ZO-F2: Low-Variance Fisher Preconditioner via Bilinear Estimation for Zeroth-Order Optimization

Hiroshi Sawada, Yuya Hikima, Kenta Niwa
NTT Communication Science Laboratories, NTT, Inc., Kyoto, Japan
hrsh.sawada@ntt.com, hikima.y.e606@m.isct.ac.jp, kenta.niwa@ntt.com

Abstract

We propose a novel zeroth-order (ZO) optimization method, ZO-F2, where Fisher information matrix (FIM)-based preconditioners are estimated in a low-variance manner via bilinear forms. Using FIM rather than Hessian allows for a larger number of estimator samples by pairwise combinations of perturbations. We further propose a numerically stable whitening-based update rule for full-matrix preconditioners, even when the FIM estimate is ill-conditioned. We provide theoretical analyses on the variance, the whitening residual, and convergence under standard and reasonable assumptions. Experimental results on black-box training of diverse models from scratch support our proposals and demonstrate the superiority of ZO-F2 over existing methods in terms of accuracy and computational efficiency.

1 Introduction

First-order (FO) backpropagation-based optimization [1–3] is a standard technique for training neural networks, in which intermediate results from the forward pass are stored for use in the backward pass to compute (stochastic) gradients $\boldsymbol{g} \in \mathbb{R}^d$ of the d-dimensional learnable parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$. However, when neural networks are directly implemented in hardware, such as optical [4–6] and quantum [7, 8] neural networks, backpropagation cannot be applied because these intermediate results cannot be observed. Moreover, computing gradients for large models using backpropagation requires a substantial amount of memory. To address these challenges, zeroth-order (ZO) optimization [9–14] has been widely adopted. It computes approximate gradients $\hat{\boldsymbol{g}} \in \mathbb{R}^d$ using perturbations $\boldsymbol{u} \in \mathbb{R}^d$, in a black-box manner for hardware-implemented neural networks [4, 15–18], and in a memory-efficient manner for fine-tuning large language models (LLMs) [19, 20].

This paper advances second-order (SO) approaches for ZO optimization. As neural networks become deeper, the curvature of their parameter space becomes increasingly ill-conditioned [21, 22]. SO approaches, such as Newton's method [23–25] and the natural gradient method [26–28], mitigate this issue by preconditioning the gradient using the Hessian $\mathbf{H} \in \mathbb{R}^{d\times d}$ or the Fisher information matrix (FIM) $\mathbf{F} \in \mathbb{R}^{d\times d}$. In a white-box, memory-abundant (FO) setting, the Hessian and FIM can be computed exactly with access to intermediate results [24, 28, 29]. However, in ZO optimization, the Hessian $\mathbf{H}$ and the FIM $\mathbf{F}$, like the gradient $\boldsymbol{g}$, can only be estimated in a black-box manner. Table 1 summarizes existing methods in this regard. Earlier methods [30–33] precondition the approximate gradient $\hat{\boldsymbol{g}}$, which is a straightforward extension of FO preconditioning but may amplify the approximation error in $\hat{\boldsymbol{g}}$. In contrast, recent methods [34–36] precondition perturbations $\boldsymbol{u}$ to mitigate this issue, and the concept of natural perturbations is proposed as an FIM preconditioner in [35]. However, the methods in [34, 36] consider only diagonal Hessian approximations and do not address the full Hessian, which may lead to an ill-conditioned matrix inversion. The method in [35] computes the FIM via the finite difference method, requiring d extra function evaluations (as indicated by "extra" and "N/A" in Table 1), which differ from those based on random perturbations used to compute the approximate gradient. Overall, these techniques are not yet sufficiently mature, and more advanced SO approaches for ZO optimization need to be developed.

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

Table 1: Existing ZO methods and the proposed method in terms of preconditioning and estimation properties. $\mathbf{H}$: Hessian, $\mathbf{F}$: FIM, $\mathbf{W}$: perturbation preconditioner.

Methods	[30, 31]	[32, 33]	[34, 36]	[35]	Proposed
Preconditioned quantity	Approximate gradient $\hat{\boldsymbol{g}}$ (3)		Perturbations $\boldsymbol{u}$ (5)		
Estimation target $\mathbf{G}$	$\mathbf{H}$	$\mathbf{H}$	$\mathbf{W}^\mathsf{T}\mathbf{H}\mathbf{W}$	$\mathbf{F}$	$\mathbf{W}^\mathsf{T}\mathbf{F}\mathbf{W}$
Matrix structure of $\mathbf{G}$	full	full	diagonal	full	full
Estimator form	bilinear (6)	quadratic (7)	quadratic	one-hot (11)	bilinear
Estimation variance (Thm. 3.1) $\downarrow$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^4)$	$\mathcal{O}(d^4)$	N/A	$\mathcal{O}(d^3)$
Required perturbations $\downarrow$	$q \geq 2$	$q \geq 1$	$q \geq 1$	extra $q = d$	$q \geq 2$
Constructed samples $\uparrow$	$\mathcal{O}(q)$	$\mathcal{O}(q)$	$\mathcal{O}(q)$	N/A	$\mathcal{O}(q^2)$

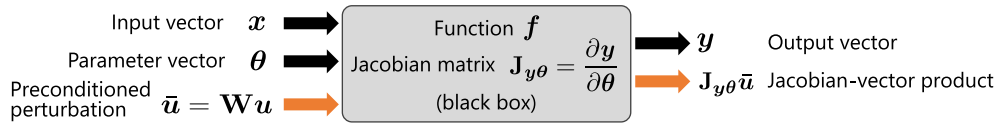


Figure 1: Black-box function $\boldsymbol{f}$: we can query the function value $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$, and approximately compute the Jacobian-vector product $\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\bar{\boldsymbol{u}}$ for perturbation $\bar{\boldsymbol{u}}$ by the finite difference method.

This paper proposes ZO-F2, a novel method for preconditioning perturbations $\boldsymbol{u}$ based on implicitly estimating the FIM $\mathbf{F}$ in a bilinear form. Our contributions are summarized as follows.

Low-variance target estimation (Subsection 3.1): We first analyze the estimation variance, which has not been explicitly analyzed in terms of d in previous works [30–34, 36], as detailed in Subsection 2.3. Our analysis (Theorem 3.1) reveals that the bilinear form achieves lower variance, scaling as $\mathcal{O}(d^3)$, compared to the quadratic form, which scales as $\mathcal{O}(d^4)$, as summarized in Table 1. Moreover, by exploiting the decomposition of the FIM into the Jacobian matrix and the output FIM (11), which allows the bilinear form to be computed via independent Jacobian-vector products as (14), the number of constructed samples for the estimation with q perturbations scales as $\mathcal{O}(q^2)$ for the FIM. This contrasts with $\mathcal{O}(q)$ for the Hessian, whose estimation is based on the Taylor expansion.

Robust computation of preconditioner (Subsection 3.2): We propose a robust update rule for the perturbation preconditioner $\mathbf{W}$ based on whitening. By avoiding the explicit computation of the FIM estimate $\hat{\mathbf{F}}$, the whitening strategy circumvents ill-conditioned matrix inversions, a problem that does not arise in the diagonal case [34, 36]. Furthermore, we account for the non-negligible estimation error and theoretically show in Theorem 3.3 that the effect of the estimation error can be controlled by a threshold τ used to clip negative or small eigenvalues in the clipping operator (19).

Strong empirical validations (Section 4): We empirically verify the advantages of ZO-F2 on diverse models, including a linear model, an RNN, a CNN, and a ResNet. The proposed ZO-F2 consistently outperforms existing methods, and the performance gap widens as the number of parameters d and perturbations q increase, which is consistent with our proposals and analyses.

Table 4 in Appendix A lists the notations used in this paper.

2 Zeroth-order (ZO) optimization

2.1 Gradient approximation

As shown in Figure 1, we consider a function (e.g., a neural network) $\boldsymbol{f}$ with a learnable parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$ that transforms an input vector $\boldsymbol{x} \in \mathbb{R}^{d_\text{in}}$ to an output vector $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^{d_\text{out}}$. Given a training dataset $\{(\boldsymbol{x}, \boldsymbol{t})\}$ consisting of input-target pairs $(\boldsymbol{x}, \boldsymbol{t})$, the per-sample $\ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})$, full-batch $\mathcal{L}(\boldsymbol{\theta})$, and mini-batch $\mathcal{L}_\mathcal{D}(\boldsymbol{\theta})$ ($\mathcal{D} \subseteq \{(\boldsymbol{x}, \boldsymbol{t})\}$) loss functions are defined as

$$\ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t}) := -\log p_{\boldsymbol{\theta}}(\boldsymbol{t} \mid \boldsymbol{x}) = -\log p(\boldsymbol{t} \mid \boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})}\,,$$
$$\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{|\{(\boldsymbol{x}, \boldsymbol{t})\}|} \sum_{(\boldsymbol{x},\boldsymbol{t}) \in \{(\boldsymbol{x},\boldsymbol{t})\}} \ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})\,, \quad \mathcal{L}_\mathcal{D}(\boldsymbol{\theta}) := \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},\boldsymbol{t}) \in \mathcal{D}} \ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})\,, \tag{1}$$

where $p(\cdot \mid \boldsymbol{y})$ is the probability distribution conditioned on the output vector $\boldsymbol{y}$, e.g., a categorical distribution with probabilities $\mathrm{softmax}(\boldsymbol{y})$ for a classification task with logits $\boldsymbol{y}$.

The gradient $\boldsymbol{g} := \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) \in \mathbb{R}^d$ of the mini-batch loss function with respect to the parameter vector $\boldsymbol{\theta}$ can be approximated via finite differences with a small smoothing parameter $\mu > 0$ as

$$\boldsymbol{g} = \mathbf{I}_d \, \boldsymbol{g} = \mathbb{E}_{\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\boldsymbol{u} \left(\boldsymbol{u}^\mathsf{T} \boldsymbol{g} \right) \right] \approx \mathbb{E}_{\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\boldsymbol{u} \frac{\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} + \mu \boldsymbol{u}) - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} - \mu \boldsymbol{u})}{2\mu} \right] . \tag{2}$$

Eq. (2) highlights a close relationship between forward gradient [37, 38] (i.e., directional derivative-based gradient estimation) and ZO optimization [12, 13, 39]. The equalities "=" indicate that the forward gradient $\boldsymbol{u} \left(\boldsymbol{u}^\mathsf{T} \boldsymbol{g} \right)$ with perturbations $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ is an unbiased estimator of $\boldsymbol{g}$. The approximation "$\approx$" indicates that the directional derivative $\boldsymbol{u}^\mathsf{T} \boldsymbol{g} = \lim_{\mu \to 0} \frac{\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} + \mu \boldsymbol{u}) - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} - \mu \boldsymbol{u})}{2\mu}$ can be approximated by the finite difference method [40, 20, 41]. Furthermore, in practice, each expectation term in (2) is estimated via a Monte Carlo approximation using q perturbations $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q$ sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. Thus, we compute the approximate gradient as

$$\hat{\boldsymbol{g}} := \frac{1}{q} \sum_{i=1}^{q} \boldsymbol{u}_i \frac{\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} + \mu \boldsymbol{u}_i) - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} - \mu \boldsymbol{u}_i)}{2\mu} , \quad \boldsymbol{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) . \tag{3}$$

The standard ZO optimization procedure iterates the computation of the approximate gradient $\hat{\boldsymbol{g}}$ via (3) and updates the parameter vector as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \, \hat{\boldsymbol{g}}$ with learning rate η.

2.2 Second-order (SO) approaches for ZO optimization

Existing SO approaches for ZO optimization can be categorized into two types. The first approach [30–33] preconditions the approximate gradient $\hat{\boldsymbol{g}}$ with a preconditioner $\mathbf{P} \in \mathbb{R}^{d \times d}$, followed by updating the parameter vector as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \, \mathbf{P} \hat{\boldsymbol{g}}$, analogous to FO optimization [23–28]. Typical preconditioners include the regularized inverses of the Hessian $\mathbf{H}$, which is symmetric, and the FIM $\mathbf{F}$, which is positive semidefinite, with a damping (regularization) coefficient $\gamma > 0$:

$$\mathbf{P} := \left(\gamma \, \mathbf{I}_d + \mathbf{H} \right)^{-1} \in \mathbb{R}^{d \times d} \quad \text{or} \quad \mathbf{P} := \left(\gamma \, \mathbf{I}_d + \mathbf{F} \right)^{-1} \in \mathbb{R}^{d \times d} . \tag{4}$$

Appendix B details the derivation of these forms and explores the relationship between $\mathbf{H}$ and $\mathbf{F}$. The second approach [34–36] instead first preconditions the perturbations $\boldsymbol{u}_i$ with a preconditioner $\mathbf{W} \in \mathbb{R}^{d \times d}$ satisfying $\mathbf{W}\mathbf{W}^\mathsf{T} = \mathbf{P}$, i.e., $\mathbf{W}$ is a matrix square root of $\mathbf{P}$, and then computes the already-preconditioned approximate gradient $\hat{\bar{\boldsymbol{g}}}$ using the preconditioned perturbations $\bar{\boldsymbol{u}}_i$ as

$$\hat{\bar{\boldsymbol{g}}} := \frac{1}{q} \sum_{i=1}^{q} \bar{\boldsymbol{u}}_i \frac{\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} + \mu \bar{\boldsymbol{u}}_i) - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} - \mu \bar{\boldsymbol{u}}_i)}{2\mu} , \quad \bar{\boldsymbol{u}}_i = \mathbf{W} \boldsymbol{u}_i , \quad \boldsymbol{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) , \tag{5}$$

followed by updating the parameter vector as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \, \hat{\bar{\boldsymbol{g}}}$. The relationship $\mathbf{W}\mathbf{W}^\mathsf{T} = \mathbf{P}$ between the gradient preconditioner $\mathbf{P}$ and the perturbation preconditioner $\mathbf{W}$ becomes clear from the first equation in (5), where $\bar{\boldsymbol{u}}_i$ appears both as a weighting vector and within the finite-difference term. The second approach is preferable, as function evaluations are performed with already-preconditioned perturbations, thereby avoiding the amplification of the approximation error in $\hat{\boldsymbol{g}}$ that may occur in the first approach.

2.3 Existing methods for estimating Hessian/FIM

Let $\mathbf{G} \in \mathbb{R}^{d \times d}$ be a symmetric matrix, which corresponds to the estimation target summarized in Table 1. One-sample estimators used in existing methods [30–34, 36], as well as in our proposed method, can be expressed in the following unified forms:

$$\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v}) := \left(\boldsymbol{u}^\mathsf{T} \mathbf{G} \boldsymbol{v} \right) \boldsymbol{u} \boldsymbol{v}^\mathsf{T} \in \mathbb{R}^{d \times d} \qquad \text{(bilinear form)} , \tag{6}$$

$$\tilde{\mathbf{G}}(\boldsymbol{u}) := \frac{1}{2} \left(\boldsymbol{u}^\mathsf{T} \mathbf{G} \boldsymbol{u} \right) \left(\boldsymbol{u} \boldsymbol{u}^\mathsf{T} - \mathbf{I}_d \right) \in \mathbb{R}^{d \times d} \qquad \text{(quadratic form)} . \tag{7}$$

The former uses two independent perturbations $\boldsymbol{u}, \boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ while the latter uses one perturbation $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. Both estimators are unbiased, i.e., $\mathbb{E}_{\boldsymbol{u}, \boldsymbol{v}} \left[\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v}) \right] = \mathbf{G}$ and $\mathbb{E}_{\boldsymbol{u}} \left[\tilde{\mathbf{G}}(\boldsymbol{u}) \right] = \mathbf{G}$, as shown in Appendix C. In [30, 31], the Hessian is estimated using (6) by setting $\mathbf{G} = \mathbf{H}$ and

$$\boldsymbol{u}^\mathsf{T} \mathbf{G} \boldsymbol{v} \approx \frac{\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} + \mu \boldsymbol{u} + \mu \boldsymbol{v}) - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} + \mu \boldsymbol{u}) - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} - \mu \boldsymbol{u} + \mu \boldsymbol{v}) + \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta} - \mu \boldsymbol{u})}{2\mu^2} . \tag{8}$$

In [32, 33], the Hessian is estimated using (7) by setting $\mathbf{G} = \mathbf{H}$ and

$$\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u} \approx \frac{\mathcal{L}_\mathcal{D}(\boldsymbol{\theta} + \mu\boldsymbol{u}) + \mathcal{L}_\mathcal{D}(\boldsymbol{\theta} - \mu\boldsymbol{u}) - 2\mathcal{L}_\mathcal{D}(\boldsymbol{\theta})}{\mu^2}\,. \tag{9}$$

In both (8) and (9), the approximation "$\approx$" is due to the Taylor expansion of $\mathcal{L}_\mathcal{D}(\boldsymbol{\theta})$ up to the second order. In these methods [30–33], the perturbations $\boldsymbol{u}$, $\boldsymbol{v}$ are used without preconditioning. The methods in [34, 36] instead use preconditioned perturbations $\mathbf{W}\boldsymbol{u}$ to estimate the Hessian via (7) by specifying $\mathbf{G} = \mathbf{W}^\mathsf{T}\mathbf{H}\mathbf{W}$ and

$$\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u} = \bar{\boldsymbol{u}}^\mathsf{T}\mathbf{H}\bar{\boldsymbol{u}} \approx \frac{\mathcal{L}_\mathcal{D}(\boldsymbol{\theta} + \mu\bar{\boldsymbol{u}}) + \mathcal{L}_\mathcal{D}(\boldsymbol{\theta} - \mu\bar{\boldsymbol{u}}) - 2\mathcal{L}_\mathcal{D}(\boldsymbol{\theta})}{\mu^2}\,, \quad \bar{\boldsymbol{u}} = \mathbf{W}\boldsymbol{u}\,. \tag{10}$$

These methods are preferable as they incorporate preconditioned perturbations. However, they consider only the diagonal Hessian and do not consider the full Hessian, which may lead to numerical instability due to a possible large condition number of the Hessian estimate $\hat{\mathbf{H}} = \mathbf{W}^{-\mathsf{T}}\hat{\mathbf{G}}\mathbf{W}^{-1}$.

