.3 permits:

$$\beta_c^2 \le L_c^2\varepsilon^2 d_c M_c^3, \quad \nu_c = \frac{2(d_c+1)M_c^2}{b_c}, \quad \omega_c = \frac{L_c^2\varepsilon^2 M_c^3}{2b_c}d_c(d_c+2)(d_c+4).$$

Hence the ideal model has $O(T^{-\frac{1}{2}}) + O(\varepsilon^2)$ stationarity, with coarse-dimensional sampling constants. Here T is the number of optimization steps, and S is the curvature refresh interval used in the method.

We now extend the same descent argument to approximate corrections. Their mean error changes the residual term, while their variance changes the sampling constants in the stationarity bound.

Additional assumption for approximate correction.

(A6) **Controlled correction error.** The predictable weight satisfies $0 \le \lambda_t \le \lambda_{\max} \le 1$, and the correction estimate obeys, uniformly in t:

$$\|\mathbb{E}_t\widehat{\tau}_t - \tau_t\| \le \beta_\tau, \qquad \mathrm{Var}_t(\widehat{\tau}_t) \le \nu_\tau\|g_{c,t}^{\mathrm{raw}}\|^2 + \omega_\tau$$

for nonnegative constants $\beta_\tau, \nu_\tau, \omega_\tau$.

Corollary C.10 (Convergence with an approximate correction)**.** *Under Assumptions A1 to A6, update with $\widehat{d}_t = \widehat{g}_{c,t} + \lambda_t\widehat{\tau}_t$. Define:*

$$\beta_{\mathrm{app}} = \beta_c + \lambda_{\max}\beta_\tau,$$
$$\nu_{\mathrm{app}} = 2\nu_c + 2\lambda_{\max}^2\nu_\tau, \qquad \omega_{\mathrm{app}} = 2\omega_c + 2\lambda_{\max}^2\omega_\tau,$$
$$\mathcal{A}_{\mathrm{app}} = 2M^2 + \nu_{\mathrm{app}}, \qquad \mathcal{B}_{\mathrm{app}} = 2\beta_{\mathrm{app}}^2 + \omega_{\mathrm{app}}.$$

For $\eta = \frac{\kappa}{2L_c\mathcal{A}_{\mathrm{app}}\sqrt{T}}$, the bound is:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|g_{c,t}^{\mathrm{raw}}\|^2 \le \frac{8L_c\mathcal{A}_{\mathrm{app}}\Delta_F}{\kappa^2\sqrt{T}} + \frac{\mathcal{B}_{\mathrm{app}}}{\mathcal{A}_{\mathrm{app}}\sqrt{T}} + \frac{2\beta_{\mathrm{app}}^2}{\kappa^2}. \tag{33}$$

Proof. The reference direction remains $d_t = (1-\lambda_t)g_{c,t} + \lambda_t g_{\star,t}$, so its alignment and norm bounds are unchanged. The conditional mean error is:

$$\mathbb{E}_t\widehat{d}_t - d_t = (\mathbb{E}_t\widehat{g}_{c,t} - g_{c,t}) + \lambda_t(\mathbb{E}_t\widehat{\tau}_t - \tau_t).$$

Its norm is at most $\beta_c + \lambda_{\max}\beta_\tau = \beta_{\text{app}}$. Let $\xi_{c,t}$ and $\xi_{\tau,t}$ be the two centered estimates, and denote their trace variances by $V_{c,t}$ and $V_{\tau,t}$. Then we have:

$$\begin{aligned}\operatorname{Var}_t(\widehat{d}_t) &= V_{c,t} + \lambda_t^2 V_{\tau,t} + 2\lambda_t \mathbb{E}_t\langle \xi_{c,t}, \xi_{\tau,t}\rangle \\ &\le V_{c,t} + \lambda_t^2 V_{\tau,t} + 2\lambda_t\sqrt{V_{c,t}V_{\tau,t}} \\ &\le 2V_{c,t} + 2\lambda_t^2 V_{\tau,t} \\ &\le \nu_{\text{app}}\|g_{c,t}^{\text{raw}}\|^2 + \omega_{\text{app}}.\end{aligned}$$

The first inequality is conditional Cauchy-Schwarz, and the second uses $2ab \le a^2 + b^2$. Together they control the cross-covariance for jointly sampled estimators. Lemma C.8 now applies with the approximate constants. Summing its inequality and substituting the new step size exactly as in the preceding theorem gives (33). □

For a deterministic scalar-broadcast correction, the squared mean error is the sum over the two branches of (27) and the conditional correction variance is zero. A uniform upper bound on that sum therefore supplies β_τ^2 in the corollary. For random scalar estimates, their mean error and variance must instead be bounded separately as in Proposition C.5. Thus the projection result supplies a correction-error bound, while the descent lemma explains how this error enters the stationarity bound.

D Implementation Details

This appendix specifies the scalar correction used by ENGINE and its relation to the approximation error in Section C.4.

