Adam or Gauss-Newton? A Comparative Study In Terms of Basis Alignment and SGD Noise

Bingbin Liu[*,†] Rachit Bansal[*,†] Depen Morwani[*,†]
Nikhil Vyas[α,β] David Alvarez-Melis[†] Sham M. Kakade[†]
[†] Kempner Institute at Harvard University [α] OpenAI

Abstract

Diagonal preconditioners are computationally feasible approximate to second-order optimizers, which have shown significant promise in accelerating training of deep learning models. Two predominant approaches are based on Adam and Gauss-Newton (GN) methods: the former leverages statistics of current gradients and is the de-factor optimizers for neural networks, and the latter uses the diagonal elements of the Gauss-Newton matrix and underpins some of the recent diagonal optimizers such as Sophia.

In this work, we compare these two diagonal preconditioning methods through the lens of two key factors: the choice of basis in the preconditioner, and the impact of gradient noise from mini-batching. To gain insights, we analyze these optimizers on quadratic objectives and logistic regression under all four quadrants. We show that regardless of the basis, there exist instances where Adam outperforms both GN^{-1} and $\text{GN}^{-1/2}$ in full-batch settings. Conversely, in the stochastic regime, Adam behaves similarly to $\text{GN}^{-1/2}$ for linear regression under a Gaussian data assumption. These theoretical results are supported by empirical studies on both convex and non-convex objectives.

1 Introduction

Modern deep learning has shifted away from vanilla (stochastic) gradient descent toward adaptive first-order optimizers with preconditioned updates of the form $\theta_{t+1} = \theta_t - \eta P g_t$, where the preconditioner $P \in \mathbb{R}^{d\times d}$ is often taken to be diagonal. Popular methods such as Adam (Kingma & Ba, 2015), RMSProp (Tieleman & Hinton, 2012), Adafactor (Shazeer & Stern, 2018), SignSGD (Bernstein et al., 2018), and Lion (Chen et al., 2023) all fall into this category. Their diagonal preconditioners are typically computed from empirical gradient statistics, such as running averages of squared gradients—rather than from true second-order curvature information. This design choice has made them efficient and practical, but it also distances them from the traditional Newton-style motivation that exploits curvature for faster convergence.

Recently, new optimizers such as Sophia (Liu et al., 2024) have revisited this connection, incorporating approximations to the diagonal of the Gauss-Newton matrix into their preconditioners. This brings the preconditioning closer to a second-order interpretation while remaining computationally feasible. A recent paper (Vyas et al., 2024) further showed that existing non-diagonal preconditioners such as Shampoo (Gupta et al., 2018) can be interpreted as applying a *diagonal preconditioner in a transformed basis* (specifically, Shampoo's eigenbasis), and proposed an optimizer called SOAP that runs Adam in Shampoo's eigenbasis. Based on this reinterpretation, it becomes natural to study diagonal preconditioners through the lens of their operating bases, comparing their behaviors when defined in the standard parameter space versus in alternative, curvature-informed bases.

[*] Equal contribution. Correspondence to {bliu, rachitbansal, dmorwani}@g.harvard.edu.
[β] Work done as a PostDoc at Harvard.

Table 1: Comparing Adam vs GN diagonal preconditioners across two axes: (i) basis choice, and (ii) gradient noise. Our theoretical results are based on quadratics (§3) and logistic regression (§4), across the *eigen* and *identity* bases on full (population) and small (single-sample) batch.

	Batch-Size Regime	
Basis Choice	**Full Batch**	**Small Batch**
Eigenbasis	$\exists$ logistic example where Adam $>$ GN^{-1}	$\text{GN}^{-1} \geq$ Adam $\approx \text{GN}^{-1/2}$ for quadratics
Identity basis	$\exists$ quadratic example where Adam $>$ GN^{-1}	Adam $\approx \text{GN}^{-1/2}$ for quadratics

This motivates our central question: can we disentangle the role of the *basis* used for preconditioning from the choice of *diagonal* scaling in that basis? In this work, we explore this question by comparing two canonical choices of diagonal scalings: one based on the running average of squared gradients, as in Adam (which approximates the diagonal of the *empirical* Fisher (Kunstner et al., 2019)), and another based on the diagonal of the Gauss-Newton (GN) matrix, which reflects curvature information derived from the Fisher or Hessian. Since many empirical-gradient-based methods (Adam, Shampoo, SOAP) effectively use the *square root* of its second-moment estimate, we additionally consider the *power* for GN and consider preconditioning with both GN^{-1} and $\text{GN}^{-1/2}$. We will formally define these choices in §2.

Importantly, these diagonal forms can be applied in arbitrary bases—including the identity basis (used by default in Adam) and the eigenbasis of the GN matrix which are the focus of this work. By decoupling the influence of the preconditioning basis from that of the diagonal approximation applied within it, we are able to investigate one guiding question: whether the empirical Fisher (used in Adam and SOAP) offers any advantage over GN-derived curvature estimates, or whether it merely serves as a tractable proxy.

Our Contributions. We show that the effectiveness of diagonal preconditioners depends on two key factors: (1) *Basis choice*: We compare preconditioning in the eigenbasis of the GN matrix versus in the identity basis. (2) *Gradient noise*: We analyze both full-batch (population gradient) and stochastic (batch size 1) regimes to isolate how preconditioner behavior is influenced by gradient variance.

We provide theoretical results for linear regression and logistic regression.

- **Sensitivity to basis choice**: For linear regression, it is well known that GN preconditioning in the eigenbasis yields optimal convergence rates for quadratics. However, when the basis is misaligned, we show that Adam can outperform both GN^{-1} and $\text{GN}^{-1/2}$. Further, in the case of logistic regression, Adam can outperform GN^{-1} *even under the eigenbasis* with full batch update (§4).
- **Equivalence under noise** (§3.2): For linear regression in the stochastic regime with Gaussian data, we show that Adam behaves similarly to $\text{GN}^{-1/2}$ regardless of basis, suggesting a surprising alignment between its empirical design and curvature-based preconditioning.

These results are summarized in a two-by-two grid in Table 1. We further discuss the distinction between GN^{-1} and $\text{GN}^{-1/2}$ in §3.1.2. The quadratic model and logistic regression provide complementary perspectives, yielding a more complete picture and highlighting the benefit of separating basis and gradient noise considerations.

We complement our theoretical findings with empirical results [1] on toy datasets and more realistic problems including CIFAR10 and Transformer experiments (§5). The results align with the theory across all empirical settings, illustrating the practical implications of basis choice and gradient noise.

[1] Code can be found at https://github.com/ClaraBing/Adam_or_GN/.

1.1 Related work

Approximate second-order optimizers Recent advances on approximate second-order methods have demonstrated success in large-scale settings, serving as efficient alternatives to classic second-order methods such as Newton's and natural gradient descent, which are computationally bottle-necked to scale to high dimensions. While first-order diagonal preconditioners are shown to be comparable in practice Kaddour et al. (2023); Zhao et al. (2024), leveraging second-order information has proven effective. Most relevant to our work are methods that can be considered as applying diagonal preconditioners in a chosen basis (Gupta et al., 2018; Liu et al., 2024; Vyas et al., 2024; Jordan et al., 2024). However, there is no clear understanding how the preconditioner and the basis interact. For instance, Sophia (Liu et al., 2024) can be considered as applying GN^{-1} in the identity basis, which as we will show, is not always desired. In contrast, our work provides a clarifying decomposition of the design space of these second-order methods by separating the choice of basis in which to perform a diagonal preconditioner, from the choice of the diagonal preconditioner itself. For results in the stochastic regime, Martens (2020) provided results the stochastic case and depends on the condition number, similar to Lemma 2. However, their setting crucially differs from ours by assuming that the covariance of the gradients is independent of the current iterate.

Efficient optimizers for large-scale training Although preconditioned methods are theoretically appealing for their faster convergence, substantial efforts have been made to translate these gains to practical speedups in wall-clock time, which is crucial in modern large-scale training. To keep each update step lightweight, it is common to approximate the Hessian using the Gauss-Newton matrix (Equation (1)), which, despite being biased, relies only on gradient information and is therefore more computationally efficient than the other commonly used Hutchinson estimator. In addition, when estimating the eigenbasis, one can use the Kronecker factorization in place of the full basis (Martens & Grosse, 2015; George et al., 2018; Vyas et al., 2024). In this work, we analyze the full eigenbasis of the Gauss-Newton matrix, while adopting the Kronecker approximation in the experiments (see §C.1 for details).

Adam vs (S)GD There have been a lot of interest understanding the comparison of Adam and (stochastic) gradient descent. Related to the preconditioning perspective in our work, Das et al. (2024) studies Adam's preconditioning effect on quadratics, and shows that it outperforms SGD when the Hessian is sufficiently ill-conditioned. A line work focuses the comparison on optimizing Transformers, which has investigated through the lens of gradient noises (Zhang et al., 2020), relation to sign descent (Kunstner et al., 2023), and the curvature of the landscape (Jiang et al., 2023; Pan & Li, 2023); Ahn et al. (2024) provides a review and a theory-friendly abstraction. Most related to our work is Maes et al. (2024), which shows that Adam's advantage over SGD for Transformers rely crucially on the choice of basis. While these results hinge on properties specific to Transformers, we are interested in understanding of algorithm design with insights that can be generally applicable.

2 Preliminaries

Consider optimizing a function $f : \mathbb{R}^d \to \mathbb{R}$ parameterized by the (vectorized) parameter $\theta \in \mathbb{R}^d$ against a loss function ℓ. Updates performed by preconditioning optimizers can be seen as

$$\theta^{(t+1)} = \theta^{(t)} - \eta \cdot (UD^pU^\top)g^{(t)},$$

where η denotes the learning rate, D is the diagonal preconditioner which is raised to the exponent $p \in \{-\frac{1}{2}, -1\}$, U is the orthonormal basis on which the preconditioned update is performed, and $g^{(t)}$ denotes the gradient at time t (which could correspond to either population gradient or stochastic gradient depending on the setting). The full update is described in Algorithm 1, and we discuss the choice of the basis and the diagonal preconditioner below.

We start with describing the *Gauss-Newton* (GN) matrix, which is the first term in the following decomposition

of the Hessian:

$$H := \nabla_\theta^2 \ell = \nabla_\theta f \nabla_f^2 \ell \nabla_\theta f^\top + \nabla_f \ell \nabla_\theta^2 f := H^{(\mathrm{GN})} + \nabla_f \ell \nabla_\theta^2 f. \tag{1}$$

Contrast to the Hessian H, the GN matrix $H^{(\mathrm{GN})}$ requires only first-order information of the network to compute and often serves as a reasonable preconditioner in practice (Sankar et al., 2021). For convex loss functions, which will be the focus of this work, the GN term is positive-semidefinite (PSD) and admits a real-valued eigendecomposition.

Basis estimation. We focus on comparing two basis choices: 1) the identity basis $U = I$, and 2) the eigenbasis of the Gauss-Newton matrix $H^{(\mathrm{GN})}$. For the experiments, we additionally consider the Kronecker-factored preconditioner (Martens & Grosse, 2015; Vyas et al., 2024) as a computationally efficient approximation of the eigenbasis: for a matrix-valued parameter $\theta \in \mathbb{R}^{n\times m}$, $H^{(\mathrm{GN})} \in \mathbb{R}^{nm\times nm}$ is approximated by the outer product of two matrices of dimension $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{m\times m}$; see §C.1 for details.

Diagonal preconditioners. Given an orthonormal basis U, we first rotate the gradient g into the basis $\tilde{g} := U^\top g$, then apply a diagonal conditioner D to the rotated gradient. We are interested in two types of D: Adam and Gauss-Newton.

For Adam, we use the following definition for theoretical analyses, where the preconditioner has diagonal entries

$$D_{ii}^{(A)} := \left(\mathbb{E}[(\tilde{g}(x)_i)^2]\right)^{-1/2} = \left(\mathbb{E}[(u_i^\top g(x))^2]\right)^{-1/2}, \tag{2}$$

where the expectation is over all possible batches. Note that in the full batch case, this simply corresponds to the rotated gradient, while in the stochastic case, it depends on the norm of the per-sample rotated gradient. This is motivated from the practical version of Adam, which maintains a running average of the gradients seen during the training.