Unlike the above estimators, the method in [35] computes the FIM by decomposing it as

$$\mathbf{F} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\Big[\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}^\mathsf{T}\mathbf{F}_{\boldsymbol{y}}\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\big|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})}\Big]\,,\ \ \mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}} := \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{d_\mathrm{out}\times d}\,, \tag{11}$$

where $\mathbf{F}_{\boldsymbol{y}} \in \mathbb{R}^{d_\mathrm{out}\times d_\mathrm{out}}$ is the FIM with respect to the output vector, which can be computed in closed form for typical distributions as shown in (42) in Appendix B. The Jacobian matrix $\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}$ is computed column-wise by the finite difference method with one-hot vectors $\boldsymbol{e}_i := [0,\dots,1\,(i\text{-th entry}),\dots,0]^\mathsf{T} \in \mathbb{R}^d$. A drawback of this method is that it requires $\mathcal{O}(d)$ additional function evaluations, which differ from the random perturbations used for estimating the gradient (5), limiting its applicability to large-scale models with a large number of parameters d.

3 Our proposed method

We propose ZO-F2 that achieves low-variance estimation with the bilinear form (6) for the estimation target $\mathbf{G} = \mathbf{W}^\mathsf{T}\mathbf{F}\mathbf{W}$. We employ the FIM $\mathbf{F}$ instead of the Hessian $\mathbf{H}$ for more constructed samples with $q \geq 2$ perturbations. We also propose a whitening strategy that robustly updates the full-matrix preconditioner $\mathbf{W}$. Algorithm 1 shows the whole procedure.

Algorithm 1 ZO-F2: ZO optimization with Fisher preconditioner via bilinear estimation

Input: training dataset $\{(\boldsymbol{x}, \boldsymbol{t})\}$, hyperparameters q, μ, η, α, γ, τ
Initialize parameter vector $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, averaged estimate $\hat{\mathbf{G}}_0 = \mathbf{I}_d$, preconditioner $\mathbf{W}_0 = \mathbf{I}_d$
for step $t = 0, 1, \dots, T-1$ **do**
  Sample mini-batch $\mathcal{D}_t$ from $\{(\boldsymbol{x}, \boldsymbol{t})\}$
  Compute the approximate gradient $\hat{\tilde{\boldsymbol{g}}}_t$ by (5) with q perturbations $\boldsymbol{u}_1, \dots, \boldsymbol{u}_q$
  Update the parameter vector by $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\,\hat{\tilde{\boldsymbol{g}}}_t$
  Compute $q(q-1)/2$ one-sample estimators $\tilde{\mathbf{G}}(\boldsymbol{u}_i, \boldsymbol{u}_j)$ by (6), (14), and (15)
  Compute the intermediate estimate $\hat{\mathbf{G}}_t^\mathrm{ie}$ by (16) and (18)
  Update the preconditioner $\mathbf{W}_{t+1}$ and the averaged estimate $\hat{\mathbf{G}}_{t+1}$ by (21) and (22)
end for

3.1 Low-variance estimation for FIM with bilinear form

The low-variance estimation of our method stems from two key design choices. The first is the use of the bilinear form (6) instead of the quadratic form (7). The second is the use of the FIM instead of the Hessian for the preconditioner basis.

We start by analyzing the first design choice, namely the estimation variance of the unified forms (6) and (7), which have not been explicitly analyzed in previous works [30–34, 36].

Assumption 3.1 (bounded spectral norm)**.** The spectral norm of a symmetric matrix $\mathbf{G} \in \mathbb{R}^{d\times d}$ is bounded as $\|\mathbf{G}\|_2 \leq C$.

Theorem 3.1 (estimation variance)**.** *For any symmetric matrix* $\mathbf{G}$ *satisfying Assumption 3.1, the Frobenius-norm variances of* $\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v})$ *in (6) and* $\tilde{\mathbf{G}}(\boldsymbol{u})$ *in (7) scale as*

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d),\boldsymbol{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\|\tilde{\mathbf{G}}(\boldsymbol{u},\boldsymbol{v})-\mathbf{G}\|_F^2\right] = \mathcal{O}(d^3) \qquad \textit{(bilinear form)}\,, \tag{12}$$

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\|\tilde{\mathbf{G}}(\boldsymbol{u})-\mathbf{G}\|_F^2\right] = \mathcal{O}(d^4) \qquad \textit{(quadratic form)}\,. \tag{13}$$

The proof is shown in Appendix C. This theorem shows that the bilinear form (6), which requires at least $q \geq 2$ perturbations, achieves lower-order variance than the quadratic form (7), which can be computed even with a single perturbation ($q = 1$). To exploit the low-variance property of the bilinear form in the FIM setting, we compute $\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v}$ in the bilinear form (6) approximately based on the FIM decomposition (11) and the Jacobian-vector product (JVP) as

$$\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v} = \boldsymbol{u}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{F}\mathbf{W}\boldsymbol{v} = \bar{\boldsymbol{u}}^\mathsf{T}\mathbf{F}\bar{\boldsymbol{v}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\left(\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\bar{\boldsymbol{u}}\right)^\mathsf{T}\mathbf{F}_{\boldsymbol{y}}\left(\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\bar{\boldsymbol{v}}\right)\Big|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})}\right]. \tag{14}$$

The JVPs $\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\bar{\boldsymbol{u}}, \mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\bar{\boldsymbol{v}}$ are approximately computed by the finite difference method as

$$\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}\bar{\boldsymbol{u}}\big|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})} \approx \frac{\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}+\mu\bar{\boldsymbol{u}})-\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}-\mu\bar{\boldsymbol{u}})}{2\mu}\,, \tag{15}$$

where the function evaluations $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}\pm\mu\bar{\boldsymbol{u}})$ are already performed in the computation of $\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}\pm\mu\bar{\boldsymbol{u}})$ for the approximate gradient (5). This approach to computing $\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v}$ in the bilinear form is fundamentally different from that for Hessian, which is based on the Taylor expansion as in (8).

Second, the use of the FIM enables us to increase the number of one-sample estimators to $q(q-1)/2 = \mathcal{O}(q^2)$ for q perturbations $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q$ by exploiting pairwise combinations of perturbations, i.e., we can construct estimators $\tilde{\mathbf{G}}(\boldsymbol{u}_1,\boldsymbol{u}_2), \tilde{\mathbf{G}}(\boldsymbol{u}_1,\boldsymbol{u}_3)\ldots, \tilde{\mathbf{G}}(\boldsymbol{u}_1,\boldsymbol{u}_q), \tilde{\mathbf{G}}(\boldsymbol{u}_2,\boldsymbol{u}_3)\ldots, \tilde{\mathbf{G}}(\boldsymbol{u}_{q-1},\boldsymbol{u}_q)$. In contrast, when using the Hessian as (8), the sample number is $\lfloor q/2 \rfloor = \mathcal{O}(q)$, i.e., constructing estimators $\tilde{\mathbf{G}}(\boldsymbol{u}_1,\boldsymbol{u}_2),\ldots,\tilde{\mathbf{G}}(\boldsymbol{u}_{q-1},\boldsymbol{u}_q)$ for even q, because the function evaluation $\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\boldsymbol{u}+\mu\boldsymbol{v})$ in (8) involves two perturbations. Let us define an $\mathcal{O}(q^2)$ aggregated estimator as

$$\hat{\mathbf{G}}^{\mathrm{agg}} = \frac{2}{q(q-1)}\sum_{i=1}^{q}\sum_{j=i+1}^{q}\frac{1}{2}\left[\tilde{\mathbf{G}}(\boldsymbol{u}_i,\boldsymbol{u}_j)+\tilde{\mathbf{G}}(\boldsymbol{u}_i,\boldsymbol{u}_j)^\mathsf{T}\right], \tag{16}$$

where averaging each estimator with its transpose ensures that $\hat{\mathbf{G}}^{\mathrm{agg}}$ is symmetric and remains unbiased for symmetric $\mathbf{G}$ as shown in Appendix D.1.

Theorem 3.2 (aggregated estimator variance)**.** *For any symmetric matrix* $\mathbf{G}$ *satisfying Assumption 3.1, the Frobenius-norm variance of* $\hat{\mathbf{G}}^{\mathrm{agg}}$ *in (16) scales as*

$$\mathbb{E}_{\{\boldsymbol{u}_k\}_{k=1}^q}\left[\|\hat{\mathbf{G}}^{\mathrm{agg}}-\mathbf{G}\|_F^2\right] = \mathcal{O}\left(\frac{d^2}{q}+\frac{d^3}{q^2}\right). \tag{17}$$

The proof is shown in Appendix D.2. The extra term $\mathcal{O}\left(d^2/q\right)$ arises from the reuse of perturbations across pairs. This term is subdominant whenever $d \gg q$, in which case the $\mathcal{O}\left(d^3/q^2\right)$ term dominates, confirming the theoretical advantage over using the Hessian, whose variance is $\mathcal{O}\left(d^3/q\right)$.

These two design choices jointly enable low-variance estimation of $\mathbf{G}$ at each step. To further reduce the variance, we perform an exponential moving average (EMA) over steps. Let $\hat{\mathbf{G}}_t$ denote the averaged estimate at the start of step t. We then compute an intermediate estimate $\hat{\mathbf{G}}_t^{\mathrm{ie}}$, which is used to update the averaged estimate $\hat{\mathbf{G}}_{t+1}$ for the next step, with an EMA coefficient $\alpha \in (0, 1]$ as

$$\hat{\mathbf{G}}_t^{\mathrm{ie}} = (1-\alpha)\,\hat{\mathbf{G}}_t + \alpha\,\hat{\mathbf{G}}^{\mathrm{agg}}\,. \tag{18}$$

3.2 Robust preconditioner update by whitening strategy

We propose a robust way to update the preconditioner $\mathbf{W}_{t+1}$ and the averaged estimate $\hat{\mathbf{G}}_{t+1}$ from the intermediate estimate $\hat{\mathbf{G}}_t^{\mathrm{ie}}$. This corresponds to line 9 of Algorithm 1, and we explicitly use step indices t and $t+1$ in what follows. We address two robustness issues. First, the intermediate estimate

$\hat{\mathbf{G}}_t^{\text{ie}}$ has non-negligible estimation error $\|\hat{\mathbf{G}}_t^{\text{ie}} - \mathbf{G}_t\|_F$ and may have negative eigenvalues, although the true $\mathbf{G}_t$ is positive semidefinite for the FIM. Second, the matrix inversion $(\gamma\,\mathbf{I}_d + \hat{\mathbf{F}})^{-1}$ related to (4) can be numerically unstable when the FIM estimate $\hat{\mathbf{F}}$ is ill-conditioned.

For the first issue, we define a clipping operation $\mathrm{clip}(\mathbf{A})$ for a symmetric matrix $\mathbf{A}$ as

$$\mathrm{clip}(\mathbf{A}) = \mathbf{U}\,\mathrm{diag}(\bar{\nu}_1, \ldots, \bar{\nu}_d)\,\mathbf{U}^\mathsf{T}, \quad \bar{\nu}_i = \max(\nu_i, \tau), \quad \mathbf{A} = \mathbf{U}\,\mathrm{diag}(\nu_1, \ldots, \nu_d)\,\mathbf{U}^\mathsf{T}, \tag{19}$$

where $\mathbf{U}$ is a matrix with eigenvectors satisfying $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_d$ and $\nu_1 \geq \cdots \geq \nu_d$ are eigenvalues. This ensures that $\mathrm{clip}(\mathbf{A})$ is positive definite with a clipping threshold $\tau > 0$. While a similar operation has been proposed in [32], we present a new analysis of the effect of τ later in Theorem 3.3.

For the second issue, we employ a whitening procedure instead of directly solving for the preconditioner. Specifically, the direct solving approach solves $\mathbf{W}_{t+1}\mathbf{W}_{t+1}^\mathsf{T} = (\gamma\,\mathbf{I}_d + \hat{\mathbf{F}}_t)^{-1}$ for $\mathbf{W}_{t+1}$, which follows from (4), $\mathbf{P} = \mathbf{W}\mathbf{W}^\mathsf{T}$, and $\hat{\mathbf{F}}_t = \mathrm{clip}(\mathbf{W}_t^{-\mathsf{T}}\hat{\mathbf{G}}_t^{\text{ie}}\mathbf{W}_t^{-1})$. However, this approach may suffer from a large condition number of $\gamma\,\mathbf{I}_d + \hat{\mathbf{F}}_t$. The whitening procedure, by contrast, is based on the fact that the optimal preconditioner $\mathbf{W}_{t+1}$ whitens the regularized FIM:

$$\mathbf{I}_d = \mathbf{W}_{t+1}^\mathsf{T}\,(\gamma\,\mathbf{I}_d + \mathbf{F}_t)\,\mathbf{W}_{t+1} = \gamma\,\mathbf{W}_{t+1}^\mathsf{T}\mathbf{W}_{t+1} + \mathbf{G}_{t+1}\,. \tag{20}$$

We therefore compute the following whitening target $\mathbf{R}_t$, and obtain the Cholesky decomposition

$$\mathbf{R}_t := \gamma\,\mathbf{W}_t^\mathsf{T}\mathbf{W}_t + \mathrm{clip}(\hat{\mathbf{G}}_t^{\text{ie}}) = \mathbf{L}_t\mathbf{L}_t^\mathsf{T}\,, \tag{21}$$

where $\mathbf{L}_t$ is a lower triangular matrix. We then update the preconditioner and averaged estimate as

$$\mathbf{W}_{t+1} = \mathbf{W}_t\mathbf{L}_t^{-\mathsf{T}}, \quad \hat{\mathbf{G}}_{t+1} = \mathbf{L}_t^{-1}\mathrm{clip}(\hat{\mathbf{G}}_t^{\text{ie}})\mathbf{L}_t^{-\mathsf{T}}\,. \tag{22}$$

We can verify that the update rule whitens the target as

$$\gamma\,\mathbf{W}_{t+1}^\mathsf{T}\mathbf{W}_{t+1} + \hat{\mathbf{G}}_{t+1} = \mathbf{L}_t^{-1}\left(\gamma\,\mathbf{W}_t^\mathsf{T}\mathbf{W}_t + \mathrm{clip}(\hat{\mathbf{G}}_t^{\text{ie}})\right)\mathbf{L}_t^{-\mathsf{T}} = \mathbf{L}_t^{-1}\mathbf{R}_t\mathbf{L}_t^{-\mathsf{T}} = \mathbf{I}_d\,. \tag{23}$$

Remark 3.1 (small condition number)**.** Our whitening strategy keeps the whitening target $\mathbf{R}_t$ close to $\mathbf{I}_d$ if the EMA coefficient α is small and the changes in $\hat{\mathbf{G}}_t$ and $\mathbf{W}_t$ from step t to $t+1$ are also small. As a result, the eigenvalues of $\mathbf{R}_t$ stay near one, yielding a small condition number and hence a numerically stable Cholesky decomposition. This contrasts with the direct approach based on $\gamma\,\mathbf{I}_d + \hat{\mathbf{F}}_t$, whose condition number can be much larger. Figure 2c provides experimental evidence for this. Furthermore, since γ is typically small, $\hat{\mathbf{G}}_t$ also stays close to $\mathbf{I}_d$ throughout the iterations from the initialization $\hat{\mathbf{G}}_0 = \mathbf{I}_d$, which supports Assumption 3.1 (bounded spectral norm).

Although the update rule (22) whitens the target, it does not perfectly whiten the true regularized FIM, $\gamma\,\mathbf{I}_d + \mathbf{F}_t$ in (20), and there remains a whitening residual. The following theorem analyzes the whitening residual and shows that it can be controlled by the clipping threshold τ.

Theorem 3.3 (whitening residual)**.** *Let $\mathbf{G}_t = \mathbf{W}_t^\mathsf{T}\mathbf{F}_t\mathbf{W}_t$ and $\mathbf{G}_{t+1} = \mathbf{W}_{t+1}^\mathsf{T}\mathbf{F}_t\mathbf{W}_{t+1}$ be the true matrices corresponding to the same true FIM $\mathbf{F}_t$ before and after the preconditioner update (22). Assume that the estimation error of the intermediate estimate $\hat{\mathbf{G}}_t^{\text{ie}}$ is bounded as $\|\hat{\mathbf{G}}_t^{\text{ie}} - \mathbf{G}_t\|_F \leq \delta$. The whitening residual is bounded as follows:*

$$\|\gamma\,\mathbf{W}_{t+1}^\mathsf{T}\mathbf{W}_{t+1} + \mathbf{G}_{t+1} - \mathbf{I}_d\|_F \leq \frac{\delta}{\tau} + \frac{1}{\tau}\left(\sum_{i=1}^{d}\max(0, \tau - \nu_i)^2\right)^{1/2}, \tag{24}$$

where ν_i, $i = 1, \ldots, d$, are the eigenvalues of $\hat{\mathbf{G}}_t^{\text{ie}}$ as defined in (19).

