

---

# ADAM OR GAUSS-NEWTON? A COMPARATIVE STUDY IN TERMS OF BASIS ALIGNMENT AND SGD NOISE

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Approximate second-order optimizers are increasingly showing promise in accelerating training of deep learning models, yet their practical performance depends critically on how preconditioning is applied. Two predominant approaches to preconditioning are based on (1) Adam, which leverages statistics of the current gradient, and (2) Gauss-Newton (GN) methods, which use approximations to the Fisher information matrix (often raised to a power). This work compares these approaches 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  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$  in full-batch settings. Conversely, in the stochastic regime, Adam behaves similarly to  $\text{GN}^{-1/2}$  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, 2014), RMSProp (Tieleman & Hinton, 2012), Adafactor (Shazeer & Stern, 2018), SignSGD (Bernstein et al., 2018), and Lion (Chen et al., 2023) all fall into this category. Recent work has shown that these optimizers perform comparably in practice (Zhao et al., 2024), prompting a natural question: what additional gains might be possible by incorporating second-order information into the diagonal preconditioner?

Second-order optimizers offer one answer to this question. Although directly using the inverse Hessian, as in Newton’s method, is typically impractical at scale, a variety of scaling-friendly approximations have been proposed that are both efficient and theoretically grounded (Martens & Grosse, 2015; Gupta et al., 2018; Liu et al., 2023; Vyas et al., 2024). These methods can be viewed as first rotating the gradient into a particular basis—often derived from an approximation to the Hessian or Fisher information matrix—and then applying a diagonal preconditioner in that rotated basis. For instance, Shampoo (Gupta et al., 2018) directly approximates the Gauss-Newton algorithm—often using an exponent on the approximated eigenvalues ranging from  $-1$  to  $-1/4$ —with a Kronecker-product structure that defines an efficient preconditioning basis (see (Morwani et al., 2024) for discussion). In contrast, SOAP (Vyas et al., 2024) can be interpreted as running Adam in a rotated basis, where the basis is potentially derived from a Kronecker-factored approximation to curvature.

This motivates a 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. 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. We formally define these choices in Section 2.

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 (Section 3) and logistic regression (Section 4), across the *eigen* and *identity* bases on full (population) and small (single-sample) batch.

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

A guiding question in our study is whether the empirical Fisher (used in Adam and SOAP) offers any advantage over GN-derived curvature estimates, or whether it merely provides a tractable proxy. Furthermore, since Adam effectively uses the *square root* of its second-moment estimate, we also investigate whether preconditioning with  $\text{GN}^{-1/2}$  is preferable to  $\text{GN}^{-1}$  under various conditions. Importantly, we aim to decouple the influence of the preconditioning basis from that of the diagonal approximation applied within it.

**Our Contributions.** We study how 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 (or more generally, bases misaligned with the eigenbasis but still satisfying structural properties such as a sparse plus low-rank structure). (2) Gradient noise: We analyze both full-batch (population gradient) and stochastic gradient (batch size 1) regimes to isolate how preconditioner behavior is influenced by gradient variance.

For linear regression, we obtain the following theoretical results:

- **Sensitivity to basis choice** (Section 3.1): It is well known that GN preconditioning in the eigenbasis yields optimal convergence rates for quadratics. However, when the basis is misaligned, Adam can outperform both  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$ , sometimes matching the speed of GN in its ideal basis.
- **Equivalence under noise** (Section 3.2): In the stochastic regime with Gaussian data, Adam behaves similarly to  $\text{GN}^{-1/2}$  regardless of basis, suggesting a surprising alignment between its empirical design and curvature-based preconditioning.

Further, in the case of logistic regression, we show Adam can even outperform  $\text{GN}^{-1}$  under the eigenbasis with full batch update (Section 4).

These results are summarized in a two-by-two grid in Table 1, and we further discuss the distinction between  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$  in Section 3.3. 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 on both simulations and more general problems (Section 5). The results align with the theory across all empirical settings, illustrating the practical implications of basis choice and gradient noise. We discuss related work in Appendix A.

## 2 PRELIMINARIES

Consider optimizing a function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  with input  $x$  and the (vectorized) parameter  $\theta \in \mathbb{R}^d$  against a loss function  $\ell$ . Updates performed by preconditioning optimizers can be seen as

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

where  $\eta$  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.

---

**Algorithm 1** Preconditioned optimizer

---

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

---

We will 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 of the Hessian:

$$H := \nabla_\theta^2 \ell = \nabla_{\theta f} \nabla_f^2 \ell \nabla_{\theta f}^\top + \nabla_f \ell \nabla_\theta^2 f := H^{(\text{GN})} + \nabla_f \ell \nabla_\theta^2 f, \quad (1)$$

Contrast to the Hessian  $H$ , the GN matrix  $H^{(\text{GN})}$  requires only first-order gradient information 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.** Our theory analyzes two basis choices: 1) the identity basis  $U = I$ , and 2) the eigenbasis of the Gauss-Newton matrix  $H^{(\text{GN})}$ ; for the mean-squared loss,  $H^{(\text{GN})} = \mathbb{E}_x[\nabla_\theta f(x) \nabla_\theta f(x)^\top]$ . 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  $\theta \in \mathbb{R}^{n \times m}$ , where  $H^{(\text{GN})} \in \mathbb{R}^{nm \times nm}$  is approximated by the outer product of two matrices of dimension  $\mathbb{R}^{n \times n}$  and  $\mathbb{R}^{m \times m}$ ; see Appendix D.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 consider two types of  $D$ : Adam and Gauss-Newton. For the theoretical part, we are going to consider the Adam’s preconditioner as described below:

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

where  $g(x)$  denotes the gradient from  $x$ , and 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 represents the expected per-sample gradient norm. 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

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

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

The preconditioners for Adam and Gauss-Newton 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 focuses on linear regression with the mean-squared loss, where we compare Adam and Gauss-Newton along both axes in Table 1: the quality of the basis choice, and the effect of batch size (gradient noise). 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  $\theta^*$  is the ground truth parameter. We consider Gaussian input  $x \sim \mathcal{N}(0, \Sigma_x)$ . 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.  $H^{(\text{GN})} = \mathbb{E}[\nabla_\theta f \nabla_\theta f^\top] = \Sigma_x$ .

#### 3.1 AXIS 1 – CHOICE OF BASIS

Deep learning optimization often encounters heterogeneous curvature, which 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., 2023). However, while  $\text{GN}^{-1}$  is known to achieve optimal convergence under the correct eigenbasis (Section 3.1.1), we show in Section 3.1.2 that  $\text{GN}^{-1}$  can be severely suboptimal under a poorly chosen basis that does not reflect the true curvature. In contrast, Adam has an “auto-tuning” effect that allows it to outperform both  $\text{GN}^{-1}$  and  $\text{GN}^{-\frac{1}{2}}$ . We support this with a constructed example.

##### 3.1.1 $\text{GN}^{-1}$ IS OPTIMAL UNDER THE CORRECT BASIS

The basis given by the eigenvectors of  $H^{(\text{GN})}$  can be considered as the “correct” basis. Under this basis, it is well-known that  $\text{GN}^{-1}$  achieves the optimal convergence rate in both full-batch and stochastic setting<sup>1</sup>: when using the full batch,  $\text{GN}^{-1}$  converges in 1 step; for the stochastic setting,  $\text{GN}^{-1}$  decreases the loss at a linear rate. Details are included in Appendix B.1 for completeness.

##### 3.1.2 INCORRECT BASIS: ADAM AUTO-TUNES TO THE CURVATURE

The previous section shows that GN is optimal under the ideal eigenbasis. What if the basis is not estimated correctly? 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 a covariance

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

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 the maximum learning rate affordable is 2.