For Gauss-Newton, it takes an additional exponent parameter $p \in \{-\frac{1}{2}, -1\}$ and computes the diagonal elements as

$$D_{ii}^{(\mathrm{GN})} := (u_i^\top H^{(\mathrm{GN})} u_i)^p, \tag{3}$$

where u_i is the i^{th} vector in the given basis. In particular, when U is the eigenbasis of $H^{(\mathrm{GN})}$, $\{u_i^\top H^{(\mathrm{GN})} u_i\}_{i\in[d]}$ give the eigenvalues of $H^{(\mathrm{GN})}$.

The preconditioners for Adam and Gauss-Newton at batch size 1 for standard losses like cross-entropy (CE) and Mean Squared Error (MSE) correspond respectively to *empirical Fisher matrix* and the *Fisher matrix*. The former is defined with gradients with respect to labels from the true data distribution, whereas the latter is defined with respect to the output of the model.

3 Theoretical Analysis on Linear Regression

This section presents results on linear regression with the mean-squared loss. The goal is to learn a function $f_\theta(x) = \theta^\top x$ with loss $\ell(\theta) = \frac{1}{2}\mathbb{E}[((\theta - \theta^*)^\top x)^2]$, where θ^* is the ground truth parameter, and the input is Gaussian following $x \sim \mathcal{N}(0, \Sigma_x)$. For this setup, the vanilla gradient descent update has $\Delta^{(t+1)} := \theta^{(t+1)} - \theta^* = (\mathbf{I} - \eta\Sigma_x)\Delta^{(t)}$, and the Gauss-Newton matrix simplifies to the data covariance matrix, i.e. $H^{(\mathrm{GN})} = \mathbb{E}[\nabla_\theta f \nabla_\theta f^\top] = \Sigma_x$.

In the following, we will refer to the basis given by the eigenvectors of $H^{(\mathrm{GN})}$ as the "correct" basis, and the identity basis as the incorrect basis. Under the correct eigenbasis, it is well-known that GN^{-1} achieves the optimal convergence rate in both full-batch and stochastic setting [2] : when using the full batch, GN^{-1} converges in 1 step; for the stochastic setting, GN^{-1} decreases the loss at a linear rate. Details are included in §A.1 for completeness.

[2] By stochastic setting we refer to updates where each batch contains a single sample.

Algorithm 1 Preconditioned optimizer

Input: $\theta^{(0)}$, BASISTYPE $\in$ {Id, EigenBasis}, PRECOND $\in$ {Adam, GN}, power $p \in \{-0.5, -1\}$, learning rate $\{\eta_t\}_{t=1}^{T}$, regularization coefficient ϵ, gradient batch size b_G, and basis estimation batch size b_H.
for $t = 1$ **to** T **do**
  Sample a batch of b_G samples $X_G := \{x_i\}_{i \in [b_G]}$.
  Compute batch loss $\ell_t(\theta^{(t)}; X_G)$ and gradient $g^{(t)} = \nabla \ell_t(\theta^{(t)}; X_G)$.
  Sample a batch of b_H samples $X_H := \{x_i\}_{i \in [b_H]}$.
  Compute the Gauss-Newton matrix $H^{(\mathrm{GN})}$ using X_H.
  if BASISTYPE == Id **then** // Basis choice
    Basis $U \leftarrow I$.
  elif BASISTYPE == EigenBasis **then**
    Basis $U \leftarrow \mathrm{EigenDecomposition}(H^{(\mathrm{GN})})$.
  Compute the basis-rotated gradient $\tilde{g}^{(t)} = U^\top g^{(t)}$.
  if PRECOND == Adam **then** // Diagonal preconditioner choice
    Compute the diagonal preconditioner as $D_{ii} = \left(\mathbb{E}[(\tilde{g}(x)_i)^2]\right)^{-1/2}$.
  elif PRECOND == GN **then**
    Compute the diagonal preconditioner as $D_{ii} = (u_i^\top H^{(\mathrm{GN})} u_i)^p$.
  $\theta^{(t+1)} = \theta_t - UD\tilde{g}^{(t)} = \theta_t - UDU^\top g^{(t)}$.

This section therefore focuses on comparing Adam and Gauss-Newton in other scenarios considered in Table 1: For the full batch setting, we show that Adam and $\mathrm{GN}^{-1/2}$ can both outperform GN^{-1} under the incorrect identity basis (§3.1). For the stochastic regime, we show that Adam and $\mathrm{GN}^{-1/2}$ behave similarly regardless of the basis choice (§3.2).

3.1 Full-batch updates under the incorrect identity basis

The heterogeneous curvature in deep learning optimization motivates the use of preconditioned methods (Sagun et al., 2016; Ghorbani et al., 2019; Yao et al., 2019; Zhang et al., 2020; Liu et al., 2024). While GN^{-1} is known to achieve optimal convergence under the correct eigenbasis, what happens when the basis is poorly chosen and fails to reflect the true curvature? This section shows that the optimality GN^{-1} is indeed sensitive to the basis choice, even in the absence of gradient noise: under the incorrect identity basis, GN^{-1} can be outperformed by Adam and $\mathrm{GN}^{-1/2}$. We show in §3.1.1 that GN^{-1} can be severely suboptimal when the basis fails to reflect the true curvature, while Adam remains adaptive with its "auto-tuning" effect. Further, the covariance Σ_x can affect the comparison of GN^{-1} and $\mathrm{GN}^{-1/2}$ (§3.1.2).

3.1.1 Adam auto-tunes to the curvature

In this section, we show that GN is sensitive to the basis choice. We provide an example where the wrong basis obscures the true curvature, voiding GN's adaptiveness.

Consider a quadratic problem with input covariance

$$\Sigma_x := \mathbb{E}[xx^\top] = \begin{bmatrix} \mathbf{1}\mathbf{1}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{2d \times 2d}, \tag{4}$$

where $\mathbf{1} \in \mathbb{R}^d$ is the all-one vector, and $\mathbf{I} \in \mathbb{R}^{d \times d}$ is the identity matrix. This problem has a block structure that is symmetric among the first d coordinates and among the last d coordinates. The two blocks have widely different maximal eigenvalues and hence different optimal learning rates: The first block has a

maximum eigenvalue of d, and thus the maximum stable learning rate is the $\frac{2}{d}$. In contrast, all eigenvalues for the second block are 1, hence can afford a maximum learning rate of 2.

The wrong basis choice we consider is the identity basis, which hides the true curvature of the problem. We will see that this makes GN fail to adapt to the curvature of the problem, while Adam remains efficient via an "auto-tuning" effect.

GN converges slowly. When taking $U = I$, the diagonal preconditioner for GNhas $D_{ii}^{(\mathrm{GN})} = 1^p = 1$ (recall that p is the exponent parameter for GN). That is, both GN^{-1} and $\mathrm{GN}^{-\frac{1}{2}}$ simply scale all coordinates by the same factor as all diagonal elements are 1, behaving the same as vanilla gradient descent.

Adam "auto-tunes" to the curvature. Given the symmetry of the problem, we can assume that the gradient norms are the same for coordinates within the same block. Therefore, Adam effectively acts as normalized gradient descent for the first block, and acts as signed gradient descent in each coordinate in the second block. Let $\|g_0^{(t)}\|$ denote the gradient norm for the first block coordinates, and $|g_i^{(t)}|$ for $i \in [d]$ denote the per-coordinate gradient norm for the coordinates in the second block which evolves independently.

Recall that for quadratic problem, the gradient is $g = \Sigma_x \Delta$ where $\Delta := \theta - \theta^*$. The update has $\Delta^{(t+1)} = (\mathbf{I} - \eta\Sigma_x)\Delta^{(t)}$ for vanilla gradient descent, and the gradient norm goes down as long as $\eta \leq \frac{2}{\lambda_{\max}}$. For Adam, the updates can be considered as gradient descent with an adaptive learning rate. Specifically, following Equation (2), the first block updates as $\Delta_0^{(t+1)} = (\mathbf{I} - \frac{\eta}{\|g_0^{(t)}\|_2} \cdot \Sigma_x)\Delta_0^{(t)}$, and the second block has per-coordinate updates $\Delta_i^{(t+1)} = (\mathbf{I} - \frac{\eta}{|g_i^{(t)}|_2} \cdot \Sigma_x)\Delta_i^{(t)}$ for $i \in [d]$. This means:

1. $\|g_0^{(t)}\|$ decreases provided $\eta / \|g_0^{(t)}\| \leq 2/d$,
2. $|g_i^{(t)}|$ decreases for $i > 0$ provided $\eta / |g_i^{(t)}| \leq 2$.

Thus, after an initial "burn-in" period, $\eta / \|g_0^{(t)}\|$ reaches $2/d$ and oscillates around this value, while $\eta / |g_i^{(t)}|$ oscillates around 2. We refer to this as the *auto-tuning* of Adam: it adapts to the curvature of different coordinates on its own, by regulating the gradient norms. After this burn-in period, we can reduce the learning rate by half at every step, reaching a target error within log number of steps.

This *auto-tuning* effect is similar to adapting to the smoothness of the curvature, which is known to be a property of normalized gradient descent (Orabona, 2023). Note that auto-tuning comes from the norm of the mean gradient in the Adam's denominator for the full batch case. Instead, Adam behaves similarly to $\mathrm{GN}^{-1/2}$ at small batch sizes, as we will show in §3.2, and does not exhibit this autotuning effect when the gradient variance dominates. [3] Concurrent work by Roulet & Agarwala (2025) provides consistent empirical evidence that at small batch sizes, keeping only the mean term in the Adam's denominator improves its performance, thus supporting the auto-tuning viewpoint of Adam.

3.1.2 Which power to use for Gauss-Newton?

Newton's method was originally proposed with a preconditioner closer to GN^{-1}. However, the square root in the denominator of Adam (Kingma & Ba, 2015) and Adagrad (Duchi et al., 2010) has spurred multiple papers (Liu et al., 2024; Lin et al., 2024; Vyas et al., 2024) questioning the correct power of the preconditioner to be used in practice. In contrast to optimality of GN^{-1} under the eigenbasis, we claim that under the

[3] Recall the definition of $D_{ii}^{(A)}$ from Equation (2). For the i_{th} basis vector u_i, let $\mu_i := u_i^\top \mathbb{E}_x[g(x)]$ and $\sigma_i^2 := \mathbb{E}[(u_i^\top g(x) - \mu_i)^2]$ denote the mean and variance of the gradient projection along u_i. Consider gradients computed on a batch of size B, then $(D_{ii}^{(A)})^2 = \mu_i^2 + \frac{1}{B}\sigma_i^2$. $(D_{ii}^{(A)})^2$ is dominated by the mean for full-batch updates ($B \to \infty$), and is often dominated by the variance in the stochastic regime ($B = 1$).

incorrect identity basis, there exists problems for which $p = 0.5$ leads to faster convergence than $p = 1$. The key idea is that the convergence rate of preconditioned gradient descent (Boyd & Vandenberghe, 2004) depends on the condition number of the preconditioned Hessian. It then suffices to construct examples where the condition number is better behaved for $p = 0.5$ than $p = 1$. We provide details in §A.3 and accompanying simulation results in §5.

3.2 Stochastic regime: Adam and $\text{GN}^{-1/2}$ are equivalent

The previous section shows that Adam and $\text{GN}^{-1/2}$ can both outperform GN^{-1} when the updates are using population gradients but under a poor basis choice. In this section, we focus on the stochastic regime (i.e. batch size 1) and show that Adam and $\text{GN}^{-\frac{1}{2}}$ behave similarly regardless of the basis choice. Proofs for this section can be found in §A.2.

We first prove a stronger result, showing that for Gaussian input distribution and quadratic loss, empirical Fisher is approximately equivalent to Fisher up to a loss scaling.