The proof is shown in Appendix E. This theorem suggests that choosing too small a value of τ risks poor performance when the estimation error δ is large, while a moderately large value of τ effectively controls the whitening residual and leads to stable performance improvement, as shown in Figures 2b and 4b.

3.3 Convergence analysis

The following theorem guarantees the convergence of Algorithm 1 to a stationary point. The formal descriptions of the assumptions and the complete proof are provided in Appendix F.

Theorem 3.4 (convergence rate)**.** *Suppose that every mini-batch loss is L-smooth (Assumption F.1), the stochastic gradient noise has variance bounded by σ^2 (Assumption F.2), and the minimum eigenvalue of $\mathbf{W}_t\mathbf{W}_t^\mathsf{T}$ satisfies $\lambda_{\min}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T}) \geq m > 0$ for all t (Assumption F.3). Let $\{\boldsymbol{\theta}_t\}_{t=0}^T$ be the iterates generated by Algorithm 1, and let R be a random variable sampled uniformly from $\{0, \dots, T-1\}$ and $\eta = \min\left(\frac{m\gamma^2}{4L(d+2)}, T^{-\frac{1}{2}}d^{-\frac{1}{2}}\right)$. Then, we have*

$$\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}_R)\|^2\right] \leq \frac{4(\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*)}{mT}\left(\frac{4L(d+2)}{m\gamma^2} + T^{\frac{1}{2}}d^{\frac{1}{2}}\right) + \frac{\mu^2L^2(d+3)^3}{2m^2\gamma^3} + \frac{4L(d+2)\sigma^2}{T^{\frac{1}{2}}d^{\frac{1}{2}}m\gamma^2} + \frac{\mu^2L^3(8d+6d^2+d^3)}{T^{\frac{1}{2}}d^{\frac{1}{2}}m\gamma^3}, \tag{25}$$

where $\mathcal{L}^* := \min_{\boldsymbol{\theta}\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta})$. Consequently, with sufficiently small μ, the iteration complexity required to achieve $\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}_R)\|^2\right] \leq \epsilon^2$ is $T = \mathcal{O}(\epsilon^{-4}d)$.

Remark 3.2. The proof strategy largely follows that of [34], where only a diagonal Hessian approximation is considered under the assumption that its diagonal elements are bounded between prescribed lower and upper bounds. Our proposed method and analysis extends this to full matrices and introduces the whitening procedure with eigenvalue clipping for the robust preconditioner update in Subsection 3.2. This procedure enables us to avoid the prescribed upper-bound assumption in [34] and to derive Lemma F.1 using the damping coefficient γ, which is naturally introduced in the SO approach. However, we still require the lower-bound assumption in Assumption F.3.

3.4 Practical considerations

Central and forward differences: The finite-difference approximations of directional derivatives used to compute approximate gradients in (3) and (5) are based on the central difference. The resulting perturbed loss function evaluations can be reused for the Hessian estimators (8), (9), and (10). As an alternative to the central difference, the forward difference is sometimes preferred in practice, especially when the number of perturbations q is large, because it halves the number of function evaluations. The forward-difference variant of (5) is given by

$$\hat{\bar{\boldsymbol{g}}} := \frac{1}{q}\sum_{i=1}^{q}\bar{\boldsymbol{u}}_i\frac{\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\bar{\boldsymbol{u}}_i) - \mathcal{L}_\mathcal{D}(\boldsymbol{\theta})}{\mu}\,, \quad \bar{\boldsymbol{u}}_i = \mathbf{W}\boldsymbol{u}_i\,, \quad \boldsymbol{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)\,. \tag{26}$$

The FIM estimation based on (14) is agnostic to the choice between central and forward differences. For the Hessian estimators, only (8) admits a forward-difference variant as

$$\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v} \approx \frac{\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\boldsymbol{u}+\mu\boldsymbol{v}) - \mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\boldsymbol{u}) - \mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\boldsymbol{v}) + \mathcal{L}_\mathcal{D}(\boldsymbol{\theta})}{\mu^2}\,, \tag{27}$$

whose perturbed loss function evaluations $\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\boldsymbol{u})$ and $\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}+\mu\boldsymbol{v})$ can be reused to compute directional derivatives along both $\boldsymbol{u}$ and $\boldsymbol{v}$, whereas in (8) they are reused only along $\boldsymbol{u}$. In contrast to the adaptability of (8), the Hessian estimators (9) and (10) inherently require the central difference. Table 2 summarizes the above discussion with respect to the required number of function evaluations. The table also summarizes the number of estimator samples obtained, which illustrates the sample efficiency of our proposed FIM estimation.

Block-diagonal preconditioner: In practical settings where the number of parameters d is in the order of thousands or larger, storing and computing the preconditioner $\mathbf{W}$ becomes infeasible. We therefore approximate the preconditioner by a block-diagonal matrix

$$\mathbf{W}_t = \begin{bmatrix} \mathbf{W}^{(1)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{W}^{(b)} \end{bmatrix} \tag{28}$$

with matrix blocks $\mathbf{W}^{(i)}$, $i = 1, \dots, b$, whose sizes are not greater than $d_{\max} \times d_{\max}$. As shown in Remark F.1, this approximation does not affect the convergence analysis (Theorem 3.4). Table 5 shows an empirical case where $d_{\max} \geq 528$ exhibited negligible performance degradation.

Table 2: Required number of function evaluations ($\times$ mini-batch size) and the number of constructed samples for estimate $\tilde{\mathbf{G}}$ and the approximate gradient $\hat{\tilde{\boldsymbol{g}}}$ with q perturbations for each estimator

Estimator	Hessian quadratic	Hessian bilinear		FIM bilinear (14)	
Difference	central (9), (10)	central (8)	forward (27)	central	forward
Function evaluations	$2q+1$	$2q$	$\lfloor 3q/2 \rfloor + 1$	$2q+1$	$q+1$
Constructed samples for $\tilde{\mathbf{G}}$	q	$\lfloor q/2 \rfloor$	$\lfloor q/2 \rfloor$	$q(q-1)/2$	$q(q-1)/2$
Constructed samples for $\hat{\tilde{\boldsymbol{g}}}$	q	$\lfloor q/2 \rfloor$	q	q	q

Less frequent preconditioner update: The update of the preconditioner $\mathbf{W}_{t+1}$ requires the eigenvalue decomposition in (19), which is computationally demanding. To alleviate this, we introduce a hyperparameter $T_{\mathrm{ud}} \in \{2, 3, \dots\}$ for the update frequency. Specifically, the update of $\mathbf{W}_{t+1}$ and its corresponding update of $\hat{\mathbf{G}}_{t+1}$ are performed only when $t \bmod T_{\mathrm{ud}} = 0$; otherwise, we skip line 9 of Algorithm 1 and simply set $\mathbf{W}_{t+1} = \mathbf{W}_t$ and $\hat{\mathbf{G}}_{t+1} = \hat{\mathbf{G}}_t^{\mathrm{ie}}$. As shown in Remark F.1, this less-frequent update does not affect the convergence analysis (Theorem 3.4).

FIM initialization: Although the FIM computation in (11) by [35] requires $\mathcal{O}(d)$ additional function evaluations, it can be made practical by performing it only once after a warm-up period for initialization, providing a relatively accurate estimate of the FIM. Specifically, we force $\mathbf{W}_t = \mathbf{I}_d$ unchanged during the warm-up period. At the end of the warm-up period $t = t_{\text{w-end}}$, we compute $\hat{\mathbf{G}}_t^{\mathrm{ie}} = \mathbf{F}$ by (11) and then compute $\mathbf{W}_{t+1}$ and $\hat{\mathbf{G}}_{t+1}$ by (21) and (22). Subsequent updates with the proposed estimator enable the method to adapt to changes in the FIM over time.

4 Experiments

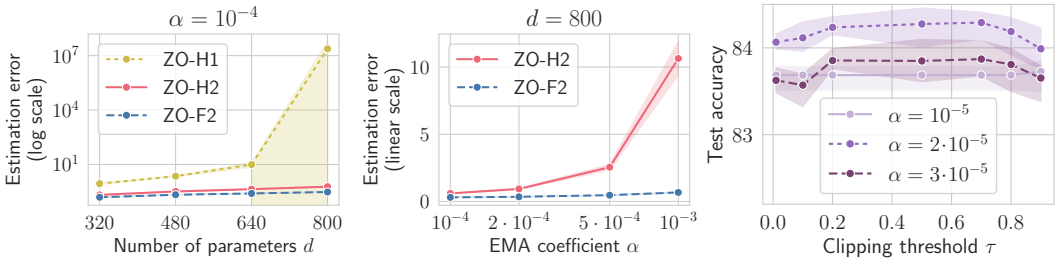
We compare four ZO optimization methods. ZO-I is the most standard method without a preconditioner, i.e., $\mathbf{W} = \mathbf{I}_d$. The remaining three methods are SO approaches with preconditioned perturbations. They differ in estimation target and form as ZO-H1 (Hessian quadratic), ZO-H2 (Hessian bilinear), and ZO-F2 (FIM bilinear), which correspond to the entries in Table 2. The number ("1" or "2") in each method name denotes the quadratic or bilinear form. For the Hessian-based methods, ZO-H1 and ZO-H2, we extend the computation of the intermediate estimate (18) by appropriately modifying $\tilde{\mathbf{G}}$, and apply the proposed whitening strategy. For these Hessian-based methods, the clipping in (19) is modified as $\bar{\nu}_i = \max(|\nu_i|, \tau)$, where the use of the absolute value follows that in [21], which suggests that it helps escape from saddle points with sign-mixed eigenvalues.

We performed experiments with four combinations of architecture and dataset/task, which are explained in the following paragraphs. We trained the parameters $\boldsymbol{\theta}$ from random initializations. Table 3 summarizes the overall results with the numbers of parameters d and perturbations q. We used central difference when $q \leq 40$, and forward difference when $q = 200$, as indicated in the "diff." column. The proposed method ZO-F2 outperformed the other methods, as indicated by the bold values. In particular, for the two largest models (CNN and ResNet, with large d and q), ZO-F2 was superior to all the other methods with statistical significance. Figure 2 shows additional results specialized for each architecture, aiming at validating the practical advantages of our proposals.

Linear model (MNIST): We consider a classification task on the MNIST dataset [42] with a simple linear model. The Hessian $\mathbf{H}$ and FIM $\mathbf{F}$ in this case are the same, and the true FIM can be easily computed, as detailed in Appendix G.1. This enables us to evaluate the accuracy of matrix estimation $\hat{\mathbf{H}}$ or $\hat{\mathbf{F}}$ for each method, by the estimation error $\|\hat{\mathbf{H}} - \mathbf{H}\|_F$ or $\|\hat{\mathbf{F}} - \mathbf{F}\|_F$. In addition to the results in Table 3, we conducted experiments with a varying number of parameters d. The left plot of Figure 2a shows that the estimation errors with the quadratic form (ZO-H1) diverged when d became large (i.e., $d = 800$), whereas the bilinear forms (ZO-H2, ZO-F2) remained stable over the examined range of d. This result is consistent with the estimation variance stated in Theorem 3.1. Furthermore, ZO-F2 achieved more accurate estimation than ZO-H2. This improvement is attributed to the larger number of constructed samples for $\tilde{\mathbf{G}}$ enabled by pairwise combinations, as shown in Table 2. We also conducted experiments with varying EMA coefficients α to observe their effect on the estimation error. The right plot shows the errors for ZO-H2 and ZO-F2 at $d = 800$ (ZO-H1 is excluded as it diverged). We observe a general tendency that the estimation error increases with α. Among

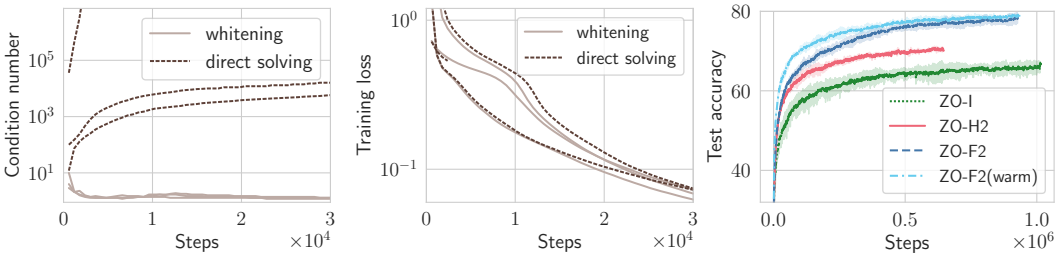
Table 3: Test accuracies (mean ± standard deviation over five runs with random seeds). **Bold**: best result. *Italic*: significantly lower accuracy than the best ($p = 0.05$, one-sided Mann-Whitney U test).

architecture	dataset/task	d	q	diff.	ZO-I	ZO-H1	ZO-H2	ZO-F2
Linear	MNIST	800	10	cen.	*83.7*±0.4	*87.3*±0.2	87.4±0.3	**87.6**±0.2
CNN	EMNIST	15,903	40	cen.	*77.3*±0.7	*78.5*±0.7	*80.0*±0.5	**82.2**±0.1
RNN	Copying	3,168	20	cen.	*94.3*±1.5	*97.1*±2.5	97.2±2.4	**97.9**±2.8
ResNet	CIFAR10	270,410	200	for.	*59.9*±0.9	N/A	*65.1*±0.8	**68.1**±0.3



(a) Results obtained with the linear model with varying (left) number of parameters d and (right) EMA coefficient α.

(b) CNN results with larger EMA α with varying clipping threshold τ.



(c) RNN results with whitening strategy and direct solving approach.

(d) ResNet results with more steps.

Figure 2: Additional experimental results. Shaded areas represent one standard deviation.

the bilinear forms, the proposed ZO-F2 consistently achieved lower estimation error than ZO-H2, demonstrating the advantage of using FIM over Hessian.

CNN (EMNIST): We evaluate EMNIST image classification [43] with a convolutional neural network (CNN). Detailed settings are deferred to Appendix G.2. The CNN results in Table 3 were obtained with the EMA coefficient $\alpha = 3{\cdot}10^{-6}$, which was relatively small and led to conservative updates of $\hat{\mathbf{G}}_t$ from the initialization $\hat{\mathbf{G}}_0 = \mathbf{I}_d$. We conducted additional experiments only for the proposed method ZO-F2 with three runs with random seeds to empirically verify Theorem 3.3. Specifically, we set larger EMA coefficients as $\alpha = 10^{-5}, 2{\cdot}10^{-5}, 3{\cdot}10^{-5}$ aiming at more aggressive updates of $\hat{\mathbf{G}}_t$. As shown in Figure 2b, all the α settings improved the test accuracies over those in Table 3. We observe that $\alpha = 2{\cdot}10^{-5}$ was the best selection among these three and $\alpha = 3{\cdot}10^{-5}$ seemed too large. However, by choosing appropriate clipping thresholds (e.g., $\tau = 0.2, 0.5, 0.7$), the performance remained high even for $\alpha = 3{\cdot}10^{-5}$, whereas a small clipping threshold (e.g., $\tau \leq 0.1$) degraded performance. These observations regarding τ are consistent with Theorem 3.3, where the estimation error δ may be large when α is too large. We conducted further experiments with more diverse settings, and the results are shown in Figure 4 in Appendix G.2.

RNN (Copying): We consider a recurrent neural network (RNN) whose input, hidden, and output layers are represented by unitary matrices. They are constructed by the Clements mesh [44], which is a specialized architecture for optical neural networks. We conducted the Copying memory task [45, 46], which is popularly tested when the RNN's hidden layer is realized by a unitary matrix. More details on the Copying memory task, the experimental settings, and the Clements mesh are deferred to Appendix G.3. Beyond the results in Table 3, we verify the robustness discussed in Subsection 3.2. The left plot of Figure 2c shows that the whitening strategy offered stable execution

whereas the direct solving approach did not, exhibiting large condition numbers with one out of three runs diverging. The right plot shows that the whitening strategy achieved lower training loss.

ResNet (CIFAR-10): We trained ResNet-20 [47] for CIFAR-10 [48] classification. We used a ResNet-20 implementation [49] and replaced batch normalization with group normalization, as the latter avoids dependence on batch statistics and provides more stable function evaluations for ZO optimization. We used forward difference for large q, and thus ZO-H1 was not applicable. Detailed settings are deferred to Appendix G.4. Beyond the results in Table 3 (with five runs, 1.5×10^5 steps each), we conducted additional experiments with two runs and up to $4 \cdot 10^5$ seconds of elapsed time. We also verified the effect of the FIM initialization with a warm-up period of $t_{\text{w-end}} = 1000$ steps (the last paragraph of Subsection 3.4), denoted by the postfix "(warm)". Figure 2d shows the results, from which we make the following observations. The FIM initialization was effective for fast improvement and maintaining high test accuracy. The elapsed time overhead of ZO-F2 and its warm-up variant was less than 10% relative to ZO-I, whereas that of ZO-H2 exceeded 50%. This is consistent with the number of function evaluations in the "forward" columns of Table 2. These results demonstrate the superiority of ZO-F2 over existing methods in terms of both accuracy and computational efficiency.

5 Conclusion

We have successfully advanced ZO optimization techniques that tackle the ill-conditioned curvature of deep models in a black-box manner. Specifically, we have proposed ZO-F2, a novel ZO optimization method that incorporates FIM-based SO preconditioning via bilinear estimation. The bilinear form achieves $\mathcal{O}(d^3)$ estimation variance, lower than the $\mathcal{O}(d^4)$ variance of the quadratic form. We exploit the FIM decomposition to increase the number of constructed samples to $\mathcal{O}(q^2)$. The proposed whitening-based update rule for full-matrix preconditioners circumvents ill-conditioned matrix inversions. Experiments on a wide range of models demonstrated that ZO-F2 consistently outperformed existing ZO methods with a low computational overhead.