D.1 Scalar correction and update

Branch notation. Boldface $\mathbf{c}_\ell, \mathbf{c}_{f,\ell} \in \mathbb{R}^{p_\ell}$ denotes the exact per-row coefficients in Section C.4, while $c_\ell, c_{f,\ell}$ denote scalar surrogates. For $\ell \in \{B, A\}$, define $\mathbf{X}_B = \mathbf{B}$ and $\mathbf{X}_A = \mathbf{A}$. For the B-branch, $p_B = m$, $\mathbf{G}_B = \mathbf{G}\mathbf{A}^\top$, $\mathbf{h}_B = \mathbf{G}\mathbf{v}_f$, $\mathbf{q}_B = \mathbf{v}_\mathbf{B}$, and $\mathbf{s}_B = \mathbf{A}\mathbf{v}_f$. For the A-branch, $p_A = r$, $\mathbf{G}_A = \mathbf{B}^\top\mathbf{G}$, $\mathbf{h}_A = \mathbf{B}^\top\mathbf{G}\mathbf{v}_f$, $\mathbf{q}_A = \mathbf{v}_\mathbf{A}$, and $\mathbf{s}_A = \mathbf{v}_f$. Here $\mathbf{G} = \nabla_\mathbf{W} f$ defines the reference contractions. The matrices $\mathbf{1}_{p_\ell}\mathbf{q}_\ell^\top$ and $\mathbf{1}_{p_\ell}\mathbf{s}_\ell^\top$ have the same shape as $\mathbf{X}_\ell$.

Scalar coefficients. The denominator-adjusted reference scalars are:

$$c_\ell = \frac{\mathbf{1}_{p_\ell}^\top \mathbf{G}_\ell \mathbf{q}_\ell}{\rho + p_\ell\|\mathbf{q}_\ell\|^2}, \qquad c_{f,\ell} = \frac{\mathbf{1}_{p_\ell}^\top \mathbf{h}_\ell}{\rho + p_\ell\|\mathbf{v}_f\|^2}. \tag{D.1}$$

These scalars differ from the arithmetic row means $\bar{c}_\ell$ and $\bar{c}_{f,\ell}$ in Section C.4. For the exact preconditioned direction $\mathbf{g}_{c,\ell} = \mathbf{\Sigma}_\ell\mathbf{G}_\ell$, Eq. (B.1) gives:

$$\langle \mathbf{1}_{p_\ell}\mathbf{q}_\ell^\top, \mathbf{g}_{c,\ell}\rangle_F = \frac{\mathbf{1}_{p_\ell}^\top \mathbf{G}_\ell \mathbf{q}_\ell}{\rho + \|\mathbf{q}_\ell\|^2}. \tag{D.2}$$

When $c_\ell \neq 0$, the ratio of this contraction to c_ℓ is $(\rho + p_\ell\|\mathbf{q}_\ell\|^2)/(\rho + \|\mathbf{q}_\ell\|^2)$, which approaches p_ℓ when $\|\mathbf{q}_\ell\|^2 \gg \rho$. For the preconditioned RLOO estimate, we use:

$$c_\ell^{\text{impl}} = \frac{\langle \mathbf{1}_{p_\ell}\mathbf{q}_\ell^\top, \widehat{\mathbf{g}}_{c,\ell}\rangle_F}{\rho + p_\ell\|\mathbf{q}_\ell\|^2 + 10^{-8}}. \tag{D.3}$$

For a fine probe, let $\mathbf{Z}_k \in \mathbb{R}^{m\times n}$ be the perturbation matrix for this weight block, f_k the loss at the k-th perturbed model, and $\bar{f} = K^{-1}\sum_{k=1}^K f_k$, with $K > 1$. Let $\mathbf{v}_{f,k}$ denote the curvature

vector when the k-th contribution is accumulated, before its next curvature update. The sampled fine coefficients used in the correction are:

$$c_{f,B}^{\mathrm{impl}} = \mathrm{EMA}_{\mathrm{bc}}\left[\frac{1}{\varepsilon(K-1)}\sum_{k=1}^{K}\frac{(f_k-\bar{f})\mathbf{1}_m^\top \mathbf{Z}_k\mathbf{v}_{f,k}}{\rho + m\|\mathbf{v}_{f,k}\|^2 + 10^{-8}}\right],$$
$$c_{f,A}^{\mathrm{impl}} = \mathrm{EMA}_{\mathrm{bc}}\left[\frac{1}{\varepsilon(K-1)}\sum_{k=1}^{K}\frac{(f_k-\bar{f})\mathbf{1}_r^\top \mathbf{B}^\top\mathbf{Z}_k\mathbf{v}_{f,k}}{\rho + r\|\mathbf{v}_{f,k}\|^2 + 10^{-8}}\right]. \tag{D.4}$$

Here $\mathrm{EMA}_{\mathrm{bc}}$ is an exponential moving average with decay 0.9, initialized at zero and divided by $1-0.9^j$ after j fine probes in the current coarse phase. Its state is reset at each new coarse phase and retained between probes. The 10^{-8} term stabilizes the sampled-coefficient denominators. The superscript impl identifies the coefficients used in the update, including their estimation, normalization, and smoothing errors.