Lemma 1. *For linear regression with Gaussian inputs, the following holds:*

$$\ell(\theta) \cdot \Sigma_x \preceq \frac{1}{2}\mathbb{E}[g(x)g(x)^\top] \preceq 3\ell(\theta) \cdot \Sigma_x.$$

We then utilize Lemma 1 to show that $\text{GN}^{-\frac{1}{2}}$ and Adam behave the same upto a scalar constant in any basis under the stochastic regime.

Corollary 1. *For single-sample updates, the updates of Adam and $GN^{-\frac{1}{2}}$ differ by a constant.*

$$\frac{1}{\sqrt{3\ell}} \cdot D^{(GN,-\frac{1}{2})} \preceq \frac{1}{2}D^{(A)} \preceq \frac{1}{\sqrt{\ell}} \cdot D^{(GN,-\frac{1}{2})}.$$

From the above lemmas, we expect Adam and $\text{GN}^{-1/2}$ to have a similar performance for small batch size. However, we still don't know how GN^{-1} and $\text{GN}^{-1/2}$ compares at small batch. To answer this, we provide a lemma quantifying the convergence rate of a general preconditioner P for linear regression with stochastic Gaussian inputs. Denoting the preconditioned Hessian as $\boldsymbol{A}(P) := P^{1/2}\Sigma_x P^{1/2}$, we have that:

Lemma 2. *For a general preconditioner P, for linear regression with stochastic Gaussian inputs, the following holds:*

$$\mathbb{E}[\ell^{(t)}] \leq O\left[\left(1 - \frac{\lambda_{\min}(\boldsymbol{A}(P))}{3\,\text{Tr}(\boldsymbol{A}(P))}\right)^t \ell^{(0)}\right].$$

Let $\kappa_s(\boldsymbol{A}(P)) = \frac{\text{Tr}(\boldsymbol{A}(P))}{\lambda_{\min}(\boldsymbol{A}(P))} = \sum_{i\in d} \frac{\lambda_i(\boldsymbol{A}(P))}{\lambda_{\min}(\boldsymbol{A}(P))}$ denote the condition-number-like quantity that the above bound depends on; note that $\kappa_s \geq d$. The bound shows that in the correct basis, GN^{-1} is the optimal preconditioner even in the stochastic regime, as it minimizes the condition number to $\kappa_s(\boldsymbol{A}(P)) = \kappa_s(\mathbf{I}) = d$. For $\text{GN}^{-1/2}$ in the correct basis, we have $\kappa_s(\boldsymbol{A}(P)) = \kappa_s((\Sigma_x)^{1/2}) = \sum_i \sqrt{\frac{\lambda_i(\Sigma_x)}{\lambda_{\min}(\Sigma_x)}}$, which is greater than d unless $\lambda_{\max}(\Sigma_x) = \lambda_{\min}(\Sigma_x)$.

4 Theoretical analysis on logistic regression

The optimality of GN^{-1} under the eigenbasis holds for quadratics (§3) but needs not be true in general. In this section, we show that for logistic regression, Adam can converge faster than GN^{-1} even under the eigenbasis with full batches.

Setup. The input x is from the set of d-dimensional one-hot vectors $\{e_i\}_{i=1}^d$, with probability $\nu_i := \mathbf{Pr}(x = e_i)$. Conditional on $x = e_i$, the label y is Bernoulli with mean $P_i := \mathbf{Pr}(y = 1|x = e_i)$. We assume $0.6 \le P_i \le 0.8$, $\forall i \in [d]$, i.e. the labels are neither deterministic nor fully random, and the optimal parameter has a bounded norm.

We optimize a weight-tied *two-layer linear network* $q : \mathbb{R}^d \to \mathbb{R}^d$, whose output depends on the *squares* of the weights: for any $\theta \in \mathbb{R}^d$ and for $i \in [d]$, we define the model's prediction as

$$q_i(\theta) = \underset{\theta}{\mathbf{Pr}}(y = 1 \mid x = e_i) = \sigma\Big(\sum_{j=1}^{d} \theta_j^2 x_j\Big) = \sigma(\theta_i^2), \qquad \sigma(z) = \frac{1}{1 + \exp(-z)}. \tag{5}$$

The square parameterization makes the problem non-convex, and is analogous to the structure of key-query multiplication in self-attention (Vaswani et al., 2017).

In the following, we show that under the natural assumption of non-increasing step sizes, there is a separation between Adam and GN^{-1} in terms of $\kappa(\nu) := \frac{\nu_{\max}}{\nu_{\min}}$. We consider local convergence near the optimum. In particular, choose ν such as $\kappa(\nu) = \Omega(d^{1/2+\delta})$ for some $\delta \in [0, \frac{1}{2}]$, and let ϵ denote the target parameter error, i.e. we want to find θ such that $\|\theta - \theta^*\|_2 \le \epsilon$. We prove that Adam enjoys dimension-free convergence, whereas GN^{-1} suffers from a polynomial-in-dimension slowdown.

Adam converges in $O(\log(1/\epsilon))$ steps. Since Adam effectively performs sign GD, the amount of parameter update is determined by the step size. The $O(\log(1/\epsilon)$ convergence hence follows directly from starting with a $O(1)$ learning rate and halving the step size every $O(1)$ steps.

GN^{-1} requires $\tilde{\Omega}(d^\delta \log(1/\epsilon))$ steps. In contrast to Adam, GN's precondtioning can result in an unboundedly large update that makes optimization diverge [4] (see Equation (22)), unless the learning rate is kept small, which in turn leads to slow convergence. To show the lower bound, we first state a more general convergence result. Recall that the GN^{-1} update is $\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)}(H^{(\mathrm{GN})}(\theta) + \alpha I)^{-1} g^{(t)}$. We assume that $\{\eta^{(t)}\}_{t\ge 0}$ are non-increasing and that the step size schedule and regularization lead to convergence, i.e. $\theta^{(t)} \to \theta^*$ as $t \to \infty$. Linearizing the update map at the limit θ^* gives

$$\theta^{(t+1)} - \theta^* = \Big(I - \eta^\infty (H_*^{(\mathrm{GN})} + \alpha I)^{-1} H_*^{(\mathrm{GN})}\Big)(\theta^{(t)} - \theta^*) \; + \; O\big(\|\theta^{(t)} - \theta^*\|^2\big),$$

where $H_*^{(\mathrm{GN})} := H^{(\mathrm{GN})}(\theta^*)$ and $\eta^\infty = \lim_{t\to\infty} \eta_t$. We are interested in lower bounding the spectral radius of the local iteration matrix:

$$\gamma(\eta^{(\infty)}, \alpha) := \big\| I - \eta^{(\infty)} (H_*^{(\mathrm{GN})} + \alpha I)^{-1} H_*^{(\mathrm{GN})} \big\|_2,$$

as it governs the ultimate *local* rate of convergence that any GN schedule can achieve. We refer to $\gamma(\eta^{(\infty)}, \alpha)$ as the local contraction factor; a value of γ close to 1 implies slow convergence.

The main result of this section is a lower bound on γ:

Theorem 2. *Suppose the weights are initialized at $\theta^{(0)} = \frac{1}{\sqrt{d}} \cdot \vec{1}$. Consider any non-increasing step size sequence $\{\eta^{(t)}\}_{t\ge 0}$ and regularization parameter $\alpha \ge 0$. If the Gauss-Newton iterates converge to θ^*, i.e. $\theta^{(t)} \to \theta^*$, then the local contraction factor $\gamma(\eta^{(\infty)}, \alpha)$ is lower bounded by:*

$$\gamma(\eta^{(\infty)}, \alpha) \ge 1 - c\sqrt{\log d}\, \max\left\{ \frac{1}{\sqrt{d}}, \sqrt{d/\kappa(\nu)} \right\},$$

for some universal constant c.

[4] Recall that the optimum is bounded given a P bounded away from 1, which differs from analyses on linearly separable data where the optimum is at infinity (Wu et al., 2024).

This theorem reveals a basic trade-off: for the Gauss-Newton method to converge globally from our chosen starting point, its final learning rate must be small. This restriction, in turn, hampers its local convergence speed and creates a bottleneck. The slowdown is substantial under the common conditions of high dimensionality and ill-conditioned data:

Corollary 3. *For imbalanced input with $\kappa(\nu) = \Omega(d^{1/2+\delta})$ for some $\delta \in [0, 1/2]$, GN^{-1} requires $t = \tilde{\Omega}(d^{\delta} \log(1/\epsilon))$ steps to reach a parameter satisfying $\|\theta^{(t)} - \theta^*\|_2 \leq \epsilon$.*

This demonstrates a polynomial slowdown in the dimension, highlighting a scenario where the theoretical power of Gauss-Newton is significantly degraded due to the practical requirement of global convergence. We empirically verify this on Transformers in §5.

Remark: Ideally, in a purely local setting, one could choose $\alpha = 0$ and use a constant final stepsize η^{∞}. For instance, setting $\eta^{\infty} = 1$ would make the iteration matrix zero, yielding $\gamma = 0$ and superlinear convergence. In fact, any other constant $\eta^{\infty} \in (0, 2)$ would similarly provide rapid linear convergence with a rate independent of the condition number. However, Theorem 2 shows this ideal scenario is not possible. As the proof (§B) shows, the requirement of ensuring convergence from a specific, natural initialization forces the algorithm's final step size, $\eta^{(\infty)}$, to be small. This constraint directly degrades the local contraction factor, preventing the rapid convergence one might expect from a Newton-like method. Further, we note that our result does not contradict with the fast convergence from line search, which does not fall under the non-increasing step size assumption.

5 Experiments

We provide experimental evidence to the theoretical results in Table 1. Our experiments are broadly divided into two categories: (i) simulations for examples in §3 and §4, (ii) non-convex examples with MLP, and (iii) Transformer experiments.

Experiment details. We use the mean square error as the objective function unless otherwise specified. For numerical stability, we optionally regularize D to be $D + \alpha I$ for some small $\alpha > 0$. We sweep over the learning rate η and the regularization coefficient α. For Adam, we also sweep over the learning rate schedule (constant or step decay) and β_2; we fix $\beta_1 = 0$, similar to Das et al. (2024). We use different samples for estimating gradient and Gauss-Newton. Due to computational considerations, our eigenbasis experiments also consider Kronecker approximation of the full eigenbasis. We report the mean and standard error based on 10 seeds. More details are provided in §C.1.

Simulations This section provides simulation results on the examples in §3 and §4.

- *Comparing Adam and GN under full-batch updates.* We empirically verify the examples provided in the theory where Adam can outperform GN both under the identity basis and the eigenbasis (Figure 1). For the identity basis, we consider a 100-dimensional linear regression task where the covariance matrix has a block-wise structure following the construction in §3.1.1. Adam converges quickly due to the auto-tuning effect, whereas Gauss-Newton converge as slowly as vanilla gradient descent. For logistic regression (§4), our results on a 2048-dimensional problem (detailed in §C.4) show that Adam converges faster even under the eigenbasis.
- *Comparing powers of Gauss-Newton.* §3.1.2 shows that the comparison of GN^{-1} and $\text{GN}^{-1/2}$ amounts to comparing the condition number of a particular matrix, for both population and stochastic settings. We empirically verify the claim on a 5-dimensional linear regression problem, controlling the choice of the covariance Σ_x. Figure 2 shows the simulation results in this case, where $\text{GN}^{-1/2}$ converges faster than GN^{-1} when using both large and small batches. Details are provided in §A.3.

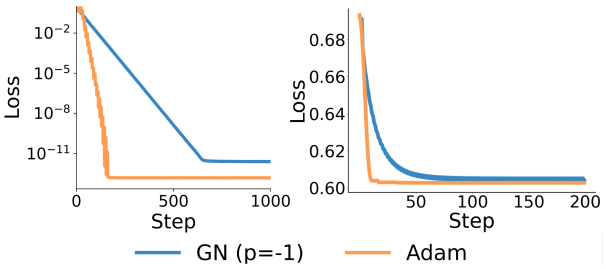


Figure 1: **Adam converges faster than GN with full batches**, (Left) under the identity basis on a linear regression task with block-wise covariance, where GN fails to adapt to the problem curvature; and (Right) under the eigenbasis, for the reparameterized logistic regression task.