References

[1] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[2] S. Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[3] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[4] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11: 441–446, 2017.

[5] F. Ashtiani, A. J. Geers, and F. Aflatouni. An on-chip photonic deep neural network for image classification. *Nature*, 606:501–506, 2022.

[6] Tingzhao Fu, Jianfa Zhang, Run Sun, Yuyao Huang, Wei Xu, Sigang Yang, Zhihong Zhu, and Hongwei Chen. Optical neural networks: progress and challenges. *Light: Science & Applications*, 13(1):263, 2024.

[7] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019.

[8] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4: 269, 2020.

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

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

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

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

[13] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[14] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. DeepZero: Scaling up zeroth-order optimization for deep model training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[15] Jiaqi Gu, Zheng Zhao, Chenghao Feng, Wuxi Li, Ray T. Chen, and David Z. Pan. FLOPS: Efficient on-chip learning for optical neural networks through stochastic zeroth-order optimization. In *Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference*, pages 1–6, 2020.

[16] J. Gu, C. Feng, Z. Zhao, Z. Ying, R. T. Chen, and D. Z. Pan. Efficient on-chip learning for optical neural networks through power-aware sparse zeroth-order optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7583–7591, 2021.

[17] S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, M. Streshinsky, M. Hochberg, and D. Englund. Single-chip photonic deep neural network with forward-only training. *Nature Photonics*, 18:1335–1343, 2024.

[18] Hiroshi Sawada, Kazuo Aoyama, and Masaya Notomi. Layered-parameter perturbation for zeroth-order optimization of optical neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20292–20301, 2025.

[19] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 53038–53075, 2023.

[20] Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 59173–59190, 2024.

[21] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.

[22] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[23] Roberto Battiti. First-and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation*, 4(2):141–166, 1992.

[24] James Martens. Deep learning via Hessian-free optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 735–742, 2010.

[25] Xi-Lin Li. Preconditioned stochastic gradient descent. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1454–1466, 2018.

[26] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[27] R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[28] J. Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[29] Felix Dangel, Frederik Kunstner, and Philipp Hennig. BackPACK: Packing more into backprop. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[30] James C Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45(10):1839–1853, 2000.

[31] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner. Simultaneous perturbation stochastic approximation of the quantum Fisher information. *Quantum*, 5:567, 2021.

[32] L. A. Prashanth, Shalabh Bhatnagar, Michael Fu, and Steve Marcus. Adaptive system optimization using random directions stochastic approximation. *IEEE Transactions on Automatic Control*, 62(5):2223–2238, 2017.

[33] Jingyi Zhu. Hessian estimation via Stein's identity in black-box problems. In *Mathematical and Scientific Machine Learning*, pages 1161–1178, 2022.

[34] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[35] Hiroshi Sawada, Kazuo Aoyama, and Yuya Hikima. Natural perturbations for black-box training of neural networks by zeroth-order optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[36] Hanzhen Zhao, Shihong Ding, Cong Fang, and Zhouchen Lin. PaZO: Preconditioned accelerated zeroth-order optimization for fine-tuning LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

[37] Atılım Güneş Baydin, Barak A Pearlmutter, Don Syme, Frank Wood, and Philip Torr. Gradients without backpropagation. *arXiv preprint arXiv:2202.08587*, 2022.

[38] Gabriel Belouze. Optimization without backpropagation. *arXiv preprint arXiv:2209.06302*, 2022.

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

[40] Jeffrey Larson, Matt Menickelly, and Stefan M Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

[41] Sohei Arisaka and Qianxiao Li. Accelerating legacy numerical solvers by non-intrusive gradient-based meta-solving. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1689–1708, 2024.

[42] Y. LeCun and C. Cortes. MNIST handwritten digit database, 2010. URL `http://yann.lecun.com/exdb/mnist`.

[43] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017.

[44] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016.

[45] M. Arjovsky, A. Shah, and Y. Bengio. Unitary evolution recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1120–1128, 2016.

[46] Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3794–3803, 2019.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[48] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

[49] Yerlan Idelbayev. Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch. `https://github.com/akamaster/pytorch_resnet_cifar10`. Accessed: 2026-04-02.

[50] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[51] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[52] J. Bae, P. Vicol, J. Z. HaoChen, and R. B. Grosse. Amortized proximal optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 8982–8997, 2022.

[53] James Martens. *Second-order optimization for neural networks*. PhD thesis, University of Toronto, 2016.

[54] F. Kunstner, P. Hennig, and L. Balles. Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[55] Hyeyoung Park, S. Amari, and Kenji Fukumizu. Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks*, 13(7):755–764, 2000.

[56] Jan R. Magnus. The moments of products of quadratic forms in normal variables. *Statistica Neerlandica*, 32(4):201–210, 1978.

[57] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

[58] Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2nd edition, 2018.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[60] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

A Notation summary

Table 4 summarizes the main notations used in this paper.

Table 4: Main notations

$\boldsymbol{f} : \mathbb{R}^{d_{\text{in}}} \times \mathbb{R}^d \to \mathbb{R}^{d_{\text{out}}}$	(learnable) function
$\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$	input vector
$\boldsymbol{\theta} \in \mathbb{R}^d$	parameter vector
$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^{d_{\text{out}}}$	output vector
$\{(\boldsymbol{x}, \boldsymbol{t})\}$	training dataset
$\mathcal{D} \subseteq \{(\boldsymbol{x}, \boldsymbol{t})\}$	mini-batch data
$\boldsymbol{t}$	target vector
$p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}) = p(\cdot \mid \boldsymbol{y})\big\vert_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})}$	probability distribution
$\ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t}) = -\log p_{\boldsymbol{\theta}}(\boldsymbol{t} \mid \boldsymbol{x})$	per-sample loss
$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{t})\sim\{(\boldsymbol{x},\boldsymbol{t})\}}[\ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})]$	full-batch loss
$\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{t})\sim\mathcal{D}}[\ell(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})]$	mini-batch loss
$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \in \mathbb{R}^d$	full-batch gradient
$\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) \in \mathbb{R}^d$	mini-batch (stochastic) gradient
$\hat{\boldsymbol{g}} \in \mathbb{R}^d$	approximate gradient
$\boldsymbol{u} \in \mathbb{R}^d$	perturbation
$\mathbf{I}_d \in \mathbb{R}^{d \times d}$	d-dimensional identity matrix
$\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$	standard normal distribution
$\mathbf{P} \in \mathbb{R}^{d \times d}$	gradient preconditioner
$\mathbf{W} \in \mathbb{R}^{d \times d}$	perturbation preconditioner
$\mathbf{H} \in \mathbb{R}^{d \times d}$	Hessian
$\mathbf{F} \in \mathbb{R}^{d \times d}$	Fisher information matrix (FIM)
$\mathbf{F}_{\boldsymbol{y}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$	FIM with respect to the output vector
$\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}} \in \mathbb{R}^{d_{\text{out}} \times d}$	Jacobian matrix
$\hat{\tilde{\boldsymbol{g}}} \in \mathbb{R}^d$	preconditioned approximate gradient
$\bar{\boldsymbol{u}} \in \mathbb{R}^d$	preconditioned perturbation
$\mathbf{G} \in \mathbb{R}^{d \times d}$	estimation target $\mathbf{W}^\mathsf{T}\mathbf{F}\mathbf{W}$ or $\mathbf{W}^\mathsf{T}\mathbf{H}\mathbf{W}$
$\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{d \times d}$	one-sample estimator (bilinear form)
$\tilde{\mathbf{G}}(\boldsymbol{u}) \in \mathbb{R}^{d \times d}$	one-sample estimator (quadratic form)
$\hat{\mathbf{G}}^{\text{ie}} \in \mathbb{R}^{d \times d}$	intermediate estimate
$\hat{\mathbf{G}} \in \mathbb{R}^{d \times d}$	averaged estimate
$q \in \{1, 2, \dots\}$	number of perturbations
$\mu > 0$	ZO smoothing parameter
$\eta > 0$	learning rate
$\alpha \in (0, 1]$	EMA coefficient
$\gamma > 0$	damping coefficient
$\tau > 0$	clipping threshold
$T_{\text{ud}} \in \{2, 3, \dots\}$	update frequency of $\mathbf{W}$

B Hessian and Fisher information matrix (FIM) and their relationship

This appendix provides a detailed explanation of the Hessian and FIM used as gradient preconditioners in (4). We consider minimizing the loss function $\mathcal{L}(\boldsymbol{\theta})$ over the training dataset $\{(\boldsymbol{x}, \boldsymbol{t})\}$ with respect to the parameter vector $\boldsymbol{\theta}$ in an iterative manner by the proximal point method [50, 51] as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \operatorname*{arg\,min}_{\delta\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) + \frac{\gamma}{2} \|\delta\boldsymbol{\theta}\|^2 \right\}, \tag{29}$$

where γ serves as the damping coefficient.

The Taylor series expansion of the loss function up to the second order gives

$$\mathcal{L}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx \mathcal{L}(\boldsymbol{\theta}) + \delta\boldsymbol{\theta}^\mathsf{T} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2} \delta\boldsymbol{\theta}^\mathsf{T} \nabla_{\boldsymbol{\theta}}^2 \mathcal{L}(\boldsymbol{\theta}) \delta\boldsymbol{\theta} \,, \tag{30}$$

where we have the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ and the Hessian

$$\mathbf{H} := \nabla_{\boldsymbol{\theta}}^2 \mathcal{L}(\boldsymbol{\theta}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{t}) \sim \{(\boldsymbol{x},\boldsymbol{t})\}} \left[\nabla_{\boldsymbol{\theta}}^2 \log p_{\boldsymbol{\theta}}(\boldsymbol{t} \mid \boldsymbol{x}) \right] \in \mathbb{R}^{d \times d} \,. \tag{31}$$

Solving (29) with (30) and (31) gives the preconditioned gradient $\mathbf{P} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ with the preconditioner

$$\mathbf{P} := \left(\gamma \, \mathbf{I}_d + \mathbf{H} \right)^{-1} \,. \tag{32}$$

Multiplying it by a learning rate η gives the damped Newton update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \, \mathbf{P} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

For the Fisher information matrix (FIM), we consider another parameter-update criterion [52]

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \arg\min_{\delta\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) + \frac{\gamma}{2} \|\delta\boldsymbol{\theta}\|^2 + \xi(\delta\boldsymbol{\theta}) \right\} \,, \tag{33}$$

with the Taylor series expansion up to the first order $\mathcal{L}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx \mathcal{L}(\boldsymbol{\theta}) + \delta\boldsymbol{\theta}^\mathsf{T} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$. The new term $\xi(\delta\boldsymbol{\theta})$ is the function space discrepancy

$$\xi(\delta\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{x} \sim \{(\boldsymbol{x},\boldsymbol{t})\}} [D_{\mathrm{KL}}(p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}), p_{\boldsymbol{\theta}+\delta\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}))] \approx \frac{1}{2} \, \delta\boldsymbol{\theta}^\mathsf{T} \mathbf{F} \, \delta\boldsymbol{\theta} \,, \tag{34}$$

which measures how the conditional distributions $p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}) = p(\cdot \mid \boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})}$ change by the parameter change $\delta\boldsymbol{\theta}$. The approximation "$\approx$" is accomplished by the Taylor series expansion of the Kullback-Leibler (KL) divergence D_{KL} up to the second order [27, 53], and we have the FIM as

$$\mathbf{F} := \mathbb{E}_{\boldsymbol{x} \sim \{(\boldsymbol{x},\boldsymbol{t})\}} \left[\mathbb{E}_{\boldsymbol{r} \sim p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x})} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{r} \mid \boldsymbol{x}) \, \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{r} \mid \boldsymbol{x})^\mathsf{T} \right] \right] \in \mathbb{R}^{d \times d} \,. \tag{35}$$

Note that, unlike the Hessian (31), the target vectors $\boldsymbol{t}$ are not used in the FIM (35) [54]. If the target vectors $\boldsymbol{t}$ are instead used as

$$\mathbf{F}_{\mathrm{emp}} := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{t}) \sim \{(\boldsymbol{x},\boldsymbol{t})\}} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{t} \mid \boldsymbol{x}) \, \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{t} \mid \boldsymbol{x})^\mathsf{T} \right] \in \mathbb{R}^{d \times d} \,, \tag{36}$$

the resulting matrix is called the empirical Fisher [54] as an approximation of the FIM. Solving (33) with (34) gives the preconditioned gradient $\mathbf{P} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ with the preconditioner

$$\mathbf{P} := \left(\gamma \, \mathbf{I}_d + \mathbf{F} \right)^{-1} \,. \tag{37}$$

Multiplying it by a learning rate η gives the natural gradient update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \, \mathbf{P} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

Next, we discuss the relationship between Hessian and FIM. Using the Jacobian matrix $\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}$ defined in (11), the Hessian and FIM [55, 28] can be expressed as

$$\mathbf{H} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{t}) \sim \{(\boldsymbol{x},\boldsymbol{t})\}} \left[\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}^\mathsf{T} \mathbf{H}_{\boldsymbol{y}} \mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}} \big|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})} + \sum_{i=1}^{d_{\mathrm{out}}} [\boldsymbol{g}_{\boldsymbol{y}}]_i \, \nabla_{\boldsymbol{\theta}}^2 \left[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}) \right]_i \right] \,, \tag{38}$$

$$\mathbf{F} = \mathbb{E}_{\boldsymbol{x} \sim \{(\boldsymbol{x},\boldsymbol{t})\}} \left[\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}}^\mathsf{T} \mathbf{F}_{\boldsymbol{y}} \mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}} \big|_{\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})} \right] \,, \tag{39}$$

with the gradient, Hessian, and FIM with respect to the output vector defined as

$$\boldsymbol{g}_{\boldsymbol{y}} := -\nabla_{\boldsymbol{y}} \log p(\boldsymbol{t} \mid \boldsymbol{y}) \in \mathbb{R}^{d_{\mathrm{out}}} \,, \tag{40}$$

$$\mathbf{H}_{\boldsymbol{y}} := -\nabla_{\boldsymbol{y}}^2 \log p(\boldsymbol{t} \mid \boldsymbol{y}) \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{out}}} \,, \tag{41}$$

$$\mathbf{F}_{\boldsymbol{y}} := \mathbb{E}_{\boldsymbol{r} \sim p(\cdot \mid \boldsymbol{y})} \left[\nabla_{\boldsymbol{y}} \log p(\boldsymbol{r} \mid \boldsymbol{y}) \, \nabla_{\boldsymbol{y}} \log p(\boldsymbol{r} \mid \boldsymbol{y})^\mathsf{T} \right] \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{out}}} \,. \tag{42}$$

When we use a normal distribution with mean $\boldsymbol{y}$ and identity covariance as $p(\cdot \mid \boldsymbol{y})$ for a regression task, the Hessian and FIM with respect to the output vector are given as $\mathbf{H}_{\boldsymbol{y}} = \mathbf{F}_{\boldsymbol{y}} = \mathbf{I}_{d_{\mathrm{out}}}$. When we use a categorical distribution with probabilities $\mathrm{softmax}(\boldsymbol{y})$ as $p(\cdot \mid \boldsymbol{y})$ with logits $\boldsymbol{y}$ for a classification task, the corresponding Hessian and FIM are given as $\mathbf{H}_{\boldsymbol{y}} = \mathbf{F}_{\boldsymbol{y}} = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\mathsf{T}$ with $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{y})$. In both these specific but typical situations, the Hessian $\mathbf{H}_{\boldsymbol{y}}$ and FIM $\mathbf{F}_{\boldsymbol{y}}$ with respect to the output vector are the same, and the Hessian (38) can thus be expressed using the FIM (39) as

$$\mathbf{H} = \mathbf{F} + \mathbb{E}_{(\boldsymbol{x},\boldsymbol{t}) \sim \{(\boldsymbol{x},\boldsymbol{t})\}} \left[\sum_{i=1}^{d_{\mathrm{out}}} [\boldsymbol{g}_{\boldsymbol{y}}]_i \, \nabla_{\boldsymbol{\theta}}^2 \left[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}) \right]_i \right] \,. \tag{43}$$

C Matrix estimation with bilinear and quadratic forms

Let $\mathbf{G} \in \mathbb{R}^{d\times d}$ be a symmetric matrix to be estimated, i.e., $\mathbf{G}^\mathsf{T} = \mathbf{G}$. We use perturbations $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and $\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ to compute the estimator $\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v})$ in (6) and $\tilde{\mathbf{G}}(\boldsymbol{u})$ in (7). We denote their elements by $G_{ij} = (\mathbf{G})_{ij}$, $\tilde{G}_{ij} = (\tilde{\mathbf{G}})_{ij}$. We use the following proposition in the proofs.