Correction and update. The scalar-corrected step, before clipping and momentum processing, is:

$$\widehat{\boldsymbol{\tau}}_\ell^{\mathrm{impl}} = \mathbf{1}_{p_\ell}(c_\ell^{\mathrm{impl}}\mathbf{q}_\ell^\top - c_{f,\ell}^{\mathrm{impl}}\mathbf{s}_\ell^\top),$$
$$\Delta\mathbf{X}_\ell = -\eta_\ell\big(\widehat{\mathbf{g}}_{c,\ell} + \lambda_\ell\widehat{\boldsymbol{\tau}}_\ell^{\mathrm{impl}}\big). \tag{D.5}$$

With the current-basis directions defined above, this correction has the form $\widetilde{\tau}_\ell$ in Proposition C.6 with $\widetilde{c}_\ell = \rho c_\ell^{\mathrm{impl}}$ and $\widetilde{c}_{f,\ell} = \rho c_{f,\ell}^{\mathrm{impl}}$. These factors account for the $1/\rho$ in the exact residual. Substituting them into Eq. (27) includes both coefficient normalization and correction scaling in the error decomposition.

Initialization safeguards. When $\mathbf{A}$ is reset to zero, B-branch feedback uses $\mathbf{A}_{\mathrm{old}}\mathbf{v}_f$ if a cached basis is available and $\max_{i,j}|A_{ij}| < 10^{-10}$. Transferred curvature vectors are normalized only when their norms exceed 10^{-10}. Otherwise, their initialized values are retained. The cached-basis fallback introduces a directional approximation in addition to the scalar coefficient error analyzed above.

E Detailed Experiments and Results

This appendix reports complete results and learning curves. Experimental settings and hyperparameters are collected in Appendix F.

E.1 Full Results and Ablation

Tables 5 and 6 expand the main results and ablation in Tables 2 and 3, reporting five-seed means and standard deviations. The main results use MCC $\times 100$ for CoLA and accuracy (%) for other tasks. The ablation uses the same metrics. Baselines are independently reproduced under the LOREN protocol. The ablation compares ENGINE with the same V-cycle without VMS-EO correction. Bold and cyan bold mark the best and second-best results in each column. The highlighted rows report ENGINE and its absolute score gain over LOREN.

E.2 Additional Convergence Curves

Figures 5 to 8 extend Figure 4 to all 16 model-task settings, reporting validation metrics (MCC $\times 100$ for CoLA and accuracy for the other tasks) and training loss. Figure 9 shows how the ablation results in Table 6 evolve with the forward-pass budget.

Table 5: Full results over five seeds, reported as mean ± standard deviation. CoLA uses MCC ×100 and other tasks use accuracy (%). Avg. averages the four task means.

GLUE (FP32)	**GPT-2-XL** (1.5B)					**OPT-2.7B**				
Method	MNLI	QNLI	SST-2	CoLA	Avg.	MNLI	QNLI	SST-2	CoLA	Avg.
MeZO	38.4 ± 1.0	59.2 ± 0.2	74.3 ± 1.0	20.1 ± 3.5	48.0	50.4 ± 3.3	**75.2 ± 1.0**	**88.7 ± 0.2**	19.8 ± 3.1	58.5
MeZO-Adam	**50.8 ± 1.4**	71.9 ± 4.4	**91.2 ± 0.5**	31.6 ± 3.1	61.4	45.9 ± 1.2	72.3 ± 3.2	86.9 ± 2.8	28.3 ± 3.5	58.4
MeZO-SVRG	49.8 ± 1.1	65.5 ± 1.0	84.5 ± 1.5	25.6 ± 2.9	56.4	41.5 ± 0.2	59.8 ± 1.0	62.9 ± 1.9	26.1 ± 3.0	47.6
LOZO	42.9 ± 0.9	60.6 ± 1.1	80.0 ± 1.0	18.1 ± 2.9	50.4	43.2 ± 1.1	59.5 ± 0.8	66.2 ± 3.0	16.9 ± 2.9	46.5
HiZOO	48.4 ± 0.2	66.8 ± 3.4	88.9 ± 0.2	27.9 ± 3.5	58.0	50.9 ± 1.0	74.7 ± 1.0	88.2 ± 0.6	22.3 ± 3.6	59.0
LOREN	50.6 ± 0.5	**74.1 ± 1.2**	89.0 ± 0.9	**32.9 ± 2.8**	**61.7**	**54.1 ± 0.7**	72.6 ± 1.8	88.2 ± 0.8	**28.6 ± 3.2**	**60.9**
ENGINE (Ours)	**52.0 ± 0.5**	**77.0 ± 1.0**	**91.8 ± 0.5**	**34.8 ± 2.8**	**63.9**	**55.5 ± 0.5**	**75.4 ± 1.2**	**91.0 ± 0.4**	**31.0 ± 2.9**	**63.2**
Δ vs LOREN	+1.4	+2.9	+2.8	+1.9	+2.3	+1.4	+2.8	+2.8	+2.4	+2.4
SuperGLUE (BF16)	**LLaMA-3-8B**					**OPT-13B**				
Method	RTE	BoolQ	WiC	CB	Avg.	RTE	BoolQ	WiC	CB	Avg.
MeZO	**59.1 ± 1.5**	64.0 ± 0.8	**59.3 ± 1.9**	71.5 ± 1.5	63.5	59.0 ± 0.5	64.4 ± 1.1	57.1 ± 0.6	72.7 ± 1.5	63.3
MeZO-Adam	58.7 ± 1.0	63.2 ± 0.2	57.1 ± 1.0	68.9 ± 0.8	62.0	**60.8 ± 1.5**	63.4 ± 1.3	55.7 ± 1.0	70.2 ± 1.7	62.2
MeZO-SVRG	58.2 ± 1.5	63.1 ± 0.2	58.0 ± 1.1	64.9 ± 3.1	61.1	59.3 ± 1.0	**65.2 ± 0.8**	56.8 ± 0.8	69.4 ± 3.0	62.7
LOZO	57.8 ± 1.2	63.6 ± 0.7	57.2 ± 2.0	71.1 ± 4.2	62.4	58.2 ± 0.7	63.4 ± 0.2	**57.7 ± 1.6**	68.8 ± 3.0	62.0
HiZOO	58.2 ± 0.5	63.6 ± 0.2	59.2 ± 1.6	67.1 ± 3.9	62.0	59.5 ± 1.2	63.7 ± 0.4	57.0 ± 1.1	72.7 ± 1.8	63.2
LOREN	58.4 ± 1.2	**64.5 ± 1.3**	58.9 ± 1.3	**72.4 ± 2.7**	**63.6**	59.5 ± 0.7	63.2 ± 0.6	57.0 ± 0.9	**73.7 ± 1.2**	**63.4**
ENGINE (Ours)	**59.8 ± 1.1**	**66.0 ± 1.0**	**59.8 ± 1.2**	**73.2 ± 2.3**	**64.7**	**62.1 ± 0.8**	**65.2 ± 0.6**	**59.0 ± 0.8**	**79.1 ± 1.2**	**66.4**
Δ vs LOREN	+1.4	+1.5	+0.9	+0.8	+1.1	+2.6	+2.0	+2.0	+5.4	+3.0

