Matrix-Free Natural Gradient via Stochastic Truncated Metrics

Ke Sun*
CSIRO
sunk@ieee.org

Abstract

Natural gradient descent has been a theoretical pillar of second-order optimization for neural networks. Existing implementations either rely on structural approximations of the metric or incur substantial computational and memory overhead. Direct application of the natural gradient remains difficult at deep-network scale. We build an optimizer based on an unbiased stochastic estimator of the Fisher information matrix. It employs a truncated exponential moving average metric with damping, which we call the Stochastic Truncated Metric (STM), and is otherwise closely aligned with Amari's original natural gradient. This gives a parameter update with $O(Kd)$ time and memory per step for fixed small K, and is matrix-free in the sense that no dense $d \times d$ or layer-wise metric matrix is constructed or inverted. It is applicable across network architectures without layer-wise assumptions. We bound the error induced by the finite-memory truncation. On two benchmarks, STM achieves mean test accuracy within 0.7 percentage points of K-FAC, with comparable or lower per-epoch training cost.

1 Introduction

Natural gradient [Amari, 1998] is a central concept in the geometric theory of neural networks. It considers the intrinsic difference between probability distributions modeled by the network using the Fisher information matrix (FIM) as the local metric. Natural gradient descent (NGD) performs parameter updates that trade off steepest loss descent against moving along local geodesics of the neuromanifold:

$$\theta_{t+1} = \theta_t - \eta\, \mathcal{F}(\theta_t)^{-1} \nabla_\theta \mathcal{L}(\theta_t), \tag{1}$$

where θ_t represents the neural network weights and biases at time t, $\mathcal{L}(\theta)$ is the loss, and $\mathcal{F}(\theta)$ is the FIM. It has influenced the development of second-order optimization, online learning, variational inference, reinforcement learning, and modern curvature-aware training methods.

NGD is computationally prohibitive when applied directly to deep neural networks (DNNs). For a model with d parameters (often in the billions for modern DNNs), the FIM is a $d \times d$ positive semi-definite (PSD) matrix, which requires $O(d^2)$ memory to store and $O(d^3)$ time to invert. Practical variants rely on diagonal, Kronecker-factored, blockwise, low-rank, or iterative approximations of $\mathcal{F}(\theta)$. These approximations are effective, despite the fact that some of them impose architecture-dependent bias on the Fisher geometry.

We re-align the natural-gradient direction with its original formulation, with no network-structure assumptions and under a strict matrix-free budget. We propose the *Stochastic Truncated Metric* (STM), which adapts Eq. (1) with two explicit approximations: ① $\mathcal{F}(\theta)$ is replaced by an unbiased stochastic rank-one estimator of the batch FIM, accumulated over time as a damped exponential moving average; ② the infinite-memory moving average is truncated, retaining only the past K

*This work was conducted while the author was affiliated with CSIRO.

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

learning steps. Apart from those approximations, we apply the exact inverse of this metric through the Woodbury identity at $O(Kd + K^3)$ parameter-update time and $O(Kd + K^2)$ auxiliary memory, reducing to $O(Kd)$ for fixed small K, plus one additional backward pass.

Our main contributions include

1. *A matrix-free and architecture-agnostic natural-gradient optimizer*, with parameter-update time and auxiliary space linear in the number of parameters d for fixed buffer size K;
2. *An error analysis.* We prove that finite-memory truncation error has an upper bound that decays geometrically in K, and, for fixed parameters and batches, bound deviations of the stochastic metric from its mean;
3. *Experiments on CIFAR-10/ResNet and Shakespeare/nanoGPT,* showing lower per-epoch cost than K-FAC on CIFAR-10 and similar cost on Shakespeare, with modestly lower test accuracy.

STM shifts the main cost of NGD from inverting a large matrix to storing K parameter-shaped gradient vectors — a more tractable constraint.

The rest of the paper unfolds as follows. Section 2 reviews curvature-aware optimization. Section 3 defines STM and derives its inverse. Sections 4 and 5 present the optimizer and the empirical results. Section 6 analyzes the error in computing the associated natural gradient. Section 7 concludes.

2 Related work

A series of prior efforts addresses the cost of curvature-aware optimization. Becker and Le Cun [1988] avoids Hessian matrix storage by utilizing a diagonal approximation. Schraudolph [2002] computes the curvature matrix-vector product directly. Martens [2010] uses iterative linear solvers for Hessian-free Newton steps. K-FAC [Martens, 2020, Martens and Grosse, 2015] approximates layer-wise FIM blocks as Kronecker products. Singh and Alistarh [2020] maintains a running estimate of the empirical Fisher and uses the Woodbury identity to update its inverse incrementally. M-FAC [Frantar et al., 2021] computes matrix-free inverse-matrix–vector products from a sliding window of empirical gradients, whereas STM uses Hutchinson's unbiased estimate of the FIM. Some of these methods make explicit structural assumptions: layer-wise Kronecker products or block-diagonal approximations.

Theoretical and practical efforts are made to reduce $\mathcal{F}(\theta)$ into tractable form. Pennington and Worah [2018] and Karakida et al. [2019] revealed that the spectrum of $\mathcal{F}(\theta)$ is highly skewed — most eigen-directions carry negligible curvature. Hayase and Karakida [2021] studied the conditional FIM spectrum in deep networks with dynamical isometry. The near-singularity of $\mathcal{F}(\theta)$ and its effect on learning have been studied [Amari et al., 2018, Sun and Nielsen, 2025]. Thomas [2014] generalizes NGD to handle degenerate PSD metrics. Lin et al. [2021] computes $\mathcal{F}(\theta)$ in a local coordinate system. Korbit et al. [2025] exploits the low-rank structure of the Gauss-Newton matrix and the Duncan-Guttman identity. Li et al. [2025] approximates the Fisher diagonal and discards all off-diagonal information. Ollivier [2015] and Sun and Nielsen [2017] use sub-components of the Fisher information of the network. First-order memory-efficient methods [Lee and Kim, 2026, Zhu et al., 2025, Zhang et al., 2025, Gonzalez Ortiz et al., 2026, Zmushko et al., 2025] reduce optimizer-state memory overhead without explicitly estimating the Fisher matrix.

Trace estimation via Hutchinson's method [Hutchinson, 1990] is widely used in DNNs. AdaCubic [Tsingalis et al., 2026] achieves $O(d)$ memory complexity for Hessian-diagonal estimation. Stochastic probes are used to estimate gradients in zeroth-order optimization [Song et al., 2026]. The accuracy of stochastic estimators of $\mathcal{F}(\theta)$ has been analyzed [Guo and Spall, 2019, Soen and Sun, 2024] and is related to Monte Carlo information geometry [Nielsen and Hadjeres, 2019]. An efficient unbiased stochastic estimator of $\mathcal{F}(\theta)$ with bounded variance was introduced in [Sun, 2026]. The Woodbury matrix identity is a standard tool for inverting large structured matrices. For example, it has been used in natural gradient since Park et al. [2000]. SPRING [Goldshlager et al., 2024] accelerates neural-network wavefunction optimization via momentum-coupled Kaczmarz projections.

STM is based on Hutchinson probes for unbiased Fisher estimation [Sun, 2026], extending the rank-one stochastic estimate to a finite-memory EMA optimizer with an efficient mechanism based on truncated Woodbury inversion for $O(Kd + K^3)$ matrix-free application, reducing to $O(Kd)$ for

fixed small K. It aligns with the original natural gradient [Amari, 1998], with a rigorous accuracy analysis and no assumption about network structure. At the cost of storing K parameter-shaped vectors, STM trades memory for computational efficiency.

3 Stochastic truncated metric

Consider a supervised DNN $p_\theta(y \mid x)$ with parameters θ, where x is an input and y ranges over C output classes. Let $d = \dim(\theta)$ denote the number of parameters. We work in the regime $K \ll d$: the buffer size K is a small constant while d grows with the model.

Following recent developments [Sun, 2026], one additional backward pass on the same forward computational graph yields an unbiased estimate of the FIM. Assume $p_\theta(y \mid x) > 0$ is always true, and that $p_\theta(y \mid x)$ is differentiable wrt θ with finite derivatives. Given a mini-batch $\mathcal{D}_t$ of size B, we sample independent Rademacher variables $\xi_{xy} \in \{\pm 1\}$ arranged as a $B \times C$ matrix, and form

$$h_t = \frac{2}{\sqrt{B}} \sum_{x \in \mathcal{D}_t} \sum_y \xi_{xy} \sqrt{p_\theta(y \mid x)}, \qquad g_t = \nabla_\theta h_t = \frac{1}{\sqrt{B}} \sum_{x \in \mathcal{D}_t} \sum_y \xi_{xy} \frac{1}{\sqrt{p_\theta(y \mid x)}} \nabla_\theta p_\theta(y \mid x).$$

Due to independence and unit variance of ξ_{xy},

$$\mathbb{E}_\xi[g_t g_t^\top] = \frac{1}{B} \sum_{x \in \mathcal{D}_t} \sum_y p_\theta(y \mid x) \nabla_\theta \log p_\theta(y \mid x) \nabla_\theta \log p_\theta(y \mid x)^\top \;:=\; \bar{\mathcal{F}}_B(\theta),$$

which is the batch-averaged FIM. We refer to g_t as the Hutchinson probe, or simply the probe. Conditioned on the parameters θ and batch $\mathcal{D}_t$, g_t has zero mean, and its outer product $g_t g_t^\top$ is an *unbiased* estimator of the batch FIM. For networks with BatchNorm, the probabilities and expectation are conditional on the input batch.