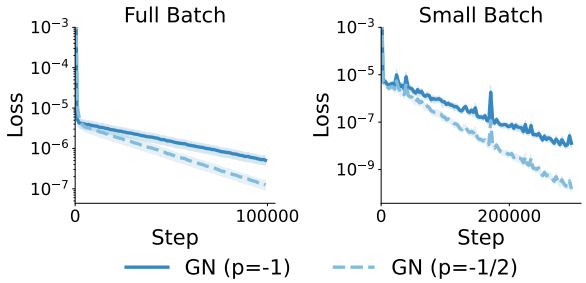


Figure 2: **Comparing GN power** $p \in \{-\frac{1}{2}, -1\}$. On a regression task where $\mathrm{GN}^{-1/2}$ leads to a more favorable condition number, $\mathrm{GN}^{-1/2}$ converges faster than GN^{-1} with both small and large batches.

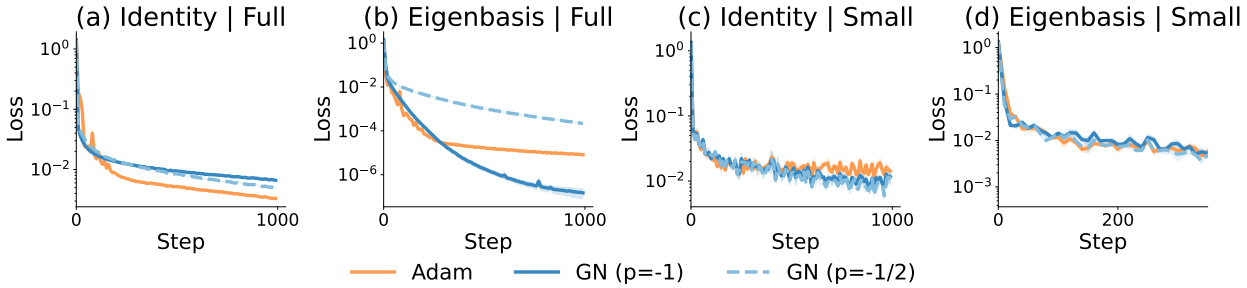


Figure 3: **Learning from a random teacher**, comparing Adam, GN^{-1} and $\mathrm{GN}^{-1/2}$ for the full 2×2 grid (Table 1).

Non-convex examples with MLP Next, we consider non-convex optimization with one-hidden-layer MLPs on the following tasks: 1) learning from a random teacher network; 2) feature learning with sparse parity and its variant "staircase", where the labels depend on a subset of input coordinates; and 3) CIFAR10 image classification, where the class labels are treated as one-hot vectors; §C.2 provides details. Experiments on these tasks cover the full 2×2 grid in Table 1, with respect to batch size (full vs small) and the basis (eigenbasis vs identity). We use the Kronecker approximation for the eigenbasis of the Hessian, which behaves similarly as the full eigenbasis (§C.1) while being more compute efficient. We provide additional experiments on intermediate basis choices by interpolating between the identity and the eigenbasis in §C.3.

As shown in Figures 3–6, our theoretical analyses empirically extend to these four non-convex tasks across both axes of interest. In particular, across all problems considered, Adam and $\mathrm{GN}^{-1/2}$ closely track one another at small batch sizes, regardless of the basis chosen (subfigures (c), (d)). Moreover, Adam is close to or better than GN^{-1} when the basis is incorrect (subfigures (a), (c)).

A logistic-like task with Transformers Finally, we experiment with Transformers, whose attention module resembles the structure of the reparametrized logistic regression in §4. We consider a selection-based regression task, where the input sequence contains a series of Gaussian vectors $\{x_t\}_{t \in [T]}$, followed by a one-hot vector that selects the position t that the regression target is based on; details are provided in §C.4). As shown in Figure 7, Adam outperforms GN with full batch updates even under the eigenbasis, consistent with our theoretical results.

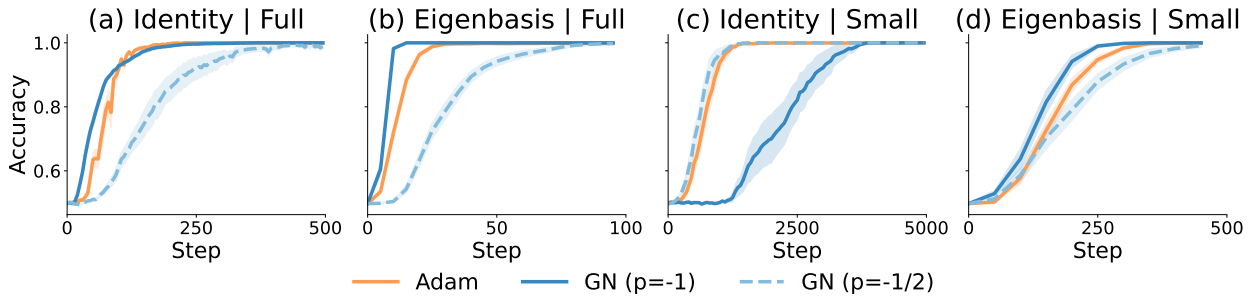


Figure 4: **Sparse parity**, comparing Adam, GN^{-1} and $\mathrm{GN}^{-1/2}$ for the full 2×2 grid (Table 1).

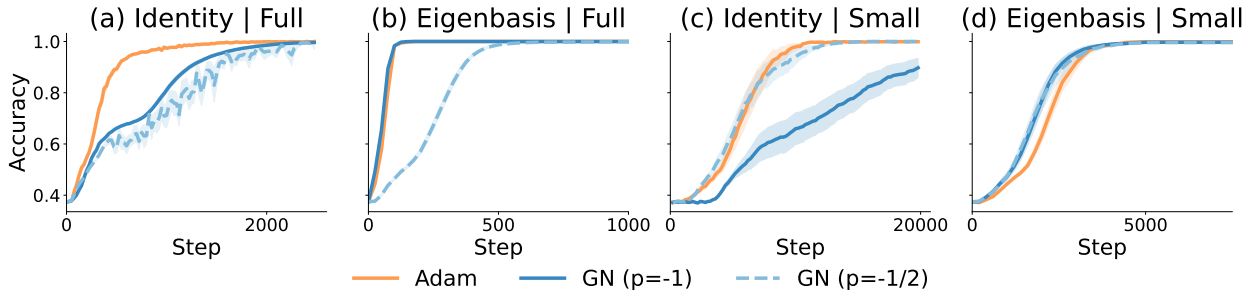


Figure 5: **Staircase**, comparing Adam, GN^{-1} and $\mathrm{GN}^{-1/2}$ for the full 2×2 grid (Table 1). Staircase is a generalization of sparse parity (Figure 4).

6 Discussion

This work studies the effectiveness of diagonal preconditioners along two key factors: the alignment to the ideal eigenbasis, and the level of gradient noise as influenced by the batch size. Our theoretical results on linear and logistic regression show that the comparison between Adam and Gauss-Newton (GN)-based diagonal preconditioners is sensitive to the change in either factor (Table 1): In the full batch setting, Adam can outperform GN in the identity basis for linear regression, and can even outperform GN in the ideal eigenbasis when considering logistic regression. In contrast, in the stochastic regime, we show that Adam and $\mathrm{GN}^{-1/2}$ exhibit similar behavior for linear regression regardless of the basis choice, thereby revealing a connection between Adam's design and curvature-based preconditioning.

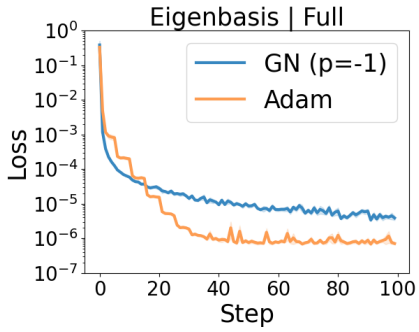


Figure 7: Transformer experiments (§5): Adam outperforms GN^{-1} under the eigenbasis with full batches.

Our empirical results support the theoretical findings. In particular, all MLP experiments align with linear regression results, and the Transformer experiments align with our logistic regression results.

It is important to understand whether phenomena and differences observed in small-scale, synthetic setups persist across scale. We hypothesize that the equivalence between Adam and $\mathrm{GN}^{-1/2}$ in the stochastic regime extends to practical, large-scale training. In particular, as training progresses, the gradient variance tends to dominate over the gradient mean, mirroring the stochastic regime in which variance drives the dynamics. Validating this hypothesis in large-scale settings is an interesting direction for future work. Finally, such equivalence combined with the benefits of Adam's *auto-tuning* at large-batch regimes suggests a promising direction: developing algorithms that exhibit similar desirable auto-tuning behavior even when operating with small batches.

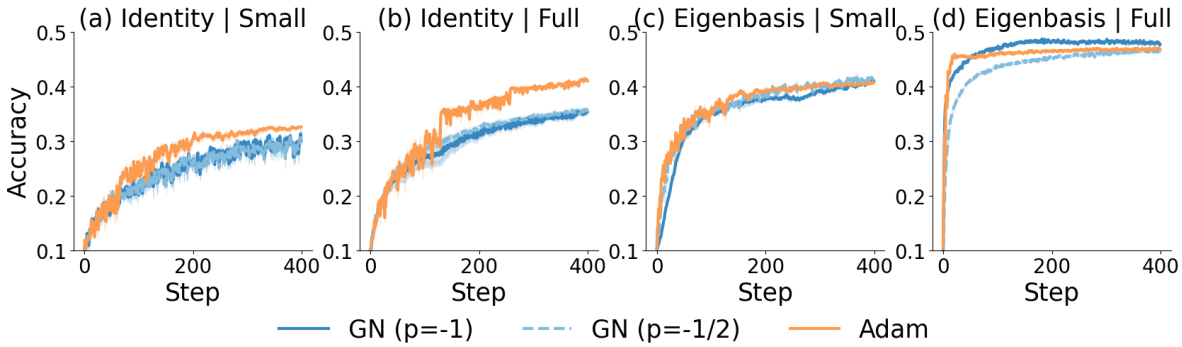


Figure 6: **CIFAR10**, comparing Adam, GN^{-1} and $\mathrm{GN}^{-1/2}$ for the full 2×2 grid (Table 1).

Acknowledgment

We thank Alex Damian for the helpful feedback. This work was enabled in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence. DM is supported by a Simons Investigator Fellowship, NSF grant DMS-2134157, DARPA grant W911NF2010021,and DOE grant DE-SC0022199. SK and DM acknowledge support from the Office of Naval Research under award N0001422-1-2377 and the National Science Foundation Grant under award #IIS 2229881.

References

E. Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. *Annual Conference Computational Learning Theory*, 2023a. doi: 10.48550/arXiv.2302.11055.

Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. *arXiv preprint arXiv: 2202.08658*, 2022.

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. Generalization on the unseen, logic reasoning and degree curriculum. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31–60. PMLR, 2023b. URL https://proceedings.mlr.press/v202/abbe23a.html.

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear attention is (maybe) all you need (to understand transformer optimization). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=0uI5415ry7.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham M. Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/884baf65392170763b27c914087bde01-Abstract-Conference.html.

Frederik Benzing. Gradient descent on neurons and its link to approximate second-order optimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato

(eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1817–1853. PMLR, 2022. URL https://proceedings.mlr.press/v162/benzing22a.html.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SIGNSGD: compressed optimisation for non-convex problems. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 559–568. PMLR, 2018. URL http://proceedings.mlr.press/v80/bernstein18a.html.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers and their ability to learn sparse Boolean functions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5767–5791, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.317. URL https://aclanthology.org/2023.acl-long.317.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9a39b4925e35cf447ccba8757137d84f-Abstract-Conference.html.

Rudrajit Das, Naman Agarwal, Sujay Sanghavi, and Inderjit S. Dhillon. Towards quantifying the preconditioning effect of adam. *arXiv preprint arXiv: 2402.07114*, 2024.

John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In Adam Tauman Kalai and Mehryar Mohri (eds.), *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pp. 257–269. Omnipress, 2010. URL http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf#page=265.

Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Pareto frontiers in neural feature learning: Data, compute, width, and luck. *arXiv preprint arXiv: 2309.03800*, 2023.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a kronecker factored eigenbasis. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9573–9583, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/48000647b315f6f00f913caa757a70b3-Abstract.html.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2232–2241. PMLR, 2019. URL http://proceedings.mlr.press/v97/ghorbani19b.html.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1837–1845. PMLR, 2018. URL http://proceedings.mlr.press/v80/gupta18a.html.

Kaiqi Jiang, Dhruv Malik, and Yuanzhi Li. How does adaptive optimization impact local neural network geometry? In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz

Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1a5e6d0441a8e1eda9a50717b0870f94-Abstract-Conference.html.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Jean Kaddour, Oscar Key, Piotr Nawrot, Pasquale Minervini, and Matt J. Kusner. No train no gain: Revisiting efficient training algorithms for transformer-based language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/51f3d6252706100325ddc435ba0ade0e-Abstract-Conference.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 and cifar-10 (canadian institute for advanced research), 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4158–4169, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/46a558d97954d0692411c861cf78ef79-Abstract.html.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=a65YK0cqH8g.

Wu Lin, Felix Dangel, Runa Eschenhagen, Juhan Bae, Richard E. Turner, and Alireza Makhzani. Can we remove the square-root in adaptive gradient methods? A second-order perspective. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=vuMD71R20q.

Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3xHDeA8Noi.

Lucas Maes, Tianyue H. Zhang, Alexia Jolicoeur-Martineau, Ioannis Mitliagkas, Damien Scieur, Simon Lacoste-Julien, and Charles Guille-Escuret. Understanding adam requires better rotation dependent assumptions. *arXiv preprint arXiv: 2410.19964*, 2024.

James Martens. New insights and perspectives on the natural gradient method. *J. Mach. Learn. Res.*, 21: 146:1–146:76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

James Martens and Roger B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2408–2417. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/martens15.html.

Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham M. Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=i9wDX850jR.

Francesco Orabona. Normalized gradients for all, 2023. URL https://arxiv.org/abs/2308.05621.

Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than sgd for transformers. *arXiv preprint arXiv: 2306.00204*, 2023.

Vincent Roulet and Atish Agarwala. Per-example gradients: a new frontier for understanding and improving optimizers. *arXiv preprint arXiv: 2510.00236*, 2025.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv: 1611.07476*, 2016.

Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N. Balasubramanian. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 9481–9488. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17142.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4603–4611. PMLR, 2018. URL http://proceedings.mlr.press/v80/shazeer18a.html.

Tijmen Tieleman and Geoffrey E. Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning. Technical report, University of Toronto, 2012. Available at: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.5.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv: 2409.11321*, 2024.

Jingfeng Wu, Peter L. Bartlett, Matus Telgarsky, and Bin Yu. Large stepsize gradient descent for logistic loss: Non-monotonicity of the loss improves optimization efficiency. *arXiv preprint arXiv: 2402.15926*, 2024.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. Pyhessian: Neural networks through the lens of the hessian. *arXiv preprint arXiv: 1912.07145*, 2019.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J. Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b05b57f6add810d3b7490866d74c0053-Abstract.html.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing what makes a good optimizer for language models. *arXiv preprint arXiv: 2407.07972*, 2024.

A Theoretical results and omitted proofs

A.1 Optimality of GN^{-1} in the correct basis

For completeness, we provide proofs for the optimality of GN^{-1} for the quadratic loss under the correct eigenbasis.

Full batch For GN^{-1}, the parameter estimation error evolves as

$$\begin{aligned}\theta^{(1)} - \theta^* =& (\theta^{(0)} - \theta^*) - \eta \cdot (H^{(\mathrm{GN})})^{-1} \cdot g^{(0)} \\ =& \theta^{(0)} - \eta \cdot \mathbb{E}[xx^\top]^{-1} \cdot \mathbb{E}[xx^\top(\theta - \theta^*)] = (1-\eta)(\theta - \theta^*).\end{aligned} \tag{6}$$

Hence GN^{-1} can reach the optimum in 1 step with $\eta = 1$.

Stochastic regime Let's consider the stochastic regime where each update step is performed with a single sample. With respect to some algorithm, define

$$M^{(t)} := \mathbb{E}[(\theta^{(t)} - \theta^*)(\theta^{(t)} - \theta^*)^\top].$$

which is the expected second-moment matrix of the distance-to-opt.

Recall that GN^{-1} has updates $\theta^{(t+1)} = \theta^{(t)} - \eta \Sigma_x^{-1} g^{(t)}$, where $g = (\theta - \theta^*)^\top x \cdot x$. We have that:

$$M^{(t+1)} = M^{(t)} - 2\eta M^{(t)} + \eta^2 \operatorname{Tr}(M^{(t)}\Sigma_x)\Sigma_x^{-1} + 2\eta^2 M^{(t)}. \tag{7}$$

Multiplying by Σ_x and taking the trace leads to:

$$\mathbb{E}[\ell^{(t+1)}] = \big(1 - 2\eta + 2\eta^2(d+1)\big)\mathbb{E}[\ell^{(t)}]. \tag{8}$$

Setting $\eta = \frac{1}{2(d+1)}$ reduces the expected error by a $\frac{1}{2}$ factor every $O(d)$ steps.

A.2 Equivalence of Adam and $\mathrm{GN}^{-0.5}$ under stochastic regime

We start with establishing the equivalence between the empirical and true Fisher (Lemma 1), which will then be used to prove the equivalence of Adam and $\mathrm{GN}^{-\frac{1}{2}}$'s updates.

A.2.1 Proof of Lemma 1: equivalence of the empirical and true Fisher

Lemma (Lemma 1, restated)**.** *For linear regression with Gaussian inputs, the following holds:*

$$\ell(\theta) \cdot \Sigma_x \preceq \frac{1}{2}\mathbb{E}[g(x)g(x)^\top] \preceq 3\ell(\theta) \cdot \Sigma_x.$$

Proof. For a given θ, let $M := (\theta - \theta^*)(\theta - \theta^*)^\top$. Then w.r.t. θ, we have

$$\mathbb{E}[g(x)g(x)^\top] = \mathbb{E}[x^\top M x \cdot xx^\top] = 2\Sigma_x M \Sigma_x + \operatorname{Tr}(\Sigma_x M)\Sigma_x \preceq 3\operatorname{Tr}(\Sigma_x M)\Sigma_x, \tag{9}$$

where the last equality follows from Wick's theorem. The lemma follows by noting that $\ell = \frac{1}{2}\mathbb{E}[x^\top M x] = \frac{1}{2}\operatorname{Tr}(\Sigma_x M)$. □

A.2.2 Proof of Theorem 1: equivalence of Adam and GN$^{-\frac{1}{2}}$

Corollary (Theorem 1, restated). *For single-sample updates, the update of Adam and GN$^{-\frac{1}{2}}$ differ by a constant.*

$$\frac{1}{\sqrt{3\ell}} \cdot D^{(GN,-\frac{1}{2})} \preceq \frac{1}{2} D^{(A)} \preceq \frac{1}{\sqrt{\ell}} \cdot D^{(GN,-\frac{1}{2})}.$$

Proof. Adam's preconditioner is based on

$$P^{(A)} := \mathbb{E}_{(x,y)\sim\mathcal{D}}[g(x)g(x)^\top]. \tag{10}$$

Given a basis U, the diagonal preconditioner given by Adam has entries

$$(D_{ii}^{(A)})^{-1} = \sqrt{u_i^\top P^{(A)} u_i}. \tag{11}$$

The relation between $D^{(A)}$ and $D^{(\mathrm{GN},-0.5)}$ follows from Lemma 1 and the fact that $(D_{ii}^{(\mathrm{GN},-0.5)})^{-1} = \sqrt{u_i^\top \Sigma_x u_i}$.

□

Part 2: when $H^{(\mathbf{GN})}$ is based on a single sample On single-sample batches, the gradient is $g(\theta) = \ell_f' \cdot \nabla_\theta f$. Then, the diagonal preconditioner for Adam (with $\beta_1 = \beta_2 = 0$) has Given a basis U, let $\tilde{g} := U^\top g$ denote the gradient rotated into the basis.

$$D_{ii}^{(A)} = (|\tilde{g}_i|)^{-1} = \left((\ell_f')^2 \cdot (U^\top \nabla_\theta f_i)^2\right)^{-0.5}.$$

The diagonal preconditioner for Gauss-Newton has

$$D_{ii}^{(GN)} = (H_{ii}^{(\mathrm{GN})})^{-0.5} = (\ell_f'' \cdot (U^\top \nabla_\theta f_i)^2)^{-0.5} = ((\ell_f')^2/\ell_f'')^{0.5} \cdot D_{ii}^{(A)}.$$

A.2.3 Proof of Lemma 2: loss convergence of general preconditioners

Lemma (Lemma 2, restated). *For a general preconditioner P, for linear regression with stochastic Gaussian inputs, the following holds:*

$$\mathbb{E}[\ell^{(t)}] \le O\left[\left(1 - \frac{\lambda_{\min}(\boldsymbol{A}(P))}{3\,\mathrm{Tr}(\boldsymbol{A}(P))}\right)^t \ell^{(0)}\right].$$

Proof. Let's define

$$M^{(t)} = \mathbb{E}[(\theta^{(t)} - \theta^\star)(\theta^{(t)} - \theta^\star)^\top]. \tag{12}$$

For any preconditioner P, given the update $\theta^{(t+1)} - \theta^* = (\mathbf{I} - \eta P x x^\top)(\theta^{(t)} - \theta^*)$, we have:

$$\begin{aligned} M^{(t+1)} &= M^{(t)} - \eta P \Sigma_x \theta^{(t)} - \eta \theta^{(t)} \Sigma_x P + \eta^2 P \left(2\Sigma_x M^{(t)} \Sigma_x + \mathrm{Tr}(\Sigma_x M^{(t)})\Sigma_x\right) P \\ &= (\mathbf{I} - \eta P \Sigma_x) M^{(t)} (\mathbf{I} - \eta P \Sigma_x)^\top + \eta^2 P \left(\Sigma_x M^{(t)} \Sigma_x + \mathrm{Tr}(\Sigma_x M^{(t)})\Sigma_x\right) P. \end{aligned} \tag{13}$$

Observe that $\mathbb{E}[\ell_t] = \mathbb{E}[\mathrm{Tr}(\Sigma_x M^{(t)})]$. This motivates us to make the following definition:

$$\widetilde{M}^{(t)} = \Sigma_x^{1/2} M^{(t)} \Sigma_x^{1/2}, \tag{14}$$

and so $\mathbb{E}[\ell_t] = \mathbb{E}[\mathrm{Tr}(\widetilde{M}^{(t)})]$.