Table 6: VMS-EO ablation over five seeds, reported as mean ± standard deviation, using MCC ×100 for CoLA and accuracy (%) for the other tasks.

GLUE (FP32)	**GPT-2-XL** (1.5B)					**OPT-2.7B**				
Method	MNLI	QNLI	SST-2	CoLA	Avg.	MNLI	QNLI	SST-2	CoLA	Avg.
LOREN	**50.6 ± 0.5**	**74.1 ± 1.2**	89.0 ± 0.9	**32.9 ± 2.8**	**61.7**	54.1 ± 0.7	72.6 ± 1.8	88.2 ± 0.8	28.6 ± 3.2	60.9
ENGINE w/o VMS-EO	48.8 ± 0.6	72.7 ± 1.1	**90.2 ± 0.5**	32.1 ± 2.9	61.0	**54.7 ± 0.8**	**73.8 ± 1.3**	**89.5 ± 0.4**	**29.4 ± 3.1**	**61.9**
ENGINE (Ours)	**52.0 ± 0.5**	**77.0 ± 1.0**	**91.8 ± 0.5**	**34.8 ± 2.8**	**63.9**	**55.5 ± 0.5**	**75.4 ± 1.2**	**91.0 ± 0.4**	**31.0 ± 2.9**	**63.2**
Δ vs LOREN	+1.4	+2.9	+2.8	+1.9	+2.3	+1.4	+2.8	+2.8	+2.4	+2.4
SuperGLUE (BF16)	**LLaMA-3-8B**					**OPT-13B**				
Method	RTE	BoolQ	WiC	CB	Avg.	RTE	BoolQ	WiC	CB	Avg.
LOREN	**58.4 ± 1.2**	64.5 ± 1.3	**58.9 ± 1.3**	**72.4 ± 2.7**	**63.6**	**59.5 ± 0.7**	63.2 ± 0.6	57.0 ± 0.9	73.7 ± 1.2	63.4
ENGINE w/o VMS-EO	58.2 ± 1.3	**65.2 ± 1.0**	57.8 ± 1.2	71.4 ± 2.5	63.2	59.0 ± 0.9	**64.1 ± 0.7**	**57.8 ± 0.9**	**75.0 ± 1.5**	**64.0**
ENGINE (Ours)	**59.8 ± 1.1**	**66.0 ± 1.0**	**59.8 ± 1.2**	**73.2 ± 2.3**	**64.7**	**62.1 ± 0.8**	**65.2 ± 0.6**	**59.0 ± 0.8**	**79.1 ± 1.2**	**66.4**
Δ vs LOREN	+1.4	+1.5	+0.9	+0.8	+1.1	+2.6	+2.0	+2.0	+5.4	+3.0