We define the stochastic metric recursively, without architectural assumptions:

$$H_t := \underbrace{\beta H_{t-1}}_{\text{Historical Curvature}} + \underbrace{\gamma(1-\beta) I}_{\text{Prior Floor}} + \underbrace{(1-\gamma)(1-\beta) g_t g_t^\top}_{\text{Stochastic Batch FIM}}, \tag{2}$$

where $\gamma \in (0, 1)$ is the identity floor, $\beta \in [0, 1)$ is the memory decay, and $H_0 = \gamma I$. The coefficients sum to 1, and thus H_t lies in the convex hull of H_0 and $\{\gamma I + (1-\gamma) g_s g_s^\top\}_{s=1}^t$ within the cone of positive definite matrices. Unrolling Eq. (2) gives the explicit sum $H_t = \gamma I + (1-\gamma)(1-\beta) \sum_{j=0}^{t-1} \beta^j g_{t-j} g_{t-j}^\top$. Each stored probe g_{t-j} was computed at its historical parameters and batch at time $t - j$. Thus truncation approximates the moving average of these historical metrics, rather than directly approximating the current FIM.

Definition 3.1 (STM)**.** *For $K \leq t$, the stochastic truncated metric is*

$$\begin{aligned} H_t^{(K)} &:= \gamma I + (1-\gamma)(1-\beta) \sum_{j=0}^{K-1} \beta^j g_{t-j} g_{t-j}^\top \\ &= \gamma I + \sum_{j=0}^{K-1} w_j g_{t-j} g_{t-j}^\top, \end{aligned} \tag{3}$$

where

$$w_j := (1-\gamma)(1-\beta)\beta^j, \qquad j = 0, \ldots, K-1, \tag{4}$$

is the weight associated with the probe of age j (age 0 means the newest probe). We let $\beta^0 = 1$ when $\beta = 0$.

The total weight of the K retained probes is $c_K := \sum_{j=0}^{K-1} w_j = (1-\gamma)(1-\beta^K)$. For example, with $\beta = 0.95$ and $K = 20$, their sum is $c_K \approx 0.64(1-\gamma)$. Due to the truncation, the retained weights are *not* renormalized and $\gamma + c_K = 1 - (1-\gamma)\beta^K \leq 1$. Unlike K-FAC or L-BFGS, whose damping terms can be adaptive, the constant γ serves as a persistent identity floor. Both H_t and $H_t^{(K)}$ are positive definite with minimum eigenvalue at least γ. The auxiliary memory is a fixed $O(Kd)$ constant independent of model structure.

The behavior of the STM is guided by the following properties:

1. *Riemannian.* $H_t^{(K)}$ is strictly positive definite, hence a valid stochastic Riemannian metric in parameter space. It is not globally invariant under arbitrary reparameterizations.
2. *Spectrum.* When $K \le d$, at most K eigenvalues exceed γ. This means STM modifies only recently observed K directions and discards stale curvature.
3. *Stationarity.* Under a stationary probe distribution with $\mathbb{E}[g_t g_t^\top] = \mathcal{F}$, taking expectations yields $\mathbb{E}[H_t^{(K)}] = \gamma I + (1-\gamma)(1-\beta^K)\mathcal{F}$. The recursion forgets past probes at a geometric rate governed by β.

Let $v_t = \nabla_\theta \mathcal{L}(\theta_t)$ be the training gradient. For $\eta > 0$, the local dynamic $-\eta[H_t^{(K)}]^{-1}v_t$ minimizes $v_t^\top \Delta + \Delta^\top H_t^{(K)} \Delta/(2\eta)$. For fixed K and with the Gram matrix maintained incrementally, the natural gradient direction $\tilde{v}_t = [H_t^{(K)}]^{-1}v_t$ can be computed in $O(Kd)$ time using the Woodbury identity.

Proposition 3.2 (Inverse-STM)**.** *We have*

$$\tilde{v}_t = \frac{1}{\gamma}\left(v_t - G_t(\gamma D^{-1} + G_t^\top G_t)^{-1} G_t^\top v_t\right), \tag{5}$$

where $G_t = [g_{t-K+1}, \dots, g_t] \in \mathbb{R}^{d\times K}$, $D = \mathrm{diag}(w_{K-1}, \dots, w_0)$ with weights from Eq. (4)*, and $\mathrm{diag}(\cdots)$ constructs a diagonal matrix with given diagonal entries. The formula assumes $0 < \beta < 1$ so that D is invertible; the edge case $\beta = 0$ is obtained directly from the one-probe metric.*

The buffer $\mathcal{B}$ is a ring of fixed length K, storing the K most recent probes. The $K \times K$ kernel $M = \gamma D^{-1} + G_t^\top G_t$ is maintained incrementally in $O(Kd)$ time and $O(K^2)$ memory: whenever a new probe g_t overwrites slot s, one row and column of the Gram matrix $G_t^\top G_t$ are refreshed via K inner products $\{g_t^\top g_{t-j}\}_{j=0}^{K-1}$ between the newly added probe g_t and all old probes.

Proposition 3.3. *Let $H_t^{(K)}$ be defined as in Eq.* (3)*, with $0 < \gamma < 1$ and $\beta \in [0, 1)$. Let G_t, D be as in Proposition 3.2. Then the ℓ_2 condition number denoted as κ satisfies*

$$\kappa\big(H_t^{(K)}\big) \;\le\; 1 + \frac{1}{\gamma}\,\lambda_{\max}\big(D^{1/2} G_t^\top G_t D^{1/2}\big), \tag{6}$$

with equality whenever $K < d$. The RHS depends on the probes through the $K \times K$ kernel Gram matrix $G_t^\top G_t$.

If $G_t D G_t^\top u = \lambda u$, then $[H_t^{(K)}]^{-1}u = u/(\gamma + \lambda)$. Hence larger curvature results in smaller updates.

When G_t has full column rank, let $P_t = G_t(G_t^\top G_t)^{-1}G_t^\top$ denote the orthogonal projector onto $\mathrm{span}\{g_{t-K+1}, \dots, g_t\}$. The Woodbury correction lives entirely in $\mathrm{col}(G_t)$. Its action on the orthogonal complement vanishes. Decomposing $v_t = P_t v_t + (I - P_t)v_t$, we have

$$[H_t^{(K)}]^{-1}(I - P_t)v_t = \frac{1}{\gamma}(I - P_t)v_t - \frac{1}{\gamma}G_t(\gamma D^{-1} + G_t^\top G_t)^{-1}G_t^\top (I - P_t)v_t = \frac{1}{\gamma}(I - P_t)v_t.$$

STM behaves as a *hybrid optimizer*: ① On the K-dimensional probe subspace: Fisher preconditioning via the $K \times K$ Woodbury kernel; ② On the $(d-K)$-dimensional orthogonal complement: Euclidean gradient descent with multiplier $1/\gamma$. The K Hutchinson probes act as a *stochastic spotlight*, illuminating K directions with second-order geometry while leaving the remaining $d - K$ directions in the dark. Hence γ is not merely a numerical damping constant but determines the learning step size on the blind directions.

Remark (Variance reduction via temporal averaging)**.** *Natural gradient can use large batch sizes [Ollivier, 2015] to reduce curvature variance. STM supplements spatial (per-step) averaging with temporal (across-step) averaging: the EMA accumulates curvature information from many small batches over time. For independent stationary probes with $\mathbb{E}[gg^\top] = \mathcal{F}$ and $\sigma^2 := \mathbb{E}\|gg^\top - \mathcal{F}\|_F^2 < \infty$, and with a fixed batch size,*

$$\mathbb{E}\left\|\frac{H_t^{(K)} - \gamma I}{c_K} - \mathcal{F}\right\|_F^2 = \sigma^2 \frac{\sum_{j=0}^{K-1} w_j^2}{c_K^2} = \frac{\sigma^2}{N_{\mathrm{eff}}(K)},$$

$$N_{\mathrm{eff}}(K) := \frac{c_K^2}{\sum_{j=0}^{K-1} w_j^2} = \frac{1+\beta}{1-\beta}\frac{1-\beta^K}{1+\beta^K} \le \min\left\{K, \frac{1+\beta}{1-\beta}\right\}.$$

The normalized FIM estimate therefore has $1/N_{\text{eff}}(K)$ times the variance of a single probe outer product, without increasing the per-step batch size ($N_{\text{eff}}(K) = 1$ when $\beta = 0$). As $K \to \infty$, $N_{\text{eff}}(K)$ approaches $(1+\beta)/(1-\beta)$.

4 The STM optimizer

We now translate the STM construction into a practical training algorithm. Algorithm 1 maintains a Hutchinson-probe buffer $\mathcal{B}$ of shape $K \times d$ whose rows are the stored probes (the row-oriented counterpart of G_t in Proposition 3.2), and applies the inverse via Proposition 3.2, avoiding any $d \times d$ FIM computation. Note that Algorithm 1 uses a zero-based update index. After inserting the probe at iteration t into the buffer, the algorithm applies the metric $H_{t+1}^{(k)}$, where $k = \min(K, t+1)$, to the gradient v_t at that same iterate.

Algorithm 1: Stochastic Truncated Metric (STM) Optimizer

Inputs : Initial parameters θ_0; buffer capacity $K \in \mathbb{Z}^+$ (e.g. 20); damping floor $\gamma \in (0,1)$ (e.g. 0.01); EMA decay $\beta \in (0,1)$ (e.g. 0.95); learning rate $\eta > 0$; momentum $\mu \in [0,1)$ (e.g. 0.9); weight decay $\lambda \geq 0$ (e.g. 5×10^{-4}).