Define $\boldsymbol{A}(P) := \Sigma_x^{1/2} P \Sigma_x^{1/2}$. The corresponding update rule is then:

$$\begin{aligned}\widetilde{M}^{(t+1)} &= (\mathbf{I} - \eta \boldsymbol{A}(P))\widetilde{M}^{(t)}(\mathbf{I} - \eta \boldsymbol{A}(P))^\top + \eta^2 \boldsymbol{A}(P)\left(\widetilde{M}^{(t)} + \mathrm{Tr}(\widetilde{M}^{(t)})\mathbf{I}\right)\boldsymbol{A}(P) \\ &\preceq (\mathbf{I} - \eta \boldsymbol{A}(P))\widetilde{M}^{(t)}(\mathbf{I} - \eta \boldsymbol{A}(P))^\top + 2\eta^2 \,\mathrm{Tr}(\widetilde{M}^{(t)})(\boldsymbol{A}(P))^2.\end{aligned} \tag{15}$$

For a given P, achieving the best loss contraction rate reduces to finding the optimal η. Rotating the left and the right hand side into the eigenbasis of $\boldsymbol{A}(P)$ and noting that the Trace of a matrix is independent of rotation, we can consider diagonal $\boldsymbol{A}(P)$ (without loss of generality) with the diagonal entries correspond to the eigenvalues. Define:

$$v_t = \mathrm{diag}(\widetilde{M}^{(t)}).$$

We have:

$$v_{t+1} = ((\mathbf{I} - \eta \boldsymbol{A}(P))^2 + \eta^2 \boldsymbol{A}(P)^2 + \eta^2 \mathrm{diag}(\boldsymbol{A}(P)^2)\vec{1}^\top)v_t. \tag{16}$$

Taking dot product with $\mathrm{diag}(\boldsymbol{A}(P)^{-1})$ on both sides, we get

$$\begin{aligned}\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_{t+1} &= \mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top((\mathbf{I} - \eta \boldsymbol{A}(P))^2 + \eta^2 \boldsymbol{A}(P)^2 + \eta^2 \mathrm{diag}(\boldsymbol{A}(P)^2)\vec{1}^\top)v_t \\ &= \mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_t - 2\eta 1^\top v_t + 2\eta^2 \mathrm{diag}(\boldsymbol{A}(P))^\top v_t + \eta^2 \,\mathrm{Tr}(\boldsymbol{A}(P))1^\top v_t.\end{aligned} \tag{17}$$

Since $\widetilde{M}^{(t)}$ is PSD, v_t has non-negative entries. Hence we have $1^\top v_t = \mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top \boldsymbol{A}(P) v_t \geq \lambda_{\min}(\boldsymbol{A}(P))\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v$ and

$$\begin{aligned}&\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_{t+1} \\ \leq &\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_t - 2\eta 1^\top v_t + \lambda_{\max}(\boldsymbol{A}(P))2\eta^2 1^\top v_t + \eta^2 \,\mathrm{Tr}(\boldsymbol{A}(P)1^\top v_t \\ = &\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_t - \eta \cdot \big(2 - (2\lambda_{\max}(\boldsymbol{A}(P)) + \mathrm{Tr}(\boldsymbol{A}(P))\eta\big)1^\top v_t \\ \leq &\left(1 - \lambda_{\min}(\boldsymbol{A}(P)) \cdot \eta\big(2 - (2\lambda_{\max}(\boldsymbol{A}(P)) + \mathrm{Tr}(\boldsymbol{A}(P))\eta\big)\right) \cdot \mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_t.\end{aligned} \tag{18}$$

The max contraction rate is achieved by setting $\eta = \frac{1}{2\lambda_{\max}(\boldsymbol{A}(P)) + \mathrm{Tr}(\boldsymbol{A}(P))}$, which gives

$$\begin{aligned}\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_{t+1} &\leq \left(1 - \frac{\lambda_{\min}(\boldsymbol{A}(P))}{2\lambda_{\max}(\boldsymbol{A}(P)) + \mathrm{Tr}(\boldsymbol{A}(P))}\right)\mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_t \\ &\leq \left(1 - \frac{\lambda_{\min}(\boldsymbol{A}(P))}{3\,\mathrm{Tr}(\boldsymbol{A}(P))}\right) \cdot \mathrm{diag}(\boldsymbol{A}(P)^{-1})^\top v_t.\end{aligned} \tag{19}$$

□

A.3 Comparing GN powers

This section discusses the comparison between GN^{-1} and $\mathrm{GN}^{-\frac{1}{2}}$. We will show that under the identity basis, $\mathrm{GN}^{-1/2}$ can outperform GN^{-1} even with full batches.

Let $\boldsymbol{A}(P) := P^{1/2}\Sigma_x P^{1/2}$ as defined in Section 3.1.2. One can show that the convergence rate of preconditioned gradient descent (Boyd & Vandenberghe, 2004) depends on the condition number of the preconditioned Hessian given by

$$\kappa(\boldsymbol{A}(P)) := \frac{\lambda_{\max}(\boldsymbol{A}(P))}{\lambda_{\min}(\boldsymbol{A}(P))}. \tag{20}$$

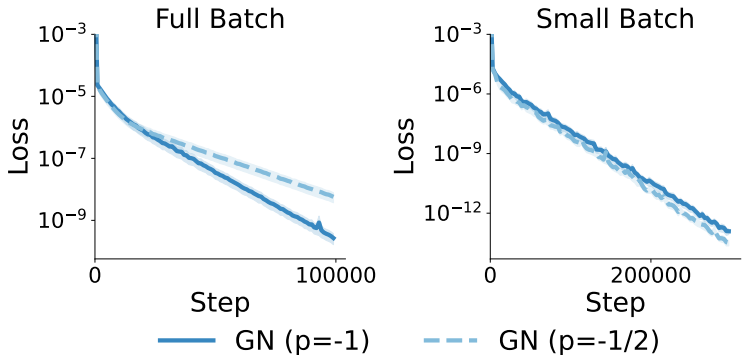


Figure 8: Comparing GN power $p \in \{-\frac{1}{2}, -1\}$. Contrary to Figure 2, when GN^{-1} has a more favorable condition number, it converges faster or close to $\mathrm{GN}^{-\frac{1}{2}}$ on both small and large batches.

By Lemma 2, to compare these two powers, it suffices to compare the condition number of for specific preconditioners Σ_x^{-1} and $\Sigma_x^{-1/2}$. We claim that there exists problems for which, even in the full batch case, in identity basis, $p = 0.5$ leads to faster convergence than $p = 1$:

Claim 1. *There exists Σ_x such that $\kappa(\boldsymbol{A}(diag(\Sigma_x)^{-1/2})) < \kappa(\boldsymbol{A}(diag(\Sigma_x)^{-1}))$.*

Denote $r(\Sigma_x) := \frac{\kappa(\Sigma_x^{1/2}\mathrm{diag}(\Sigma_x^{-1})\Sigma_x^{1/2})}{\kappa(\Sigma_x^{1/2}\mathrm{diag}(\Sigma_x^{-1/2})\Sigma_x^{1/2})}$. We empirically show that there exists Σ_x such that $r(\Sigma_x) > 1$. We obtain such Σ_x by fixing the diagonal matrix of eigenvalues Λ and randomly sampling orthonormal matrices U, and setting $\Sigma_x = U\Lambda U^\top$.

In particular, we construct two covariance matrices $\Sigma_{\text{half}}, \Sigma_{\text{one}} \in \mathbb{R}5 \times 5$, such that $r(\Sigma_{\text{half}}) > 1$ (i.e. $\mathrm{GN}^{-1/2}$ is more favorable), and $r(\Sigma_{\text{one}}) < 1$ (i.e. GN^{-1} is more favorable). As shown in Figure 2 and Figure 8, $\mathrm{GN}^{-\frac{1}{2}}$ indeed converges faster on data from Σ_{half}, whereas GN^{-1} converges faster with Σ_{one}, consistent with the theory.

Characterizing covariance matrices for which $r(\Sigma_x) > 1$ is left as future work.

B Proof for the logistic lower bound for GN (Theorem 2)

We optimize a weight-tied two-layer linear network $q : \mathbb{R}^d \to \mathbb{R}^d$, where

$$q_i(\theta) = \underset{\theta}{\mathbf{Pr}}(y = 1 \mid x = e_i) = \sigma\Big(\sum_{j=1}^{d} \theta_j^2 x_j\Big) = \sigma(\theta_i^2), \qquad \sigma(z) = \frac{1}{1+\exp(-z)}. \tag{21}$$

For logistic regression, the loss is $\ell(\theta) = -\mathbb{E}_{x,y}[\sum_{i\in[d]} \mathbb{I}(x = e_i)\big(y \log q_i(\theta) + (1-y)\log(1-q_i(\theta))\big)]$. The updates separate by dimension; the population gradient $g(\theta)$ and the diagonal Gauss-Newton matrix $H^{(\mathrm{GN})}(\theta)$ are given by

$$[g(\theta)]_i = 2\nu_i\,\theta_i\big(\sigma(\theta_i^2) - P_i\big), \quad [H^{(\mathrm{GN})}(\theta)]_{ii} = 4\nu_i\,\theta_i^2\,\sigma(\theta_i^2)\big(1-\sigma(\theta_i^2)\big), \tag{22}$$

where $P_i := \mathbf{Pr}(y = 1|x = e_i)$.

For a step size sequence $\{\eta^{(t)}\}$ and ridge regularization $\alpha \geq 0$, the Gauss-Newton iteration is

$$\theta^{(t+1)} = M_{\eta^{(t)},\alpha}(\theta^{(t)}), \qquad \text{where } M_{\eta,\alpha}(\theta) = \theta - \eta\,(H^{(\mathrm{GN})}(\theta) + \alpha I)^{-1} g(\theta).$$

Coordinate-wise, this update reads

$$[\, M_{\eta,\alpha}(\theta)\,]_i \;=\; \theta_i - \frac{2\theta_i \nu_i (\sigma(\theta_i^2) - P_i)}{4(\theta_i)^2 \nu_i\, \sigma(\theta_i^2)(1-\sigma(\theta_i^2)) + \alpha}. \tag{23}$$

The proof strategy for Theorem 2 hinges on the following key technical lemma that establishes a learning rate threshold for a single coordinate, where the threshold is a function of both initialization $\theta^{(0)}$ and the regularization parameter α. Above this threshold, the Gauss-Newton update diverges. By requiring that all coordinates avoid this divergence to ensure global convergence, we use the lemma to derive a strict upper bound on the algorithm's final learning rate, η^{∞}. Substituting this necessary restriction into the definition of the local contraction factor will directly yield the lower bound in Theorem 2, showing that slow convergence is an unavoidable consequence of global stability from the chosen initialization.

Lemma 3. *For constants $\eta, \alpha > 0$, define the one-dimensional update map $M_{\eta,\alpha} : \mathbb{R} \to \mathbb{R}$ corresponding to a regularized Gauss-Newton step:*

$$M_{\eta,\alpha}(\theta) = \theta - \eta \frac{2\theta(\sigma(\theta^2) - P)}{4\theta^2\sigma(\theta^2)(1-\sigma(\theta^2)) + \alpha}.$$

Consider the update rule $\theta^{(t+1)} = M_{\eta^{(t)},\alpha}(\theta^{(t)})$.

There exists a universal constant c such that for any target probability $P \in [0.6, 0.8]$ and any initial weight $\theta^{(0)} > 0$ satisfying $\sigma((\theta^{(0)})^2) \le 0.55$, if the non-increasing learning rate sequence satisfies

$$\eta^{(t)} \ge c\sqrt{\log \frac{1}{\theta^{(0)}}} \left(\theta^{(0)} + \frac{\alpha}{\theta^{(0)}}\right) \quad \textit{for all } t,$$

then $|\theta^{(t)}|$ diverges geometrically.

Proof. The proof proceeds in three parts. First, we show that the first step, $\theta^{(1)}$, becomes large. Second, we establish a key property satisfied by $\theta^{(1)}$. Finally, we use this to show that all subsequent iterates grow geometrically.

Part 1: The first step makes $\theta^{(1)}$ large. The condition $\sigma((\theta^{(0)})^2) \le 0.55$ implies $\theta^{(0)} \le 0.5$. Since $P \ge 0.6$, the term $\sigma((\theta^{(0)})^2) - P$ is negative, ensuring $\theta^{(1)} > \theta^{(0)}$. We can lower bound $\theta^{(1)}$ as follows:

$$\begin{aligned}
\theta^{(1)} &= \theta^{(0)} - \eta^{(0)} \frac{2\theta^{(0)}(\sigma((\theta^{(0)})^2) - P)}{4(\theta^{(0)})^2\sigma((\theta^{(0)})^2)(1-\sigma((\theta^{(0)})^2)) + \alpha} \\
&\ge \theta^{(0)} + \eta^{(0)} \frac{2\theta^{(0)}(0.6 - 0.55)}{4(\theta^{(0)})^2 \cdot 0.5^2 + \alpha} \\
&\ge \eta^{(0)} \frac{0.1\theta^{(0)}}{(\theta^{(0)})^2 + \alpha} \\
&= \eta^{(0)} \frac{0.1}{\theta^{(0)} + \alpha/\theta^{(0)}}.
\end{aligned}$$

Substituting our lower bound for $\eta^{(0)}$ from the lemma statement yields:

$$\theta^{(1)} \ge \left(c\sqrt{\log \frac{1}{\theta^{(0)}}} \left(\theta^{(0)} + \frac{\alpha}{\theta^{(0)}}\right)\right) \frac{0.1}{\theta^{(0)} + \alpha/\theta^{(0)}} = 0.1c\sqrt{\log \frac{1}{\theta^{(0)}}}.$$

As $\theta^{(0)} \to 0$, this lower bound grows, so we can choose the universal constant c large enough to make $\theta^{(1)}$ arbitrarily large.