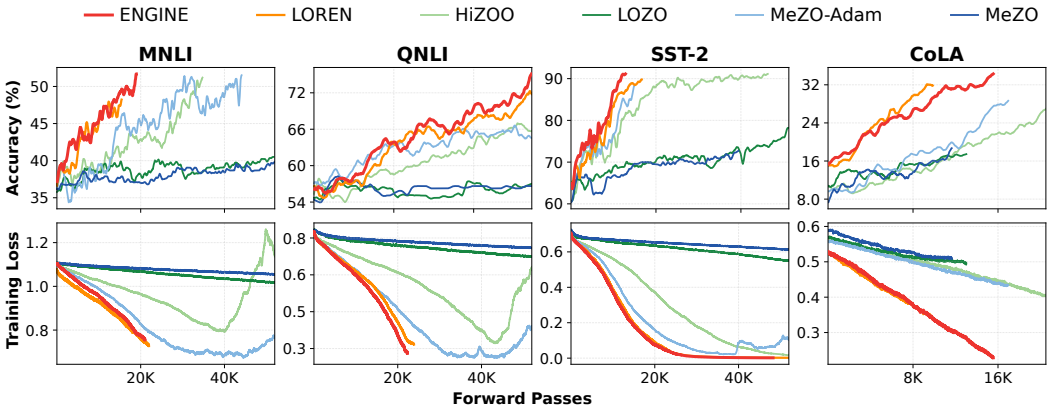


Figure 5: **GPT-2-XL (1.5B) training dynamics on four GLUE tasks (MNLI, QNLI, SST-2, CoLA).** *Top row*: validation MCC ×100 for CoLA and accuracy (%) for the other tasks. *Bottom row*: training loss.

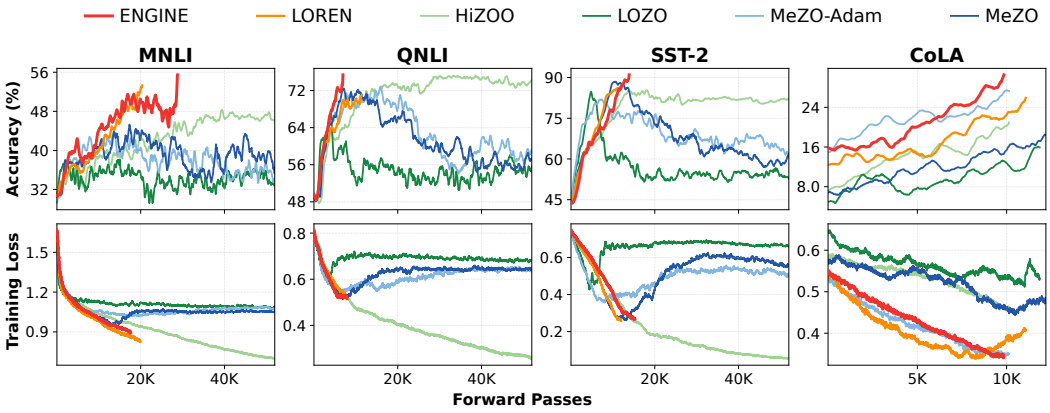


Figure 6: **OPT-2.7B training dynamics on four GLUE tasks (MNLI, QNLI, SST-2, CoLA).** *Top row*: validation MCC $\times 100$ for CoLA and accuracy (%) for the other tasks. *Bottom row*: training loss.

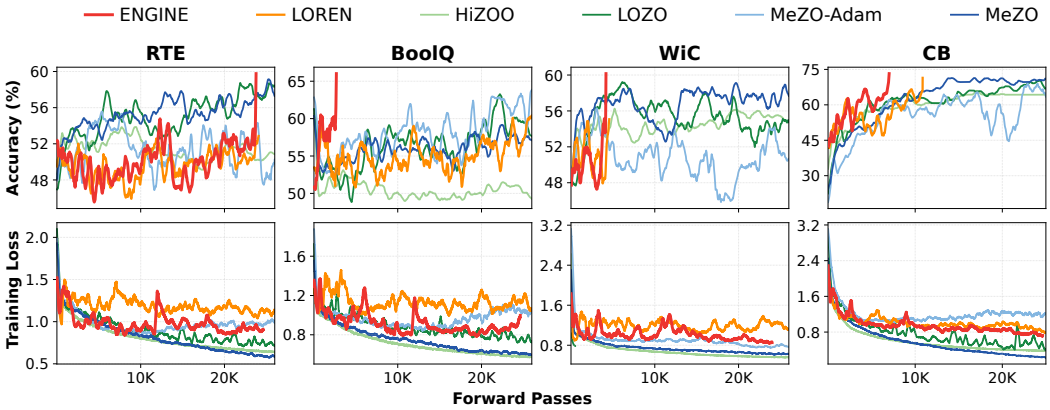


Figure 7: **LLaMA-3-8B training dynamics on four SuperGLUE tasks (RTE, BoolQ, WiC, CB).** *Top row*: validation accuracy. *Bottom row*: training loss.

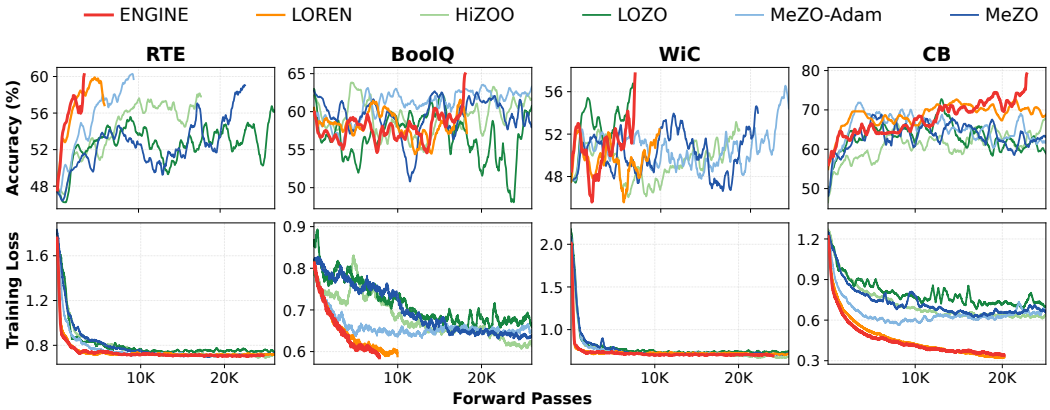


Figure 8: **OPT-13B training dynamics on four SuperGLUE tasks (RTE, BoolQ, WiC, CB).** *Top row*: validation accuracy. *Bottom row*: training loss.