Initialize buffer $\mathcal{B} \leftarrow 0_{K \times d}$ and Gram $Q \leftarrow 0_{K \times K}$; set buffer head $s \leftarrow 1$, active $k \leftarrow 0$, momentum $m \leftarrow \varnothing$

for $t = 0, 1, \ldots$ *until the training budget is exhausted* **do**

- $v_t \leftarrow \nabla_\theta \mathcal{L}(\theta_t)$ `// vanilla gradient`
- sample Rademacher matrix $\xi_t \in \{\pm 1\}^{B \times C}$ and form h_t as in Section 3
- $g_t \leftarrow \nabla_\theta h_t$ `// Hutchinson probe`
- $\mathcal{B}_s \leftarrow g_t$ `// append probe to the buffer`
- $d_{\text{new}} \leftarrow \mathcal{B}\, g_t$ `// dot products for Gram update`
- $Q_{s,:} \leftarrow d_{\text{new}}^\top$, $Q_{:,s} \leftarrow d_{\text{new}}$ `// rolling Gram`
- $s \leftarrow (s \bmod K) + 1$, $k \leftarrow \min(k+1, K)$
- $\omega_i \leftarrow w_{\text{age}(i)}$ from Eq. (4) for each active row i `// age 0 = newest; same row order as` $\mathcal{B}$
- $B_t \leftarrow$ active rows of $\mathcal{B}$ in storage order `// row-permuted` $G_t^\top$
- $Q_t \leftarrow$ active $k \times k$ submatrix of Q in the same row order `// active Gram`
- $q \leftarrow B_t v_t$
- $M \leftarrow \gamma\, \text{diag}(\omega^{-1}) + Q_t$ `// Woodbury kernel`
- $a \leftarrow \text{CholeskySolve}(M, q)$
- $\tilde{v}_t \leftarrow \frac{1}{\gamma}\left(v_t - B_t^\top a\right)$ `// natural-gradient direction`
- $m \leftarrow \tilde{v}_t$ **if** $m = \varnothing$ **else** $\mu m + (1-\mu)\tilde{v}_t$ `// momentum`
- $u_t \leftarrow m$ **if** $\mu > 0$ **else** $\tilde{v}_t$
- $\theta_{t+1} \leftarrow (1 - \eta\lambda)\theta_t - \eta\, u_t$ `// update with weight decay`

Each STM iteration performs one standard backward pass, one probe backward pass, $O(Kd)$ vector work for the rolling Gram update and matrix-free preconditioning, and an $O(K^3)$ Cholesky solve for the $K \times K$ Woodbury kernel. Thus the optimizer-update time complexity is $O(Kd + K^3)$, which reduces to $O(Kd)$ for small K. The auxiliary space complexity is $O(Kd + K^2)$: $O(Kd)$ for the probe buffer and $O(K^2)$ for the rolling Gram matrix.

The complexity comparison is detailed in Table 1. L-BFGS [Liu and Nocedal, 1989] has similar $O(Kd)$ vector complexity, but its secant pairs approximate Euclidean Hessian geometry rather than the FIM, while M-FAC [Frantar et al., 2021] uses a sliding window of empirical gradients with $O(Kd + K^3)$ update time. Shampoo [Gupta et al., 2018], SOAP [Vyas et al., 2025], and K-FAC [Martens, 2020] exploit layer-wise structured preconditioning. These methods require storing and decomposing preconditioner factors. The associated costs scale with the dimensions of each layer (e.g., $O(p^2) = O(d)$ space and $O(p^3) = O(d^{1.5})$ factor-decomposition time for a single $p \times p$ layer). STM is matrix-free in the stricter sense: it stores only parameter-shaped probe vectors as well as a small $K \times K$ Gram matrix.

Table 1: Optimizer-update complexity comparison for total parameter count d and buffer size K (excluding forward/backward passes; Shampoo, SOAP, and K-FAC assume squared layers with factor decompositions at every step).

Method	**Time/Iter**	**Space**	**Matrix-Free?**
SGD / Adam	$O(d)$	$O(d)$	Yes
L-BFGS	$O(Kd)$	$O(Kd)$	Yes
Shampoo	$O(d^{1.5})$	$O(d)$	No
SOAP	$O(d^{1.5})$	$O(d)$	No
K-FAC	$O(d^{1.5})$	$O(d)$	Partial
Explicit FIM inversion	$O(d^3)$	$O(d^2)$	No
STM (Ours)	$\mathbf{O(Kd + K^3)}$	$\mathbf{O(Kd + K^2)}$	**Yes**

In distributed settings, synchronizing the vanilla gradient v_t and probe g_t across workers allows each of them to maintain an identical, local probe buffer $\mathcal{B}$ and Gram matrix Q. Thus, the per-step communication complexity is $O(d)$, identical to SGD and independent of K. In contrast, K-FAC must additionally synchronize layer-wise Kronecker factors.

With fixed small K, the $K \times K$ Cholesky solve is dimension-independent; the dominant overhead is the extra Hutchinson-probe backward pass plus streaming the K stored probe vectors for the $O(Kd)$ Gram update. These operations are sequential over the buffer but highly parallelizable across parameter shards, leading to scalable implementations via GPU kernel fusion. Although STM is more expensive per iteration than first-order methods, Fisher preconditioning may reduce the number of iterations required to reach a target loss. In the following section, we evaluate accuracy and training cost on two benchmarks.

5 Experiments

We test the proposed STM optimizer on CIFAR-10 image classification with a small ResNet ($d \approx 272$k) [He et al., 2016, Krizhevsky, 2009] and character-level Shakespeare language modeling with nanoGPT ($d \approx 10.7$M) [Karpathy, 2023]. We compare against SGD with momentum [Sutskever et al., 2013], AdamW with decoupled weight decay [Loshchilov and Hutter, 2019], and K-FAC [Martens, 2020, Pauloski et al., 2020, 2021] as a commonly used second-order optimizer. We use the `kfac-pytorch` implementation of [Pauloski et al., 2020, 2021]. All reported results are produced on a single NVIDIA L4 GPU (24GB VRAM).

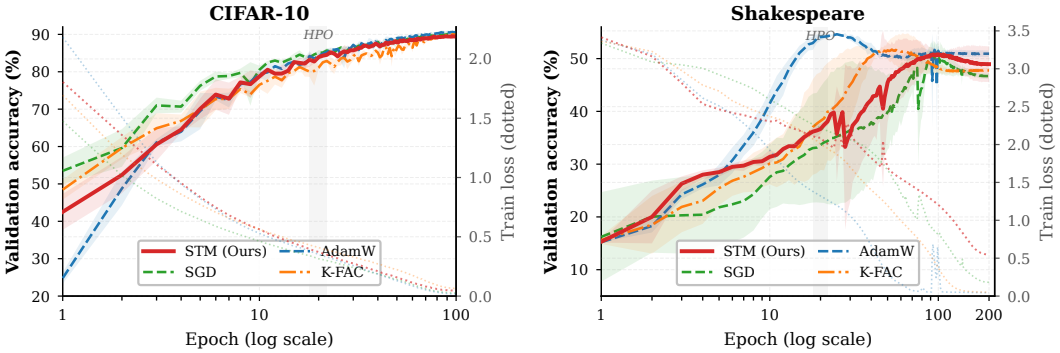
Each optimizer is selected by a validation sweep over learning rate, weight decay, and optimizer-specific parameters. For CIFAR-10, two candidate configurations per optimizer are promoted after 20 epochs to a 100-epoch full validation run using a cosine learning rate schedule. Final selection for CIFAR-10 uses mean best validation accuracy across HPO seeds, preferring lower sample standard deviation for near ties. Then, the selected configuration is re-run for 100 epochs on CIFAR-10 and 200 epochs on Shakespeare using three new initializations. Test accuracy is reported based on the validation-best checkpoint. See Table 2 for test accuracy, mean training time per epoch, and peak GPU memory. See Fig. 1 for training loss and validation accuracy.

On CIFAR-10, STM attains $88.70\% \pm 0.24\%$, slightly below K-FAC, while costing 9.80 s per epoch versus 29.39 s for K-FAC (about $3\times$ faster) and 7.35–7.43 s for the first-order methods. K-FAC refreshes its layer-wise inverse factors every 10 steps; STM solves its Woodbury system every step yet has lower observed per-epoch cost. On Shakespeare, STM attains $49.71\% \pm 0.73\%$, above SGD at $49.16\% \pm 0.39\%$ in mean test accuracy but below AdamW and K-FAC. STM and K-FAC have similar per-epoch costs (23.38 s and 23.03 s, respectively), while SGD and AdamW cost 12.50 s and 13.35 s. STM's mean peak allocated GPU memory is below K-FAC's on CIFAR-10 (321.6 vs 344.3 MiB), but above it on Shakespeare (3865.4 vs 3226.1 MiB), and above both first-order baselines on both tasks. The ablations (Fig. 2) show a buffer-size plateau (88.69–88.96% across $K \in \{5, 10, 20, 40\}$), moderate sensitivity to the identity floor with $\gamma = 0.1$ lowest at 87.74%, and small differences across memory-decay values (88.61–88.96% across $\beta \in \{0.80, 0.90, 0.95, 0.99\}$).