The wrong basis choice we consider is the identity basis. We will see that this makes GN fail to adapt to the curvature of the problem, while Adam remains efficient.

**GN converges slowly.** When taking  $U = I$ , the diagonal preconditioner for GN has  $D_{ii}^{(\text{GN})} = 1^p = 1$ . That is, both  $\text{GN}^{-1}$  and  $\text{GN}^{-\frac{1}{2}}$  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, Adam effectively acts as normalized gradient descent for the first block, and acts as signed gradient descent in each coordinate in the second block. As we train using Adam with a constant learning rate  $\eta$ , let  $\|g_0^{(t)}\|$  represent the gradient norm for the first block coordinates, and  $|g_i^{(t)}|$  for  $i \in [d]$  represent the per-coordinate gradient norm for the coordinates in the second block which evolves independently.

<sup>1</sup>By stochastic setting we refer to updates where each batch contains a single sample.

Recall that for quadratic problem, the gradient is  $g = \Sigma_x \Delta$  where  $\Delta := \theta - \theta^*$ , and  $\Delta^{(t+1)} = (\mathbf{I} - \eta \Sigma_x) \Delta^{(t)}$  for vanilla gradient descent; 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 essentially, 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, where 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, and reach a target error within log number of steps. We note that 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  $\text{GN}^{-1/2}$  at small batch sizes, as we will show in Section 3.2, and does not exhibit this autotuning effect when the gradient variance dominates.<sup>2</sup> 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.

### 3.2 AXIS 2 – GRADIENT NOISE AND BATCH SIZE

The previous section shows an example where Adam outperforms GN when the updates are using population gradients but a poor basis choice. In this section, we focus on the stochastic regime 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 Appendix B.2.

We first prove a stronger result, showing that for Gaussian input distribution and quadratic loss, empirical Fisher is approximately equal 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 can utilize the lemma above to show that  $\text{GN}^{-\frac{1}{2}}$  and Adam behave the same upto a scalar constant in any basis for our theoretical setup at batch size 1.

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

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

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 if  $\text{GN}^{-1}$  performs better than  $\text{GN}^{-1/2}$  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  $\mathbf{A}(P) := P^{1/2} \Sigma_x P^{1/2}$ , we have that:

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

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

<sup>2</sup>Recall the definition of  $D_{ii}^{(\text{A})}$  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  $(D_{ii}^{(\text{A})})^2 = \mu_i^2 + \frac{1}{B} \sigma_i^2$ .  $(D_{ii}^{(\text{A})})^2$  is dominated by the mean for full-batch updates ( $B \rightarrow \infty$ ), and is often dominated by the variance in the stochastic regime ( $B = 1$ ).

270 Defining a version of the condition number  $\kappa_s(\mathbf{A}(P)) = \frac{\text{Tr}(\mathbf{A}(P))}{\lambda_{\min}(\mathbf{A}(P))} = \sum_{i \in d} \frac{\lambda_i(\mathbf{A}(P))}{\lambda_{\min}(\mathbf{A}(P))} \geq d$ . We  
 271 can see that the convergence rate depends on  $\kappa_s(\mathbf{A}(P))$ .  
 272

273 The above bound shows that in the correct basis,  $\text{GN}^{-1}$  is the optimal preconditioner even in the  
 274 stochastic regime, as it minimizes the condition number to  $\kappa_s(\mathbf{A}(P)) = \kappa_s(\mathbf{I}) = d$ . For  $\text{GN}^{-1/2}$   
 275 in the correct basis, we have  $\kappa_s(\mathbf{A}(P)) = \kappa_s((\Sigma_x)^{1/2}) = \sum_i \sqrt{\frac{\lambda_i(\Sigma_x)}{\lambda_{\min}(\Sigma_x)}}$ , which is greater than  $d$   
 276 unless  $\lambda_{\max}(\Sigma_x) = \lambda_{\min}(\Sigma_x)$ .  
 277

### 279 3.3 WHICH POWER TO USE FOR GAUSS-NEWTON? 280

281 Newton’s method was originally proposed with a preconditioner closer to  $\text{GN}^{-1}$ . However, the  
 282 square root in the denominator of Adam (Kingma & Ba, 2014) and Adagrad (Duchi et al., 2011) has  
 283 spurred multiple papers (Liu et al., 2023; Lin et al., 2024; Vyas et al., 2024) questioning the correct  
 284 power of the preconditioner to be used in practice. As shown before, for the quadratic model, in  
 285 the correct basis,  $\text{GN}^{-1}$  is optimal. However, under the incorrect identity basis, we claim that there  
 286 exists problems for which, even in the full batch case,  $p = 0.5$  leads to faster convergence than  $p = 1$ .  
 287 The key idea is that the convergence rate of preconditioned gradient descent (Boyd & Vandenberghe,  
 288 2004) depends on the condition number of the preconditioned Hessian. It then suffices to construct  
 289 examples where the condition number is better behaved for  $p = 0.5$  than  $p = 1$ . We provide details  
 290 in Appendix B.3 and accompanying simulation results in Section 5.  
 291

## 292 4 THEORETICAL ANALYSIS ON LOGISTIC REGRESSION 293

294 For quadratics, Section 3.1.1 shows that  $\text{GN}^{-1}$  is optimal under the eigenbasis. However, this needs  
 295 not hold in general. In this section, we provide an example with logistic regression, where Adam  
 296 converges faster than  $\text{GN}^{-1}$  even under the eigenbasis with full batches.  
 297

298 **Setup.** Since we are operating under the eigenbasis, the updates can be considered per dimension.  
 299 We hence take the input  $x$  from the set of  $d$ -dimensional one-hot vectors  $\{e_i\}_{i=1}^d$ , with probability  
 300  $\nu_i := \Pr(x = e_i)$ . Conditional on  $x = e_i$ , the label  $y$  is Bernoulli with mean  $P_i := \Pr(y = 1 | x =$   
 301  $e_i)$ . We assume  $0.6 \leq P_i \leq 0.8, \forall i \in [d]$ , i.e. the labels are neither deterministic nor fully random,  
 302 and the optimal parameter has a bounded norm.

303 We optimize a *two-layer linear network*  $q : \mathbb{R}^d \rightarrow \mathbb{R}^d$ , whose output depends on the *squares* of the  
 304 weights: for any  $\theta \in \mathbb{R}^d$  and for  $i \in [d]$ , we define the model’s prediction as  
 305

$$306 q_i(\theta) = \Pr(y = 1 | x = e_i) = \sigma\left(\sum_{j=1}^d \theta_j^2 x_j\right) = \sigma(\theta_i^2), \quad \sigma(z) = \frac{1}{1 + \exp(-z)}. \quad (5)$$

309 The square parameterization makes the problem non-convex, and is analogous to the structure of  
 310 key-query multiplication in self-attention (Vaswani et al., 2017).  
 311

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

317 **Adam converges in  $O(\log(1/\epsilon))$  steps.** Adam effectively performs sign GD, hence the amount of  
 318 parameter update is determined by the step size. The  $O(\log(1/\epsilon))$  convergence hence follows directly  
 319 from starting with a  $O(1)$  learning rate and halving the step size every  $O(1)$  steps.  
 320

321  **$\text{GN}^{-1}$  requires  $\tilde{\Omega}(d^\delta \log(1/\epsilon))$  steps.** In contrast to Adam,  $\text{GN}$ ’s preconditioning can result in an  
 322 unboundedly large update (see Equation (21)), unless the learning rate is kept small, which in turn  
 323 leads to slow convergence. To show the lower bound, we first state a more general convergence  
 result. Recall that the  $\text{GN}^{-1}$  update is  $\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)}(H^{(\text{GN})}(\theta) + \alpha I)^{-1}g^{(t)}$ . We assume that