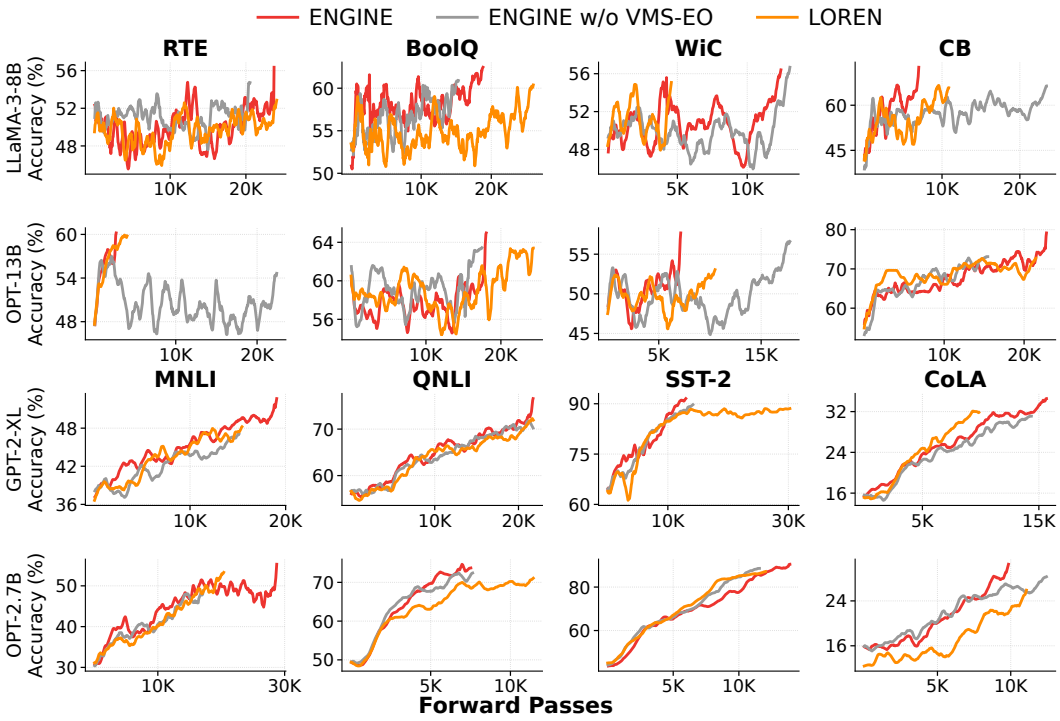


Figure 9: **VMS-EO ablation across four models and 16 tasks.** Validation scores (MCC $\times 100$ for CoLA and accuracy (%) for other tasks) versus forward passes for ENGINE (red), ENGINE w/o VMS-EO (grey), and LOREN (orange). From top to bottom, the rows show LLaMA-3-8B, OPT-13B, GPT-2-XL, and OPT-2.7B. In the first two rows, columns show RTE, BoolQ, WiC, and CB. In the last two rows, columns show MNLI, QNLI, SST-2, and CoLA.

F Experimental Setup and Hyperparameters

Tasks and metrics. We follow the LOREN protocol for datasets, splits, and training. GPT-2-XL and OPT-2.7B use FP32 on MNLI, QNLI, SST-2, and CoLA. LLaMA-3-8B and OPT-13B use BF16 on RTE, BoolQ, WiC, and CB. The main results use MCC $\times 100$ for CoLA and accuracy (%) for the other tasks. Avg. is the arithmetic mean of the four task scores for each model. Ablation tables and learning curves use the same metrics. Sensitivity analyses retain accuracy for all tasks, including CoLA.

Training budget. Each run uses one NVIDIA A100 80GB GPU. All methods use batch size 64, perturbation radius $\epsilon = 10^{-3}$, and direction count $K = 6$. Training lasts 8,000 steps for GPT-2-XL and OPT-2.7B and 4,000 steps for LLaMA-3-8B and OPT-13B.

Method configurations. Across models, MeZO-Adam uses betas $(0.9, 0.999)$. MeZO-SVRG uses full-batch update interval 2. LOZO searches ranks $\{\mathbf{2}, 4, 8\}$ and intervals $\{\mathbf{50}, 100\}$. HiZOO uses Hessian smoothing 10^{-8}. LOREN uses curvature learning rate $\nu = 10^{-3}$ and damping 10^{-1}.

Tables 7 and 8 give the baseline and ENGINE settings. Baseline settings follow LOREN, and bold entries mark selected values. The ablation uses the same ENGINE settings with $\lambda = 0$.