These preliminary empirical results are not intended as SOTA contests, but hint that STM is a matrix-free architecture-agnostic second-order optimizer, with lower mean test accuracy than AdamW

Table 2: Test accuracy, training time, and peak GPU memory. Accuracy is measured at each reloaded validation-best checkpoint and reported as mean ± sample standard deviation. Time is the mean per training epoch including data loading. Memory is the mean peak allocated GPU memory across three seeds, where each seed's peak is the maximum over training epochs. Sample standard deviations are below 0.01 MiB except Shakespeare K-FAC (0.40 MiB).

	SGD	AdamW	K-FAC	STM ($K = 20$)
CIFAR-10 / small ResNet	89.83 ± 0.36	90.07 ± 0.11	88.96 ± 0.09	88.70 ± 0.24
Time / epoch (s)	7.35	7.43	29.39	9.80
Peak GPU RAM (MiB)	267.3	268.4	344.3	321.6
Shakespeare / nanoGPT	49.16 ± 0.39	51.63 ± 0.12	50.32 ± 0.59	49.71 ± 0.73
Time / epoch (s)	12.50	13.35	23.03	23.38
Peak GPU RAM (MiB)	2866.2	2907.1	3226.1	3865.4



(a) CIFAR-10 / small ResNet (b) Shakespeare / nanoGPT

Figure 1: Training and validation curves on CIFAR-10 with a small ResNet (left) and character-level Shakespeare with nanoGPT (right). The final training runs use three new seeds for 100 and 200 epochs, respectively. Validation accuracy lines and shaded bands show the mean and sample standard deviation. Dotted curves show mean training loss. The gray band marks the CIFAR-10 20-epoch HPO promotion boundary for reference. STM uses fixed memory $K = 20$. For nanoGPT, K-FAC is configured to skip `wte`, `wpe`, `attn`, and `lm_head`.

and K-FAC on both tasks, and than SGD on CIFAR-10. STM provides Fisher preconditioning across convolutional networks and transformers without explicit $d \times d$ curvature matrices, while refreshing the implicit curvature computation at every learning step. Its value lies in providing Fisher preconditioning at $O(Kd + K^3)$ cost, and in offering theoretical insights into natural-gradient performance on modern DNNs.

6 Theoretical analysis

We provide a theoretical analysis of the STM update. This analysis concerns the quality of the preconditioned direction at each step and expected-metric conditioning in a simplified quadratic setting, but does not justify training stability or superiority to K-FAC, nor provide convergence guarantees for general nonconvex training objectives. Throughout this section, $\|\cdot\|$ and $\|\cdot\|_F$ denote the operator norm and the Frobenius norm for matrices, respectively, while $\|\cdot\|_2$ denotes the Euclidean norm for vectors.

Theorem 6.1 (Truncation Error). *$\{\theta_s\}_{s=1}^{t}$ denotes the learning trajectory up to step t. g_{t-j} denotes the Hutchinson probe gradient computed at θ_{t-j}. Let $1 \le K \le t$, $\beta \in [0, 1)$, $\gamma \in (0, 1)$, and $v_t \neq 0$ almost surely. At each learning step, the Rademacher variables are mutually independent and independent of the current parameters and mini-batch. Let $\bar{\mathcal{F}}_B(\theta_s)$ denote the batch-averaged FIM at step s, and assume $\sup_{1 \le s \le t} \mathbb{E}[\mathrm{tr}(\bar{\mathcal{F}}_B(\theta_s))] \le T_{\max} < \infty$. Let*

$$\mathcal{T}_{t,K}^{(\beta)} := (1 - \beta) \sum_{j=K}^{t-1} \beta^{j-K} \, \mathbb{E}\big[\mathrm{tr}\big(\bar{\mathcal{F}}_B(\theta_{t-j})\big)\big] \, . \tag{7}$$

Then, the input-normalized truncation error satisfies

$$\mathbb{E}\left[\frac{\|H_t^{-1}v_t - [H_t^{(K)}]^{-1}v_t\|_2}{\|v_t\|_2}\right] \;\le\; \frac{(1-\gamma)\,\beta^K\,\mathcal{T}_{t,K}^{(\beta)}}{\gamma^2}. \tag{8}$$

When $\beta = 0$ or $K = t$, the truncation error is zero.

Corollary 6.2 (Geometric Decay of Truncation Error)**.** *Under the assumptions of Theorem 6.1, with fixed $\beta \in (0,1)$, γ and $T_{\max} > 0$, $H_t^{(K)}$ and H_t are constructed from the same probe history $g_1, \ldots, g_t$. Let $\varepsilon_{\mathrm{trunc}}(K)$ denote the expectation on the LHS of Eq.* (8)*. Since $\mathcal{T}_{t,K}^{(\beta)} \le T_{\max}$, the error bound decays geometrically with K:*

$$\varepsilon_{\mathrm{trunc}}(K) \le \frac{(1-\gamma)\beta^K\mathcal{T}_{t,K}^{(\beta)}}{\gamma^2} \le \frac{(1-\gamma)T_{\max}}{\gamma^2}\,\beta^K, \qquad \forall\, 1 \le K \le t, \tag{9}$$

and $\varepsilon_{\mathrm{trunc}}(t) = 0$. For any given tolerance $\varepsilon > 0$ and step $t \ge 1$, the buffer size

$$K = \min\left\{t,\ \max\left\{1,\ \left\lceil \frac{\log\big((1-\gamma)T_{\max}/(\gamma^2\varepsilon)\big)}{\log(1/\beta)} \right\rceil\right\}\right\}$$

ensures $\varepsilon_{\mathrm{trunc}}(K) \le \varepsilon$. If the trace bound in Theorem 6.1 holds uniformly in t, the RHS of Eq. (9) *is uniformly $O(\beta^K)$, and the sufficient buffer size is $O(\log(1/\varepsilon))$ as $\varepsilon \to 0^+$.*

Proposition 6.3 (Residual Error Bound)**.** *For a fixed STM $H_t^{(K)}$ and input gradient v_t, let $\tilde{v}_t = [H_t^{(K)}]^{-1}v_t$ be the exact STM-preconditioned direction. For any $\hat{v}_t \in \mathbb{R}^d$ aiming to approximate $\tilde{v}_t$, define the residual*

$$r_t = v_t - H_t^{(K)}\hat{v}_t.$$

Then

$$\|\hat{v}_t - \tilde{v}_t\|_2 \le \frac{\|r_t\|_2}{\gamma}. \tag{10}$$

With G_t and D defined in Proposition 3.2, the residual has the matrix-free expression

$$r_t = v_t - \gamma\hat{v}_t - G_t\big[D(G_t^\top \hat{v}_t)\big],$$

which requires $O(Kd)$ arithmetic operations because D is diagonal. Given the STM and the input gradient v_t, the bound assesses the accuracy of a computed direction $\hat{v}_t$ through its residual. When the residual is evaluated numerically, the evaluation error should also be considered. Errors in computing the probes g_{t-j} and input gradient v_t require a more careful analysis of their effect on the preconditioned direction $[H_t^{(K)}]^{-1}v_t$.

Remark. *The factor $1/\gamma$ is sharp when $K < d$, as $\ker(G_t^\top)$ contains a nonzero vector r_t. Choosing $\hat{v}_t = \tilde{v}_t - r_t/\gamma$ gives residual r_t and attains equality in Eq.* (10)*.*

Corollary 6.4. *Let $1 \le K \le t$, $\beta \in [0,1)$, and $\gamma \in (0,1)$. Let H_t and $H_t^{(K)}$ be constructed from the same arbitrary probe history $g_1, \ldots, g_t$. For any $v_t \in \mathbb{R}^d \setminus \{0\}$ and $\hat{v}_t \in \mathbb{R}^d$, define*

$$\rho_t = \frac{\|v_t - H_t^{(K)}\hat{v}_t\|_2}{\|v_t\|_2}, \qquad \tau_{t,K} = \mathrm{tr}(H_t - H_t^{(K)}) = (1-\gamma)(1-\beta)\sum_{j=K}^{t-1}\beta^j\|g_{t-j}\|_2^2,$$

where $\tau_{t,K} = 0$ when $K = t$ or $\beta = 0$. Then

$$\frac{\|\hat{v}_t - H_t^{-1}v_t\|_2}{\|v_t\|_2} \le \frac{\rho_t}{\gamma} + \frac{\tau_{t,K}}{\gamma(\gamma+\tau_{t,K})}. \tag{11}$$

The bound is sharp for every feasible $(\gamma, \rho_t, \tau_{t,K})$, in the sense that there exist probes $g_1, \ldots, g_t$ and vectors $v_t, \hat{v}_t$ to realize $(\rho_t, \tau_{t,K})$ and make the inequality tight.

Corollary 6.4 decomposes the error relative to $H_t^{-1}v_t$ into solver residual (Proposition 6.3) and truncation error. Setting $\hat{v}_t = [H_t^{(K)}]^{-1}v_t$ ($\rho_t = 0$), relaxing $\tau_{t,K}/[\gamma(\gamma+\tau_{t,K})] \le \tau_{t,K}/\gamma^2$, and taking expectations under the assumptions of Theorem 6.1 recovers Eq. (8). The truncation term is bounded above by $1/\gamma$. If $\rho_t \ll \tau_{t,K}/(\gamma+\tau_{t,K})$, truncation dominates the bound on the RHS of Eq. (11); further reducing ρ_t is less helpful to lower the error bound.