324  $\{\eta^{(t)}\}_{t \geq 0}$  are non-increasing and that the step size schedule and regularization lead to convergence,  
 325 i.e.  $\theta^{(t)} \rightarrow \theta^*$  as  $t \rightarrow \infty$ . Linearizing the update map at the limit  $\theta^*$  gives  
 326

$$327 \quad \theta^{(t+1)} - \theta^* = \left( I - \eta^\infty (H_*^{(\text{GN})} + \alpha I)^{-1} H_*^{(\text{GN})} \right) (\theta^{(t)} - \theta^*) + O(\|\theta^{(t)} - \theta^*\|^2),$$

328  
 329 where  $H_*^{(\text{GN})} := H^{(\text{GN})}(\theta^*)$  and  $\eta^\infty = \lim_{t \rightarrow \infty} \eta_t$ . We are interested in lower bounding the spectral  
 330 radius of the local iteration matrix:  
 331

$$332 \quad \gamma(\eta^{(\infty)}, \alpha) := \left\| I - \eta^{(\infty)} (H_*^{(\text{GN})} + \alpha I)^{-1} H_*^{(\text{GN})} \right\|_2,$$

333 as it governs the ultimate *local* rate of convergence that any GN schedule can achieve. We refer to  
 334  $\gamma(\eta^{(\infty)}, \alpha)$  as the local contraction factor; a value of  $\gamma$  close to 1 implies slow convergence.  
 335

336 The main result of this section is a lower bound on  $\gamma$ :  
 337

338 **Theorem 2.** *Suppose the weights are initialized at  $\theta^{(0)} = \frac{1}{\sqrt{d}} \cdot \vec{1}$ . Consider any non-increasing step*  
 339 *size sequence  $\{\eta^{(t)}\}_{t \geq 0}$  and regularization parameter  $\alpha \geq 0$ . If the Gauss-Newton iterates converge*  
 340 *to  $\theta^*$ , i.e.  $\theta^{(t)} \rightarrow \theta^*$ , then the local contraction factor  $\gamma(\eta^{(\infty)}, \alpha)$  is lower bounded by:*  
 341

$$342 \quad \gamma(\eta^{(\infty)}, \alpha) \geq 1 - c\sqrt{\log d} \max \left\{ \frac{1}{\sqrt{d}}, \sqrt{d/\kappa(\nu)} \right\},$$

343  
 344 for some universal constant  $c$ .  
 345

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

350 **Corollary 3.** *For imbalanced input with  $\kappa(\nu) = \Omega(d^{1/2+\delta})$  for some  $\delta \in [0, 1/2]$ ,  $\text{GN}^{-1}$  requires*  
 351  *$t = \tilde{\Omega}(d^\delta \log(1/\epsilon))$  steps to reach a parameter satisfying  $\|\theta^{(t)} - \theta^*\|_2 \leq \epsilon$ .*  
 352

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

356 *Remark:* Ideally, in a purely local setting, one could choose  $\alpha = 0$  and use a constant final stepsize  
 357  $\eta^\infty$ . For instance, setting  $\eta^\infty = 1$  would make the iteration matrix zero, yielding  $\gamma = 0$  and  
 358 superlinear convergence. In fact, any other constant  $\eta^\infty \in (0, 2)$  would similarly provide rapid linear  
 359 convergence with a rate independent of the condition number. However, Theorem 2 shows this ideal  
 360 scenario is not possible. As the proof (Appendix C) shows, the requirement of ensuring convergence  
 361 from a specific, natural initialization forces the algorithm’s final step size,  $\eta^{(\infty)}$ , to be small. This  
 362 constraint directly degrades the local contraction factor, preventing the rapid convergence one might  
 363 expect from a Newton-like method. Further, we note that our result does not contradict with the fast  
 364 convergence from line search, which does not fall under the non-increasing step size assumption.  
 365

## 366 5 EXPERIMENTS

367  
 368 In this section, we provide experimental evidence to the theoretical claims made in the previous  
 369 section. Our experiments are broadly divided into two categories: (i) simulations for examples in  
 370 Section 3 and Section 4, (ii) non-convex examples with MLP, and (iii) Transformer experiments.  
 371

372 **Experiment details.** We use the mean square error as the objective function unless otherwise  
 373 specified. For numerical stability, we optionally regularize  $D$  to be  $D + \alpha I$  for some small  $\alpha > 0$ . We  
 374 sweep over the learning rate  $\eta$  and the regularization coefficient  $\alpha$ . For Adam, we also sweep over the  
 375 learning rate schedule (constant or step decay) and  $\beta_2$ ; we fix  $\beta_1 = 0$ , similar to Das et al. (2024). We  
 376 use different samples for estimating gradient and Gauss-Newton. Due to computational considerations,  
 377 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 Appendix D.1.

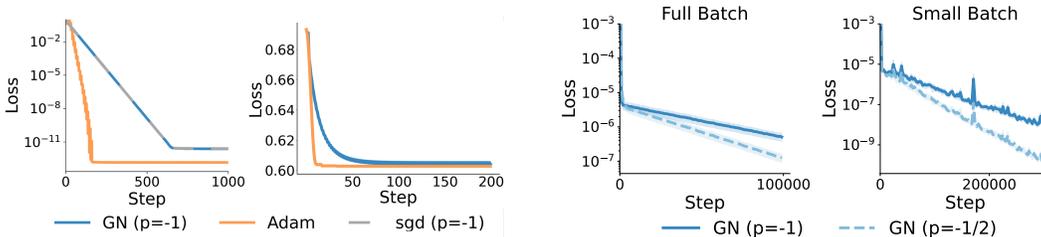


Figure 1: **Simulation results.** (Left) Adam converges faster than GN with full batches 1) under the identity basis on a linear regression task with block-wise covariance (left subplot), where GN fails to adapt to the problem curvature and behaves identically to SGD; and 2) under the eigenbasis, for the reparameterized logistic regression task (right subplot). (Right) Comparing GN power  $p \in \{-\frac{1}{2}, -1\}$ . On a regression task where  $\text{GN}^{-\frac{1}{2}}$  leads to a more favorable condition number,  $\text{GN}^{-\frac{1}{2}}$  converges faster than  $\text{GN}^{-1}$  with both small and large batches.

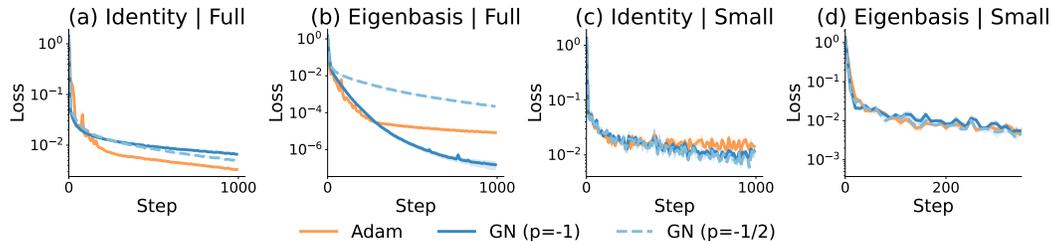


Figure 2: **Learning from a random teacher**, comparing Adam,  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$  for the full  $2 \times 2$  grid (Table 1).