Part 2: Establishing a key property of $\theta^{(1)}$. We now show we can choose c large enough to ensure two conditions hold simultaneously for $\theta^{(1)}$:

(i) $\sigma((\theta^{(1)})^2) \geq 0.9$.

(ii) $4(\theta^{(1)})^2(1-\sigma((\theta^{(1)})^2)) \leq \theta^{(0)}$.

Condition (i) is met by choosing c sufficiently large. For condition (ii), we use the facts that $1-\sigma(z) \leq e^{-z}$ and that $z(1-\sigma(z))$ is a decreasing function for $z \geq 2$. [5] Let $\theta^{(1)}_{\text{low}} := 0.1c\sqrt{\log(1/\theta^{(0)})}$. This implies:

$$\begin{aligned}
4(\theta^{(1)})^2(1-\sigma((\theta^{(1)})^2)) &\leq 4(\theta^{(1)}_{\text{low}})^2(1-\sigma((\theta^{(1)}_{\text{low}})^2)) \leq 4(\theta^{(1)}_{\text{low}})^2 e^{-(\theta^{(1)}_{\text{low}})^2} \\
&= 4(0.01c^2)\log(1/\theta^{(0)}) \cdot \exp\left(-(0.01c^2)\log(1/\theta^{(0)})\right) \\
&= (0.04c^2)\log(1/\theta^{(0)}) \cdot (\theta^{(0)})^{0.01c^2} \\
&= \left((0.04c^2)\log(1/\theta^{(0)})(\theta^{(0)})^{0.01c^2-1}\right) \cdot \theta^{(0)}.
\end{aligned}$$

For a sufficiently large constant c (e.g., $0.01c^2 > 2$), the term in the large parenthesis is less than 1, because for a fixed $\theta^{(0)} \in (0, 0.5]$, the polynomial term $(\theta^{(0)})^{0.01c^2-1}$ decays much faster than the logarithmic term $\log(1/\theta^{(0)})$ grows. This establishes condition (ii).

Part 3: Proving geometric divergence for $t \geq 1$. Consider any η satisfying the learning rate lower bound, i.e. suppose:

$$\eta \geq c\sqrt{\log\frac{1}{\theta^{(0)}}}\left(\theta^{(0)} + \frac{\alpha}{\theta^{(0)}}\right).$$

We show that for any θ where $\theta^2 \geq (\theta^{(1)})^2$, it follows that $|M_{\eta,\alpha}(\theta)| \geq \sqrt{2}|\theta|$. First, rewrite the update as

$$M_{\eta,\alpha}(\theta) = \theta\left(1 - \frac{2\eta(\sigma(\theta^2)-P)}{4\theta^2\sigma(\theta^2)(1-\sigma(\theta^2))+\alpha}\right).$$

Let $K(\theta) := \frac{2\eta(\sigma(\theta^2)-P)}{4\theta^2\sigma(\theta^2)(1-\sigma(\theta^2))+\alpha}$; note that $K(\theta) > 0$ provided that $\theta^2 \geq (\theta^{(1)})^2$. We seek to show $|1-K(\theta)| \geq \sqrt{2}$, which is true if $K(\theta) \geq 1+\sqrt{2}$.

We lower bound $K(\theta)$ for any θ where $\theta^2 \geq (\theta^{(1)})^2$. The numerator can be lower-bounded using condition (i):

$$2\eta(\sigma(\theta^2)-P) \geq 2\eta(\sigma((\theta^{(1)})^2)-P) \geq 2\eta(0.9-0.8) = 0.2\eta. \tag{24}$$

For the denominator, we use the fact that $z(1-\sigma(z))$ is decreasing (for $z \geq 2$), condition (ii), and that $\theta^{(0)} \in [0, 0.5]$:

$$4\theta^2\sigma(\theta^2)(1-\sigma(\theta^2))+\alpha \leq 4\theta^2(1-\sigma(\theta^2))+\alpha \leq 4(\theta^{(1)})^2\big(1-\sigma((\theta^{(1)})^2)\big)+\alpha \leq \theta^{(0)}+\alpha \leq \theta^{(0)} + \frac{\alpha}{\theta^{(0)}}.$$

Combining these bounds gives:

$$K(\theta) \geq \frac{0.2\eta}{\theta^{(0)}+\alpha/\theta^{(0)}} \geq 0.2c\sqrt{\log\frac{1}{\theta^{(0)}}},$$

where the last step follows by substituting the lower bound for η. Since $\theta^{(0)} \leq 0.5$, we have $\log(1/\theta^{(0)}) \geq \log(2)$. We can choose the universal constant c large enough such that $0.2c\sqrt{\log 2} \geq 1+\sqrt{2}$.

Thus, for any $t \geq 1$, we have $|\theta^{(t+1)}| = |M_{\eta^{(t)},\alpha}(\theta^{(t)})| \geq \sqrt{2}|\theta^{(t)}|$, which shows that $|\theta^{(t)}|$ diverges geometrically. □

[5] Indeed, $\frac{\mathrm{d}}{\mathrm{d}z}[z(1-\sigma(z))] = (1-\sigma(z)) - z\sigma(z)(1-\sigma(z)) < 0$ once $z \geq 2$.

We are now ready to complete the proof of Theorem 2.

Proof. (Theorem 2) The proof proceeds by using Lemma 3 to find an upper bound on the final learning rate $\eta^{(\infty)}$, and then substituting this bound into the definition of the local contraction factor γ.

At the optimum θ^*, the diagonal entries of the Fisher matrix $H_*^{(\mathrm{GN})} = H^{(\mathrm{GN})}(\theta^*)$ are

$$\lambda_i(H_*^{(\mathrm{GN})}) = 4(\theta_i^*)^2 \nu_i \sigma((\theta_i^*)^2)\big(1 - \sigma((\theta_i^*)^2)\big). \tag{25}$$

Recall the target probabilities $P_i = \sigma((\theta_i^*)^2)$ are assumed to lie in $[0.6, 0.8]$. This implies that $(\theta_i^*)^2$'s are bounded by a universal constant. Thus, the term $4(\theta_i^*)^2\sigma((\theta_i^*)^2)\big(1 - \sigma((\theta_i^*)^2)\big)$ is also bounded by universal constants, and we conclude that the Fisher eigenvalues are proportional to the sampling probabilities $\{\nu_i\}$. In particular,

$$\lambda_{\min}(H_*^{(\mathrm{GN})}) \geq \nu_{\min}/c_1,$$

where c_1 is a universal constant.

The Gauss-Newton update for each coordinate θ_i can be analyzed independently. For the sequence $\theta^{(t)}$ to converge θ^*, the iterates for each coordinate $\theta_i[t]$ must also converge to θ_i^*. From Equation (23) and taking α/ν_i as the regularization parameter, the update rule for each coordinate can be written as:

$$[\, M_{\eta,\alpha}(\theta)\,]_i \;=\; \theta_i - \eta\, \frac{2\theta_i(\sigma(\theta_i^2) - P_i)}{4(\theta_i)^2\,\sigma(\theta_i^2)(1-\sigma(\theta_i^2)) + \alpha/\nu_i}.$$

We can apply Lemma 3 coordinate wise. Since $\{\eta^{(t)}\}$ is non-increasing, Lemma 3 implies the limiting stepsize $\eta^{(\infty)}$ must satisfy the following for each coordinate $i \in [d]$:

$$\eta^{(\infty)} \leq c\sqrt{\log \frac{1}{\theta_i^{(0)}}}\left(\theta_i^{(0)} + \frac{\alpha/\nu_i}{\theta_i^{(0)}}\right).$$

In our setting, the initial weights are $\theta_i^{(0)} = 1/\sqrt{d}$. To get a single upper bound on $\eta^{(\infty)}$, we take the tightest possible constraint derived from above, which is when ν_i is at its maximum, $\nu_{\max}$. Thus, for convergence to be possible, $\eta^{(\infty)}$ must be bounded by:

$$\eta^{(\infty)} \leq c\sqrt{\log d}\left(\frac{1}{\sqrt{d}} + \frac{\sqrt{d}\alpha}{\nu_{\max}}\right).$$

The local contraction factor is the spectral radius of $I - \eta^{(\infty)}(H_*^{(\mathrm{GN})} + \mathrm{diag}(\{\alpha/\nu_i\}))^{-1}H_*^{(\mathrm{GN})}$, whose eigenvalues are $1 - \eta^{(\infty)}\frac{\lambda_i(H_*^{(\mathrm{GN})})}{\lambda_i(H_*^{(\mathrm{GN})})+\alpha/\nu_i} \geq 1 - 1 - \eta^{(\infty)}\frac{\lambda_i(H_*^{(\mathrm{GN})})}{\lambda_i(H_*^{(\mathrm{GN})})+\alpha}$ (recall Equation (25)). We can lower bound the spectral radius by considering the smallest eigenvalue of $H_*^{(\mathrm{GN})}$, $\lambda_{\min}$:

$$\gamma(\eta^{\infty}, \alpha) \geq 1 - \eta^{\infty}\frac{\lambda_{\min}(H_*^{(\mathrm{GN})})}{\lambda_{\min}(H_*^{(\mathrm{GN})}) + \alpha}.$$

Substituting the upper bound on $\eta^{(\infty)}$ from above:

$$\begin{aligned}\gamma(\eta^{(\infty)}, \alpha) &\geq 1 - \left(c\sqrt{\log d}\left(\frac{1}{\sqrt{d}} + \frac{\sqrt{d}\alpha}{\nu_{\max}}\right)\right)\frac{\lambda_{\min}}{\lambda_{\min}+\alpha} \\ &= 1 - c\sqrt{\log d}\left(\frac{1}{\sqrt{d}}\cdot\frac{\lambda_{\min}}{\lambda_{\min}+\alpha} + \frac{\sqrt{d}\alpha}{\nu_{\max}}\cdot\frac{\lambda_{\min}}{\lambda_{\min}+\alpha}\right).\end{aligned}$$

We can bound the terms in the parenthesis using $\frac{\lambda_{\min}}{\lambda_{\min}+\alpha} \leq 1$ and $\frac{\alpha}{\lambda_{\min}+\alpha} \leq 1$:

$$\gamma(\eta^{(\infty)}, \alpha) \geq 1 - c\sqrt{\log d}\left(\frac{1}{\sqrt{d}} + \frac{\sqrt{d}\lambda_{\min}}{\nu_{\max}}\right).$$

Using the our lower bound $\lambda_{\min} \geq \nu_{\min}/c_1$ from above, and absorbing constants into c':

$$\gamma(\eta^{(\infty)}, \alpha) \geq 1 - c'\sqrt{\log d}\left(\frac{1}{\sqrt{d}} + \sqrt{d}\frac{\nu_{\min}}{\nu_{\max}}\right),$$

which implies the claimed result. □

C Experiments

C.1 Additional experiment information

Hyperparameters, hardware, and runtime The learning rate (η) search is first performed at factors of 3 (e.g. $0.01, 0.003, 0.001$) and then at factors of 2 or finer around the optimal value. The regularization (α) search is at factors at 10 (e.g. $10^{-3}, 10^{-4}$). For Adam, we additionally sweep over $\beta_2 \in \{0, 0.9, 0.95, 0.99\}$. When comparing batch sizes, we vary the batch size used for computing gradient, and always use a large batch size (4096) for Gauss-Newton matrix to ensure an accurate basis estimation. Experiments were run on NVIDIA A100 GPUs. Simulation runs in §3.1.1 each completes within 1min. Simulation runs in §3.1.2 takes 9min for every 100k steps. The parity and staircase runs take around 10min for every 1k steps. For CIFAR experiments, runs under the identity basis take less than 5min each, and runs under the Kronecker approximation of the eigenbasis take around 80min each.