Theorem 6.5 (Concentration of STM). *Fix $1 \le K \le t$, parameters θ_{t-j}, and batches $\mathcal{D}_{t-j}$ of size B, for $j = 0, \dots, K-1$. Assume all score vectors are finite. The probes g_{t-j} are constructed as in Section 3, using Rademacher variables mutually independent across all j, x, y. For $\gamma \in (0,1)$ and $\beta \in [0,1)$, write $F_j = \bar{\mathcal{F}}_B(\theta_{t-j})$. Define the expected STM*

$$\bar{H}_t^{(K)} := \mathbb{E}_\xi[H_t^{(K)}] = \gamma I + \sum_{j=0}^{K-1} w_j F_j$$

and

$$S_{t,K} := \sum_{j=0}^{K-1} w_j^2 \left[(\operatorname{tr} F_j)^2 + \operatorname{tr}(F_j^2)\right].$$

For every $\delta \in (0,1)$, with probability at least $1-\delta$, both the STM and its inverse satisfy

$$\|H_t^{(K)} - \bar{H}_t^{(K)}\| \le \sqrt{S_{t,K}/\delta}, \tag{12}$$

$$\|[H_t^{(K)}]^{-1} - [\bar{H}_t^{(K)}]^{-1}\| \le \frac{\sqrt{S_{t,K}/\delta}}{\gamma^2}. \tag{13}$$

Remark. *By Theorem 6.5, for every $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\|[H_t^{(K)}]^{-1} v - [\bar{H}_t^{(K)}]^{-1} v\|_2 \le \frac{\sqrt{S_{t,K}/\delta}}{\gamma^2} \|v\|_2, \tag{14}$$

for all $v \in \mathbb{R}^d$ on the same event. This bound also applies when v depends on the probes.

Theorem 6.5 considers a simplified setting: the parameters and batches are fixed *before* the independent probes are drawn. In practice, earlier probes influence later parameters. Extending Theorem 6.5 to the learning trajectory requires a more complex analysis left for future work. In Eq. (14), the reference direction $[\bar{H}_t^{(K)}]^{-1} v$ is obtained by applying the inverse of the expected STM to v; it generally differs from $\mathbb{E}_\xi\left[[H_t^{(K)}]^{-1} v\right]$, even for fixed v.

Proposition 6.6 (Quadratic Contraction). *Consider $F \succ 0$ has eigenvalues $0 < \lambda_{\min} = \lambda_1 \le \cdots \le \lambda_d = \lambda_{\max}$, with $\kappa(F) := \lambda_{\max}/\lambda_{\min}$. For $K \ge 1$, $\gamma \in (0,1)$, and $\beta \in [0,1)$, set $c_K = (1-\gamma)(1-\beta^K) > 0$ and $\bar{H}_K = \gamma I + c_K F$. Then, $\bar{H}_K^{-1} F$ has eigenvalues $\mu_i = \lambda_i/(\gamma + c_K \lambda_i)$, with*

$$\kappa_{\mathrm{pre}} := \frac{\mu_{\max}}{\mu_{\min}} = \kappa(F) \, \frac{\gamma + c_K \lambda_{\min}}{\gamma + c_K \lambda_{\max}} \le \kappa(F). \tag{15}$$

To minimize $L(e) = \frac{1}{2} e^\top F e$, let $e_{n+1} = (I - \eta \bar{H}_K^{-1} F) e_n$ be defined using the fixed step size $\eta = 2/(\mu_{\min} + \mu_{\max})$, which gives the fastest worst-case loss decrease. Then

$$L(e_{n+1}) \le q(\kappa_{\mathrm{pre}})^2 L(e_n), \quad q(\kappa) := (\kappa - 1)/(\kappa + 1). \tag{16}$$

Gradient descent with its optimal step size has worst-case loss factor $q(\kappa(F))^2 \ge q(\kappa_{\mathrm{pre}})^2$, strictly larger iff $\lambda_{\max} > \lambda_{\min}$. If $F \succeq 0$ is singular and nonzero, the result holds on its positive eigenspace.

Remark. *Let $Y \mid X \sim p_{\theta^\star}(\cdot \mid X)$. Let $\theta^\star$ minimize $\mathcal{L}(\theta) = \mathbb{E}[-\log p_\theta(Y \mid X)]$. Under regularity conditions, $\nabla^2 \mathcal{L}(\theta^\star) = \mathcal{F}(\theta^\star)$. Assume batches are sampled from the distribution of X. With probes computed at $\theta^\star$, $\mathbb{E}[g_s g_s^\top] = \mathcal{F}(\theta^\star)$ and $\bar{H}_K = \mathbb{E}[H_t^{(K)}] = \gamma I + c_K \mathcal{F}(\theta^\star)$ for $t \ge K$. Proposition 6.6 applies to $L(\theta) = \frac{1}{2}(\theta - \theta^\star)^\top \mathcal{F}(\theta^\star)(\theta - \theta^\star)$.*

7 Conclusion

We introduce the Stochastic Truncated Metric, a matrix-free natural-gradient optimizer without explicit $d \times d$ matrices. STM differs from NGD in two aspects: it replaces the exact FIM with a damped temporal average of rank-one estimators, each unbiased for its historical batch FIM, and it truncates the infinite memory to a recent buffer of size K. The inverse of the truncated metric is applied through a $K \times K$ Woodbury system, which is algebraically exact, yielding $O(Kd + K^3)$ optimizer-update time and $O(Kd + K^2)$ auxiliary memory per step, which is linear in the parameter count d for fixed small K. Empirically, STM trains on CIFAR-10/ResNet and Shakespeare/nanoGPT

with mean test accuracy 0.26 and 0.61 percentage points below K-FAC, respectively, while mean training time per epoch is about one-third of K-FAC's on CIFAR-10 and similar on Shakespeare. Our analysis bounds finite-memory truncation and fixed-parameter and fixed-batch stochastic-probe deviations.

For deep neural networks, the exact natural-gradient update can be ill-posed, as the Fisher information matrix of an over-parameterized DNN can be singular [Sun and Nielsen, 2025]. Practical methods often work with a regularized or compressed surrogate. The bottleneck moves from computing a large matrix inverse to choosing which Fisher directions can be kept under a finite memory budget. K-FAC addresses this via layer-wise Kronecker factors. STM uses a time-windowed buffer of stochastic Fisher probes, while remaining agnostic to network structure. One design principle for scalable natural gradient in this regime is finite-memory Fisher representation across space and time.

A few limitations remain. STM only preconditions the subspace spanned by the stored probes. Other directions are handled by the identity floor, so performance can degrade when useful curvature is not captured by the buffer. Although the per-step cost is linear in d for fixed K, storing K full parameter-shaped probes is memory intensive for billion-parameter models. BF16 probe-buffer compression would halve probe-buffer storage relative to FP32, while Kronecker, sharded, or random-sketched buffers may adapt the same Woodbury structure to large-language-model (LLM) pre-training and parameter-efficient fine-tuning. We evaluate stability and efficiency on two benchmarks but not SOTA accuracy. With three seeds on two learning tasks, these results establish feasibility rather than superiority. Broader empirical studies and improvements are left to future work.

Acknowledgments and Disclosure of Funding

The experiments used Google Cloud research credits. The author declares no other competing interests.

References

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

S. Amari, T. Ozeki, R. Karakida, Y. Yoshida, and M. Okada. Dynamics of learning in MLP: Natural gradient and singularity revisited. *Neural Computation*, 30(1):1–33, 2018.

S. Becker and Y. Le Cun. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 Connectionist Models Summer School*, pages 29–37, 1988.

E. Frantar, E. Kurtic, and D. Alistarh. M-FAC: Efficient matrix-free approximations of second-order information. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

G. Goldshlager, N. Abrahamsen, and L. Lin. A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions. *Journal of Computational Physics*, 516:113351, 2024. URL `https://www.sciencedirect.com/science/article/pii/S0021999124005990`.

J. J. Gonzalez Ortiz, A. Gupta, C. Rinard, and D. Blalock. FlashOptim: Optimizers for memory-efficient training. *arXiv preprint arXiv:2602.23349*, 2026.

S. Guo and J. C. Spall. Relative accuracy of two methods for approximating observed Fisher information. In *Data-Driven Modeling, Filtering and Control: Methods and applications*, pages 189–211. IET Press, London, 2019.

V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/gupta18a.html`.

T. Hayase and R. Karakida. The spectrum of Fisher information of deep networks achieving dynamical isometry. In *International Conference on Artificial Intelligence and Statistics*, pages 334–342. PMLR, 2021.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '16, pages 770–778. IEEE, 2016.

M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990.

R. Karakida, S. Akaho, and S. Amari. Universal statistics of Fisher information in deep neural networks: Mean field approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1032–1041. PMLR, 2019.

A. Karpathy. nanoGPT. `https://github.com/karpathy/nanoGPT`, 2023.

M. Korbit, A. D. Adeoye, A. Bemporad, and M. Zanon. Exact Gauss-Newton optimization for training deep neural networks. *Neurocomputing*, 658:131738, 2025. URL `https://www.sciencedirect.com/science/article/pii/S0925231225024105`.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

W. Lee and H.-T. Kim. SAGE: Sign-adaptive gradient for memory-efficient LLM optimization, 2026. URL `https://arxiv.org/abs/2604.07663`.

Y. X. Li, F. Dangel, D. Tam, and C. Raffel. Fishers for free? Approximating the Fisher information matrix by recycling the squared gradient accumulator. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 34252–34270. PMLR, 13–19 Jul 2025.