**Simulations** This section provides simulation results on the examples in Section 3 and Section 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 left). For the identity basis, we consider a 100-dimensional linear regression task (Section 3.1.2) where the covariance matrix has a block-wise structure. Adam converges quickly due to the auto-tuning effect, whereas Gauss-Newton converge as slowly as vanilla gradient descent. For logistic regression (Section 4), our results on a 2048-dimensional problem (detailed in Appendix D.4) show that Adam converges faster even under the eigenbasis.
- *Comparing powers of Gauss-Newton.* Section 3.3 shows that the comparison of  $\text{GN}^{-1}$  and  $\text{GN}^{-\frac{1}{2}}$  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  $\Sigma_x$ . Figure 1 (right) shows the simulation results in this case, where  $\text{GN}^{-\frac{1}{2}}$  converges faster than  $\text{GN}^{-1}$  when using both large and small batches. Details are provided in Appendix B.3.

**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; Appendix D.2 provides details. Experiments on these tasks cover the full  $2 \times 2$  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 (Appendix D.1) while being more compute efficient. We provide additional experiments on intermediate basis choices by interpolating between the identity and the eigenbasis in Appendix D.3.

As shown in Figure 2–4, our theoretical analyses empirically extend to these four non-convex tasks across both axes of interest. In particular, across all problems considered, Adam and  $\text{GN}^{-1/2}$  closely track one another at small batch sizes, regardless of the basis chosen (subfigures (c), (d)). Moreover, Adam is close to or better than  $\text{GN}^{-1}$  when the basis is incorrect (subfigures (a), (c)).

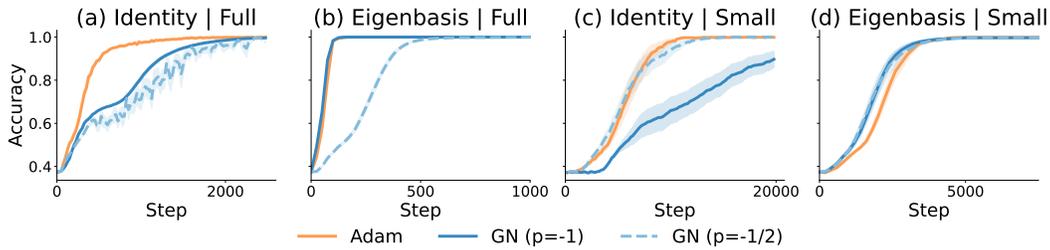


Figure 3: **Staircase**, comparing Adam,  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$  for the full  $2 \times 2$  grid (Table 1). Staircase is a generalization of sparse parity (Figure 7).

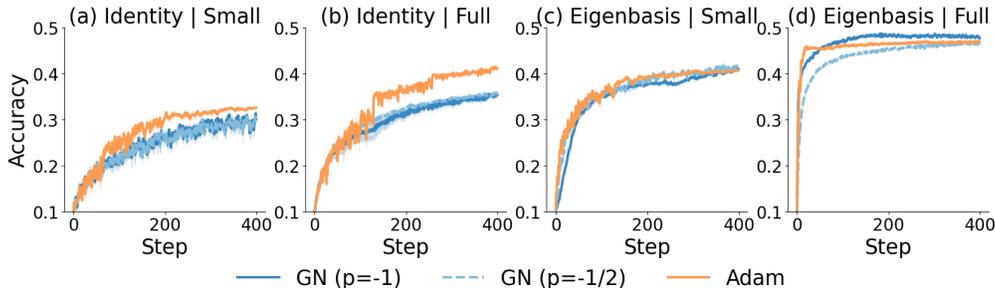


Figure 4: **CIFAR10**, comparing Adam,  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$  for the full  $2 \times 2$  grid (Table 1).

**A logistic-like task with Transformers** We consider a selection-based regression task learned with a 1-layer Transformer, whose attention module resembles the structure of the reparametrized logistic regression in Section 4 (detailed in Appendix D.4). Results in Figure 5 are consistent with our theory, where Adam outperforms GN under the eigenbasis with full batch updates.

## 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  $\text{GN}^{-1/2}$  exhibit similar behavior for linear regression regardless of the basis choice, thereby revealing a connection between Adam’s design and curvature-based preconditioning.

It is important to understand whether phenomena and differences observed in small-scale, synthetic setups persist across scale. Our empirical results on simulation, synthetic, and image datasets are consistent with the theoretical results. In particular, all MLP experiments align with findings for linear regression, and the Transformer experiments align with our logistic regression results. [A more thorough study of implications on larger-scale neural networks is an interesting next step.](#) For instance, it has been observed that layer-based block-wise approximation of the full Gauss-Newton matrix can lead to similar or sometimes superior performance [Benzing \(2022\)](#); [Zhang et al. \(2025\)](#); [Abreu et al. \(2025\)](#). Moreover, we hypothesize that the equivalence between Adam and  $\text{GN}^{-1/2}$  in the stochastic regime extends to practical, large-scale training. In particular, as training progresses, the gradient variance

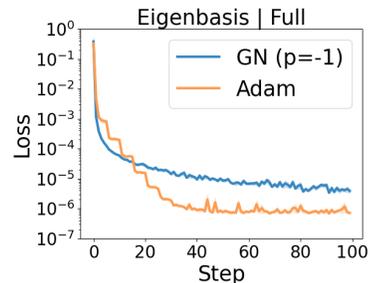


Figure 5: Transformer experiments (Section 5): Adam outperforms  $\text{GN}^{-1}$  under the eigenbasis with full batches.

---

486 tends to dominate over the gradient mean, mirroring the stochastic regime in which variance drives the  
487 dynamics. Validating this hypothesis in large-scale settings is an interesting direction for future work.  
488 Finally, such equivalence combined with the benefits of Adam’s *auto-tuning* at large-batch regimes  
489 suggests a promising direction: developing algorithms that exhibit similar desirable auto-tuning  
490 behavior even when operating with small batches.

491  
492  
493

## REFERENCES

- 494 E. Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. Sgd learning on neural networks: leap  
495 complexity and saddle-to-saddle dynamics. *Annual Conference Computational Learning Theory*,  
496 2023. doi: 10.48550/arXiv.2302.11055.
- 497 Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. The merged-staircase property: a  
498 necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural  
499 networks. *arXiv preprint arXiv: 2202.08658*, 2022.
- 500 Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. Generalization on the unseen, logic  
501 reasoning and degree curriculum. *Journal of Machine Learning Research*, 25(331):1–58, 2024.
- 502  
503 Natalie Abreu, Nikhil Vyas, Sham Kakade, and Depen Morwani. The potential of second-order  
504 optimization for llms: A study with full gauss-newton. *arXiv preprint arXiv: 2510.09378*, 2025.
- 505  
506 Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear  
507 attention is (maybe) all you need (to understand transformer optimization). In *The Twelfth Inter-*  
508 *national Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.  
509 OpenReview.net, 2024. URL <https://openreview.net/forum?id=0uI5415ry7>.
- 510  
511 B. Barak, Benjamin L. Edelman, Surbhi Goel, S. Kakade, Eran Malach, and Cyril Zhang. Hidden  
512 progress in deep learning: Sgd learns parities near the computational limit. *Neural Information*  
513 *Processing Systems*, 2022. doi: 10.48550/arXiv.2207.08799.
- 514 Frederik Benzing. Gradient descent on neurons and its link to approximate second-order optimization.  
515 *arXiv preprint arXiv: 2201.12250*, 2022.
- 516  
517 Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd:  
518 Compressed optimisation for non-convex problems. *arXiv preprint arXiv: 1802.04434*, 2018.
- 519  
520 S. Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers and  
521 their ability to learn sparse boolean functions. *Annual Meeting of the Association for Computational*  
522 *Linguistics*, 2022. doi: 10.48550/arXiv.2211.12316.
- 523  
524 Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- 525 Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham,  
526 Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of  
527 optimization algorithms. *Neural Information Processing Systems*, 2023. doi: 10.48550/arXiv.2302.  
528 06675.
- 529 Rudrajit Das, Naman Agarwal, Sujay Sanghavi, and Inderjit S. Dhillon. Towards quantifying the  
530 preconditioning effect of adam. *arXiv preprint arXiv: 2402.07114*, 2024.
- 531  
532 John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and  
533 stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL  
534 <http://jmlr.org/papers/v12/duchilla.html>.
- 535  
536 Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Pareto frontiers in  
537 neural feature learning: Data, compute, width, and luck. *arXiv preprint arXiv: 2309.03800*, 2023.
- 538  
539 Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast  
approximate natural gradient descent in a kronecker-factored eigenbasis. *arXiv preprint arXiv:*  
*1806.03884*, 2018.