Proposition C.1 ($A(s)$-polynomial [56])**.** *For s symmetric matrices $\mathbf{A}_1, \dots, \mathbf{A}_s$,*

$$A(s) := \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\prod_{j=1}^{s} \boldsymbol{w}^\mathsf{T}\mathbf{A}_j\boldsymbol{w}\right]$$

can be written for $s = 1, 2, 3, 4$ as

$$A(1) = \mathrm{tr}(\mathbf{A}_1)\,, \tag{44}$$

$$A(2) = \mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_2) + 2\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2)\,, \tag{45}$$

$$\begin{aligned} A(3) = {} & \mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_3) + 8\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3) \\ & + 2\left[\mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_2\mathbf{A}_3) + \mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_3) + \mathrm{tr}(\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2)\right]\,, \end{aligned} \tag{46}$$

$$\begin{aligned} A(4) = {} & \mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_4) + 16[\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_4\mathbf{A}_3) \\ & + \mathrm{tr}(\mathbf{A}_1\mathbf{A}_3\mathbf{A}_2\mathbf{A}_4)] + 8[\mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_3\mathbf{A}_4) \\ & + \mathrm{tr}(\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_4)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3)] \\ & + 4\left[\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_3\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_1\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_2\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_1\mathbf{A}_4)\mathrm{tr}(\mathbf{A}_2\mathbf{A}_3)\right] \\ & + 2[\mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_3\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_2\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_1)\mathrm{tr}(\mathbf{A}_4)\mathrm{tr}(\mathbf{A}_2\mathbf{A}_3) \\ & + \mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_4) + \mathrm{tr}(\mathbf{A}_2)\mathrm{tr}(\mathbf{A}_4)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_3) + \mathrm{tr}(\mathbf{A}_3)\mathrm{tr}(\mathbf{A}_4)\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2)]\,. \end{aligned} \tag{47}$$

C.1 Unbiasedness

The bilinear form (6), $\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v}) = \left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v}\right)\boldsymbol{u}\boldsymbol{v}^\mathsf{T}$, is an unbiased estimator of $\mathbf{G}$ as

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d),\boldsymbol{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\tilde{\mathbf{G}}(\boldsymbol{u}, \boldsymbol{v})\right] = \mathbb{E}_{\boldsymbol{u}}\left[\boldsymbol{u}\boldsymbol{u}^\mathsf{T}\right]\mathbf{G}\mathbb{E}_{\boldsymbol{v}}\left[\boldsymbol{v}\boldsymbol{v}^\mathsf{T}\right] = \mathbf{G}\,. \tag{48}$$

The quadratic form (7), $\tilde{\mathbf{G}}(\boldsymbol{u}) = \frac{1}{2}\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)\left(\boldsymbol{u}\boldsymbol{u}^\mathsf{T} - \mathbf{I}_d\right)$, is an unbiased estimator of $\mathbf{G}$ because

$$\begin{aligned} \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\tilde{G}_{ij}\right] &= \frac{1}{2}\left\{\mathbb{E}_{\boldsymbol{u}}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)\left(\boldsymbol{u}^\mathsf{T}\mathbf{E}_{ij}\boldsymbol{u}\right)\right] - \delta_{ij}\mathbb{E}_{\boldsymbol{u}}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)\right]\right\} \\ &\overset{\text{(a)}}{=} \frac{1}{2}\left\{\mathrm{tr}(\mathbf{G})\mathrm{tr}(\mathbf{E}_{ij}) + 2\mathrm{tr}(\mathbf{G}\mathbf{E}_{ij}) - \delta_{ij}\mathrm{tr}(\mathbf{G})\right\} \\ &= G_{ij}\,. \end{aligned} \tag{49}$$

where δ_{ij} is the Kronecker delta and $\mathbf{E}_{ij} \in \mathbb{R}^{d\times d}$ is a matrix having only two nonzero scalars as $1/2$ in the ij- and ji-elements and 0 elsewhere (when $i = j$, $\mathbf{E}_{ij}$ has only one nonzero scalar as 1 in the ii-element). (a) follows from $A(2)$-polynomial with $\mathbf{A}_1 = \mathbf{G}$ and $\mathbf{A}_2 = \mathbf{E}_{ij}$ and also from $A(1)$-polynomial with $\mathbf{A}_1 = \mathbf{G}$.

C.2 Proof of Theorem 3.1 (estimation variance)

We first present the proof of Theorem 3.1, and then give the associated lemmas.

Proof. The results are special cases of Lemma C.2 and Lemma C.1, which have the terms $\mathrm{tr}(\mathbf{G})$ and $\mathrm{tr}(\mathbf{G}^2)$. Under Assumption 3.1, $\|\mathbf{G}\|_2 \le C$ holds. Thus,

$$|\mathrm{tr}(\mathbf{G})| \le \sum_{i=1}^{d}|\lambda_i(\mathbf{G})| \le \sum_{i=1}^{d} C = Cd\,, \quad \mathrm{tr}(\mathbf{G}^2) = \sum_{i=1}^{d}\lambda_i(\mathbf{G})^2 \le \sum_{i=1}^{d} C^2 = C^2 d\,, \tag{50}$$

where $\lambda_i(\mathbf{G})$ is the i-th eigenvalue of $\mathbf{G}$. Substituting (50) into (55) and (51) gives (12) and (13). □

Lemma C.1. *The Frobenius-norm MSE of $\tilde{\mathbf{G}}(\boldsymbol{u})$ in (7) is evaluated as*

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\|\tilde{\mathbf{G}}(\boldsymbol{u})-\mathbf{G}\|_F^2\right]=\frac{1}{4}\left(d^2+9d+16\right)\mathrm{tr}(\mathbf{G})^2+\frac{1}{2}\left(d^2+9d+14\right)\mathrm{tr}(\mathbf{G}^2)\,. \quad (51)$$

Proof. The squared Frobenius norm can be written as

$$\begin{aligned}\|\tilde{\mathbf{G}}(\boldsymbol{u})-\mathbf{G}\|_F^2=&\frac{1}{4}\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)^2\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{u}\right)^2+\left(\frac{d}{4}-1\right)\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)^2+\mathrm{tr}(\mathbf{G}^2)\\&-\frac{1}{2}\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)^2\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{u}\right)+\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)\mathrm{tr}(\mathbf{G})\,. \quad (52)\end{aligned}$$

Now, we evaluate the expectations using $A(s)$-polynomials with $\mathbf{A}_1=\mathbf{A}_2=\mathbf{G}$, $\mathbf{A}_3=\mathbf{A}_4=\mathbf{I}_d$ as

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)\right]&=\mathrm{tr}(\mathbf{G})\,,\\
\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)^2\right]&=\mathrm{tr}(\mathbf{G})^2+2\mathrm{tr}(\mathbf{G}^2)\,,\\
\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)^2\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{u}\right)\right]&=(d+4)\mathrm{tr}(\mathbf{G})^2+(2d+8)\mathrm{tr}(\mathbf{G}^2)\,,\\
\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{u}\right)^2\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{u}\right)^2\right]&=(d^2+10d+24)\mathrm{tr}(\mathbf{G})^2+(2d^2+20d+48)\mathrm{tr}(\mathbf{G}^2)\,. \quad (53)\end{aligned}$$

Therefore, we have that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\|\tilde{\mathbf{G}}(\boldsymbol{u})-\mathbf{G}\|_F^2\right]=&\frac{1}{4}\left[(d^2+10d+24)\mathrm{tr}(\mathbf{G})^2+(2d^2+20d+48)\mathrm{tr}(\mathbf{G}^2)\right]\\&+\left(\frac{d}{4}-1\right)\left[\mathrm{tr}(\mathbf{G})^2+2\mathrm{tr}(\mathbf{G}^2)\right]+\mathrm{tr}(\mathbf{G}^2)\\&-\frac{1}{2}\left[(d+4)\mathrm{tr}(\mathbf{G})^2+(2d+8)\mathrm{tr}(\mathbf{G}^2)\right]+\mathrm{tr}(\mathbf{G})^2\,. \quad (54)\end{aligned}$$

Simplifying the above equation gives the conclusion (51). □

Lemma C.2. *The Frobenius-norm MSE of $\tilde{\mathbf{G}}(\boldsymbol{u},\boldsymbol{v})$ in (6) is evaluated as*

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d),\boldsymbol{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\|\tilde{\mathbf{G}}(\boldsymbol{u},\boldsymbol{v})-\mathbf{G}\|_F^2\right]=\left(d^2+4d+3\right)\mathrm{tr}(\mathbf{G}^2)\,. \quad (55)$$

Proof. The squared Frobenius norm can be written as

$$\|\tilde{\mathbf{G}}(\boldsymbol{u},\boldsymbol{v})-\mathbf{G}\|_F^2=\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v}\right)^2\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{u}\right)\left(\boldsymbol{v}^\mathsf{T}\boldsymbol{v}\right)+\mathrm{tr}(\mathbf{G}^2)-2\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v}\right)^2\,. \quad (56)$$

Now, to utilize $A(s)$-polynomials in Proposition C.1 even for these bilinear forms, let us define

$$\mathbf{A}_{\mathbf{G}}:=\frac{1}{2}\begin{pmatrix}0&\mathbf{G}\\\mathbf{G}&0\end{pmatrix}\,,\quad \mathbf{A}_{\text{up}}:=\begin{pmatrix}\mathbf{I}_d&0\\0&0\end{pmatrix}\,,\quad \mathbf{A}_{\text{low}}:=\begin{pmatrix}0&0\\0&\mathbf{I}_d\end{pmatrix}\in\mathbb{R}^{2d\times 2d}\,. \quad (57)$$

By letting $\boldsymbol{w}=[\boldsymbol{u}^\mathsf{T},\boldsymbol{v}^\mathsf{T}]^\mathsf{T}$, it follows from $A(2)$-polynomial (45) with $\mathbf{A}_1=\mathbf{A}_2=\mathbf{A}_{\mathbf{G}}$ that

$$\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}\left[(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v})^2\right]=\mathrm{tr}(\mathbf{G}^2)\,. \quad (58)$$

Similarly, it follows from $A(4)$-polynomial (47) with $\mathbf{A}_1=\mathbf{A}_2=\mathbf{A}_{\mathbf{G}}$, $\mathbf{A}_3=\mathbf{A}_{\text{up}}$, $\mathbf{A}_4=\mathbf{A}_{\text{low}}$ that

$$\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}\left[\left(\boldsymbol{u}^\mathsf{T}\mathbf{G}\boldsymbol{v}\right)^2\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{u}\right)\left(\boldsymbol{v}^\mathsf{T}\boldsymbol{v}\right)\right]=(d^2+4d+4)\mathrm{tr}(\mathbf{G}^2)\,.$$

Therefore, we have that

$$\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}\left[\|\tilde{\mathbf{G}}(\boldsymbol{u},\boldsymbol{v})-\mathbf{G}\|_F^2\right]=(d^2+4d+4)\mathrm{tr}(\mathbf{G}^2)+\mathrm{tr}(\mathbf{G}^2)-2\mathrm{tr}(\mathbf{G}^2)\,. \quad (59)$$

Simplifying the above equation gives the conclusion (55). □

D Aggregated estimator

D.1 Unbiasedness

Let us define

$$\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j) := \frac{1}{2}\left[\tilde{\mathbf{G}}(\boldsymbol{u}_i, \boldsymbol{u}_j) + \tilde{\mathbf{G}}(\boldsymbol{u}_i, \boldsymbol{u}_j)^\mathsf{T}\right], \tag{60}$$

which is an unbiased estimator of a symmetric matrix $\mathbf{G}$ as

$$\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j)\right] = \frac{1}{2}(\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\tilde{\mathbf{G}}(\boldsymbol{u}_i, \boldsymbol{u}_j)\right] + \mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\tilde{\mathbf{G}}(\boldsymbol{u}_i, \boldsymbol{u}_j)\right]^\mathsf{T}) = \frac{1}{2}(\mathbf{G}+\mathbf{G}^\mathsf{T}) = \mathbf{G}\,. \tag{61}$$

Therefore, the aggregated estimator (16) is an unbiased estimator of symmetric $\mathbf{G}$ as

$$\mathbb{E}\left[\hat{\mathbf{G}}^{\mathrm{agg}}\right] = \frac{2}{q(q-1)}\sum_{i=1}^{q}\sum_{j=i+1}^{q}\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j)\right] = \frac{2}{q(q-1)}\sum_{i=1}^{q}\sum_{j=i+1}^{q}\mathbf{G} = \mathbf{G}\,. \tag{62}$$

D.2 Proof of Theorem 3.2 (aggregated estimator variance)

We first present the proof of Theorem 3.2, and then give the associated propositions and lemmas.

Proof. The result is a special case of Lemma D.1. Under Assumption 3.1, we have (50) satisfied, i.e., $|\mathrm{tr}(\mathbf{G})| \leq Cd$ and $\mathrm{tr}(\mathbf{G}^2) \leq C^2 d$. By substituting these into (71), we obtain (17). □

In the following lemmas, we use the following propositions regarding the Hoeffding decomposition.

Proposition D.1 (Hoeffding decomposition [57] of the kernel)**.** *For*

$$\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j) - \mathbf{G} = g_1(\boldsymbol{u}_i) + g_1(\boldsymbol{u}_j) + g_2(\boldsymbol{u}_i, \boldsymbol{u}_j) \tag{63}$$

with

$$g_1(\boldsymbol{u}_i) := \mathbb{E}_{\boldsymbol{u}_j}\left[\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j)\right] - \mathbf{G}\,, \tag{64}$$

$$g_2(\boldsymbol{u}_i, \boldsymbol{u}_j) := \tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j) - g_1(\boldsymbol{u}_i) - g_1(\boldsymbol{u}_j) - \mathbf{G}\,, \tag{65}$$

we have the variance in the decomposed form as

$$V_{\mathrm{full}} := \mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\left\|\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j) - \mathbf{G}\right\|_F^2\right] = 2V_1 + V_2 \tag{66}$$

with

$$V_1 := \mathbb{E}_{\boldsymbol{u}_i}\left[\|g_1(\boldsymbol{u}_i)\|_F^2\right], \qquad V_2 := \mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\|g_2(\boldsymbol{u}_i, \boldsymbol{u}_j)\|_F^2\right]. \tag{67}$$

Proposition D.2 (Hoeffding decomposition [57] of the aggregated estimator)**.** *For the Hoeffding decomposition*

$$\hat{\mathbf{G}}^{\mathrm{agg}} - \mathbf{G} = \frac{2}{q}\sum_{i=1}^{q} g_1(\boldsymbol{u}_i) + \frac{2}{q(q-1)}\sum_{i=1}^{q}\sum_{j=i+1}^{q} g_2(\boldsymbol{u}_i, \boldsymbol{u}_j) \tag{68}$$

of the aggregated estimator (16)*, denoted by using* (60) *as*

$$\hat{\mathbf{G}}^{\mathrm{agg}} = \frac{2}{q(q-1)}\sum_{i=1}^{q}\sum_{j=i+1}^{q}\tilde{\mathbf{G}}^s(\boldsymbol{u}_i, \boldsymbol{u}_j)\,, \tag{69}$$

we have the variance evaluated as

$$\mathbb{E}_{\{\boldsymbol{u}_k\}_{k=1}^q}\left[\left\|\hat{\mathbf{G}}^{\mathrm{agg}} - \mathbf{G}\right\|_F^2\right] = \frac{4}{q}V_1 + \frac{2}{q(q-1)}V_2\,. \tag{70}$$

Lemma D.1. *The Frobenius-norm MSE of* $\hat{\mathbf{G}}^{\mathrm{agg}}$ *in (16) is evaluated as*

$$\mathbb{E}_{\{\boldsymbol{u}_k\}_{k=1}^q}\left[\|\hat{\mathbf{G}}^{\mathrm{agg}} - \mathbf{G}\|_F^2\right] = \frac{2}{q}\mathrm{tr}(\mathbf{G})^2 + \left[\frac{2(d+2)}{q} + \frac{d^2+3d+4}{q(q-1)}\right]\mathrm{tr}(\mathbf{G}^2)\,. \tag{71}$$

Proof. It follows from (66), (83), and (74) that

$$V_2 = V_{\text{full}} - 2V_1 = \text{tr}(\mathbf{G})^2 + \frac{1}{2}(d^2+5d+8)\text{tr}(\mathbf{G}^2) - \left[\text{tr}(\mathbf{G})^2 + (d+2)\text{tr}(\mathbf{G}^2)\right]$$
$$= \frac{1}{2}(d^2+3d+4)\text{tr}(\mathbf{G}^2)\,. \tag{72}$$

Therefore, from (70), (74), and the above, we have that

$$\mathbb{E}_{\{\boldsymbol{u}_k\}_{k=1}^q}\left[\left\|\hat{\mathbf{G}}^{\text{agg}} - \mathbf{G}\right\|_F^2\right] = \frac{4}{q}V_1 + \frac{2}{q(q-1)}V_2$$
$$= \frac{2}{q}\left[\text{tr}(\mathbf{G})^2 + (d+2)\text{tr}(\mathbf{G}^2)\right] + \frac{1}{q(q-1)}(d^2+3d+4)\text{tr}(\mathbf{G}^2)$$
$$= \frac{2}{q}\text{tr}(\mathbf{G})^2 + \left[\frac{2(d+2)}{q} + \frac{d^2+3d+4}{q(q-1)}\right]\text{tr}(\mathbf{G}^2)\,. \tag{73}$$

This completes the proof. □

Lemma D.2. *For* (64)*, we have that*

$$V_1 = \mathbb{E}_{\boldsymbol{u}_i}\left[\|g_1(\boldsymbol{u}_i)\|_F^2\right] = \frac{1}{2}\left[\text{tr}(\mathbf{G})^2 + (d+2)\text{tr}(\mathbf{G}^2)\right]\,. \tag{74}$$