W. Lin, F. Nielsen, M. E. Khan, and M. Schmidt. Tractable structured natural-gradient descent using local parameterizations. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6680–6691. PMLR, 18–24 Jul 2021.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

J. Martens. Deep learning via Hessian-free optimization. In *International Conference on Machine Learning (ICML)*, volume 27, pages 735–742, 2010.

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

J. Martens and R. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, pages 2408–2417. PMLR, 2015.

F. Nielsen and G. Hadjeres. Monte Carlo information-geometric structures. In F. Nielsen, editor, *Geometric Structures of Information*, pages 69–103. Springer International Publishing, Cham, 2019.

Y. Ollivier. Riemannian metrics for neural networks I: feedforward networks. *Information and Inference: A Journal of the IMA*, 4(2):108–153, 2015.

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

J. G. Pauloski, Z. Zhang, L. Huang, W. Xu, and I. T. Foster. Convolutional Neural Network Training with Distributed K-FAC. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986. doi: 10.5555/3433701.3433826.

J. G. Pauloski, Q. Huang, L. Huang, S. Venkataraman, K. Chard, I. Foster, and Z. Zhang. KAISA: An Adaptive Second-Order Optimizer Framework for Deep Neural Networks. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384421. doi: 10.1145/3458817.3476152. URL https://doi.org/10.1145/3458817.3476152.

J. Pennington and P. Worah. The spectrum of the Fisher information matrix of a single-hidden-layer neural network. In *Advances in Neural Information Processing Systems*, pages 5415–5424, 2018.

N. N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14(7):1723–1738, 2002.

S. P. Singh and D. Alistarh. WoodFisher: Efficient second-order approximation for neural network compression. In *Advances in Neural Information Processing Systems*, volume 33, pages 18098–18109. Curran Associates, Inc., 2020.

A. Soen and K. Sun. Trade-Offs of diagonal Fisher information matrix estimators. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 5870–5912. Curran Associates, Inc., 2024.

M. Song, L. Zhang, B. Li, N. He, M. Muehlebach, and S. Oh. Zeroth-order optimization at the edge of stability. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026. URL https://openreview.net/forum?id=InkMd89Pju.

K. Sun. Deterministic bounds and random estimates of metric tensors on neuromanifolds. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=Ssevs8KCsU.

K. Sun and F. Nielsen. Relative Fisher information and natural gradient for learning large modular models. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 3289–3298. PMLR, 2017.

K. Sun and F. Nielsen. A geometric modeling of Occam's razor in deep learning. *Information Geometry*, 8:233–273, 2025. Special Issue: Half a Century of Information Geometry, Part 2. Formerly titled "Lightlike neuromanifolds, Occam's razor and deep learning".

I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147. PMLR, 2013.

P. Thomas. GeNGA: A generalization of natural gradient ascent with positive and negative convergence results. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1575–1583. PMLR, 2014.

I. Tsingalis, C. Kotropoulos, and C. Briat. AdaCubic: An adaptive cubic regularization optimizer for deep learning. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=pZBQ7J371k. J2C Certification.

N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

Y. Zhang, C. Chen, Z. Li, T. Ding, C. Wu, D. P. Kingma, Y. Ye, Z.-Q. Luo, and R. Sun. Adam-mini: Use fewer learning rates to gain more. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iBExhaU3Lc.

H. Zhu, Z. Zhang, W. Cong, X. Liu, S. Park, V. Chandra, B. Long, D. Z. Pan, Z. Wang, and J. Lee. APOLLO: SGD-like memory, AdamW-level performance. In *Eighth Conference on Machine Learning and Systems*, 2025. URL https://openreview.net/forum?id=mJrPkdcZDj.

P. Zmushko, A. Beznosikov, M. Takáč, and S. Horváth. FRUGAL: Memory-efficient optimization by reducing state overhead for scalable training. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 80708–80739. PMLR, 13–19 Jul 2025.

A Ablation study

All optimizers use the same data split for each seed. Main runs use batch size $B = 128$ and cosine decay. Final runs use seeds 51–53; ablations use 54–56. Test accuracy is evaluated at each seed's reloaded validation-best checkpoint. We exclude runs with non-finite values and report only complete three-seed results. Timing includes solver checks but excludes validation and checkpoint saving.

CIFAR-10 uses a deterministic 90/10 training/validation split and the official test set. Its reported standard deviations include variation across splits. ResNet starts with a 3×3, stride-1 convolution and no initial max-pooling. It has three stages of basic blocks with widths 16/32/64 and 272,474 parameters. CIFAR-10 tuning uses seeds 48–50. The near-tie threshold in Section 5 is 0.5 percentage points.

Shakespeare uses a fixed contiguous 90/5/5 training/validation/test split. NanoGPT uses 6 layers, 6 heads, embedding dimension 384, sequence length $T = 128$, and vocabulary size 65. Each epoch covers all non-overlapping training windows once, with targets shifted by one character. Language-model loss and Fisher probes average over BT target tokens; probes use the factor $1/\sqrt{BT}$. For K-FAC, the shared embedding/output weights use SGD.

SGD uses Nesterov momentum. K-FAC uses observed targets to compute curvature factors. It removes loss-averaging factors from this calculation but keeps the training gradient averaged.

Tables 3 and 4 list the tuning values. The exact CIFAR-10 candidates below specify the non-Cartesian combinations in Table 3; all other settings are as listed there:

- SGD (η, λ): $(0.01, 0.0005)$, $(0.02, 0.0005)$, $(0.03, 0.0005)$, $(0.02, 0.0001)$, $(0.03, 0.0001)$.
- AdamW (η, λ): $(0.003, 0.0005)$, $(0.005, 0.0005)$, $(0.01, 0.0005)$, $(0.005, 0.0001)$, $(0.01, 0.0001)$.
- K-FAC $(\eta, \text{damping}, \lambda)$: $(0.0001, 0.03, 0.0005)$, $(0.0003, 0.03, 0.0005)$, $(0.001, 0.03, 0.0005)$, $(0.0003, 0.01, 0.0005)$, $(0.0003, 0.03, 0.0001)$; momentum is 0.9.
- STM (η, γ, λ): $(0.003, 0.010, 0.0005)$, $(0.003, 0.015, 0.0005)$, $(0.005, 0.015, 0.0005)$, $(0.003, 0.030, 0.0005)$, $(0.003, 0.015, 0.0001)$, $(0.005, 0.030, 0.0001)$.

Selected CIFAR-10 settings are SGD $(\eta, \lambda) = (0.03, 0.0001)$; AdamW $(\eta, \lambda) = (0.01, 0.0001)$; K-FAC $(\eta, \text{damping}, \lambda) = (0.0003, 0.03, 0.0005)$; and STM $(\eta, \gamma, \lambda) = (0.003, 0.015, 0.0005)$. Shakespeare candidates are the Cartesian products within each row of Table 4. HPO uses seeds 48–50. Selected settings are SGD $(\eta, \mu) = (0.01, 0.95)$; AdamW $(\eta, \lambda) = (0.001, 0.1)$; K-FAC $(\eta, \text{damping}) = (0.01, 0.01)$; and STM $(\eta, \gamma) = (0.0005, 0.005)$; remaining settings are as listed in the table.

Table 3: CIFAR-10 / ResNet configurations. "#" counts tested settings.

Method	#	LR(s)	Opt-specific params	Weight decay / momentum
STM	6	$\{0.003, 0.005\}$	$\beta = 0.95$, $K = 20$, $\gamma \in \{0.010, 0.015, 0.030\}$	$\lambda \in \{10^{-4}, 5\times10^{-4}\}$, $\mu = 0.9$
SGD	5	$\{0.01, 0.02, 0.03\}$	momentum SGD	$\lambda \in \{10^{-4}, 5\times10^{-4}\}$, $\mu = 0.9$
AdamW	5	$\{0.003, 0.005, 0.01\}$	warmup epochs $= 5$	$\lambda \in \{10^{-4}, 5\times10^{-4}\}$
K-FAC	5	$\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$	damping $\in \{0.01, 0.03\}$	$\lambda \in \{10^{-4}, 5\times10^{-4}\}$

The CIFAR-10 ablations vary K, γ, or β one at a time (Table 5 and Fig. 2). Other parameters stay at the selected values: learning rate 0.003, weight decay 0.0005, momentum 0.9, and $(K, \gamma, \beta) = (20, 0.015, 0.95)$. We rerun the selected setting and reuse its results in all three sweeps.

Table 4: Shakespeare / nanoGPT configurations; "#" counts tested configurations.