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

545 Vineet Gupta, Tomer Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization.  
546 *International Conference on Machine Learning*, 2018.

547 Kaiqi Jiang, Dhruv Malik, and Yuanzhi Li. How does adaptive optimization impact local neural  
548 network geometry? *Advances in Neural Information Processing Systems*, 36:8305–8384, 2023.

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

552 Jean Kaddour, Oscar Key, Piotr Nawrot, Pasquale Minervini, and Matt J. Kusner. No train no  
553 gain: Revisiting efficient training algorithms for transformer-based language models. In Alice  
554 Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine  
555 (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural*  
556 *Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 -*  
557 *16, 2023*, 2023. URL [http://papers.nips.cc/paper\\_files/paper/2023/hash/](http://papers.nips.cc/paper_files/paper/2023/hash/51f3d6252706100325ddc435ba0ade0e-Abstract-Conference.html)  
558 [51f3d6252706100325ddc435ba0ade0e-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/51f3d6252706100325ddc435ba0ade0e-Abstract-Conference.html).

559 Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*  
560 *arXiv:1412.6980*, 2014.

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

563 Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher ap-  
564 proximation for natural gradient descent. In Hanna M. Wallach, Hugo Larochelle, Alina  
565 Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in*  
566 *Neural Information Processing Systems 32: Annual Conference on Neural Information Pro-*  
567 *cessing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp.  
568 4158–4169, 2019. URL [https://proceedings.neurips.cc/paper/2019/hash/](https://proceedings.neurips.cc/paper/2019/hash/46a558d97954d0692411c861cf78ef79-Abstract.html)  
569 [46a558d97954d0692411c861cf78ef79-Abstract.html](https://proceedings.neurips.cc/paper/2019/hash/46a558d97954d0692411c861cf78ef79-Abstract.html).

570 Frederik Kunstner, Jacques Chen, J. Lavington, and Mark W. Schmidt. Noise is not the main factor  
571 behind the gap between sgd and adam on transformers, but sign descent might be. *International*  
572 *Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2304.13960.

573 Wu Lin, Felix Dangel, Runa Eschenhagen, Juhan Bae, Richard E. Turner, and Alireza Makhzani. Can  
574 we remove the square-root in adaptive gradient methods? A second-order perspective. In Ruslan  
575 Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and  
576 Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*,  
577 volume 235 of *Proceedings of Machine Learning Research*, pp. 29949–29973. PMLR, 21–27 Jul  
578 2024. URL <https://proceedings.mlr.press/v235/lin24e.html>.

579 Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic  
580 second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

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

584 James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:*  
585 *1412.1193*, 2014.

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

---

594 Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Rosie Zhao, and S. Kakade. Feature  
595 emergence via margin maximization: case studies in algebraic tasks. *International Conference on*  
596 *Learning Representations*, 2023. doi: 10.48550/arXiv.2311.07568.

597  
598 Depen Morwani, Itai Shapira, Nikhil Vyas, Eran Malach, Sham Kakade, and Lucas Janson. A new  
599 perspective on shampoo’s preconditioner. *arXiv preprint arXiv:2406.17748*, 2024.

600  
601 Francesco Orabona. Normalized gradients for all, 2023. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2308.05621)  
602 [2308.05621](https://arxiv.org/abs/2308.05621).

603  
604 Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than sgd for transformers.  
*arXiv preprint arXiv: 2306.00204*, 2023.

605  
606 Vincent Roulet and Atish Agarwala. Per-example gradients: a new frontier for understanding and  
607 improving optimizers. *arXiv preprint arXiv: 2510.00236*, 2025.

608  
609 Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity  
and beyond. *arXiv preprint arXiv: 1611.07476*, 2016.

610  
611 Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N. Balasubramanian. A deeper  
612 look at the hessian eigenspectrum of deep neural networks and its applications to regularization.  
613 In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference*  
614 *on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on*  
615 *Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp.  
616 9481–9488. AAAI Press, 2021. doi: 10.1609/AAAI.V35I11.17142. URL [https://doi.org/](https://doi.org/10.1609/aaai.v35i11.17142)  
[10.1609/aaai.v35i11.17142](https://doi.org/10.1609/aaai.v35i11.17142).

617  
618 Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost.  
619 In *International Conference on Machine Learning*, pp. 4596–4604. PMLR, 2018.

620  
621 Tijmen Tieleman and Geoffrey E. Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for  
622 machine learning. Technical report, University of Toronto, 2012. Available at: [https://www.](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.5.pdf)  
[cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.5.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.5.pdf).

623  
624 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz  
625 Kaiser, and Illia Polosukhin. Attention is all you need. *NEURIPS*, 2017.

626  
627 Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas  
628 Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint*  
*arXiv: 2409.11321*, 2024.

629  
630 Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. Pyhessian: Neural networks  
631 through the lens of the hessian. *arXiv preprint arXiv: 1912.07145*, 2019.

632  
633 Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv  
634 Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural*  
*Information Processing Systems*, 33:15383–15393, 2020.

635  
636 Thomas T. Zhang, Behrad Moniri, Ansh Nagwekar, Faraz Rahman, Anton Xue, Hamed Hassani, and  
637 Nikolai Matni. On the concurrence of layer-wise preconditioning methods and provable feature  
638 learning. *arXiv preprint arXiv: 2502.01763*, 2025.

639  
640 Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing  
641 what makes a good optimizer for language models. *arXiv preprint arXiv: 2407.07972*, 2024.

642  
643  
644  
645  
646  
647

---

648 **The Use of Large Language Models** In this work, LLMs are used for rephrasing and polishing the  
649 writing, assisting with basic algebraic manipulation in the proof, and assisting with the plotting code.  
650

## 651 A RELATED WORK

652  
653 **Approximate second-order optimizers** Recent advances on approximate second-order methods  
654 have demonstrated success in large-scale settings, serving as efficient alternatives to classic second-  
655 order methods such as Newton’s and natural gradient descent, which are computationally bottle-  
656 necked to scale to high dimensions. While first-order diagonal preconditioners are shown to be  
657 comparable in practice Kaddour et al. (2023); Zhao et al. (2024), leveraging second-order information  
658 has proven effective. Most relevant to our work are methods that can be considered as applying  
659 diagonal preconditioners in a chosen basis (Gupta et al., 2018; Liu et al., 2023; Vyas et al., 2024;  
660 Jordan et al., 2024). However, there is no clear understanding how the preconditioner and the basis  
661 interact. For instance, Sophia (Liu et al., 2023) can be considered as applying  $\text{GN}^{-1}$  in the identity  
662 basis, which as we will show, is not always desired. In contrast, our work provides a clarifying  
663 decomposition of the design space of these second-order methods by separating the choice of basis in  
664 which to perform a diagonal preconditioner, from the choice of the diagonal preconditioner itself.  
665

666 For results in the stochastic regime, Martens (2014) provided results the stochastic case and depends  
667 on the condition number, similar to Lemma 2. However, their setting crucially differs from ours by  
668 assuming that the covariance of the gradients is independent of the current iterate.