Table 7: Specific hyperparameters for the baselines. Learning rate search grids are listed by model, with selected values in bold. The two MeZO-SVRG rows give full-batch and mini-batch rates.

Method	GPT-2-XL	OPT-2.7B	LLaMA-3-8B, OPT-13B
MeZO	{1e-4, 5e-5, 1e-5, **5e-6**, 1e-6}	{1e-4, 5e-5, 1e-5, **5e-6**, 1e-6}	{1e-5, 5e-6, **1e-6**, 5e-7, 1e-7}
MeZO-Adam	{1e-3, 5e-4, **1e-4**, 5e-5, 1e-5}	{1e-3, 5e-4, 1e-4, **5e-5**, 1e-5}	{1e-3, 5e-4, 1e-4, **5e-5**, 1e-5}
MeZO-SVRG full	{1e-4, **5e-5**, 1e-5, 5e-6, 1e-6}	{1e-4, **5e-5**, 1e-5, 5e-6, 1e-6}	{1e-5, 5e-6, **1e-6**, 5e-7, 1e-7}
MeZO-SVRG mini	{1e-5, 5e-6, **1e-6**, 5e-7, 1e-7}	{1e-5, 5e-6, **1e-6**, 5e-7, 1e-7}	{1e-6, 5e-7, **1e-7**, 5e-8, 1e-8}
LOZO	{1e-4, 5e-5, 1e-5, **5e-6**, 1e-6}	{1e-4, 5e-5, 1e-5, **5e-6**, 1e-6}	{1e-5, 5e-6, 1e-6, **5e-7**, 1e-7}
HiZOO	{1e-3, 5e-4, 1e-4, **5e-5**, 1e-5}	{1e-4, 5e-5, 1e-5, **5e-6**, 1e-6}	{1e-5, 5e-6, **1e-6**, 5e-7, 1e-7}
LOREN	{**1e-5**, 5e-6, 1e-6, 5e-7, 1e-7}	{1e-5, 5e-6, 1e-6, **5e-7**, 1e-7}	{1e-5, 5e-6, 1e-6, **5e-7**, 1e-7}
ENGINE	{**1e-5**, 5e-6, 1e-6, 5e-7, 1e-7}	{1e-5, 5e-6, 1e-6, **5e-7**, 1e-7}	{1e-5, 5e-6, 1e-6, **5e-7**, 1e-7}

Table 8: Specific hyperparameters for ENGINE. Each row gives a model's learning rate, rank, and refresh intervals S in coarse steps. All four models use $\nu = 10^{-3}$, $\rho = 10^{-3}$, C60F60, and mixing bounds $[0.3, 0.8]$.

			Refresh interval S			
Model	η	r	MNLI	QNLI	CoLA	SST-2
GPT-2-XL	1e-5	16	5	20	20	20
OPT-2.7B	1e-5	32	20	20	20	5
Model	η	r	CB	RTE	BoolQ	WiC
LLaMA-3-8B	5e-7	32	20	20	20	20
OPT-13B	5e-7	64	20	20	10	100

F.1 Hyperparameter Sensitivity

Let $\mathcal{I} = \{5, 10, 20, 100, 1000\}$ be the candidate refresh intervals. The index j identifies one of the 16 model and task settings, and $a_j(S)$ denotes its raw validation accuracy at refresh interval S. For a positive accuracy range across the candidates, we define the normalized score:

$$q_j(S) = \frac{a_j(S) - \min_{u \in \mathcal{I}} a_j(u)}{\max_{u \in \mathcal{I}} a_j(u) - \min_{u \in \mathcal{I}} a_j(u)}, \qquad S \in \mathcal{I}. \tag{34}$$

Table 9 reports these per-setting Min-Max scores. At $S = 20$, they range from 0.91 to 1.00 across all 16 settings, supporting a common default of 20 coarse steps. The per-task intervals used for the main results are listed in Table 8.

We also vary the coarse and fine phase lengths $M, N \in \{10, 30, 60\}$ epochs, following Algorithm 1. In this schedule, each epoch comprises eight minibatch updates, so C60F60 allocates 480 coarse and 480 fine updates per cycle. For each schedule, we average validation accuracy over the 16 model and task settings, then apply Min-Max normalization across the nine schedules. Table 10 shows that C60F60 achieves the highest average validation accuracy.

Table 9: Sensitivity to the refresh interval S in coarse steps. Entries are the normalized scores $q_j(S)$ in Eq. (34). Bold marks the highest score for each model and task.

GPT-2-XL					OPT-2.7B				
S	MNLI	QNLI	CoLA	SST-2	S	MNLI	QNLI	CoLA	SST-2
5	**1.00**	**1.00**	0.94	**1.00**	5	**1.00**	**1.00**	**1.00**	**1.00**
10	0.95	**1.00**	**1.00**	0.94	10	**1.00**	0.92	0.94	0.91
20	0.92	**1.00**	**1.00**	**1.00**	20	**1.00**	**1.00**	**1.00**	0.91
100	0.34	0.27	0.35	0.38	100	0.31	0.36	0.45	0.22
1000	0.00	0.00	0.00	0.00	1000	0.00	0.00	0.00	0.00
LLaMA-3-8B					OPT-13B				
S	RTE	BoolQ	WiC	CB	S	RTE	BoolQ	WiC	CB
5	**1.00**	**1.00**	0.96	**1.00**	5	**1.00**	**1.00**	**1.00**	**1.00**
10	0.96	**1.00**	0.93	0.91	10	0.92	**1.00**	**1.00**	**1.00**
20	**1.00**	**1.00**	**1.00**	**1.00**	20	**1.00**	0.92	0.93	**1.00**
100	0.41	0.53	0.61	0.57	100	0.61	0.67	**1.00**	0.77
1000	0.00	0.00	0.00	0.00	1000	0.00	0.00	0.00	0.00