Method	#	LR(s)	Opt-specific params	Weight decay / momentum
STM	6	$\{5\times10^{-4}, 10^{-3}, 0.003\}$	$\beta = 0.95$, $K = 20$, $\gamma \in \{0.005, 0.010\}$	$\lambda = 0.1$, $\mu = 0.9$
SGD	6	$\{0.005, 0.01, 0.02\}$	momentum SGD	$\lambda = 0$, $\mu \in \{0.9, 0.95\}$
AdamW	6	$\{3\times10^{-4}, 6\times10^{-4}, 10^{-3}\}$	warmup epochs $= 0$	$\lambda \in \{0.01, 0.1\}$
K-FAC	6	$\{0.01, 0.03, 0.1\}$	damping $\in \{0.003, 0.01\}$	$\lambda = 0$, $\mu = 0.9$

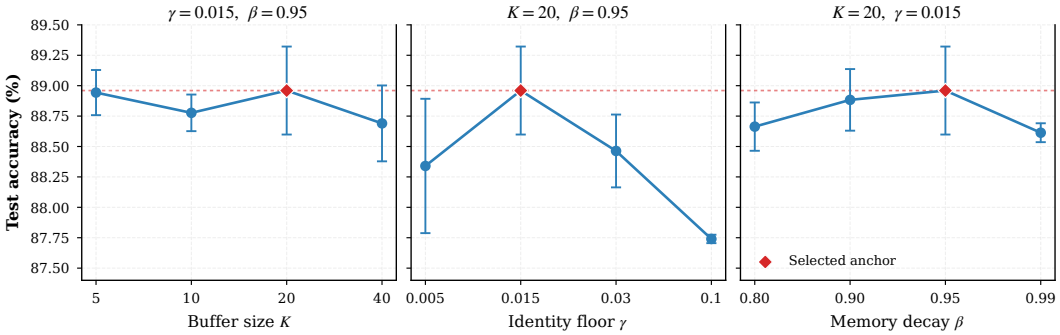


Figure 2: CIFAR-10 / ResNet ablations: buffer size K (left), identity floor γ (center), and memory decay β (right). Error bars show sample standard deviation across seeds 54–56.

Table 5: CIFAR-10 / ResNet ablations (100 epochs, seeds 54–56). Accuracy is the mean across seeds; Std is the sample standard deviation. Bold rows repeat the selected configuration.

K	γ	β	Test Acc (%)	Std (%)
Buffer size K ($\gamma = 0.015$, $\beta = 0.95$)				
5	0.015	0.95	88.94	±0.19
10	0.015	0.95	88.78	±0.15
20	**0.015**	**0.95**	**88.96**	**±0.36**
40	0.015	0.95	88.69	±0.31
Identity floor γ ($K = 20$, $\beta = 0.95$)				
20	0.005	0.95	88.34	±0.55
20	**0.015**	**0.95**	**88.96**	**±0.36**
20	0.030	0.95	88.46	±0.30
20	0.100	0.95	87.74	±0.03
Memory decay β ($K = 20$, $\gamma = 0.015$)				
20	0.015	0.80	88.66	±0.20
20	0.015	0.90	88.88	±0.25
20	**0.015**	**0.95**	**88.96**	**±0.36**
20	0.015	0.99	88.61	±0.08

B Proof of Proposition 3.2

Proof. For $0 < \gamma < 1$ and $0 < \beta < 1$, $D \succ 0$ is invertible and $H_t^{(K)} = \gamma I + G_t D G_t^\top$. Applying the Woodbury identity

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

with $A = \gamma I$, $U = G_t$, $C = D$, and $V = G_t^\top$ gives

$$[H_t^{(K)}]^{-1} = \frac{1}{\gamma} I - \frac{1}{\gamma^2} G_t \big(D^{-1} + \gamma^{-1} G_t^\top G_t\big)^{-1} G_t^\top = \frac{1}{\gamma}\Big(I - G_t \big(\gamma D^{-1} + G_t^\top G_t\big)^{-1} G_t^\top\Big).$$

Equation (5) follows immediately. □

C Proof of Proposition 3.3

Proof. Note that $G_t D G_t^\top = (G_t D^{1/2})(G_t D^{1/2})^\top \succeq 0$ shares its nonzero eigenvalues with $D^{1/2} G_t^\top G_t D^{1/2}$. We have $\lambda_{\min}(H_t^{(K)}) \geq \gamma$, which is tight when $\operatorname{rank}(G_t D G_t^\top) \leq K < d$. Hence,

$$\kappa\big(H_t^{(K)}\big) = \frac{\lambda_{\max}(H_t^{(K)})}{\lambda_{\min}(H_t^{(K)})} = \frac{\gamma + \lambda_{\max}(D^{1/2} G_t^\top G_t D^{1/2})}{\lambda_{\min}(H_t^{(K)})} \leq 1 + \frac{1}{\gamma}\,\lambda_{\max}\big(D^{1/2} G_t^\top G_t D^{1/2}\big).$$

□

D Proof of Theorem 6.1

Proof. By Eqs. (2) and (3), and $\|g_{t-j} g_{t-j}^\top\| = \|g_{t-j}\|_2^2$, we have

$$\|H_t - H_t^{(K)}\| = \left\| (1-\gamma)(1-\beta) \sum_{j=K}^{t-1} \beta^j g_{t-j} g_{t-j}^\top \right\| \leq (1-\gamma)(1-\beta) \sum_{j=K}^{t-1} \beta^j \|g_{t-j}\|_2^2.$$

By Section 3,

$$\mathbb{E}[\|g_{t-j}\|_2^2] = \mathbb{E}\big[\mathbb{E}_\xi[\operatorname{tr}(g_{t-j} g_{t-j}^\top) \mid \theta_{t-j}, \mathcal{D}_{t-j}]\big] = \mathbb{E}[\operatorname{tr}(\bar{\mathcal{F}}_B(\theta_{t-j}))].$$

Therefore

$$\mathbb{E}\big[\|H_t - H_t^{(K)}\|\big] \leq (1-\gamma)(1-\beta) \sum_{j=K}^{t-1} \beta^j \mathbb{E}\big[\operatorname{tr}(\bar{\mathcal{F}}_B(\theta_{t-j}))\big] = (1-\gamma)\beta^K \mathcal{T}_{t,K}^{(\beta)}. \tag{17}$$

Since $H_t \succeq H_t^{(K)} \succeq \gamma I \succ 0$, we have $\|H_t^{-1}\| \leq \gamma^{-1}$ and $\|[H_t^{(K)}]^{-1}\| \leq \gamma^{-1}$, hence

$$\begin{aligned}\frac{\|H_t^{-1} v_t - [H_t^{(K)}]^{-1} v_t\|_2}{\|v_t\|_2} &\leq \big\|H_t^{-1} - [H_t^{(K)}]^{-1}\big\| = \big\|H_t^{-1}\big(H_t^{(K)} - H_t\big)[H_t^{(K)}]^{-1}\big\| \\ &\leq \|H_t^{-1}\|\,\|H_t - H_t^{(K)}\|\,\|[H_t^{(K)}]^{-1}\| \leq \frac{1}{\gamma^2}\|H_t - H_t^{(K)}\|.\end{aligned}$$

Taking expectations on both sides gives Eq. (8). □

E Proof of Proposition 6.3

Proof. By Eq. (3), $H_t^{(K)} \succeq \gamma I$ with $\gamma > 0$, thus

$$\|[H_t^{(K)}]^{-1}\| \leq \gamma^{-1}.$$

By the definitions of $\tilde{v}_t$ and r_t,

$$H_t^{(K)}(\hat{v}_t - \tilde{v}_t) = -r_t.$$

Hence

$$\|\hat{v}_t - \tilde{v}_t\|_2 = \|[H_t^{(K)}]^{-1} r_t\|_2 \leq \frac{\|r_t\|_2}{\gamma}.$$

□

F Proof of Corollary 6.4

Proof. By Eqs. (2) and (3),

$$0 \preceq H_t - H_t^{(K)} = (1-\gamma)(1-\beta)\sum_{j=K}^{t-1}\beta^j g_{t-j}g_{t-j}^\top \preceq \operatorname{tr}\big(H_t - H_t^{(K)}\big)I = \tau_{t,K} I,$$

thus

$$\gamma I \preceq H_t^{(K)} \preceq H_t \preceq H_t^{(K)} + \tau_{t,K} I.$$

Hence

$$\begin{aligned} 0 \preceq [H_t^{(K)}]^{-1} - H_t^{-1} &\preceq [H_t^{(K)}]^{-1} - \big(H_t^{(K)} + \tau_{t,K} I\big)^{-1} \\ &= \tau_{t,K}[H_t^{(K)}]^{-1}\big(H_t^{(K)} + \tau_{t,K} I\big)^{-1} \preceq \frac{\tau_{t,K}}{\gamma(\gamma+\tau_{t,K})} I. \end{aligned}$$

By Proposition 6.3,

$$\begin{aligned} \frac{\|\hat{v}_t - H_t^{-1}v_t\|_2}{\|v_t\|_2} &\le \frac{\|\hat{v}_t - [H_t^{(K)}]^{-1}v_t\|_2 + \|([H_t^{(K)}]^{-1} - H_t^{-1})v_t\|_2}{\|v_t\|_2} \\ &\le \frac{\|\hat{v}_t - [H_t^{(K)}]^{-1}v_t\|_2}{\|v_t\|_2} + \big\|[H_t^{(K)}]^{-1} - H_t^{-1}\big\| \le \frac{\rho_t}{\gamma} + \frac{\tau_{t,K}}{\gamma(\gamma+\tau_{t,K})}. \end{aligned}$$

For sharpness, choose an arbitrary unit vector $u \in \mathbb{R}^d$ such that $\|u\|_2 = 1$, and set

$$g_{t-j} = \begin{cases} \sqrt{\dfrac{\tau_{t,K}}{(1-\gamma)(1-\beta)\beta^K}}\, u, & K < t,\ \beta > 0,\ j = K, \\ 0, & \text{otherwise}, \end{cases} \qquad v_t = u, \qquad \hat{v}_t = \frac{1+\rho_t}{\gamma}u.$$

Since $g_{t-j} = 0$ for $0 \le j < K$, Eqs. (2) and (3) give

$$H_t^{(K)} = \gamma I, \qquad H_t = H_t^{(K)} + (1-\gamma)(1-\beta)\sum_{j=K}^{t-1}\beta^j g_{t-j}g_{t-j}^\top = \gamma I + \tau_{t,K} uu^\top.$$