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

679  
680 **Adam vs (S)GD** There have been a lot of interest understanding the comparison of Adam and  
681 (stochastic) gradient descent. Related to the preconditioning perspective in our work, Das et al.  
682 (2024) studies Adam’s preconditioning effect on quadratics, and shows that it outperforms SGD  
683 when the Hessian is sufficiently ill-conditioned. A line work focuses the comparison on optimizing  
684 Transformers, which has investigated through the lens of gradient noises (Zhang et al., 2020), relation  
685 to sign descent (Kunstner et al., 2023), and the curvature of the landscape (Jiang et al., 2023; Pan  
686 & Li, 2023); Ahn et al. (2024) provides a review and a theory-friendly abstraction. Most related to  
687 our work is Maes et al. (2024), which shows that Adam’s advantage over SGD for Transformers rely  
688 crucially on the choice of basis. While these results hinge on properties specific to Transformers, we  
689 are interested in understanding of algorithm design with insights that can be generally applicable.

## 690 B THEORETICAL RESULTS AND OMITTED PROOFS

### 691 B.1 OPTIMALITY OF $\text{GN}^{-1}$ IN THE CORRECT BASIS

692  
693 For completeness, we provide proofs for the optimality of  $\text{GN}^{-1}$  for the quadratic loss under the  
694 correct eigenbasis.  
695

696  
697 **Full batch** For  $\text{GN}^{-1}$ , the parameter estimation error evolves as

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

700 Hence  $\text{GN}^{-1}$  can reach the optimum in 1 step with  $\eta = 1$ .  
701

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

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

Multiplying by  $\Sigma_x$  and taking the trace leads to:

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

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

## B.2 EQUIVALENCE OF ADAM AND $\text{GN}^{-0.5}$ 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  $\text{GN}^{-\frac{1}{2}}$ 's updates.

### B.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  $\theta$ , let  $M := (\theta - \theta^*)(\theta - \theta^*)^\top$ . Then w.r.t.  $\theta$ , we have

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

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

### B.2.2 PROOF OF COROLLARY 1: EQUIVALENCE OF ADAM AND $\text{GN}^{-\frac{1}{2}}$

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

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

*Proof.* Adam's preconditioner is based on

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

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

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

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

$\square$

**Part 2: when  $H^{(\text{GN})}$  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.

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

The diagonal preconditioner for Gauss-Newton has

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

### B.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:*

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

*Proof.* Let's define

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

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

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

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

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

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

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

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

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

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

We have:

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

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

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

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

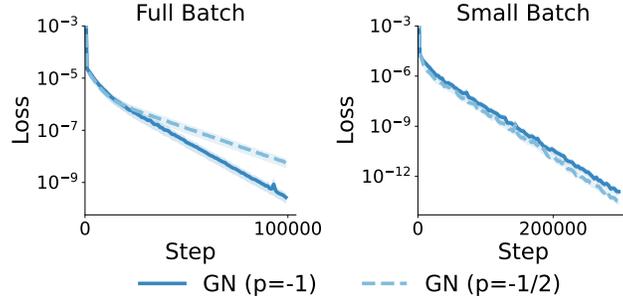


Figure 6: Comparing GN power  $p \in \{-\frac{1}{2}, -1\}$ . Contrary to Section 5, when  $\text{GN}^{-1}$  has a more favorable condition number, it converges faster or close to  $\text{GN}^{-\frac{1}{2}}$  on both small and large batches.

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

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

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

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

□

### B.3 COMPARING GN POWERS

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

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

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

By Lemma 2, to compare these two powers, it suffices to compare the condition number of for specific preconditioners  $\Sigma_x^{-1}$  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  $\Sigma_x$  such that  $\kappa(\mathbf{A}(\text{diag}(\Sigma_x)^{-1/2})) < \kappa(\mathbf{A}(\text{diag}(\Sigma_x)^{-1}))$ .*

Denote  $r(\Sigma_x) := \frac{\kappa(\Sigma_x^{1/2} \text{diag}(\Sigma_x^{-1}) \Sigma_x^{1/2})}{\kappa(\Sigma_x^{1/2} \text{diag}(\Sigma_x^{-1/2}) \Sigma_x^{1/2})}$ . We empirically show that there exists  $\Sigma_x$  such that  $r(\Sigma_x) > 1$ . We obtain such  $\Sigma_x$  by fixing the diagonal matrix of eigenvalues  $\Lambda$  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.  $\text{GN}^{-1/2}$  is more favorable), and  $r(\Sigma_{\text{one}}) < 1$  (i.e.  $\text{GN}^{-1}$  is more favorable). As shown in Figure 1

(right) and Figure 6,  $\text{GN}^{-\frac{1}{2}}$  indeed converges faster on data from  $\Sigma_{\text{half}}$ , whereas  $\text{GN}^{-1}$  converges faster with  $\Sigma_{\text{one}}$ , consistent with the theory.

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

## C PROOF FOR THE LOGISTIC EXAMPLE

For logistic regression, recall that the gradient  $g(\theta)$  and the diagonal Gauss–Newton matrix  $H^{(\text{GN})}(\theta)$  are given by

$$[g(\theta)]_i = 2\nu_i \theta_i (\sigma(\theta_i^2) - P_i), \quad [H^{(\text{GN})}(\theta)]_{ii} = 4\nu_i \theta_i^2 \sigma(\theta_i^2) (1 - \sigma(\theta_i^2)). \quad (21)$$

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

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

Coordinate-wise, this update reads

$$[M_{\eta, \alpha}(\theta)]_i = \theta_i - \frac{2\theta_i \nu_i (\sigma(\theta_i^2) - P_i)}{4(\theta_i^2)^2 \nu_i \sigma(\theta_i^2) (1 - \sigma(\theta_i^2)) + \alpha}. \quad (22)$$

### C.1 PROOF FOR THEOREM 2

The proof strategy 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  $\alpha$ . 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,  $\eta^\infty$ . Substituting this necessary restriction into the definition of the local contraction factor directly yields the theorem’s lower bound, 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  $M_{\eta, \alpha} : \mathbb{R} \rightarrow \mathbb{R}$  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  $\sigma((\theta^{(0)})^2) \leq 0.55$ , if the learning rate sequence satisfies

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

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

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

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

918 follows:  
919

$$\begin{aligned}
920 \quad \theta^{(1)} &= \theta^{(0)} - \eta^{(0)} \frac{2\theta^{(0)}(\sigma((\theta^{(0)})^2) - P)}{4(\theta^{(0)})^2\sigma((\theta^{(0)})^2)(1 - \sigma((\theta^{(0)})^2)) + \alpha} \\
921 & \\
922 &\geq \theta^{(0)} + \eta^{(0)} \frac{2\theta^{(0)}(0.6 - 0.55)}{4(\theta^{(0)})^2(0.55)(0.45) + \alpha} \\
923 & \\
924 &\geq \eta^{(0)} \frac{0.1\theta^{(0)}}{(\theta^{(0)})^2 + \alpha} \\
925 & \\
926 &= \eta^{(0)} \frac{0.1}{\theta^{(0)} + \alpha/\theta^{(0)}}. \\
927 & \\
928 &
\end{aligned}$$