Proof. Let us denote the random part of $g_1(\boldsymbol{u}_i)$ as

$$\mathbf{M}(\boldsymbol{u}_i) := \mathbb{E}_{\boldsymbol{u}_j}\left[\tilde{\mathbf{G}}^s(\boldsymbol{u}_i,\boldsymbol{u}_j)\right] = \frac{1}{2}(\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T}\mathbf{G} + \mathbf{G}\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T})\,. \tag{75}$$

Then, by using the cyclic property of the trace, we have that

$$\text{tr}(\mathbf{M}(\boldsymbol{u}_i)^2) = \frac{1}{4}\left[2\text{tr}(\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T}\mathbf{G}) + 2\text{tr}(\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\mathbf{G}\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T})\right] \tag{76}$$
$$= \frac{1}{2}\left[(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_i)^2 + (\boldsymbol{u}_i^\mathsf{T}\boldsymbol{u}_i)(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}^2\boldsymbol{u}_i)\right]\,. \tag{77}$$

Additionally noting that $\mathbf{M}$ and $\mathbf{G}$ are symmetric, we have that

$$\text{tr}(\mathbf{M}(\boldsymbol{u}_i)^\mathsf{T}\mathbf{G}) = \text{tr}(\mathbf{G}^\mathsf{T}\mathbf{M}(\boldsymbol{u}_i)) = \text{tr}(\mathbf{M}(\boldsymbol{u}_i)\mathbf{G}) = \boldsymbol{u}_i^\mathsf{T}\mathbf{G}^2\boldsymbol{u}_i\,. \tag{78}$$

Therefore, we have that

$$\|g_1(\boldsymbol{u}_i)\|_F^2 = \text{tr}(\mathbf{M}(\boldsymbol{u}_i)^2) - 2\text{tr}(\mathbf{M}(\boldsymbol{u}_i)\mathbf{G}) + \text{tr}(\mathbf{G}^2) \tag{79}$$
$$= \frac{1}{2}\left[(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_i)^2 + (\boldsymbol{u}_i^\mathsf{T}\boldsymbol{u}_i)(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}^2\boldsymbol{u}_i)\right] - 2(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}^2\boldsymbol{u}_i) + \text{tr}(\mathbf{G}^2)\,. \tag{80}$$

Employing $A(2)$-polynomial (45) with $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{G}$ and with $\mathbf{A}_1 = \mathbf{I}_d, \mathbf{A}_2 = \mathbf{G}^2$, and also employing $A(1)$-polynomial (44) with $\mathbf{A}_1 = \mathbf{G}^2$, we have that

$$\mathbb{E}_{\boldsymbol{u}_i}\left[\|g_1(\boldsymbol{u}_i)\|_F^2\right] = \frac{1}{2}\left[\text{tr}(\mathbf{G})^2 + 2\text{tr}(\mathbf{G}^2) + d\text{tr}(\mathbf{G}^2) + 2\text{tr}(\mathbf{G}^2)\right] - 2\text{tr}(\mathbf{G}^2) + \text{tr}(\mathbf{G}^2) \tag{81}$$
$$= \frac{1}{2}\left[\text{tr}(\mathbf{G})^2 + (d+2)\text{tr}(\mathbf{G}^2)\right]\,. \tag{82}$$

This completes the proof. □

Lemma D.3. *We have that*

$$V_{\text{full}} = \mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\left\|\tilde{\mathbf{G}}^s(\boldsymbol{u}_i,\boldsymbol{u}_j) - \mathbf{G}\right\|_F^2\right] = \text{tr}(\mathbf{G})^2 + \frac{1}{2}(d^2+5d+8)\text{tr}(\mathbf{G}^2)\,. \tag{83}$$

Proof. Let us denote $\mathbf{D} = \tilde{\mathbf{G}}(\boldsymbol{u}_i,\boldsymbol{u}_j) - \mathbf{G}$ so that $\tilde{\mathbf{G}}^s(\boldsymbol{u}_i,\boldsymbol{u}_j) - \mathbf{G} = \frac{1}{2}(\mathbf{D}+\mathbf{D}^\mathsf{T})$. We have that

$$\left\|\tilde{\mathbf{G}}^s(\boldsymbol{u}_i,\boldsymbol{u}_j) - \mathbf{G}\right\|_F^2 = \frac{1}{4}\|\mathbf{D}+\mathbf{D}^\mathsf{T}\|_F^2 = \frac{1}{2}\left[\|\mathbf{D}\|_F^2 + \text{tr}(\mathbf{D}^2)\right]\,. \tag{84}$$

The trace term can be written as

$$\mathrm{tr}(\mathbf{D}^2) = (\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_j)^2(\boldsymbol{u}_i^\mathsf{T}\boldsymbol{u}_j)^2 - 2(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_j)^2 + \mathrm{tr}(\mathbf{G}^2)\,. \tag{85}$$

Now, to utilize $A(s)$-polynomials in Proposition C.1 even for these bilinear forms, let us define

$$\mathbf{A}_\mathbf{G} = \frac{1}{2}\begin{pmatrix} \mathbf{0} & \mathbf{G} \\ \mathbf{G} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{2d\times 2d}, \quad \mathbf{A}_{\mathbf{I}_d} = \frac{1}{2}\begin{pmatrix} \mathbf{0} & \mathbf{I}_d \\ \mathbf{I}_d & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{2d\times 2d}. \tag{86}$$

By letting $\boldsymbol{u} = [\boldsymbol{u}_i^\mathsf{T}, \boldsymbol{u}_j^\mathsf{T}]^\mathsf{T}$, it follows from $A(2)$-polynomial (45) with $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}_\mathbf{G}$ that

$$\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_j)^2\right] = \mathrm{tr}(\mathbf{G}^2)\,. \tag{87}$$

Similarly, it follows from $A(4)$-polynomial (47) with $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}_\mathbf{G}$, $\mathbf{A}_3 = \mathbf{A}_4 = \mathbf{A}_{\mathbf{I}_d}$ that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[(\boldsymbol{u}_i^\mathsf{T}\mathbf{G}\boldsymbol{u}_j)^2(\boldsymbol{u}_i^\mathsf{T}\boldsymbol{u}_j)^2\right] &= 16\left[\frac{3\mathrm{tr}(\mathbf{G}^2)}{8}\right] + 4\left[\frac{\mathrm{tr}(\mathbf{G}^2)}{2}\frac{d}{2} + 2\left(\frac{\mathrm{tr}(\mathbf{G})}{2}\right)^2\right] \\ &= (d+6)\mathrm{tr}(\mathbf{G}^2) + 2\mathrm{tr}(\mathbf{G})^2\,. \end{aligned} \tag{88}$$

Therefore, the expectation of the trace term is given as

$$\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\mathrm{tr}(\mathbf{D}^2)\right] = (d+6)\mathrm{tr}(\mathbf{G}^2)+2\mathrm{tr}(\mathbf{G})^2-2\mathrm{tr}(\mathbf{G}^2)+\mathrm{tr}(\mathbf{G}^2) = (d+5)\mathrm{tr}(\mathbf{G}^2)+2\mathrm{tr}(\mathbf{G})^2\,. \tag{89}$$

Since $\mathbf{D} = \tilde{\mathbf{G}}(\boldsymbol{u}_i, \boldsymbol{u}_j) - \mathbf{G}$, we already have $\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\|\mathbf{D}\|_F^2\right]$ in (55). Therefore,

$$\mathbb{E}_{\boldsymbol{u}_i,\boldsymbol{u}_j}\left[\left\|\tilde{\mathbf{G}}^s(\boldsymbol{u}_i,\boldsymbol{u}_j) - \mathbf{G}\right\|_F^2\right] = \frac{1}{2}\left[(d^2+4d+3)\mathrm{tr}(\mathbf{G}^2) + (d+5)\mathrm{tr}(\mathbf{G}^2) + 2\mathrm{tr}(\mathbf{G})^2\right]\,. \tag{90}$$

Simplifying the above leads to the conclusion (83). □

E Proof of Theorem 3.3 (whitening residual)

The proof of Theorem 3.3 is given.

Proof. The whitening result of the true regularized FIM can be written as

$$\begin{aligned}\gamma\,\mathbf{W}_{t+1}^\mathsf{T}\mathbf{W}_{t+1} + \mathbf{G}_{t+1} &= \mathbf{W}_{t+1}^\mathsf{T}\left(\gamma\,\mathbf{I}_d + \mathbf{F}_t\right)\mathbf{W}_{t+1} \\ &= \mathbf{L}_t^{-1}\mathbf{W}_t^\mathsf{T}\left(\gamma\,\mathbf{I}_d + \mathbf{F}_t\right)\mathbf{W}_t\mathbf{L}_t^{-\mathsf{T}} \\ &= \mathbf{L}_t^{-1}\left(\gamma\,\mathbf{W}_t^\mathsf{T}\mathbf{W}_t + \mathbf{G}_t\right)\mathbf{L}_t^{-\mathsf{T}}\,. \end{aligned} \tag{91}$$

From (23), we have $\mathbf{I}_d = \mathbf{L}_t^{-1}\left(\gamma\,\mathbf{W}_t^\mathsf{T}\mathbf{W}_t + \mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})\right)\mathbf{L}_t^{-\mathsf{T}}$. Therefore, the whitening residual can be bounded as

$$\begin{aligned}\|\gamma\,\mathbf{W}_{t+1}^\mathsf{T}\mathbf{W}_{t+1} + \mathbf{G}_{t+1} - \mathbf{I}_d\|_F &= \|\mathbf{L}_t^{-1}\left(\mathbf{G}_t - \mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})\right)\mathbf{L}_t^{-\mathsf{T}}\|_F \\ &\le \|\mathbf{L}_t^{-1}\|_2^2\,\|\mathbf{G}_t - \mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})\|_F \\ &\le \|\mathbf{L}_t^{-1}\|_2^2\left(\delta + \|\hat{\mathbf{G}}_t^{\mathrm{ie}} - \mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})\|_F\right)\,, \end{aligned} \tag{92}$$

where $\|\cdot\|_2$ denotes the spectral norm. Since $\mathbf{R}_t$ in (21) is positive definite (as $\mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})$ has minimum eigenvalue at least $\tau > 0$ and $\gamma\,\mathbf{W}_t^\mathsf{T}\mathbf{W}_t$ is positive semidefinite), its Cholesky decomposition $\mathbf{R}_t = \mathbf{L}_t\mathbf{L}_t^\mathsf{T}$ yields a full-rank lower triangular matrix $\mathbf{L}_t$, and thus $\|\mathbf{L}_t^{-1}\|_2^2 = \|(\mathbf{L}_t\mathbf{L}_t^\mathsf{T})^{-1}\|_2 = \|\mathbf{R}_t^{-1}\|_2$. The minimum eigenvalue of $\mathbf{R}_t$ is lower bounded by that of $\mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})$, which is at least τ, and thus

$$\|\mathbf{L}_t^{-1}\|_2^2 = \|\mathbf{R}_t^{-1}\|_2 \le \|\mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})^{-1}\|_2 \le \tau^{-1}. \tag{93}$$

Because $\hat{\mathbf{G}}_t^{\mathrm{ie}}$ and $\mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})$ share the same eigenvectors and the only differences are in the clipped eigenvalues as shown in (19), it follows that

$$\left\|\hat{\mathbf{G}}_t^{\mathrm{ie}} - \mathrm{clip}(\hat{\mathbf{G}}_t^{\mathrm{ie}})\right\|_F = \left(\sum_{i=1}^{d}\max(0,\tau-\nu_i)^2\right)^{1/2}\,. \tag{94}$$

Substituting (93) and (94) into (92) yields the bound (24). □

F Convergence analysis

This appendix details the convergence analysis stated as Theorem 3.4 in Subsection 3.3.

F.1 Assumptions

We begin with two standard assumptions. For notational simplicity, we abbreviate $\nabla_{\boldsymbol{\theta}}$ as ∇.

Assumption F.1 (L-smooth)**.** For any mini-batch $\mathcal{D}$, the mini-batch loss $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$ is L-smooth, i.e.,

$$\left|\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}') - \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) - \nabla\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})^{\mathsf{T}}(\boldsymbol{\theta}' - \boldsymbol{\theta})\right| \leq \frac{L}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2 \tag{95}$$

for any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$ [58].

Consequently, the full-batch loss $\mathcal{L}(\boldsymbol{\theta})$ is also L-smooth:

$$\mathcal{L}(\boldsymbol{\theta}') - \mathcal{L}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta})^{\mathsf{T}}(\boldsymbol{\theta}' - \boldsymbol{\theta}) \leq \frac{L}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2 . \tag{96}$$

Assumption F.2 (σ^2-bounded stochastic gradient noise)**.** The mini-batch stochastic gradient $\nabla\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$ is an unbiased estimator of the full-batch gradient $\nabla\mathcal{L}(\boldsymbol{\theta})$, and its variance is bounded as

$$\mathbb{E}_{\mathcal{D}}[\nabla\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})] = \nabla\mathcal{L}(\boldsymbol{\theta}) , \quad \mathbb{E}_{\mathcal{D}}\left[\|\nabla\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta})\|^2\right] \leq \sigma^2 . \tag{97}$$

The next assumption is specific to the SO approach, where we use preconditioners $\mathbf{W}$ or $\mathbf{P} = \mathbf{W}\mathbf{W}^{\mathsf{T}}$.

Assumption F.3 (positive-definite preconditioner covariance matrix)**.** For $\mathbf{W}_t$ computed in line 9 of Algorithm 1, the minimum eigenvalue of $\mathbf{W}_t\mathbf{W}_t^{\mathsf{T}}$ is bounded as $\lambda_{\min}(\mathbf{W}_t\mathbf{W}_t^{\mathsf{T}}) \geq m > 0$ for all t.

F.2 Proof of Theorem 3.4 (convergence rate)

Now, we present the proof of Theorem 3.4, followed by supporting lemmas.

Proof. It follows from (96) by letting $\boldsymbol{\theta}' = \boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ that

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t) + \nabla\mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + \frac{L}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2. \tag{98}$$

Since $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\hat{\tilde{\boldsymbol{g}}}_t$,

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}}\hat{\tilde{\boldsymbol{g}}}_t + \frac{L\eta^2}{2}\|\hat{\tilde{\boldsymbol{g}}}_t\|^2. \tag{99}$$

Let $\mathcal{D}_t$ be the mini-batch sampled at the t-th step in line 4 of Algorithm 1 and $\{\boldsymbol{u}_{i,t}\}_{i=1}^q$ be the random perturbation vectors sampled in line 5 of Algorithm 1. Taking expectation for (99) with respect to $\{\boldsymbol{u}_{i,t}\}_{i=1}^q$ and $\mathcal{D}_t$, we have

$$\begin{aligned}
&\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] && (100)\\
&\leq \mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\mathcal{L}(\boldsymbol{\theta}_t) - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}}\hat{\tilde{\boldsymbol{g}}}_t + \frac{L\eta^2}{2}\|\hat{\tilde{\boldsymbol{g}}}_t\|^2\right] \\
&= \mathcal{L}(\boldsymbol{\theta}_t) - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}}\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\tilde{\boldsymbol{g}}}_t\right] + \frac{L\eta^2}{2}\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\|\hat{\tilde{\boldsymbol{g}}}_t\|^2\right] . && (101)
\end{aligned}$$

The second term in (101) can be bounded as

$$
\begin{aligned}
&- \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}} \mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\bar{\boldsymbol{g}}}_t\right] \\
&= -\eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}} \mathbf{W}_t \mathbf{W}_t^{\mathsf{T}} \nabla \mathcal{L}(\boldsymbol{\theta}_t) - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}} (\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\bar{\boldsymbol{g}}}_t\right] - \mathbf{W}_t \mathbf{W}_t^{\mathsf{T}} \nabla \mathcal{L}(\boldsymbol{\theta}_t)) \\
&\overset{\text{(a)}}{\leq} -\eta m \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)^{\mathsf{T}} (\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\bar{\boldsymbol{g}}}_t\right] - \mathbf{W}_t \mathbf{W}_t^{\mathsf{T}} \nabla \mathcal{L}(\boldsymbol{\theta}_t)) \\
&\leq -\eta m \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \eta \sqrt{m} \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\| \frac{1}{\sqrt{m}} \|\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\bar{\boldsymbol{g}}}_t\right] - \mathbf{W}_t \mathbf{W}_t^{\mathsf{T}} \nabla \mathcal{L}(\boldsymbol{\theta}_t)\| \\
&\overset{\text{(b)}}{\leq} -\eta m \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta m \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2}{2} + \frac{\eta \|\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\bar{\boldsymbol{g}}}_t\right] - \mathbf{W}_t \mathbf{W}_t^{\mathsf{T}} \nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2}{2m} \\
&\overset{\text{(c)}}{\leq} -\frac{\eta m}{2} \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta \mu^2 L^2 (d+3)^3}{8 m \gamma^3},
\end{aligned} \tag{102}
$$

where (a) follows from Assumption F.3, (b) from Young's inequality, and (c) from Lemma F.2.