Hence $H_t u = (\gamma + \tau_{t,K})u$, and

$$v_t - H_t^{(K)}\hat{v}_t = u - \gamma\frac{1+\rho_t}{\gamma}u = -\rho_t u, \qquad H_t^{-1}v_t = H_t^{-1}u = \frac{1}{\gamma+\tau_{t,K}}u.$$

Therefore

$$\frac{\|\hat{v}_t - H_t^{-1}v_t\|_2}{\|v_t\|_2} = \left\|\left(\frac{1+\rho_t}{\gamma} - \frac{1}{\gamma+\tau_{t,K}}\right)u\right\|_2 = \frac{\rho_t}{\gamma} + \frac{\tau_{t,K}}{\gamma(\gamma+\tau_{t,K})}.$$

□

G Proof of Theorem 6.5

Proof. For $(x, y) \in \mathcal{D}_{t-j} \times \{1, \dots, C\}$, let

$$b_{j,xy} = \frac{1}{\sqrt{B}}\sqrt{p_{\theta_{t-j}}(y \mid x)}\,\nabla_\theta \log p_{\theta_{t-j}}(y \mid x).$$

The probe in Section 3 satisfies

$$g_{t-j} = \sum_{x,y}\xi_{j,xy}b_{j,xy}, \qquad \mathbb{E}_\xi[g_{t-j}g_{t-j}^\top] = \sum_{x,y} b_{j,xy}b_{j,xy}^\top = F_j.$$

Since $\operatorname{tr} F_j = \sum_{x,y}\|b_{j,xy}\|_2^2$ and $\operatorname{tr}(F_j^2) = \sum_{x,y,x',y'}(b_{j,xy}^\top b_{j,x'y'})^2$, we have

$$\begin{aligned} \mathbb{E}_\xi\|g_{t-j}\|_2^4 &= \sum_{x,y}\|b_{j,xy}\|_2^4 + \sum_{(x,y)\neq(x',y')}\Big(\|b_{j,xy}\|_2^2\|b_{j,x'y'}\|_2^2 + 2(b_{j,xy}^\top b_{j,x'y'})^2\Big) \\ &= (\operatorname{tr} F_j)^2 + 2\operatorname{tr}(F_j^2) - 2\sum_{x,y}\|b_{j,xy}\|_2^4. \end{aligned}$$

Hence

$$\begin{aligned}\mathbb{E}_\xi\|g_{t-j}g_{t-j}^\top - F_j\|_F^2 &= \mathbb{E}_\xi\big[\|g_{t-j}\|_2^4 - 2\,\mathrm{tr}(g_{t-j}g_{t-j}^\top F_j) + \mathrm{tr}(F_j^2)\big] = \mathbb{E}_\xi\|g_{t-j}\|_2^4 - \mathrm{tr}(F_j^2)\\ &= (\mathrm{tr}\,F_j)^2 + \mathrm{tr}(F_j^2) - 2\sum_{x,y}\|b_{j,xy}\|_2^4 \le (\mathrm{tr}\,F_j)^2 + \mathrm{tr}(F_j^2).\end{aligned}$$

Therefore

$$\mathbb{E}_\xi\|H_t^{(K)} - \bar H_t^{(K)}\|^2 \le \mathbb{E}_\xi\|H_t^{(K)} - \bar H_t^{(K)}\|_F^2 = \sum_{j=0}^{K-1} w_j^2\,\mathbb{E}_\xi\|g_{t-j}g_{t-j}^\top - F_j\|_F^2 \le S_{t,K}.$$

For $S_{t,K} > 0$, Markov's inequality yields Eq. (12):

$$\Pr_\xi\left(\|H_t^{(K)} - \bar H_t^{(K)}\| > \sqrt{S_{t,K}/\delta}\right) \le \frac{\mathbb{E}_\xi\|H_t^{(K)} - \bar H_t^{(K)}\|^2}{S_{t,K}/\delta} \le \delta.$$

The case when $S_{t,K} = 0$ is trivial.

Finally, since $H_t^{(K)} \succeq \gamma I$ and $\bar H_t^{(K)} \succeq \gamma I$,

$$\big\|[H_t^{(K)}]^{-1} - [\bar H_t^{(K)}]^{-1}\big\| = \big\|[H_t^{(K)}]^{-1}\big(\bar H_t^{(K)} - H_t^{(K)}\big)[\bar H_t^{(K)}]^{-1}\big\| \le \gamma^{-2}\big\|H_t^{(K)} - \bar H_t^{(K)}\big\|,$$

which gives Eq. (13) on the same event. □

H Proof of Proposition 6.6

Proof. Since $\bar H_K = \gamma I + c_K F$, $\bar H_K^{-1}F = (\gamma I + c_K F)^{-1}F$ has eigenvalues

$$\mu_i = \frac{\lambda_i}{\gamma + c_K\lambda_i}, \qquad i = 1, \dots, d.$$

Since $\frac{\lambda}{\gamma + c_K\lambda}$ is monotonically increasing, we have $0 < \mu_{\min} = \mu_1 \le \cdots \le \mu_d = \mu_{\max}$ and

$$\kappa_{\mathrm{pre}} = \frac{\mu_{\max}}{\mu_{\min}} = \frac{\lambda_{\max}}{\lambda_{\min}}\,\frac{\gamma + c_K\lambda_{\min}}{\gamma + c_K\lambda_{\max}} = \kappa(F)\,\frac{\gamma + c_K\lambda_{\min}}{\gamma + c_K\lambda_{\max}} \le \kappa(F),$$

with strict inequality iff $\lambda_{\max} > \lambda_{\min}$.

For any $\eta > 0$, $I - \eta\bar H_K^{-1}F$ is symmetric with eigenvalues $1 - \eta\mu_i$. Therefore

$$\min_{\eta>0}\|I - \eta\bar H_K^{-1}F\| = \min_{\eta>0}\max\big\{|1-\eta\mu_{\min}|, |1-\eta\mu_{\max}|\big\} = \frac{\mu_{\max} - \mu_{\min}}{\mu_{\max} + \mu_{\min}} = q(\kappa_{\mathrm{pre}}),$$

attained when $1 - \eta\mu_{\min} = \eta\mu_{\max} - 1$, i.e., $\eta = 2/(\mu_{\min} + \mu_{\max})$.

We have

$$\begin{aligned}L(e_{n+1}) &= \frac{1}{2}\|F^{1/2}e_{n+1}\|_2^2 = \frac{1}{2}\big\|(I - \eta\bar H_K^{-1}F)F^{1/2}e_n\big\|_2^2\\ &\le \|I - \eta\bar H_K^{-1}F\|^2\,\frac{1}{2}\|F^{1/2}e_n\|_2^2 = q(\kappa_{\mathrm{pre}})^2 L(e_n).\end{aligned}$$

Replacing $(\bar H_K^{-1}F, \mu_{\min}, \mu_{\max})$ by $(F, \lambda_{\min}, \lambda_{\max})$ gives $\min_{\eta>0}\|I - \eta F\| = q(\kappa(F))$ at $\eta_{\mathrm{GD}} = 2/(\lambda_{\min} + \lambda_{\max})$.

Since $q(\kappa) = 1 - \frac{2}{\kappa+1}$ is monotonically increasing on $[1, \infty)$,

$$q(\kappa_{\mathrm{pre}}) \le q(\kappa(F)), \qquad q(\kappa_{\mathrm{pre}})^2 \le q(\kappa(F))^2,$$

with strict inequality iff $\lambda_{\max} > \lambda_{\min}$. The singular F case is trivial. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the scope as a matrix-free STM natural-gradient optimizer with stochastic probes, finite memory, theory, and experiments; these claims match the theoretical bounds, experiments, and limitations discussed later.

   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: The conclusion discusses subspace coverage, memory scaling for billion-parameter models, and the limited empirical scope which establishes feasibility rather than superiority.

   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: The theoretical section states the assumptions for truncation, concentration, inverse stability, and residual-based error; the corresponding proofs are provided in the appendix.

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: The experiments specify datasets, models, baselines, seeds, validation selection, test-at-validation-best reporting, batch size, and HPO protocol, with additional HPO grids and protocol details in the appendix.

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: [No]

   Justification: Code is not released with this paper; the paper provides algorithmic details, HPO grids, and reporting protocol.

   Guidelines:

   - The answer [N/A] means that the 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: The main text and appendix specify datasets, architectures, optimizers, train/validation/test splits, HPO grids, selection rule, seeds, epochs, batch size, and reporting protocol.

   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: The main tables report mean and standard deviation over three fixed random seeds, and the appendix states the seed set and aggregation protocol.

   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: [No]

   Justification: The experiments state that reported runs use a single NVIDIA L4 GPU (24GB), and the main table reports mean wall-clock time per epoch for each method and benchmark. Total GPU time is not provided.

   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: The work is an optimization-method paper using standard public datasets and does not involve human subjects, private data, sensitive attributes, or deployment in a high-risk setting.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The work is foundational optimizer research; any downstream societal impact would arise only through generic use in training neural networks and is not tied to a specific deployment or application domain.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained model, image generator, scraped dataset, or other artifact requiring controlled release safeguards.

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: [No]

Justification: The paper cites the datasets, architectures, and baseline methods used, but it does not explicitly list license names or terms of use for all existing assets.

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: The submission does not release a new dataset, model, or code package; the optimizer is documented as an algorithm in the paper.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [N/A]

Justification: The work does not involve human subjects research.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core method, theory, or experiments.

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.