Table 10: Sensitivity to the V-cycle schedule. Rows give coarse phase length M and columns give fine phase length N, both in epochs. Scores are Min-Max normalized averages over 16 model and task settings. Bold marks the best schedule.

	$N = 60$	$N = 30$	$N = 10$
$M = 60$	**1.00**	0.65	0.44
$M = 30$	0.91	0.76	0.34
$M = 10$	0.82	0.71	0.00

G Efficiency Analysis

In Table 11, ENGINE has the highest query efficiency in all four settings and the highest wall-clock efficiency on GPT-2-XL and OPT-2.7B.

Table 11: Query and wall-clock speed-ups at common target accuracies. Each entry divides the largest query count or time in its column by that method's value, assigning the least efficient method $1.0\times$. Higher is better, and bold marks the best result.

Method	GPT-2-XL SST-2 (70%)		OPT-2.7B QNLI (70%)		LLaMA-3-8B CB (65%)		OPT-13B CB (65%)	
	Queries	Time	Queries	Time	Queries	Time	Queries	Time
MeZO	1.0	1.0	1.0	1.0	1.2	1.2	2.3	2.5
MeZO-Adam	10.3	9.8	9.7	9.2	1.0	1.0	1.0	1.0
MeZO-SVRG	3.9	2.1	4.1	4.5	2.1	1.8	3.2	2.9
LOZO	1.6	4.7	2.2	6.2	2.2	**6.4**	3.5	**7.3**
HiZOO	7.5	3.2	8.0	3.7	1.6	1.3	1.9	1.5
LOREN	12.7	9.5	11.7	8.3	4.6	2.0	5.0	3.4
ENGINE	**16.5**	**11.4**	**22.2**	**16.7**	**10.0**	4.1	**5.3**	4.0

H Limitations and Future Work

Adaptive query allocation. The refresh interval S and coarse/fine phase lengths are fixed in advance. Curvature drift can make cached fine-space information stale, while the relative benefit of

each phase can change during training. Adapting refreshes and phase lengths to curvature drift or validation-loss plateaus could improve query allocation.

Learning-rate selection. The branch learning rates $\eta_{\mathbf{B}}$, $\eta_{\mathbf{A}}$, and η_f share one value to limit tuning, leaving separate rates to be explored.

Access requirements and task coverage. ENGINE requires access to model parameters to perturb weights and transfer states between spaces, limiting its use through prediction-only APIs. Our experiments cover classification fine-tuning across four models and 16 model-task settings. Evaluation on generation tasks would establish whether the observed benefits extend to other fine-tuning objectives.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state all claims, contributions, and experimental scope (abstract and Sections 1 and 2).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Three limitations are explicitly discussed in Appendix H, each paired with a concrete future direction.

   Guidelines:

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

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Assumptions are stated with the results and their setup. Complete structural proofs are provided in Appendix B, and statistical and conditional convergence proofs in Appendix C.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Algorithm 1, Appendix D (implementation details), Appendix E (detailed experiments and full results), and Appendix F (setup and hyperparameter tables) together provide sufficient information to reproduce all results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code is available at the repository `https://github.com/AI4Science-WestlakeU/ENGINE`. All benchmark datasets (GLUE, SuperGLUE) are publicly available.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix F gives the datasets, reproduction protocol, ablation settings, and hyperparameter grids. The evaluation protocol follows LOREN.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Main-body Table 2 reports mean accuracy over 5 independent seeds, and full mean $\pm$ std results are provided in Appendix E (Table 5).

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: GPU model (NVIDIA A100 80GB) is documented in Appendix F. Appendix G reports common-target query and wall-clock efficiency, and Figure 1 reports speed-ups relative to LOREN.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work proposes an optimization algorithm for LLM fine-tuning. It raises no ethical concerns and complies fully with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper presents a foundational optimization algorithm with no direct path to negative societal impacts. There are no application-specific harms to discuss.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release new pre-trained models or datasets, so no safeguards are required.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All assets are cited with their original papers. Licenses: GPT-2-XL (MIT), OPT-2.7B/13B (Meta OPT License), LLaMA-3-8B (Meta LLaMA 3 Community License), GLUE/SuperGLUE tasks (CC-BY 4.0 or task-specific open licenses), LOREN baseline code (MIT).

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The ENGINE implementation is released as a new software asset at `https://github.com/AI4Science-WestlakeU/ENGINE`, documented with a README covering installation, supported models/tasks, and run commands.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper involves no human subjects and therefore no IRB review is required.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs (GPT-2-XL, OPT, LLaMA-3) are the objects of study, not components of the proposed method itself. No LLM was used to generate research content.

    Guidelines:

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