The third term in (101) can be bounded using Lemma F.3 and Assumption F.2 (which gives $\mathbb{E}_{\mathcal{D}_t}\left[\|\nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2\right] \leq \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \sigma^2$) as

$$
\begin{aligned}
\frac{L\eta^2}{2} \mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\|\hat{\bar{\boldsymbol{g}}}_t\|^2\right] &\leq \frac{L\eta^2}{2} \mathbb{E}_{\mathcal{D}_t}\left[\frac{2}{\gamma^2}(d+2)\|\nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2 + \frac{\mu^2 L^2}{2\gamma^3}(8d + 6d^2 + d^3)\right] \\
&\leq \frac{L\eta^2 (d+2)}{\gamma^2}\left(\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \sigma^2\right) + \frac{\eta^2 \mu^2 L^3 (8d + 6d^2 + d^3)}{4\gamma^3}.
\end{aligned} \tag{103}
$$

By letting the learning rate as $\eta = \min\left(\frac{m\gamma^2}{4L(d+2)}, T^{-\frac{1}{2}} d^{-\frac{1}{2}}\right)$, it follows from (101), (102), and (103) that

$$
\begin{aligned}
\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] \leq \mathcal{L}(\boldsymbol{\theta}_t) &- \frac{\eta m}{4} \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta \mu^2 L^2 (d+3)^3}{8 m \gamma^3} \\
&+ \frac{L\eta^2 (d+2)\sigma^2}{\gamma^2} + \frac{\eta^2 \mu^2 L^3 (8d + 6d^2 + d^3)}{4\gamma^3}.
\end{aligned} \tag{104}
$$

Summing this inequality from $t = 0$ to $T-1$ and dividing it by $\frac{\eta m T}{4}$ yields

$$
\begin{aligned}
&\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2\right] \\
&\leq \frac{4(\mathcal{L}(\boldsymbol{\theta}_0) - \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)])}{\eta m T} + \frac{\mu^2 L^2 (d+3)^3}{2m^2 \gamma^3} + \frac{4L\eta(d+2)\sigma^2}{m\gamma^2} + \frac{\eta \mu^2 L^3 (8d + 6d^2 + d^3)}{m\gamma^3} \\
&\overset{\text{(a)}}{\leq} \frac{4(\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*)}{\eta m T} + \frac{\mu^2 L^2 (d+3)^3}{2m^2 \gamma^3} + \frac{4L\eta(d+2)\sigma^2}{m\gamma^2} + \frac{\eta \mu^2 L^3 (8d + 6d^2 + d^3)}{m\gamma^3} \\
&\overset{\text{(b)}}{\leq} \frac{4(\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*)}{\eta m T} + \frac{\mu^2 L^2 (d+3)^3}{2m^2 \gamma^3} + \frac{4L(d+2)\sigma^2}{T^{\frac{1}{2}} d^{\frac{1}{2}} m\gamma^2} + \frac{\mu^2 L^3 (8d + 6d^2 + d^3)}{T^{\frac{1}{2}} d^{\frac{1}{2}} m\gamma^3},
\end{aligned} \tag{105}
$$

where (a) is due to the fact that $\mathcal{L}(\boldsymbol{\theta}) \geq \mathcal{L}^*$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$ and (b) comes from $\eta \leq T^{-\frac{1}{2}} d^{-\frac{1}{2}}$.

Here, since $\eta = \min\left(\frac{m\gamma^2}{4L(d+2)}, T^{-\frac{1}{2}} d^{-\frac{1}{2}}\right)$, we have $\eta^{-1} \leq \max\left(\frac{4L(d+2)}{m\gamma^2}, T^{\frac{1}{2}} d^{\frac{1}{2}}\right) \leq \frac{4L(d+2)}{m\gamma^2} + T^{\frac{1}{2}} d^{\frac{1}{2}}$. Moreover, $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2\right] = \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{\theta}_R)\|^2\right]$ from the definition of R. It thus follows from (105) that

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{\theta}_R)\|^2\right] \leq \frac{4(\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*)}{mT}&\left(\frac{4L(d+2)}{m\gamma^2} + T^{\frac{1}{2}} d^{\frac{1}{2}}\right) + \frac{\mu^2 L^2 (d+3)^3}{2m^2 \gamma^3} \\
&+ \frac{4L(d+2)\sigma^2}{T^{\frac{1}{2}} d^{\frac{1}{2}} m\gamma^2} + \frac{\mu^2 L^3 (8d + 6d^2 + d^3)}{T^{\frac{1}{2}} d^{\frac{1}{2}} m\gamma^3}.
\end{aligned} \tag{106}
$$

When μ is sufficiently small,

$$
\mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{\theta}_R)\|^2\right] = O\left(T^{-1} d + T^{-\frac{1}{2}} d^{\frac{1}{2}}\right). \tag{107}
$$

Therefore, the iteration complexity required to achieve $\mathbb{E}\big[\|\nabla\mathcal{L}(\boldsymbol{\theta}_R)\|^2\big] \leq \epsilon^2$ is $T = O(\epsilon^{-4}d)$. $\square$

Lemma F.1. *Let $\{\mathbf{W}_t\}_{t=1}^T$ be the iterates generated by Algorithm 1. For any step $t \geq 1$, the maximum eigenvalue of $\mathbf{W}_t\mathbf{W}_t^\mathsf{T}$ is bounded as*

$$\lambda_{\max}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T}) \leq \gamma^{-1}, \tag{108}$$

where γ is the damping coefficient used in (4)*.*

Proof. From the second equation of (22), $\hat{\mathbf{G}}_{t+1}$ is positive definite, and it follows from (23) that $\lambda_{\max}(\gamma\mathbf{W}_{t+1}^\mathsf{T}\mathbf{W}_{t+1}) \leq 1$ for any step $t \geq 0$. Therefore, for any step $t \geq 1$, we have that

$$\lambda_{\max}(\gamma\mathbf{W}_t^\mathsf{T}\mathbf{W}_t) \leq 1 \;\Leftrightarrow\; \lambda_{\max}(\mathbf{W}_t^\mathsf{T}\mathbf{W}_t) \leq \gamma^{-1}.$$

Since $\lambda_{\max}(\mathbf{W}_t^\mathsf{T}\mathbf{W}_t) = \lambda_{\max}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T})$, we have the conclusion (108). $\square$

Remark F.1 (spectral bounds for block-diagonal preconditioner and less frequent preconditioner update)**.** The convergence analysis discussed here uses the following spectral bound from Assumption F.3 and (108) in Lemma F.1:

$$0 < m \leq \lambda_{\min}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T}), \quad \lambda_{\max}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T}) \leq \gamma^{-1}. \quad \text{(spectral bound)} \tag{109}$$

This bound is maintained even for the block-diagonal preconditioner and the less frequent preconditioner update discussed in Subsection 3.4, as shown below. **Block-diagonal preconditioner:** We have $m \leq \lambda_{\min}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T}) = \min_{i=1,\dots,b}\lambda_{\min}\big(\mathbf{W}_t^{(i)}\mathbf{W}_t^{(i)\top}\big)$. The clipping/whitening update (19), (21), and (22) is applied independently to each block i. Hence the derivation of (108) in Lemma F.1, which relies on (23) giving $\lambda_{\max}\big(\gamma\mathbf{W}_t^{(i)}\mathbf{W}_t^{(i)\top}\big) \leq 1$, applies to each block, yielding $\lambda_{\max}(\mathbf{W}_t\mathbf{W}_t^\top) = \max_{i=1,\dots,b}\lambda_{\max}\big(\mathbf{W}_t^{(i)}\mathbf{W}_t^{(i)\top}\big) \leq \gamma^{-1}$. **Less frequent preconditioner update:** Since $\mathbf{W}_{t+1} = \mathbf{W}_t$ at any iteration t for which the update is skipped, the spectral bound is preserved for all t.

Let us denote $\|\boldsymbol{g}\|_\mathbf{P}^2 = \boldsymbol{g}^\mathsf{T}\mathbf{P}\boldsymbol{g}$ for any vector $\boldsymbol{g} \in \mathbb{R}^d$ and symmetric matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$ in the proofs of the following lemmas.

Lemma F.2. *Let $\{\hat{\tilde{\boldsymbol{g}}}_t\}_{t=0}^{T-1}$ and $\{\mathbf{W}_t\}_{t=0}^{T-1}$ be the iterates generated by Algorithm 1. Under Assumptions F.1 and F.2, the bias of $\hat{\tilde{\boldsymbol{g}}}_t$ with respect to the preconditioned gradient $\mathbf{W}_t\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}(\boldsymbol{\theta}_t)$ is bounded by*

$$\|\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q,\mathcal{D}_t}\big[\hat{\tilde{\boldsymbol{g}}}_t\big] - \mathbf{W}_t\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}(\boldsymbol{\theta}_t)\| \leq \frac{\mu L}{2}\left(\frac{d+3}{\gamma}\right)^{3/2}. \tag{110}$$

Proof. It follows from Lemma F.4 with $|\varepsilon_{i,t}| \leq \frac{\mu L}{2}\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^2$ that

$$\begin{aligned}\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q,\mathcal{D}_t}\big[\hat{\tilde{\boldsymbol{g}}}_t\big] &= \mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q,\mathcal{D}_t}\left[\frac{1}{q}\sum_{i=1}^q \mathbf{W}_t\boldsymbol{u}_{i,t}\left(\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + \varepsilon_{i,t}\right)\right] \\ &= \frac{1}{q}\sum_{i=1}^q \mathbb{E}_{\boldsymbol{u}_{i,t}}\big[\mathbf{W}_t\boldsymbol{u}_{i,t}\left(\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}(\boldsymbol{\theta}_t) + \varepsilon_{i,t}\right)\big], \end{aligned} \tag{111}$$

where we used $\mathbb{E}_{\mathcal{D}_t}[\nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)] = \nabla \mathcal{L}(\boldsymbol{\theta}_t)$ from Assumption F.2 in the second equality. We thus have that

$$
\begin{aligned}
&\|\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\tilde{\boldsymbol{g}}}_t\right] - \mathbf{W}_t \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}(\boldsymbol{\theta}_t)\| \\
&= \left\| \mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q, \mathcal{D}_t}\left[\hat{\tilde{\boldsymbol{g}}}_t\right] - \frac{1}{q}\sum_{i=1}^q \mathbf{W}_t \mathbb{E}_{\boldsymbol{u}_{i,t}}\left[\boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T}\right] \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}(\boldsymbol{\theta}_t) \right\| \\
&= \left\| \frac{1}{q}\sum_{i=1}^q \mathbb{E}_{\boldsymbol{u}_{i,t}}[\mathbf{W}_t \boldsymbol{u}_{i,t} \varepsilon_{i,t}] \right\| \\
&\le \frac{1}{q}\sum_{i=1}^q \left\| \mathbb{E}_{\boldsymbol{u}_{i,t}}[\mathbf{W}_t \boldsymbol{u}_{i,t} \varepsilon_{i,t}] \right\| \quad \text{(triangle inequality)} \\
&\le \frac{1}{q}\sum_{i=1}^q \mathbb{E}_{\boldsymbol{u}_{i,t}}[\|\mathbf{W}_t \boldsymbol{u}_{i,t}\| |\varepsilon_{i,t}|] \quad \text{(Jensen's inequality)} \\
&\le \frac{\mu L}{2q}\sum_{i=1}^q \mathbb{E}_{\boldsymbol{u}_{i,t}}\left[\|\mathbf{W}_t \boldsymbol{u}_{i,t}\| \|\boldsymbol{u}_{i,t}\|^2_{\mathbf{W}_t \mathbf{W}_t^\mathsf{T}}\right] \\
&\le \frac{\mu L}{2q}\|\mathbf{W}_t\|_2 \|\mathbf{W}_t \mathbf{W}_t^\mathsf{T}\|_2 \sum_{i=1}^q \mathbb{E}_{\boldsymbol{u}_{i,t}}\left[\|\boldsymbol{u}_{i,t}\| \|\boldsymbol{u}_{i,t}\|^2\right] \\
&\overset{\text{(a)}}{\le} \frac{\mu L}{2q}\gamma^{-3/2}\sum_{i=1}^q \mathbb{E}_{\boldsymbol{u}_{i,t}}\left[\|\boldsymbol{u}_{i,t}\| \|\boldsymbol{u}_{i,t}\|^2\right] \\
&\overset{\text{(b)}}{\le} \frac{\mu L}{2}\left(\frac{d+3}{\gamma}\right)^{3/2},
\end{aligned} \tag{112}
$$

where (a) follows from Lemma F.1 and (b) comes from Lemma 1 of [12] for evaluating $\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\|\boldsymbol{u}\|\|\boldsymbol{u}\|^2\right] \le (d+3)^{3/2}$ and the fact that $\boldsymbol{u}_{i,t}$ are independently and identically distributed. □

Lemma F.3. *Let $\{\hat{\tilde{\boldsymbol{g}}}_t\}_{t=0}^{T-1}$ be the iterates generated by Algorithm 1. Under Assumption F.1, Lemma F.4 holds, and the second-order moment of $\hat{\tilde{\boldsymbol{g}}}_t$ is bounded as*

$$
\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q}\left[\|\hat{\tilde{\boldsymbol{g}}}_t\|^2\right] \;\le\; \frac{2}{\gamma^2}(d+2)\|\nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2 + \frac{\mu^2 L^2}{2\gamma^3}(8d + 6d^2 + d^3)\,. \tag{113}
$$

Proof. From Lemma F.4 with $|\varepsilon_{i,t}| \le \frac{\mu L}{2}\|\mathbf{W}_t \boldsymbol{u}_{i,t}\|^2$ and the convexity of $\|\cdot\|^2$ (i.e., $\|\frac{1}{q}\sum_{i=1}^q \boldsymbol{a}_i\|^2 \le \frac{1}{q}\sum_{i=1}^q \|\boldsymbol{a}_i\|^2$) and $\|\boldsymbol{a}+\boldsymbol{b}\|^2 \le 2\|\boldsymbol{a}\|^2 + 2\|\boldsymbol{b}\|^2$, we have

$$
\begin{aligned}
\|\hat{\tilde{\boldsymbol{g}}}_t\|^2 &\le \frac{1}{q}\sum_{i=1}^q \|\mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + \mathbf{W}_t \boldsymbol{u}_{i,t}\varepsilon_{i,t}\|^2 \\
&\le \frac{1}{q}\sum_{i=1}^q \left(2\|\mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2 + 2\|\mathbf{W}_t \boldsymbol{u}_{i,t}\varepsilon_{i,t}\|^2\right) \\
&\le \frac{1}{q}\sum_{i=1}^q \left(2\|\mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2 + \frac{\mu^2 L^2}{2}\|\mathbf{W}_t \boldsymbol{u}_{i,t}\|^6\right).
\end{aligned} \tag{114}
$$

Now, for evaluating the expectations of the right-hand terms, we use $A(s)$-polynomial in Proposition C.1. For that purpose, we first rearrange $\|\mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2$ as

$$
\begin{aligned}
\|\mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2 &= \mathrm{tr}(\nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)^\mathsf{T} \mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)) \\
&= \boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \mathbf{W}_t \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T} \mathbf{W}_t^\mathsf{T} \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t) \nabla \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)^\mathsf{T} \mathbf{W}_t \boldsymbol{u}_{i,t}\,.
\end{aligned} \tag{115}
$$

Setting $\mathbf{A}_1 = \mathbf{W}_t^\mathsf{T}\mathbf{W}_t$ and $\mathbf{A}_2 = \mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)^\mathsf{T}\mathbf{W}_t$ in the $A(2)$-polynomial (45) gives

$$\begin{aligned}
&\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q}\left[\|\mathbf{W}_t\boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2\right] \\
&= \mathrm{tr}(\mathbf{W}_t^\mathsf{T}\mathbf{W}_t)\,\mathrm{tr}(\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)^\mathsf{T}\mathbf{W}_t) + 2\mathrm{tr}(\mathbf{W}_t^\mathsf{T}\mathbf{W}_t\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)^\mathsf{T}\mathbf{W}_t) \\
&= \mathrm{tr}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T})\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2_{\mathbf{W}_t\mathbf{W}_t^\mathsf{T}} + 2\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2_{(\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^2}\,. \qquad (116)
\end{aligned}$$

We then use the $A(3)$-polynomial (46) by setting $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}_3 = \mathbf{W}_t^\mathsf{T}\mathbf{W}_t$ to evaluate the expectation of $\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^6 = (\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\mathbf{W}_t\boldsymbol{u}_{i,t})^3$, giving

$$\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q}\left[\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^6\right] = \mathrm{tr}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^3 + 8\mathrm{tr}((\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^3) + 6\mathrm{tr}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T})\mathrm{tr}((\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^2)\,. \qquad (117)$$

Now, according to Lemma F.1, we have that

$$\begin{aligned}
\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2_{\mathbf{W}_t\mathbf{W}_t^\mathsf{T}} &\le \|\mathbf{W}_t\mathbf{W}_t^\mathsf{T}\|_2\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2 \le \gamma^{-1}\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2\,, \\
\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2_{(\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^2} &\le \|\mathbf{W}_t\mathbf{W}_t^\mathsf{T}\|_2\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2_{\mathbf{W}_t\mathbf{W}_t^\mathsf{T}} \le \gamma^{-2}\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2\,, \\
\mathrm{tr}(\mathbf{W}_t\mathbf{W}_t^\mathsf{T}) &\le \gamma^{-1}d\,, \quad \mathrm{tr}((\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^2) \le \gamma^{-2}d\,, \quad \mathrm{tr}((\mathbf{W}_t\mathbf{W}_t^\mathsf{T})^3) \le \gamma^{-3}d\,.
\end{aligned}$$

Therefore, we have the following bounds for the expectations:

$$\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q}\left[\|\mathbf{W}_t\boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2\right] \;\le\; (d+2)\gamma^{-2}\|\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\|^2\,, \qquad (118)$$

$$\mathbb{E}_{\{\boldsymbol{u}_{i,t}\}_{i=1}^q}\left[\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^6\right] \;\le\; \gamma^{-3}(8d + 6d^2 + d^3)\,. \qquad (119)$$

Taking the expectations for (114), using (118) and (119), and noting that $\boldsymbol{u}_{i,t}$ are independent and identically distributed gives (113). This completes the proof. □

Lemma F.4. *With Assumption F.1, the preconditioned approximate gradient* (5) *satisfies*

$$\hat{\tilde{\boldsymbol{g}}}_t = \frac{1}{q}\sum_{i=1}^q \mathbf{W}_t\boldsymbol{u}_{i,t}\left(\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + \varepsilon_{i,t}\right)\,, \qquad (120)$$

where $\varepsilon_{i,t}$ is a scalar satisfying $|\varepsilon_{i,t}| \le \frac{\mu L}{2}\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^2$.