929 Substituting our lower bound for  $\eta^{(0)}$  from the lemma statement yields:  
930

$$931 \quad \theta^{(1)} \geq \left( c\sqrt{\log \frac{1}{\theta^{(0)}}} \left( \theta^{(0)} + \frac{\alpha}{\theta^{(0)}} \right) \right) \frac{0.1}{\theta^{(0)} + \alpha/\theta^{(0)}} = 0.1c\sqrt{\log \frac{1}{\theta^{(0)}}}.$$

932 As  $\theta^{(0)} \rightarrow 0$ , this lower bound grows, so we can choose the universal constant  $c$  large enough to  
933 make  $\theta^{(1)}$  arbitrarily large.  
934

935 *Part 2: Establishing a key property of  $\theta^{(1)}$ .* We now show we can choose  $c$  large enough to ensure  
936 two conditions hold simultaneously for  $\theta^{(1)}$ :  
937

- 938 (i)  $\sigma((\theta^{(1)})^2) \geq 0.9$ .  
939  
940 (ii)  $4(\theta^{(1)})^2(1 - \sigma((\theta^{(1)})^2)) \leq \theta^{(0)}$ .  
941  
942

943 Condition (i) is met by choosing  $c$  sufficiently large. For condition (ii), we use the facts that  $1 - \sigma(z) \leq$   
944  $e^{-z}$  and that  $z(1 - \sigma(z))$  is a decreasing function for  $z \geq 2$ .<sup>3</sup> Let  $\theta_{\text{low}}^{(1)} := 0.1c\sqrt{\log(1/\theta^{(0)})}$ . This  
945 implies:  
946

$$\begin{aligned}
947 \quad 4(\theta^{(1)})^2(1 - \sigma((\theta^{(1)})^2)) &\leq 4(\theta_{\text{low}}^{(1)})^2(1 - \sigma((\theta_{\text{low}}^{(1)})^2)) \leq 4(\theta_{\text{low}}^{(1)})^2 e^{-(\theta_{\text{low}}^{(1)})^2} \\
948 &= 4(0.01c^2) \log(1/\theta^{(0)}) \cdot \exp\left(-0.01c^2 \log(1/\theta^{(0)})\right) \\
949 &= (0.04c^2) \log(1/\theta^{(0)}) \cdot (\theta^{(0)})^{0.01c^2} \\
950 &= \left( (0.04c^2) \log(1/\theta^{(0)}) (\theta^{(0)})^{0.01c^2 - 1} \right) \cdot \theta^{(0)}.
\end{aligned}$$

951 For a sufficiently large constant  $c$  (e.g.,  $0.01c^2 > 2$ ), the term in the large parenthesis is less than 1,  
952 because for a fixed  $\theta^{(0)} \in (0, 0.5]$ , the polynomial term  $(\theta^{(0)})^{0.01c^2 - 1}$  decays much faster than the  
953 logarithmic term  $\log(1/\theta^{(0)})$  grows. This establishes condition (ii).  
954

955 *Part 3: Proving geometric divergence for  $t \geq 1$ .* Consider any  $\eta$  satisfying the learning rate lower  
956 bound, i.e. suppose:  
957

$$958 \quad \eta \geq c\sqrt{\log \frac{1}{\theta^{(0)}}} \left( \theta^{(0)} + \frac{\alpha}{\theta^{(0)}} \right).$$

959 We show that for any  $\theta$  where  $\theta^2 \geq (\theta^{(1)})^2$ , it follows that  $|M_{\eta, \alpha}(\theta)| \geq \sqrt{2}|\theta|$ . First, rewrite the  
960 update as  
961

$$962 \quad M_{\eta, \alpha}(\theta) = \theta \left( 1 - \frac{2\eta(\sigma(\theta^2) - P)}{4\theta^2\sigma(\theta^2)(1 - \sigma(\theta^2)) + \alpha} \right).$$

963 Let  $K(\theta) = \frac{2\eta(\sigma(\theta^2) - P)}{4\theta^2\sigma(\theta^2)(1 - \sigma(\theta^2)) + \alpha}$ . We seek to show  $|1 - K(\theta)| \geq \sqrt{2}$ , which is true if  $K(\theta) \geq 1 + \sqrt{2}$ .  
964

965 We lower bound  $K(\theta)$  for any  $\theta$  where  $\theta^2 \geq (\theta^{(1)})^2$ . The numerator is positive and lower-bounded  
966 using condition (i):  $2\eta(\sigma(\theta^2) - P) \geq 2\eta(\sigma((\theta^{(1)})^2) - P) \geq 2\eta(0.9 - 0.8) = 0.2\eta$ . For the  
967

971 <sup>3</sup>Indeed,  $\frac{d}{dz}[z(1 - \sigma(z))] = (1 - \sigma(z)) - z\sigma'(z)(1 - \sigma(z)) < 0$  once  $z \geq 2$ .

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

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

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  $\eta$ . 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.  $\square$

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  $\gamma$ .

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

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

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  $4(\theta_i^*)^2 \sigma((\theta_i^*)^2)(1 - \sigma((\theta_i^*)^2))$  is also bounded by universal constants, and we conclude that the Fisher eigenvalues are proportional to the sampling probabilities. In particular,

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

where  $c_1$  is a universal constant.

The Gauss-Newton update for each coordinate  $\theta_i$  can be analyzed independently. For the sequence  $\theta^{(t)}$  to converge  $\theta^*$ , the iterates for each coordinate  $\theta_i[t]$  must also converge to  $\theta_i^*$ . From equation 22, the update rule for each coordinate can be written as:

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

This shows that we can apply Lemma 3 coordinate wise, where we take  $\alpha/\nu_i$  as the regularization parameter. 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  $\nu_i$  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  $I - \eta^{(\infty)}(H_*^{(\text{GN})} + \alpha I)^{-1}H_*^{(\text{GN})}$ . Its eigenvalues are  $1 - \eta^{(\infty)} \frac{\lambda_i(H_*^{(\text{GN})})}{\lambda_i(H_*^{(\text{GN})} + \alpha)}$ . We can lower bound the spectral radius by considering the smallest

eigenvalue of  $H_*^{(\text{GN})}$ ,  $\lambda_{\min}$ :

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

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

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

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

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

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.  $\square$

## D EXPERIMENTS

### D.1 ADDITIONAL EXPERIMENT INFORMATION

**Hyperparameters, hardware, and runtime** The learning rate ( $\eta$ ) search is first performed [within  \$\[10^{-5}, 0.1\]\$](#)  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 ( $\alpha$ ) search is at factors at 10 (e.g.  $10^{-3}$ ,  $10^{-4}$ ). [The optimal values typically lie within  \$10^{-2}\$  to  \$10^{-6}\$ , and  \$\text{GN}^{-1}\$  tends to require larger  \$\alpha\$  than  \$\text{GN}^{-1/2}\$ .](#) 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 Section 3.1.2 each completes within 1min. Simulation runs in Section 3.3 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

$$H^{(\text{GN})} := \mathbb{E}[gg^\top] \approx \mathbb{E}[GG^\top] \otimes \mathbb{E}[G^\top G], \quad (23)$$

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  $H^{(\text{GN})} \in \mathbb{R}^{mn \otimes mn}$  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.