Kronecker factorization Inspired by prior work (Martens & Grosse, 2015; Gupta et al., 2018; Vyas et al., 2024), our experiments use Kronecker factorization as a computationally efficient approximation to the full eigenbasis. Given a matrix-valued parameter $W \in \mathbb{R}^{m\times n}$, let $g \in \mathbb{R}^{mn}$ denote the flattened gradient, and $G \in \mathbb{R}^{m\times n}$ denote the unflattened gradient. The $mn \times mn$ Gauss-Newton matrix can be approximated by a Kronecker factorization as

$$H^{(\mathrm{GN})} := \mathbb{E}[gg^\top] \approx \mathbb{E}[GG^\top] \otimes \mathbb{E}[G^\top G], \tag{26}$$

where $\otimes$ denote the Kronecker product. The eigenvalues and eigenvectors of the Kronecker product are the products and Kronecker products of the factors; hence the eigenbasis of $H^{(\mathrm{GN})} \in \mathbb{R}^{mn\otimes mn}$ can be approximated by computing the eigenbasis of the smaller $\mathbb{E}[GG^\top] \in \mathbb{R}^{m\times m}$, $\mathbb{E}[G^\top G] \in \mathbb{R}^{n\times n}$.

Is Kronecker approximation a good proxy for the full eigenbasis? Benzing (2022) showed that Kronecker-factored approximation such as KFAC (Martens & Grosse, 2015) can lead to better performance than using the full eigenbasis in some cases. They attributed the gain to heuristic damping, which effectively controls the step sizes and was beneficial in their experiments. Our experiments do not use such heuristic damping, and we find the Kronecker approximation to behave similarly to the full eigenbasis, while being much more compute-efficient.

C.2 Details for MLP experiments with squared loss

This section provides details for the MLP experiments in §5.

We learn all tasks with single-hidden-layer MLPs given by $\hat{y} = f(x;\theta) = a \cdot \sigma(w^\top x + b)$, where $w \in \mathbb{R}^{d\times m}, a, b \in \mathbb{R}^d$, with d and m being the input and hidden dimensions respectively. The non-linearity σ defaults to ReLU unless specified otherwise.

Below we provide detailed descriptions of the tasks:

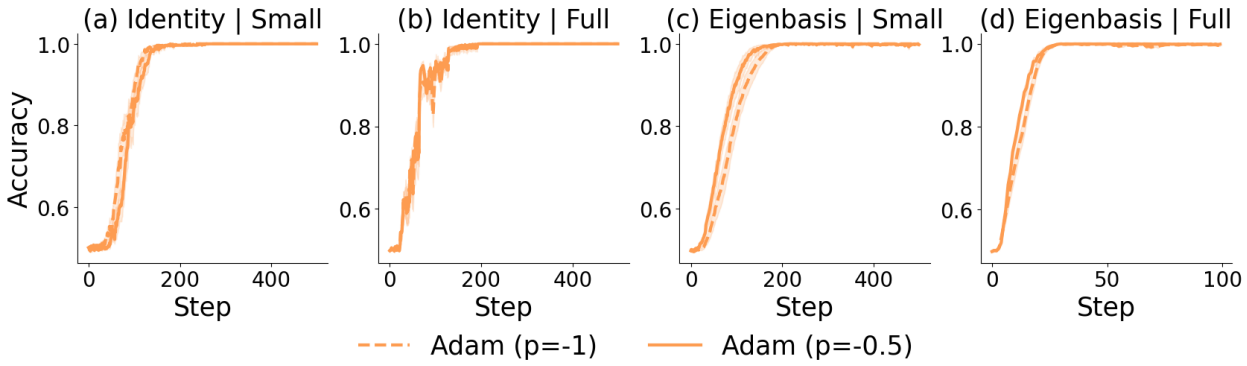


Figure 9: **Sparse parity**, comparing Adam, with power -1 or $-\frac{1}{2}$ for the full 2×2 grid (Table 1).

- **Learning from a random teacher network.** We first construct a teacher-student setting to evaluate our hypotheses on a non-convex example. Consider input vectors, $\mathbf{x}_i \sim \mathcal{N}(0, \Sigma_x) \in \mathbb{R}^d$ where $\Sigma_x \in \mathbb{R}^{d \times d}$ is a random covariance matrix. We initialize a random teacher model, $f_{\mathcal{T}} : \mathbb{R}^d \to \mathbb{R}$, as a single hidden layer MLP. For each input vector, we sample output labels from the teacher: $y_i = f(x_i; \theta_{\mathcal{T}}) = a_{\mathcal{T}} \cdot \sigma(w_{\mathcal{T}}^\top x_i + b_{\mathcal{T}})$, where, $w_{\mathcal{T}} \in \mathbb{R}^{m \times d}$; $a_{\mathcal{T}}, b_{\mathcal{T}} \in \mathbb{R}$, and $n_{\mathcal{T}}, h_{\mathcal{T}}$ are the teacher's input and hidden dimensions, respectively. The objective is to learn this (x, y) mapping using an identical student model which has a hidden dimension $d_{\mathcal{S}} = 2 \times d_{\mathcal{T}}$.
- **Feature learning with sparse parity.** Sparse parity is well-studied and widely adopted for understanding neural network optimization (Barak et al., 2022; Bhattamishra et al., 2023; Edelman et al., 2023; Morwani et al., 2024; Abbe et al., 2023b). It can be viewed as learning a sparse "feature" embedded in a much higher ambient dimension. Specifically, (d, k)-parity is a function from $\boldsymbol{x} \in \{\pm 1\}^d$ to $y = \prod_{i \in \mathcal{S}} x_i \in \{\pm 1\}$, where $\mathcal{S} = \{s_1, s_2, \ldots, s_k\} \subseteq [d]$ is the unknown support of relevant coordinates. In our experiments, we set $d = 20$, $k = 6$.
- **Feature learning with staircase.** We consider a multi-feature generalization of sparse parity called the staircase function (Abbe et al., 2022, 2023a). Given input $x \in \{\pm 1\}^d$, the label y is the sum of several parity functions, whose supports are specified by k segments. Specifically, $y = \sum_{(s_i, e_i) \in \mathcal{P}} \prod_{j=s_i}^{e_i - 1} x_j$, where $\mathcal{P} = \{(s_i, e_i)\}_{i \in [k]}$ are the start (inclusive) and stop (exclusive) indices of a segment. For our experiments, we set each segment to be of the same size and choose $d = 21, k = 3$, i.e., $\mathcal{P} = \{(0, 7), (7, 14), (14, 21)\}$, and $y \in \{-3, -1, 1, 3\}$.
- **CIFAR-10** (Krizhevsky et al., 2009). The input images are flattened to a length-3072 vector and the labels are treated as 10-dimensional one-hot vectors. We use 400 steps in all experiments, which is sufficient for large-batch eigenbasis experiments to reach around 47% accuracy, a reasonable performance for 2-layer MLPs.

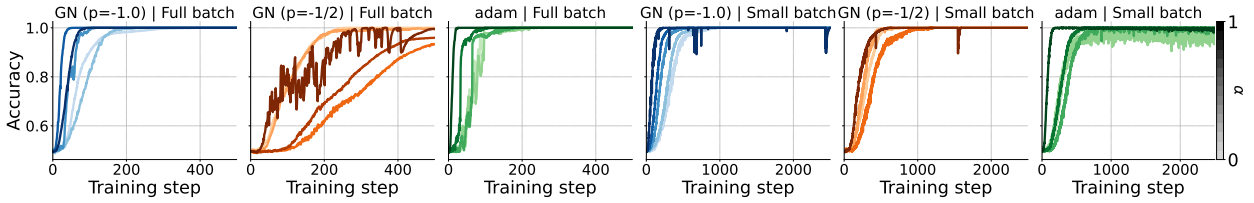
What about using power $p = -1$ for Adam? In §2, we introduced the power $p \in \{-\frac{1}{2}, -1\}$ as a hyperparameter for Gauss-Newton (GN) but kept the power Adam to be $-\frac{1}{2}$, following the standard definition of the Adam algorithm. For completeness, we experiment on Adam with $p = -1$ on sparse parity. Our results in Figure 9 is consistent with Lin et al. (2024), which finds that $p = -1$ shows comparable empirical performance to the standard choice of $p = -0.5$, especially under low-precision.

C.3 Interpolating between basis

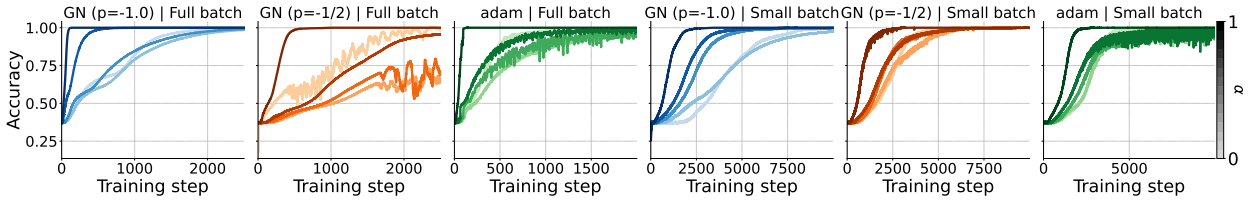
In §5, we discussed the behavior of GN and Adam on two kinds of basis: identity and full-GN, depicting an incorrect and a correct basis to precondition the gradient, respectively. To provide a more complete picture of the effect of basis, we provide results with more granularity with respect to the choice of basis.

Algorithm 2 Geodesic interpolation between bases

Input: full GN basis U, interpolation factor α.
  Compute the matrix log $K := \text{logm}(U)$.
  Compute the matrix exponent $\hat{U} := \exp(\alpha \cdot K)$.
  Obtain the real part $U_\alpha := \text{real}(\hat{U})$.
Output: U_α.



(a) (20, 6)-Sparse parity



(b) (21,3)-Staircase

Figure 10: **Basis interpolation**: Comparing GN^{-1}, $\text{GN}^{-1/2}$, and Adam under various bases, for parity and staircase (§5). Each basis is obtained by a geometric interpolation between the eigenbasis (darker colors) and the identity basis (lighter colors), parmaeterized by a factor $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$.

Particularly, we compare GN and Adam on bases of "intermediate" quality by interpolating between the identity and full-GN basis. Given the identity basis I and the eigenbasis U, we construct an interpolation U_γ, parameterized by some interpolation factor $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$, using geodesic interpolation (Algorithm 2). In particular, $U_0 = I$ and $U_1 = U$. Results are shown in Figure 10. In particular, Adam and $\text{GN}^{-\frac{1}{2}}$ behave similarly under the stochastic regime across basis choices, as predicted by the theory (§3.2).

C.4 Details for logistic experiments

Simulation for §4 We run simulation following the 2-layer linear network example in §4. The inputs are 2048-dimensional one-hot vectors following a power law decay, with $v_i := \mathbf{Pr}(x = e_i) \propto i^{-c}$. We set $c = 0.6$ in the experiments. The label distributions are set to $P_i = p(y = 1|x = e_i) = 0.75$ for all i.

Transformer experiments This section provides details for the Transformer experiments in §5. The attention module shares a similar structure as the logistic regression results in §4: the inner product of query and key matrices resembles the reparameterization, and the softmax function resembles the logistic function.

We consider a selection task motivated by the example in §4: The input is a sequence of T Gaussian vectors followed by a length-d one-hot vector ($d \geq T$) specifying which input is used in the regression task, i.e. $[x_1, \cdots, x_T, s]$, with the label given by $y = \langle \theta_*, x_i \rangle$ if $s = e_i$. In the experiments, we set $T = 32$ and use a 1-layer 1-head Transformer with dimension 128. Figure 7 shows results the comparison of GN and Adam under the Kronecker-approximated eigenbasis with large batches (batch size = 16384), where GN shows slower convergence, consistent with the theory. Results are aggregated over 10 seeds, and each run takes around 90min to complete.