Proof. From Assumption F.1 applied twice, with $\boldsymbol{\theta}' = \boldsymbol{\theta}_t + \mu\mathbf{W}_t\boldsymbol{u}_{i,t}$, $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ and with $\boldsymbol{\theta}' = \boldsymbol{\theta}_t - \mu\mathbf{W}_t\boldsymbol{u}_{i,t}$, $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, and the triangle inequality, we have that

$$\left|\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t + \mu\mathbf{W}_t\boldsymbol{u}_{i,t}) - \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t - \mu\mathbf{W}_t\boldsymbol{u}_{i,t}) - 2\mu\boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t)\right| \le \mu^2 L\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^2\,. \qquad (121)$$

This means that we can write

$$\frac{\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t + \mu\mathbf{W}_t\boldsymbol{u}_{i,t}) - \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t - \mu\mathbf{W}_t\boldsymbol{u}_{i,t})}{2\mu} = \boldsymbol{u}_{i,t}^\mathsf{T}\mathbf{W}_t^\mathsf{T}\nabla\mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + \varepsilon_{i,t}\,, \qquad (122)$$

where $|\varepsilon_{i,t}| \le \frac{\mu L}{2}\|\mathbf{W}_t\boldsymbol{u}_{i,t}\|^2$. Substituting this into (5), we have the conclusion (120). □

G Detailed experimental settings and further experimental results

This appendix describes detailed settings of the experiments reported in Section 4. Further experimental results are also provided here. We coded the programs with PyTorch [59], and ran them on an NVIDIA RTX 6000 (48 GB memory). The source code is accessible at `https://github.com/nttcslab/ZO-F2`.

G.1 Linear model (MNIST)

For a 28×28 pixel image, we applied fixed random projections to reduce the dimensionality from 784 to d_{rp}, yielding the input vector $\boldsymbol{x} \in \mathbb{R}^{d_{\mathrm{rp}}}$, where d_{rp} varied as $d_{\mathrm{rp}} = 32, 48, 64, 80$. We applied a linear module without a bias parameter, defined as $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^{d_{\mathrm{out}}}$ with $d_{\mathrm{out}} = 10$, for classifying 10 classes. Consequently, the dimensionality of the parameter vector $\boldsymbol{\theta}$ varied as

$d = 320, 480, 640, 800$. Table 3 reports the results with $d = 800$. We specified $d_{\max} = 800$, making the preconditioner $\mathbf{W}$ a full matrix rather than block-diagonal for all the settings of d.

The Hessian $\mathbf{H}$ and FIM $\mathbf{F}$ in this case are the same because the expression (43) in Appendix B holds and $\nabla^2_{\boldsymbol{\theta}} [\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})]_i$, $i = 1, \ldots, d_{\text{out}}$, therein are zero matrices. Furthermore, the true FIM can be easily computed based on the decomposition (11) and the fact that the Jacobian of the linear model is simply given as $\mathbf{J}_{\boldsymbol{y}\boldsymbol{\theta}} = \boldsymbol{x}^\mathsf{T} \otimes \mathbf{I}_{d_{\text{out}}}$.

The mini-batch size and the number of epochs were 100 and 200, respectively, giving 120,000 steps in total. We used stochastic gradient descent (SGD) as the optimizer with a learning rate of $\eta = 0.001$. The other hyperparameter settings were $\mu = 10^{-3}$, $\alpha = 10^{-4}$, $\gamma = 0.01$, $\tau = 0.5$, and $T_{\text{ud}} = 100$, unless otherwise specified.

Figure 3 shows the estimation error (mean ± standard deviation over five runs) of the Hessian for ZO-H2 and the FIM for ZO-F2 when varying the number q of perturbations. We observe the following. With the minimum number $q = 2$ of perturbations, the number of constructed samples for $\tilde{\mathbf{G}}$ is only one for both methods ZO-H2 and ZO-F2. The estimation errors were large for both methods. However, by increasing the number of perturbations as $q \geq 4$, the number of constructed samples for $\tilde{\mathbf{G}}$ becomes larger in ZO-F2 than in ZO-H2, as shown in Table 2. Consequently, the estimation errors were smaller in ZO-F2 than in ZO-H2.

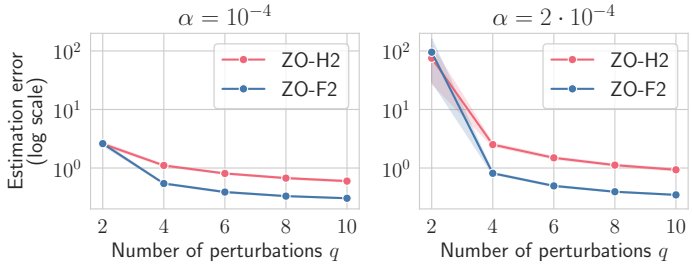


Figure 3: Estimation errors with the linear model (MNIST) with varying number of perturbations q.

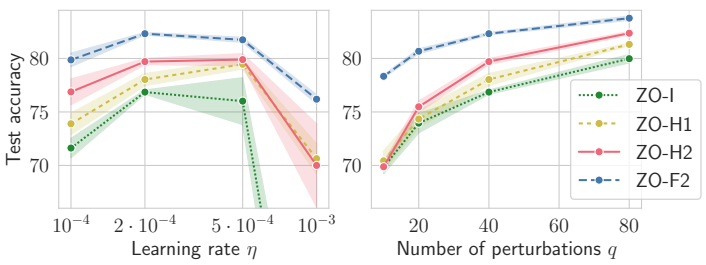
G.2 CNN (EMNIST)

We designed a CNN with $d = 15{,}903$ parameters to handle the 47-class EMNIST Balanced dataset. The architecture consists of three convolutional layers with a channel configuration of 8-16-16, followed by two fully connected layers with a width transition of 144-64-47. We specified $d_{\max} = 497$, making the preconditioner $\mathbf{W}$ block-diagonal with $b = 32$ matrix blocks. The mini-batch size and the number of epochs were 100 and 300, respectively, giving 338,400 steps in total. We adopted Adam [60] instead of SGD as the optimizer due to its robustness to hyperparameter choices and its stable convergence behavior. The hyperparameter settings were $\mu = 10^{-3}$, $\eta = 2{\cdot}10^{-4}$, $\alpha = 3{\cdot}10^{-6}$, $\gamma = 0.01$, $\tau = 0.5$, and $T_{\text{ud}} = 100$, unless otherwise specified.

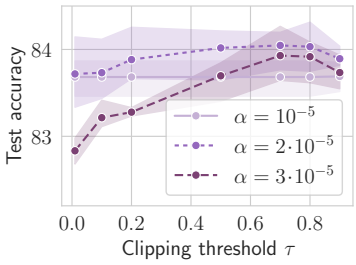
Figure 4a shows the CNN results by varying the learning rate η and the number of perturbations q, where $\eta = 2{\cdot}10^{-4}$ and $q = 40$ were the default settings for Table 3. We observe that the proposed method ZO-F2 outperformed the other methods in all settings.

Figure 4b shows the results corresponding to Figure 2b but with forward difference instead of central difference. We observe that the performance degradations with small clipping thresholds (e.g., $\tau \leq 0.2$) were more prominent when $\alpha = 3{\cdot}10^{-5}$ than those in Figure 2b. These observations regarding τ are again consistent with Theorem 3.3, and clearly demonstrate the importance of choosing a moderately large clipping threshold (e.g., $0.5 \leq \tau \leq 0.8$).

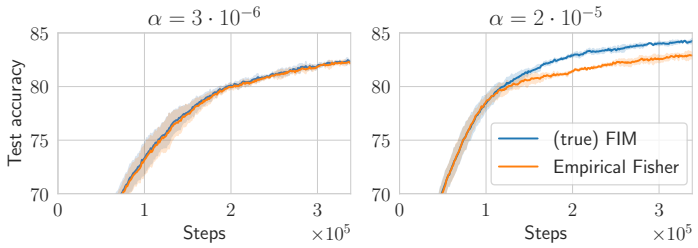
Figure 4c shows the CNN results by the empirical Fisher (36) together with those of the (true) FIM (35). When the EMA coefficient is conservative as in the left plot, the true and empirical Fisher yielded nearly identical accuracies across all steps. However, when the EMA coefficient is aggressive as in the right plot, the (true) FIM performed clearly better than the empirical Fisher, where the known pathologies of the empirical Fisher [54] were observed.



(a) CNN results varying (left) learning rate η and (right) the number of perturbations q.



(b) CNN results (forward difference): larger EMA α with varying clipping threshold τ.



(c) CNN results with the (true) FIM and the empirical Fisher (36), with two EMA coefficients $\alpha = 3 \cdot 10^{-6}$ (left, conservative) and $\alpha = 2 \cdot 10^{-5}$ (right, aggressive).

Figure 4: Further experimental results on the CNN. Shaded areas represent one standard deviation over three runs with random seeds.

G.3 RNN (Copying)

The Copying memory task tests how well an RNN remembers data seen many steps before. An example of input target sequence is

Input: `7318242831----------:---------`

Target: `----------7318242831`

where `7318242831` is the data to remember, with possible alphabet symbols from `1` to `8`. Here, `-` and `:` are special symbols meaning 'blank' and 'start', respectively. In the experiment, the data length to remember and the total length were 10 and 120, respectively. The training and test datasets contained 60,000 and 10,000 sequences, respectively.

To handle the above 10 symbols, the RNN had a 10-dimensional output. The hidden dimensionality of the RNN was 32. The total number of parameters was $d = 3{,}168$, and we set $d_{\max} = 512$, resulting in $b = 7$ matrix blocks. The mini-batch size and the number of epochs were 100 and 100, respectively, giving 60,000 steps in total. We adopted Adam [60] for its robustness to hyperparameter choices and its stable convergence behavior. The hyperparameter settings were $\mu = 10^{-3}$, $\eta = 10^{-4}$, $\alpha = 10^{-5}$, $\gamma = 0.1$, $\tau = 0.5$, and $T_{\mathrm{ud}} = 100$.

Figure 5 shows the structure of the Clements mesh [44], which is a specialized architecture for optical neural networks. It consists of a layered array of Mach-Zehnder interferometers (MZIs) colored in gray. An MZI consists of two pairs of phase shifters (PS) and beam splitters (BS), whose 2×2 transfer matrices are shown alongside. An MZI realizes a 2×2 complex unitary matrix depending on the settings of the phase parameters θ colored in orange. The Clements mesh realizes a unitary matrix of larger size by combining multiple MZIs in a layered structure.

Table 5 shows how test accuracy decreases as the block size decreases for the RNN(Copying) task. We selected this task from Table 3 because the number of parameters is the maximum number, among the four tasks, that we can compute the full-matrix preconditioner within practical computational cost. We observe the following. The full-matrix preconditioner, $d = d_{\max} = 3168$, attained the best test accuracy with a non-negligible memory footprint overhead. For $d_{\max} \geq 528$ (cf. $d_{\max} = 512$ in Table 3), the test accuracies were very close to that of the full-matrix preconditioner, but with far less memory footprint overhead. For $d_{\max} \leq 352$, the test accuracies were getting worse as the block size decreased, though even $d_{\max} = 1$ (diagonal) still outperformed ZO-I (94.3 ± 1.5).

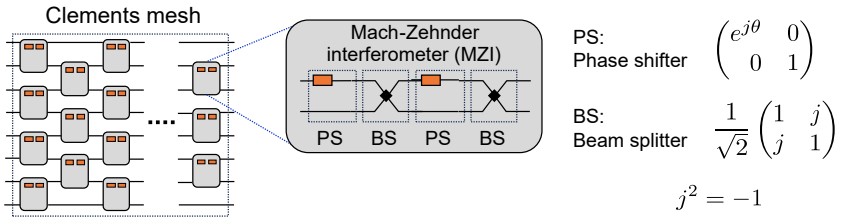


Figure 5: Clements mesh.

Table 5: Test accuracies (mean ± standard deviation over five runs) and computational cost (seconds/epoch and memory footprint in GB) by ZO-F2 with varying $d_{\max}$ for RNN(Copying).

$d_{\max}$	1	4	13	39	117	352	528	1056	1584	2112	2640	3168
Test accuracy	96.5	97.0	97.1	97.3	97.7	97.8	98.0	98.1	98.0	98.0	98.0	98.1
	±2.3	±2.3	±2.4	±2.5	±2.6	±2.8	±2.8	±2.8	±2.8	±2.8	±2.8	±2.9
Seconds/epoch	121.6	121.6	121.3	122.5	122.5	125.4	124.0	122.1	124.2	126.4	123.8	129.8
Memory footprint	0.36	0.36	0.36	0.36	0.37	0.37	0.37	0.39	0.40	0.40	0.40	0.56

G.4 ResNet (CIFAR-10)

The total number of parameters was $d = 270{,}410$, and we specified $d_{\max} = 498$, resulting in $b = 543$ matrix blocks. The mini-batch size was 100. The number of epochs for the results in Table 3 was 300, making the number of steps 150,000. We adopted Adam [60] for its robustness to hyperparameter choices and its stable convergence behavior. The hyperparameter settings were $\mu = 10^{-3}$, $\eta = 10^{-3}$, $\alpha = 10^{-5}$, $\gamma = 0.01$, $\tau = 0.5$, and $T_{\mathrm{ud}} = 500$.

H Limitations and future work

First, as summarized in Table 1, the proposed ZO-F2 achieves a low-variance Fisher preconditioner in a full-matrix form. However, it requires $q \geq 2$ perturbations as [30, 31], thereby excluding the $q = 1$ setting.

Second, the largest experiment we conducted was for ResNet-20 (270K parameters), and we did not perform experiments for LLMs. In the research of ZO optimization, we understand that the following two types of problems are essentially different.

1. Training neural networks from scratch with random initialization, where the estimated gradient variance scales as $\mathcal{O}(d)$ with the number of parameters d.

2. Fine-tuning pre-trained neural networks like LLMs, whose loss landscape exhibits favorable conditions like local r-effective rank assumption in [19] and the estimated gradient variance does not directly depend on d.

The main focus of our paper is the first problem, where d largely affects the problem hardness. To our knowledge, the largest from-scratch experiment to date is ResNet-20 (270K parameters), per [14].

Third, Theorem 3.3 shows that the clipping threshold τ controls the whitening residual; however, τ is tuned manually in Section 4, and developing an adaptive tuning rule remains future work.

Fourth, we discuss the limitations of our convergence analysis for ZO-F2 (Theorem 3.4). Our proof strategy follows [34], and thus inherits several technical limitations from this line of work. While our convergence analysis avoids the bounded-maximum-eigenvalue assumption imposed in [34], it still requires Assumption F.3, which assumes a strictly positive lower bound on the minimum eigenvalue of the preconditioner matrix. Furthermore, the practical techniques in Subsection 3.4 are not reflected in Theorem 3.4.

I Impact statement

We have proposed a zeroth-order (ZO) optimization method that incorporates second-order approaches. While the proposed method, ZO-F2, is broadly applicable to ZO optimization problems, it is particularly relevant to training neural networks implemented directly in hardware, such as optical and quantum neural network systems, where backpropagation is infeasible, as noted in the Introduction. More broadly, advances in ZO optimization with second-order approaches may serve as a key technique enabling hardware-implemented neural network training.

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: In the abstract and introduction, our main contributions are accurately summarized.

   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: In Section H, limitations and future work are summarized.

   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: Our main theoretical results are Theorems 3.1, 3.2, 3.3 and 3.4 in Section 3. For Theorem 3.1, we have Assumption 3.1 and a complete proof in Appendix C.2. For Theorem 3.2, we have Assumption 3.1 and a complete proof in Appendix D.2. For Theorem 3.3, we have a complete proof in Appendix E. For Theorem 3.4, we have Assumptions F.1, F.2 and F.3 and a complete proof in Appendix F.

   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: In Section 4 and Appendix G, detailed information regarding our experiments is summarized.

   Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: The source code is accessible at `https://github.com/nttcslab/ZO-F2`.

   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: In Appendix G, the details of experimental settings (e.g., hyperparameters) are summarized.

   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: Table 3 reports the standard deviation of five runs, and also the statistical significance ($p = 0.05$, one-sided Mann-Whitney U test). Figures 2a, 2b, and 2d show error bars.

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: In Appendix G, we report the computing resource information.

   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: We have reviewed the NeurIPS Code of Ethics and confirm that the research in this paper conforms to it, including with respect to responsible use of data, privacy, potential societal impacts, and reproducibility.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In Appendix I, potential societal impacts are mentioned.

    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: We have not released new models in this paper.

    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: We used open-access datasets (e.g., EMNIST), and they are appropriately cited.

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: [N/A]

Justification: This paper does not release new assets.

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: Crowdsourcing experiments and research with human subjects are not included in this paper.

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: This paper does not involve crowdsourcing nor research with human subjects.

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: LLM is used only for writing and editing.

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.