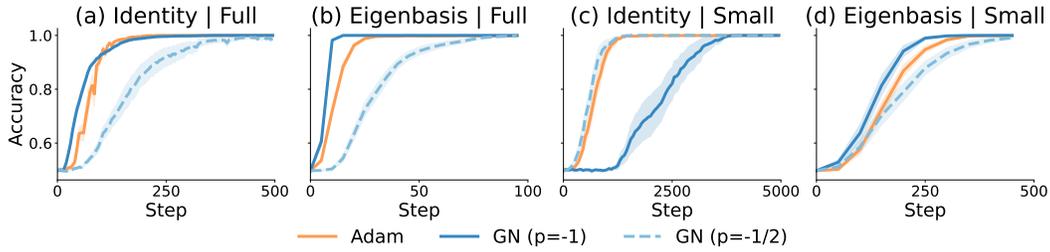


Figure 7: **Sparse parity**, comparing Adam,  $\text{GN}^{-1}$  and  $\text{GN}^{-1/2}$  for the full  $2 \times 2$  grid (Table 1).

## D.2 DETAILS FOR MLP EXPERIMENTS WITH SQUARED LOSS

This section provides details for the MLP experiments in Section 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  $\sigma$  defaults to ReLU unless specified otherwise.

Below we provide detailed descriptions of the tasks:

- **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,  $x_i \sim \mathcal{N}(0, \Sigma_x) \in \mathbb{R}^d$  where  $\Sigma_x \in \mathbb{R}^{d \times d}$  is a random covariance matrix. We initialize a random teacher model,  $f_{\mathcal{T}}: \mathbb{R}^d \rightarrow \mathbb{R}$ , 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,  $w_{\mathcal{T}} \in \mathbb{R}^{m \times d}$ ;  $a_{\mathcal{T}}, b_{\mathcal{T}} \in \mathbb{R}^d$ , 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  $d_S = 2 \times d_{\mathcal{T}}$ .
- **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., 2022; Edelman et al., 2023; Morwani et al., 2023; Abbe et al., 2024). It can be viewed as learning a sparse “feature” embedded in a much higher ambient dimension. Specifically,  $(d, k)$ -parity is a function from  $x \in \{\pm 1\}^d$  to  $y = \prod_{i \in \mathcal{S}} x_i \in \{\pm 1\}$ , where  $\mathcal{S} = \{s_1, s_2, \dots, s_k\} \subseteq [d]$  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; 2023). 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 Section 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 8 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.

## D.3 INTERPOLATING BETWEEN BASIS

In Section 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

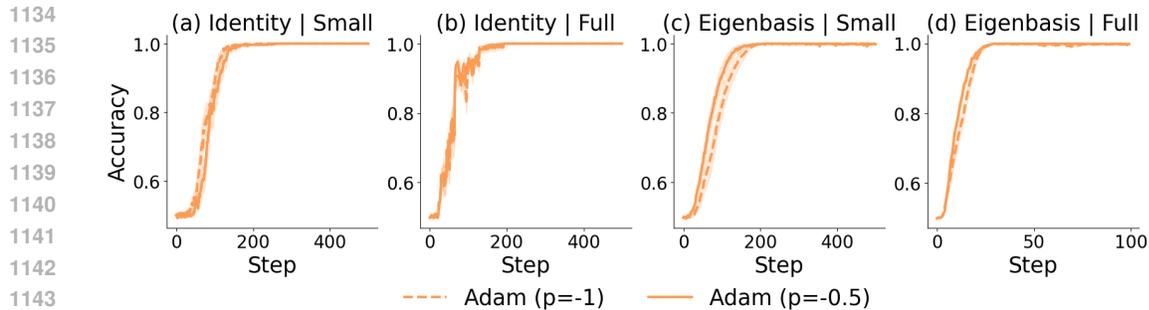


Figure 8: **Sparse parity**, comparing Adam, with power  $-1$  or  $-\frac{1}{2}$  for the full  $2 \times 2$  grid (Table 1).

---

**Algorithm 2** Geodesic interpolation between bases

---

- 17: **Input:** full GN basis  $U$ , interpolation factor  $\alpha$ .  
18:     Compute the matrix  $\log K := \text{logm}(U)$ .  
19:     Compute the matrix exponent  $\hat{U} := \exp(\alpha \cdot K)$ .  
20:     Obtain the real part  $U_\alpha := \text{real}(\hat{U})$ .  
21: **Output:**  $U_\alpha$ .
- 

more complete picture of the effect of basis, we provide results with more granularity with respect to the choice of basis.

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  $U_\gamma$ , 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 9. In particular, Adam and  $\text{GN}^{-\frac{1}{2}}$  behave similarly under the stochastic regime across basis choices, as predicted by the theory (Section 3.2).

#### D.4 DETAILS FOR LOGISTIC EXPERIMENTS

**Simulation for Section 4** We run simulation following the 2-layer linear network example in Section 4. The inputs are 2048-dimensional one-hot vectors following a power law decay, with  $\nu_i := \Pr(x = e_i) \propto i^{-c}$ . 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 Section 5. The attention module shares a similar structure as the logistic regression results in Section 4: the inner product of query and key matrices resembles the reparameterization, and the softmax function resembles the logistic function. As a result, Gauss-Newton is forced to take conservative step sizes and suffers from slow convergence, as shown in Figure 5.

We consider a selection task: 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.  $[x_1, \dots, x_T, s]$ . For  $s = e_i$ , the label is given by  $y = \langle \theta_*, x_i \rangle$ . We set  $T = 32$  in the experiments. The task is learned with a 1-layer 1-head Transformer with dimension 128. Figure 5 shows results comparing GN and Adam under the Kronecker-approximated eigenbasis with large batches (batch size = 16384), each aggregated over 10 seeds. Each run takes around 90min to complete.

1188  
 1189  
 1190  
 1191  
 1192  
 1193  
 1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218  
 1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229  
 1230  
 1231  
 1232  
 1233  
 1234  
 1235  
 1236  
 1237  
 1238  
 1239  
 1240  
 1241

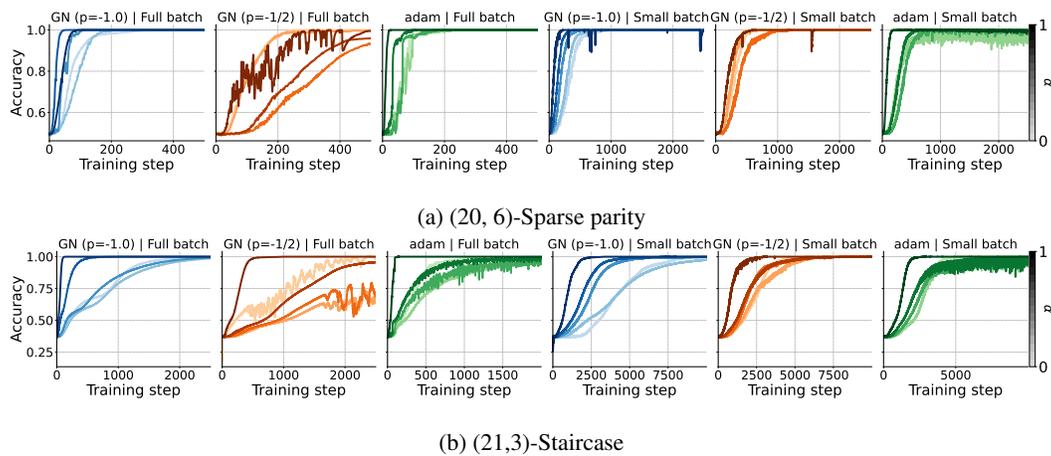


Figure 9: **Basis interpolation**: Comparing  $\text{GN}^{-1}$ ,  $\text{GN}^{-1/2}$ , and Adam under various bases, for parity and staircase (Section 5). Each basis is obtained by a geometric interpolation between the eigenbasis (darker colors) and the identity basis (lighter colors), parameterized by a factor  $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$ .